REPORT R-719 FEBRUARY, 1976 UILU-ENG 76-2207

CSL COORDINATED SCIENCE LABORATORY

ON THE COMPLEXITY OF DECODERS FOR GOPPA CODES

DILIP V. SARWATE

UNIVERSITY OF ILLINOIS - URBANA, ILLINOIS

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* ON THE COMPLEXITY OF DECODERS FOR GOPPA CODES		5. TYPE OF REPORT & PERIOD COVERED Technical Report 6. PERFORMING ORG. REPORT NUMBER R-719; UILU-ENG 76-2207
7. AUTHOR(s) Dilip V. Sarwate		8. CONTRACT OR GRANT NUMBER(s) DAAB-07-72-C-0259 NSF GK-24879
9. PERFORMING ORGANIZATION NAME AND ADDRESS Coordinated Science Laboratory University of Illinois at Urbana-Champaign Urbana, Illinois 61801		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Joint Services Electronics Program		12. REPORT DATE February, 1976 13. NUMBER OF PAGES 23
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Goppa Codes
Error Correcting Codes
Computational Complexity
Decoding Algorithms

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

Sugiyama et al. have shown that, for a t-error-correcting Goppa code of block length n, the key equation for errors-only decoding as well as for errors-and-erasures decoding can be solved in $O(t^2)$ arithmetic operations. Their algorithms use the extended version of Euclid's algorithm for the greatest common division (GCD) of two polynomials and have the same order of complexity as Berlekamp's algorithm for BCH codes. It is shown here that if a more efficient algorithm for computing polynomial GCDs is used,

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

20. ABSTRACT (continued)

then the key equation can be solved in $0(t \log^2 t)$ arithmetic operations. Also, determining the syndrome, the error locations and the error or erasure values all require $0(n \log n)$ arithmetic operations. Thus, for a fixed ratio of t/n, errors-only decoding as well as errors-and-erasures decoding of a Goppa code can be done in $0(n \log^2 n)$ arithmetic operations. It is also shown that long primitive binary BCH codes can be decoded in $0(n \log n)$ arithmetic operations.

UILU-ENG 76-2207

ON THE COMPLEXITY OF DECODERS FOR GOPPA CODES

by

Dilip V. Sarwate

This work was supported in part by the Joint Services Electronics Program (U.S. Army, U.S. Navy, and U.S. Air Force) under Contract DAAB-07-72-C-0259, and in part by the National Science Foundation under Grant GK-24879.

ON THE COMPLEXITY OF DECODERS FOR GOPPA CODES*

DILIP V. SARWATE†

February 20, 1976

ABSTRACT

Sugiyama et al. have shown that, for a t-error-correcting Goppa code of block length n, the key equation for errors-only decoding as well as for errors-and-erasures decoding can be solved in $O(t^2)$ arithmetic operations. Their algorithms use the extended version of Euclid's algorithm for the greatest common division (GCD) of two polynomials and have the same order of complexity as Berlekamp's algorithm for BCH codes. It is shown here that if a more efficient algorithm for computing polynomial GCDs is used, then the key equation can be solved in $O(t \log^2 t)$ arithmetic operations. Also, determining the syndrome, the error locations and the error or erasure values all require $O(n \log n)$ arithmetic operations. Thus, for a fixed ratio of t/n, errors-only decoding as well as errors-and-erasures decoding of a Goppa code can be done in $O(n \log^2 n)$ arithmetic operations. It is also shown that long primitive binary BCH codes can be decoded in $O(n \log n)$ arithmetic operations.

---

*This work was supported in part by the Joint Services Electronics Program (U.S. Army, U.S. Navy, and U.S. Air Force) under Contract DAAB-07-72-C-0259, and in part by the National Science Foundation under Grant GK-24879.

†Dilip V. Sarwate is with the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801.

ON THE COMPLEXITY OF DECODERS FOR GOPPA CODES

by

Dilip V. Sarwate
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

I. INTRODUCTION

Sugiyama et al. [1] have shown that, for a t-error-correcting Goppa code [2-4], the key equation for errors-only decoding can be solved for the error-locator polynomial $\sigma(z)$ and the error-evaluator polynomial $\eta(z)$ by use of the extended version of Euclid's algorithm for the greatest common divisor (GCD) of two polynomials. They also show that the number of multiplications required for this algorithm is approximately $7.5\ t^2$ whereas Burton's modification [5] of Berlekamp's algorithm for BCH codes [6] requires approximately $5\ t^2$ multiplications. Thus, both these algorithms require $O(t^2)$ arithmetic operations. Patterson's algorithm [7] for decoding Goppa codes uses Berlekamp's algorithm and hence has complexity at least $O(t^2)$. In Section II, it is shown that if an efficient computational technique for polynomial GCDs is used [8,9], then $\sigma(z)$ and $\eta(z)$ can be determined in $O(t \log^2 t)$ arithmetic operations. As one might expect, this algorithm is faster than the other only if t (and hence, by implication, the block length n) is large. In Section III, it is shown that the syndrome can be computed from the received vector in $O(n \log n)$

