REPORT R-516 JUNE, 1971 UILU-ENG 71-2219

COORDINATED SCIENCE LABORATORY

POLYNOMIAL ROOT COMPUTATION WITH A STORED TABLE

R.T. CHIEN
A. MOY

UNIVERSITY OF ILLINOIS - URBANA, ILLINOIS

UILU-ENG 71-2219

POLYNOMIAL ROOT COMPUTATION WITH A STORED TABLE

by

R. T. Chien and A. Moy

This work was supported in part by the Joint Serivces Electronics Program (U. S. Army, U. S. Navy and U. S. Air Force) under Contract DAAB-07-67-C-0199, and in part by the National Science Foundation under Grant GK-2339 and GK-24879.

(This will appear in the Proceedings of the 5th Annual Princeton Conference on Information Science and Systems.)

POLYNOMIAL ROOT COMPUTATION WITH A STORED TABLE*

R. T. Chien and A. Moy
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign

Abstract

A method of finding the roots of a polynomial over a finite field is presented. The proposed method uses a small table to help reduce the computational complexity.

This method is applicable to algebraic decoding techniques, particularly toward the computation of error locations. The stored table approach is attractive due to its high speed.

---

*This work is supported by the National Science Foundation under Grant GK-2339 and GK-24879; auxiliary support is provided by the Joint Services Electronics Program (U.S. Army, U.S. Navy, and U.S. Air Force) under contract DAAB-07-67-C-0199.

POLYNOMIAL ROOT COMPUTATION WITH A STORED TABLE*

R. T. Chien and A. Moy
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign

I. Introduction

The purpose of this paper is to present methods to find roots of polynomials over $GF(2^n)$. A particularly important application of this work is in the decoding process of BCH codes, a class of cyclic codes in which the generator polynomial g(x) is the polynomial of least degree for which

$$\alpha^{m_o}, \alpha^{m_o+1}, \ldots, \alpha^{m_o+d-2}$$

are roots, m_o an integer and α is an element in $GF(p^m)$.[1,2]

The decoding procedure can be reduced to the process of solving the equation

$$e(\alpha^j) = \Sigma\, Y_i X_i^{j+m_o-1} = S_j,\ j = 1,2,\ldots,2t,$$

where the error pattern e(x) is described by values Y_i and location X_i, and the locations given in terms of an error-location number α^{j-1} for the jth symbol.[1]

The error location polynomial $\sigma(x)$ is defined as $\sigma(x) = X^t + \sigma_1 X^{t-1} + \ldots + \sigma_t = \prod_{i=1}^{t} (x - X_i)$. Presently, there are methods to find σ_1, σ_2, ..., σ_t from S_1, S_2, ..., S_{2t} and to calculate the values Y_i if the values of X_i are known.[3]

However, finding the values of X_i, i.e., the roots of $\sigma(x) = 0$ still poses an immediate important problem.

Of the available methods to handle this problem, two results will be discussed and used as comparison to this work. One is in the work of Berlekamp, Ramsey, and Solomon[2] and more recently, with the additional material, in Berlekamp's Algebraic Coding Theory.[3] A brief summary is as follows:

(i) If the given polynomial f(x) is not an affine polynomial, i.e.,

$$f(x) \neq \sum_i L_i Z^{p^i} - u, \; L_i, \; u \in GF(p^m),$$

then, if possible, multiply f(x) by a suitable factor to transform f(x) into an affine polynomial A(x). If this is not possible, then use the algorithm in [2] to obtain an affine multiple of f(x).

(ii) Find the roots of the affine polynomial by solving m simultaneous equations over GF(p).

(iii) Substitute these roots in f(x) to determine which are the roots of f(x).

A more recent work is that of Chien, Cunningham, and Oldham.[4] The results, briefly, are:

(i) The given polynomial is transformed to a standard form.

(ii) The polynomial in standard form is factors conceptually.

(iii) Coefficients of the factors are found with the aid of a stored table.

The goal of this paper is to give a more efficient alternative to the methods mentioned above for polynomials of degree 3. The methods will avoid the need for an affine polynomial and maintain the size of the table at a minimal. For $GF(2^n)$ where n is an even integer, the methods presented are efficient and easy to manipulate. When n is odd, the efficiency is still comparable to the other methods if n is not large.