arithmetic operations and that the error locations and error values also can be computed from $\sigma(z)$ and $\eta(z)$ in $O(n \log n)$ arithmetic operations. The results of these sections have also been independently proposed by Justesen [15] in the context of the decoding of Reed-Solomon codes. Thus, for a fixed ratio of t/n, errors-only decoding of a Goppa code can be done in $O(n \log^2 n)$ arithmetic operations. Using Berlekamp's estimates of the minimum distance of long primitive binary BCH codes [11], it is shown that these codes can be decoded in $O(n \log n)$ arithmetic operations. In Section IV, errors-and-erasures decoding of Goppa codes is considered. The efficient GCD algorithm is applied to the errors-and-erasures version of the Sugiyama et al. algorithm [12] and it is shown that in this case also, the key equation can be solved in $O(t \log^2 t)$ arithmetic operations. Some other computations necessary in this case are also shown to be of complexity at most $O(t \log^2 t)$ or $O(n \log n)$. Thus errors-and-erasures decoding of Goppa codes is shown to be of the same order of complexity as errors-only decoding of Goppa codes.

II. ERRORS-ONLY DECODING

Following the notation in [4], let $g(z)$ be a polynomial of degree $2t$ with coefficients in $GF(q^m)$, L the subset of elements of $GF(q^m)$ that are not roots of $g(z)$, and n the number of elements in L. Then the Goppa code of length n, symbol field GF(q), location field $GF(q^m)$, and Goppa

polynomial g(z) is the set of all vectors $\underline{c}$ that satisfy

$$\sum_{\gamma \in L} \frac{c_\gamma}{z - \gamma} \equiv 0 \bmod g(z).$$

This code has minimum distance at least 2t + 1. Let $\underline{e}$ be the error vector, $\underline{r} = \underline{c} + \underline{e}$ the received vector, Then, the syndrome polynomial S(z) is the polynomial of degree 2t - 1 or less such that

$$S(z) \equiv \sum_{\gamma \in L} \frac{r_\gamma}{z - \gamma} \bmod g(z) .$$

Thus

$$S(z) = - \sum_{\gamma \in L} r_\gamma \cdot \frac{g(z) - g(\gamma)}{(z - \gamma)} \cdot \frac{1}{g(\gamma)} . \tag{1}$$

Let $e_\gamma \neq 0$ iff $\gamma \in M$, and define

$$\sigma(z) = \prod_{\gamma \in M} (z - \gamma) \tag{2}$$

$$\eta(z) = \sum_{\gamma \in M} e_\gamma \prod_{\delta \in M - \{\gamma\}} (z - \delta). \tag{3}$$

Then, the key equation for errors-only decoding is

$$S(z)\sigma(z) \equiv \eta(z) \bmod g(z) \tag{4}$$

and

$$e_\gamma = \begin{cases} 0 & \text{if } \sigma(\gamma) \neq 0 \\ \dfrac{\eta(\gamma)}{\sigma'(\gamma)} & \text{if } \sigma(\gamma) = 0 \end{cases} \tag{5}$$

where $\sigma'(z)$ is the formal derivative of $\sigma(z)$ [1-4].

The decoding algorithm for the Goppa code then consists of

(i) Computation of $S(z)$ from (1).

(ii) Solution of (4) for $\sigma(z)$ and $\eta(z)$.

(iii) Determination of the roots of $\sigma(z)$.

(iv) Determination of the error values e_γ using (5).

In the remainder of this section, an efficient algorithm for (ii) is discussed.

Following [9], define a sequence of "remainder polynomials" $a_0(z), a_1(z), \ldots, a_k(z), a_{k+1}(z)$ with degree $a_i(z) < \deg a_{i-1}(z)$ as follows. $a_0(z) = g(z)$, $a_1(z) = S(z)$. By successive divisions

$$\left.\begin{aligned} a_0(z) &= a_1(z)q_1(z) + a_2(z) \\ a_1(z) &= a_2(z)q_2(z) + a_3(z) \\ &\cdots \\ a_{k-1}(z) &= a_k(z)q_k(z) \end{aligned}\right\} \quad . \tag{6}$$

Since $a_k(z) \mid a_{k-1}(z)$, we set $a_{k+1}(z) = 0$.

Then $\mathrm{GCD}[a_0(z), a_1(z)] = a_k(z)$.

Two other sequences of polynomials $x_i(z), y_i(z)$ are defined as

$$x_0(z) = 1 \qquad x_1(z) = 0$$

$$y_0(z) = 0 \qquad y_1(z) = 1$$

$$\left.\begin{aligned} x_i(z) &= x_{i-2}(z) - q_{i-1}(z)x_{i-1}(z) \\ y_i(z) &= y_{i-2}(z) - q_{i-1}(z)y_{i-1}(z) \end{aligned}\right\} \qquad i \geq 2 \tag{7}$$

where the q_i's are defined in (6). Then, for $i \geq 0$

$$a_0(z)x_i(z) + a_1(z)y_i(z) = a_i(z) \tag{8}$$

i.e.,

$$S(z)y_i(z) \equiv a_i(z) \bmod g(z).$$

The following theorem is due to Sugiyama et al. [1].

Theorem 1: Given $a_0(z) = g(z)$ and $a_1(z) = S(z)$, let i be the unique integer such that $\deg a_i(z) < t$ and $\deg a_{i-1}(z) \geq t$. If t or fewer errors have occurred, then

$$\eta(z) = \delta a_i(z)$$

$$\sigma(z) = \delta y_i(z)$$

where δ is a nonzero constant chosen to make $\delta y_i(z)$ monic.

A fast decoding algorithm must find a_i and y_i in an efficient manner. An algorithm due to Moenck [8] can be used for this purpose. Given $a_0(z)$ of degree $2t$ and $a_1(z)$ of degree $2t - 1$ or less, define j as the unique integer such that $\deg a_j(z) > t$ and $\deg a_{j+1}(z) \leq t$. The algorithm is a recursive procedure that computes the matrix R_j where

$$R_j = \begin{bmatrix} x_j(z) & y_j(z) \\ x_{j+1}(z) & y_{j+1}(z) \end{bmatrix} . \tag{9}$$

From (8),

$$\begin{bmatrix} a_j(z) \\ \\ a_{j+1}(z) \end{bmatrix} = R_j \begin{bmatrix} a_0(z) \\ \\ a_1(z) \end{bmatrix} . \qquad (10)$$

The algorithm for the matrix R_j is as follows [9], where, by analogy to the integer function, $\lfloor d/e \rfloor$ denotes the quotient when d(z) is divided by e(z).

Algorithm 1:

procedure HGCD(a_0,a_1);

if deg (a_1) $\leq \frac{1}{2}$ deg (a_0) *then return* $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

else begin

let $a_0 = b_0 z^m + c_0$ where $m = \lfloor \deg(a_0)/2 \rfloor$ and deg (c_0) < m;

let $a_1 = b_1 z^m + c_1$ where deg (c_1) < m;

comment b_0 and b_1 are the leading terms of a_0 and a_1;

R ← HGCD(b_0,b_1);

$$\begin{bmatrix} d \\ \\ e \end{bmatrix} \leftarrow R \begin{bmatrix} a_0 \\ \\ a_1 \end{bmatrix} ;$$

f ← d modulo e;

let $e = g_0 x^{\lfloor m/2 \rfloor} + h_0$ where deg (h_0) < $\lfloor m/2 \rfloor$;

let $f = g_1 x^{\lfloor m/2 \rfloor} + h_1$ where deg (h_1) < $\lfloor m/2 \rfloor$;

$S \leftarrow$ HGCD (g_0, g_1);

$q \leftarrow \lfloor d/e \rfloor$;

return $S * \begin{bmatrix} 0 & 1 \\ 1 & -q \end{bmatrix} * R$

end

After computing R_j by Algorithm 1, $a_{j+1}(z)$ can be computed from (10). If deg $a_{j+1}(z) < t$, then $a_{j+1}(z)$ is the same as $a_i(z)$ of Theorem 1. If deg $a_{j+1}(z) = t$, the one can divide $a_j(z)$ by $a_{j+1}(z)$ to get

$$a_{j+2}(z) = a_j(z) - q_{j+1}(z)a_{j+1}(z). \tag{11}$$

Then deg $a_{j+2}(z) < t$ and this polynomial is the $a_i(z)$ of Theorem 1. From (7)

$$y_i(z) = y_{j+2}(z) = y_j(z) - q_{j+1}(z)y_{j+1}(z). \tag{12}$$

The number of arithmetic operations required to compute $\sigma(z)$ and $\eta(z)$ from $g(z)$ and $S(z)$ can be found using the following results from [8-10].

Theorem 2: HGCD requires $O[M(n) \log n]$ arithmetic operations if its arguments are of degree at most n, where $M(n)$ is the number of arithmetic operations required to multiply two polynomials of degree n.

Theorem 3: Two polynomials of degree n can be multiplied in O(n log n) arithmetic operations.

Theorem 4: Division of a 2nth-degree polynomial by a nth degree polynomial requires O(n log n) arithmetic operations.

Thus, HGCD requires $O(t \log^2 t)$ arithmetic operations to produce R_j. Following this, the multiplication in (10) requires O(t log t) operations. The division in (11) requires O(t log t) operations and computing y_i in (12) also requires O(t log t) operations. Finally, multiplying $a_i(z)$ and $y_i(z)$ by δ requires O(t) arithmetic operations. This result is summarized in the following theorem.

Theorem 5: Given the Goppa polynomial g(z) of degree 2t and the syndrome polynomial S(z) of degree at most 2t - 1, $\sigma(z)$ and $\eta(z)$ can be computed in $O(t \log^2 t)$ arithmetic operations.

III. SYNDROME COMPUTATION AND ERROR CORRECTION

Given the received vector $\underline{r}$, the decoder must first determine S(z) from (1). Let

$$S(z) = \sum_{i=0}^{2t-1} S_i z^i \quad \text{and} \quad \underline{S} = [S_{2t-1}, S_{2t-2}, \ldots, S_0].$$

Let $g(z) = \sum_{i=0}^{2t} g_i z^i$.

If $\gamma_1, \gamma_2, \ldots, \gamma_n$ are the n elements in L, then

$$\underline{S}^T = H\underline{r}^T \tag{13}$$

and

$$H = \begin{bmatrix} \frac{g_{2t}}{g(\gamma_1)} & \frac{g_{2t}}{g(\gamma_2)} & \cdots & \frac{g_{2t}}{g(\gamma_n)} \\ \frac{g_{2t-1} + g_{2t}\gamma_1}{g(\gamma_1)} & \frac{g_{2t-1} + g_{2t}\gamma_2}{g(\gamma_2)} & \cdots & \frac{g_{2t-1} + g_{2t}\gamma_n}{g(\gamma_n)} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{g_1 + g_2\gamma_1 + \ldots + g_{2t}\gamma_1^{2t-1}}{g(\gamma_1)} & \frac{g_1 + g_2\gamma_2 + \ldots + g_{2t}\gamma_2^{2t-1}}{g(\gamma_2)} & \cdots & \frac{g_1 + g_2\gamma_n + \ldots + g_{2t}\gamma_n^{2t-1}}{g(\gamma_n)} \end{bmatrix}$$

$$= XYZ \text{ where}$$

$$X = \begin{bmatrix} g_{2t} & 0 & 0 & \cdots & 0 \\ g_{2t-1} & g_{2t} & 0 & \cdots & 0 \\ g_{2t-2} & g_{2t-1} & g_{2t} & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ g_1 & g_2 & g_3 & \cdots & g_{2t} \end{bmatrix} \tag{14}$$

$$Y = \begin{bmatrix} 1 & 1 & \cdots\cdots & 1 \\ \gamma_1 & \gamma_2 & \cdots\cdots & \gamma_n \\ \gamma_1^2 & \gamma_2^2 & \cdots\cdots & \gamma_n^2 \\ \vdots & \vdots & \cdots & \vdots \\ \gamma_1^{2t-1} & \gamma_2^{2t-1} & \cdots\cdots & \gamma_n^{2t-1} \end{bmatrix} \tag{15}$$

and

$$Z = \text{diag}\left(\frac{1}{g(\gamma_1)}, \frac{1}{g(\gamma_2)}, \ldots, \frac{1}{g(\gamma_n)}\right). \tag{16}$$

Let us assume that $L = GF(q^m)$. Let α be a primitive element of $GF(q^m)$. Without loss of generality, one can take Y as

$$Y = \begin{bmatrix} 1 & 1 & 1 & \cdots\cdots\cdots\cdots\cdots & 1 \\ 0 & & \alpha & \cdots\cdots\cdots\cdots\cdots & \alpha^{q^m-2} \\ 0 & 1 & \alpha^2 & \cdots\cdots\cdots\cdots\cdots & \alpha^{2(q^m-2)} \\ \vdots & & & & \\ 0 & 1 & \alpha^{2t-1} & \cdots\cdots\cdots\cdots\cdots & \alpha^{(2t-1)(q^m-2)} \end{bmatrix} \tag{17}$$

$$Z = \text{diag}\left(\frac{1}{g(0)}, \frac{1}{g(1)}, \frac{1}{g(\alpha)}, \ldots, \frac{1}{g(\alpha^{q^m-2})}\right). \tag{18}$$

Now, the intermediate quantity, $\underline{r}^{*T} = Z\underline{r}^T$ can be computed using $n = q^m$ arithmetic operations. Let

$$\underline{r}^* = [r_\infty^*, r_0^*, r_1^*, r_2^*, \ldots, r_{q^m-2}^*]$$

and define $r^*(z) = \sum_{i=0}^{q^m-2} r_i^* z^i$.

Then, computing $Y\underline{r}^{*T}$ is the same as evaluating $r^*(z)$ at $z = 1,\alpha,\alpha^2,\ldots,\alpha^{2t-1}$. (Of course, r_∞^* must be added to $r^*(1)$.) Polynomial evaluations at the nth roots of unity are best done using the Fast Fourier Transforms [9,10] and this requires O(n log n) arithmetic operations.

Let $\underline{S}^{*T} = Y\underline{r}^{*T} = YZ\underline{r}^T$, where $\underline{S}^* = [S_{2t-1}^*,\ldots,S_0^*]$ and define the polynomial $S^*(z)$ as

$$S^*(z) = \sum_{i=0}^{2t-1} S_i^* z^i.$$

It is straightforward to verify that computing $X\underline{S}^{*T}$ is the same as computing the higher order 2t coefficients of $g(z)S^*(z)$. This polynomial multiplication requires O(t log t) arithmetic operations only. Thus, the syndrome can be computed in O(n log n) arithmetic operations for $L = GF(q^m)$.

When L is a proper subset of $GF(q^m)$ and the received vector is given $\underline{r} = [r_{\gamma_1}, r_{\gamma_2}, \ldots, r_{\gamma_n}]$, let

$$\underline{r}' = [r_0', r_1', r_\alpha', \ldots, r'_{\alpha^{q^m-2}}]$$

where $r_\gamma' = \begin{cases} r_\gamma & \text{if } \gamma \in L \\ 0 & \text{if } \gamma \notin L \end{cases}$

and

$$Z' = \mathrm{diag}(z_0, z_1, z_\alpha, \ldots, z_{\alpha^{q^m-2}})$$

$$\text{where } z_\gamma = \begin{cases} \dfrac{1}{g(\gamma)} & \text{if } \gamma \in L \\ 0 & \text{if } \gamma \notin L \end{cases}$$

Obviously $\underline{S}^T = H\underline{r}^T = XYZ'\underline{r}'^T$ where Y is given by (17). Thus, the syndrome can always be computed in $O(mq^m) = O(n \log n)$ arithmetic operations.

To correct the errors that have occurred, the decoder must find the roots of $\sigma(z)$ and then find the error values from (5). The Fourier Transform of $\sigma(z)$ gives the values of $\sigma(z)$ at $z = \alpha^i$, $i = 0,.,...,q^m-2$ and hence the roots of $\sigma(z)$ can be found in $O(n \log n)$ arithmetic operations by use of the Fast Fourier Transform. Similarly, the Fourier Transforms of $\eta(z)$ and $\sigma'(z)$ can be computed in $O(n \log n)$ steps and the error values can be computed in t arithmetic operations. It has thus been shown that the errors-only decoding of a Goppa code requires

(i) $O(n \log n)$ arithmetic operations to compute the syndrome.

(ii) $O(t \log^2 t)$ airthmetic operations to compute the polynomial $\sigma(z)$ and $\eta(z)$.

(iii) $O(n \log n)$ arithmetic operations to compute the error locations.

(iv) $O(n \log n)$ arithmetic operations to compute the error values.

The use of Fast Fourier Transforms to compute the syndrome, error locations and error values can substantially reduce the time spent in decoding even for quite short block lengths [13,14]. On the other hand, the computation of σ and η using HGCD is more efficient than using Euclid's algorithm or Berlekamps' algorithm only when t (and, hence, by implication, n) is quite large. Thus, the decoding algorithm is efficient asymptotically. If the ratio t/n is fixed, then the following result has been proved.

Theorem 6: Errors-only decoding of a Goppa code of block length n can be done in $O(n \log^2 n)$ arithmetic operations.

Corollary 1: Errors-only decoding of a long primitive binary BCH code can be done in $O(n \log n)$ arithmetic steps.

Proof: It is well-known that the BCH codes are a subclass of the Goppa codes. Berlekamp [11] has proved that for long primitive binary BCH codes of rate R and block length n, the designed distance is approximately $2n \ln R^{-1}/\log n$, i.e., t is $O(\frac{n}{\log n})$. Hence the solution of the key equation requires

$$O(t \log^2 t) = O(\frac{n}{\log n} \log(\frac{n}{\log n}) \log(\frac{n}{\log n})) = O(\frac{n}{\log n}(\log n - \log \log n)^2) =$$

$$= O(n \log n)$$

arithmetic operations. All other computations necessary are also of the same order of complexity. Q.E.D.

Corollary 2: Justesen [15]: Errors-only decoding of a Reed-Solomon code can be done in $O(n \log^2 n)$ arithmetic steps.

IV. ERRORS-AND-ERASURES DECODING

Sugiyama et al. [12] have shown that the key equation for errors-and-erasures decoding can be solved in a manner similar to that of their errors-only decoding algorithm. (Their paper contains an error which is patched up in this section,) Here, the relevant polynomials are the error-locator polynomial $\sigma_e(z)$ and the error-evaluator polynomial $\eta_e(z)$ (which were defined without subscripts in (2-3)) and the erasure-locator and erasure-evaluator polynomials defined analogously as

$$\sigma_\varepsilon(z) = \prod_{\gamma \in N} (z - \gamma) \tag{19}$$

$$\eta_\varepsilon(z) = \sum_{\gamma \in N} e_\gamma \prod_{\delta \in N - \{\gamma\}} (z - \delta)$$

where N is the set of erasure locations (N and M in (2) are disjoint sets) and the e_γ's are erasure values, i.e., the difference between the (arbitrary) value assigned by the decoder to r_γ and the transmitted symbol C_γ. The key equation in this case is

$$S(z) \equiv \frac{\eta_e(z)}{\sigma_e(z)} + \frac{\eta_\varepsilon(z)}{\sigma_\varepsilon(z)} \mod g(z) \tag{20}$$

and the decoder knows $S(z)$ as well as $\sigma_\varepsilon(z)$.

Let deg $\sigma_e = n_e$ and deg $\sigma_\varepsilon = n_\varepsilon$ with $1 \le 2n_e + + n_\varepsilon < 2t + 1$. The errata-locator and errata-evaluator polynomials are defined as

$$\sigma(z) = \sigma_e(z)\sigma_\varepsilon(z)$$
$$\eta(z) = \eta_e(z)\sigma_\varepsilon(z) + \eta_\varepsilon(z)\sigma_e(z). \tag{21}$$

The modified syndrome polynomial $S_\varepsilon(z)$ of degree $2t - 1$ or less is defined as

$$S_\varepsilon(z) \equiv \sigma_\varepsilon(z)S(z) \bmod g(z) \tag{22}$$

and the key equation can be rewritten as

$$\sigma_e(z)S_\varepsilon(z) \equiv \eta(z) \bmod g(z). \tag{23}$$

In [12], it is shown that deg $\sigma_e \le t - \frac{n_\varepsilon}{2}$ and deg $\eta \le t - 1 + \frac{n_\varepsilon}{2}$ and the following solution of (23) is proposed.

Theorem 7:

(i) If $n_\varepsilon = 0$ i.e., no erasures occurred, the decoder can follow the errors-only decoding procedure.

(ii) deg $S_\varepsilon < n_\varepsilon$ iff $n_e = 0$ and the solution is thus $\sigma_e(z) = 1$, $\eta_e(z) = 0$, $\eta(z) = \eta_\varepsilon(z) = S_\varepsilon(z)$.

(iii) Otherwise, take $a_0(z) = g(z)$ and $a_1(z) = S_\varepsilon(z)$ and let the remainder sequence $a_i(z)$ be as defined in (6-8). Let k be the unique integer such that deg $a_{k-1} \ge t + \frac{n_\varepsilon}{2}$ and deg $a_k \le t + \frac{n_\varepsilon}{2} - 1$. Then,

$$\eta(z) = \delta a_k(z)$$
$$\sigma_e(z) = \delta y_k(z)$$

where δ is a nonzero constant chosen to make δy_k monic.

The proof of (iii) in [12] is valid only if n_ε is an even number. When n_ε is odd, the bounds above can be replaced by $\deg a_{k-1} \geq t + \frac{n_\varepsilon}{2} + \frac{1}{2}$ and $\deg a_k \leq t + \frac{n_\varepsilon}{2} - \frac{3}{2}$. However, it is entirely possible that one of the polynmoials in the remainder sequence has degree $t + \frac{n_\varepsilon}{2} - \frac{1}{2}$. In this case, there is no integer k such that the degrees of both a_{k-1} and a_k are bounded as above. If, instead, one defines k as the unique integer such that $\deg a_{k-1} > t + \frac{n_\varepsilon}{2} - 1$ and $\deg a_k \leq t + \frac{n_\varepsilon}{2} - 1$, then, (for odd n_ε), $\deg a_{k-1} \geq t + \frac{n_\varepsilon}{2} - \frac{1}{2}$. However,

$$\begin{aligned} \deg y_k &= \deg a_0 - \deg a_{k-1} \\ &\leq 2t - (t + \tfrac{n_\varepsilon}{2} - \tfrac{1}{2}) = t - \tfrac{n_\varepsilon}{2} + \tfrac{1}{2}. \end{aligned}$$

In this case, the solution for σ_e may have degree $t - \frac{n_\varepsilon}{2} + \frac{1}{2}$ while the degree of σ_e is actually at most $t - \frac{n_\varepsilon}{2} - \frac{1}{2}$.

The way around this difficulty is quite straightforward. If n_ε is odd, so is $2n_e + n_\varepsilon$. However, all that is necessary for correct decoding is that $2n_e + n_\varepsilon < 2t + 1$, i.e., n_ε can be increased by 1 and still satisfy the constraint. Thus, the decoder can simply keep count of the number of erasures in the received symbols as they are received from the demodulator. If the first (n - 1) symbols contain an odd number of erasures, the decoder simply ignores the last received symbol and treats it as an erasure. In this manner, one can always ensure that the number of erasures is even.

In applying the fast computational algorithms, it is easy to see that case (i) of Theorem 7 has been dealt with in Section II while

case (ii) involves only polynomial multiplication and division and hence can be done in $O(t \log t)$ arithmetic operations. However, HGCD cannot be applied directly to case (iii) of Theorem 7. In this case, (20) can be rewritten as

$$\sigma_e(z)[\sigma_\varepsilon(z)S(z)] \equiv \eta(z) \bmod g(z). \tag{24}$$

Any solution of (24) is a solution of (20) and (23) and vice versa. Furthermore, if $2n_e + n_\varepsilon < 2t + 1$, the solution of (20) is unique [12, Theorem 1] and hence it suffices to solve (24) for a pair of relatively prime polynomials $\sigma_e(z)$ and $\eta(z)$ of degrees at most $t - \frac{n_\varepsilon}{2}$ and $t - 1 + \frac{n_\varepsilon}{2}$, respectively.

Lemma: If at least one erasure has occurred, then the erasure value(s) can be chosen so that deg $S = 2t - 1$.

Proof: $S_{2t-1} = \sum_{\gamma \in L} \frac{g_{2t} r_\gamma}{g(\gamma)}$

If S_{2t-1} is zero, one of the erasure values (which are arbitrarily assigned) can be changed (to $r_\gamma + 1$, say). For this modified received vector, $S_{2t-1} \neq 0$ and deg $S = 2t - 1$. Note that this checking and forcing of S_{2t-1} to be nonzero takes $O(n)$ arithmetic operations and can be done before the rest of the syndrome is computed.

Theorem 8: If deg S = 2t - 1 and the number of erasures n_ε is even, the congruence (24) can be solved for σ_e and η in $O(t \log^2 t)$ steps using HGCD.

Proof: Let $a_0 = \sigma_\varepsilon S$ and $a_1 = g$. Then deg $a_0 = 2t - 1 + n_\varepsilon$ and deg $a_1 = 2t$. Invoking HGCD gives the matrix R_j defined in (9) where j is the unique integer such that deg $a_j > \frac{1}{2}(2t - 1 + n_\varepsilon)$ and deg $a_{j+1} \leq \frac{1}{2}(2t - 1 + n_\varepsilon)$. Since n_ε is even, deg $a_j \geq t + \frac{n_\varepsilon}{2}$ and deg $a_{j+1} \leq t + \frac{n_\varepsilon}{2} - 1$.

Also,

$$x_{j+1}\sigma_\varepsilon S + y_{j+1} g = a_{j+1}$$

and deg x_{j+1} = deg g - deg $a_j \leq 2t - (t + \frac{n_\varepsilon}{2})$

i.e., deg $x_{j+1} \leq t - \frac{n_\varepsilon}{2}$.

Thus, $\eta(z) = \delta a_{j+1}(z)$

$\sigma_e(z) = \delta x_{j+1}(z)$

where δ is a constant chosen to make δx_{j+1} monic. The proof that a_{j+1} and x_{j+1} are relatively prime follows along the lines of the proof of [12, Theorem 2] and is omitted.

The multiplication $\sigma_\varepsilon S$ requires $O(t \log t)$ arithmetic operations. HGCD requires $O(t \log^2 t)$ arithmetic operations, the computation of a_{j+1} from (10) can be done in $O(t \log t)$ arithmetic operations and the multiplication by δ above requires $O(t)$ arithmetic operations

Q.E.D.

Notice that, from the Lemma and prior discussion, the decoder can always ensure that the hypotheses of Theorem 8 are satisfied. Following the solution of (24), the decoder can compute $\sigma(z)$ from (21), find the roots of σ and use (5) to determine the errata values. All of these operations require $O(n \log n)$ time only.

The next theorem considers the following problem. The demodulator output may be either a symbol from GF(q) or a special symbol denoting an erasure, for which the decoder substitutes some symbol from GF(q). The decoder thus knows the erasure-locations i.e., the set N. However, the computations required of the decoder make use of $\sigma_\varepsilon(z)$ and hence the decoder must first find $\sigma_\varepsilon(z)$ from the set N.

Theorem 9: Given the set of erasures N, the erasure-locator polynomial $\sigma_\varepsilon(z)$ can be determined by procedures of complexity $O(n \log n)$ and $O(t \log^2 t)$.

Proof: $n_\varepsilon = |N| \leq 2t$. Consider the Goppa code length n and minimum distance $\geq 4t + 1$ which has the Goppa polynomial $g^2(z)$. Suppose that the all-zeroes codeword was transmitted and the vector $\underline{v}$ was received where $v_\gamma = \begin{cases} 0 & \text{if } \gamma \notin N \\ 1 & \text{if } \gamma \in N \end{cases}$. Using an errors-only decoding algorithm for this code, one can find the syndrome in $O(n \log n)$ steps and the error-locator polynomial in $O(t \log^2 t)$ steps. This error-locator must be $\prod_{\gamma \in N} (z - \gamma)$, since this Goppa code has

minimum distance at least $4t + 1$ and at most $2t$ errors occurred. Hence $\sigma_\varepsilon(z)$ can be computed by procedures of complexity $O(n \log n)$ and $O(t \log^2 t)$. Q.E.D.

The results of this section can be summarized as:

Theorem 10: For a fixed ratio of t/n, errors-and-erasures decoding of a Goppa code requires $O(n \log^2 n)$ arithmetic steps.

Corollary: Errors-and-erasures decoding of a Reed-Solomon code of block length n requires $O(n \log^2 n)$ arithmetic steps.

In [16], Justesen describes his well-known asymptotically good codes. For these, the inner decoder uses $O(\frac{n^2}{\log^2 n})$ arithmetic operations while the outer decoder uses $O(\frac{n^2}{\log n})$ arithmetic operations. The corresponding bit complexities are $O_B(n^2)$ and $O_B(n^2 \log n)$. The outer decoder uses an errors-and-erasures decoding algorithm for Reed-Solomon codes. If the algorithm described in this section is used, the outer decoder requires $O(n \log^2 n)$ arithmetic operations and the corresponding bit complexity is $O_B(n \log^4 n)$. Thus, the complexity of the decoder for Justesen codes is dominated by the complexity of the inner, rather than the outer, decoder, and the bit complexity is $O_B(n^2)$ rather than $O_B(n^2 \log n)$. Thus, the following results has been proved.

Theorem 11: A Justesen code of block length n can be decoded in $O(n^2)$ bit operations.

V. CONCLUSIONS

Asymptotically efficient decoding algorithms have been presented for errors-only and errors-and-erasures decoding of Goppa codes. The methods suggested for syndrome computation and error-correction can be used to improve decoder efficiency even at moderate block lengths. However, the algorithm for the computation of the errata-locator polynomial and the errata-evaluator polynomial is more efficient than Berlekamp's algorithm or Euclid's algorithm only for very large block lengths. Therefore, a fruitful area of research would be in improving the coefficient of the arithmetic complexity of the HGCD algorithm, and also in investigating the complexity of the inner decoder for Justesen codes.

VI. ACKNOWLEDGEMENTS

It is a pleasure to acknowledge several very helpful conversations with S. J. Hong and F. P. Preparata. I am also grateful to an anonymous reviewer of an earlier version of this paper for pointing out [15].

VI. REFERENCES

1. Y. Sugiyama, M. Kasahara, S. Hirasawa, and T. Namekawa, "A Method for Solving Key Equation for Decoding Goppa Codes," *Information and Control*, vol. 27, pp. 87-99 (1975).

2. V. D. Goppa, "A New Class of Linear Error-Correcting Codes," *Problems of Information Transmission*, vol. 6, No. 4, pp. 24-30, (1970) (In Russian).

3. V. D. Goppa, "A Rational Representation of Codes and (L,g) Codes," *Problems of Information Transmission*, vol. 7, No. 3, pp. 41-49 (1971) (In Russian).

4. E. R. Berlekamp, "Goppa Codes," *IEEE Transactions on Information Theory*, vol. IT-19, No. 5, pp. 590-592 (1973).

5. H. O. Burton, "Inversionless Decoding of Binary BCH Codes," *IEEE Transactions on Information Theory*, vol. IT-17, No. 4, pp. 464-466 (1971).

6. E. R. Berlekamp, *Algebraic Coding Theory*, McGraw-Hill, New York (1968).

7. N. J. Patterson, "The Algebraic Decoding of Goppa Codes," *IEEE Transactions on Information Theory*, vol. IT-21, No. 2, pp. 203-208 (1975).

8. R. Moenck, "Fast Computation of GCD's," *Proceedings of the Fifth Annual ACM Symposium on the Theory of Computing*, pp. 142-151 (1973).

9. A. V. Aho, J. E. Hopcraft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, Mass. (1974).

10. J. M. Pollard, "The Fast Fourier Transform in a Finite Field," *Mathematics of Computation*, vol. 25, pp. 365-374 (1971).

11. E. R. Berlekamp, "Long Primitive Binary BCH Codes Have Distance $d \sim 2n \ln R^{-1}/\log n...$," *IEEE Transactions on Information Theory*, vol IT-18, No. 3, pp. 415-426 (1972).

12. Y. Sugiyama, M. Kasahara, S. Hirasawa, and T. Namekawa, "An Erasures-and-Errors Decoding Algorithm for Goppa Codes," *IEEE Transactions on Information Theory*, vol. IT-22, No. 2, (1976).

13. R. H. Paschburg, "Software Implementation of Error-Correcting Codes," Coordinated Science Laboratory Report R-659, University of Illinois at Urbana, August 1974.

14. R. T. Chien, C. L. Chen, R. B. Brown, and D. V. Sarwate, "Final Technical Report: Contract F30602-72-C-0031," Rome Air Development Center Report, RADC-TR-75-217, Rome, New York (1975).

15. J. Justesen, "On the Complexity of Decoding Reed-Solomon Codes," *IEEE Transactions on Information Theory*, (in press).

16. J. Justesen, "A Class of Constructive Asymptotically Good Algebraic Codes," *IEEE Transactions on Information Theory*, vol. IT-18, No. 5, pp. 652-656 (1972).