II. Basic Theory

Suppose, over $GF(2^n)$, n an even positive integer, it is given that

$$f(x) = x^3 + px^2 + qx + r = 0,$$

where p, q, r are elementary symmetric functions of the roots β_1, β_2, β_3 of f(x). Then,

$$\beta_1 + \beta_2 + \beta_3 = p \qquad (1)$$

$$\beta_1\beta_2 + \beta_1\beta_3 + \beta_2\beta_3 = q \qquad (2)$$

$$\beta_1\beta_2\beta_3 = r. \qquad (3)$$

Using a method from the theory of equations, first define the functions ϕ and ξ by

$$\phi = \beta_1 + \omega\beta_2 + \omega^2\beta_3 \qquad (4)$$

$$\xi = \beta_1 + \omega^2\beta_2 + \omega\beta_3 \qquad (5)$$

where $\omega \neq 1$ is a cubic root of unity.

Now, perform the following operations on equations (1), (4), and (5):

(i) Add equations (1), (4), and (5)

(ii) Add equations $\omega^2 p$, $\omega\phi$, and ξ

(iii) Add equations ωp, $\omega^2\phi$, and ξ

These operations will give, respectively,

$$\begin{aligned} \beta_1 &= p + \phi + \xi \\ \beta_2 &= p + \omega^2\phi + \omega\xi \\ \beta_3 &= p + \omega\phi + \omega^2\xi \end{aligned} \qquad (6)$$

The roots of f(x), then, can be found if ϕ and ξ are known. Now, it can easily be shown that

$$\phi\xi = p^2 + q \tag{7}$$

$$\phi^3 + \xi^3 = pq + r \tag{8}$$

Equations (7) and (8) give

$$(\xi^3)^2 + (pq+r)\xi^3 + (p^2+q)^3 = 0 \tag{9}$$

Since (9) is a second degree equation in ξ^3, it follows that the only roots of the equation must belong to

$$K_3 \equiv \{\text{elements with multiple 3 exponents}\}$$

It should also be noted that (9) has solutions if and only if f(x) has solutions. (9) can be solved with the aid of a table constructed from the elements of K_3. It will consist of the sums S and products P of elements of K_3. Although the table will be large if the field is large, an algorithm will be provided to shorten the table considerably.

III. An Example

Consider

$$(x + \alpha^3)(x + \alpha^5)(x + \alpha^{10}) = x^3 + \alpha^{14}x^2 + \alpha^{14}x + \alpha^3 = 0$$

over $GF(2^4)$. Here,

$$p = \alpha^{14} \quad , \quad q = \alpha^{14} \quad , \quad r = \alpha^3 \text{ and}$$

$$pq + r = \alpha^8, \quad (p^2 + q)^3 = (\alpha^2)^3 = \alpha^6 .$$

Substituting in (9) gives

$$(\xi^3)^2 + \alpha^8\xi^3 + \alpha^6 = 0$$

Table 1: Stored Table for GF(2^4)

	A	P	B	S
1	$\alpha^3 \cdot \alpha^{12}$	1	$\alpha^3 + \alpha^{12}$	α^{10}
2	$\alpha^6 \cdot \alpha^9$	1	$\alpha^6 + \alpha^9$	α^5
3	$1 \cdot \alpha^3$	α^3	$1 + \alpha^3$	α^{14}
4	$\alpha^6 \cdot \alpha^{12}$	α^3	$\alpha^6 + {}^{12}$	α^4
5	$1 \cdot \alpha^6$	α^6	$1 + \alpha^6$	α^{13}
6	$\alpha^9 \cdot \alpha^{12}$	α^6	$\alpha^9 + \alpha^{12}$	α^8
7	$1 \cdot \alpha^9$	α^9	$1 + \alpha^9$	α^7
8	$\alpha^3 \cdot \alpha^6$	α^9	$\alpha^3 + \alpha^6$	α^2
9	$1 \cdot \alpha^{12}$	α^{12}	$1 + \alpha^{12}$	α^{11}
10	$\alpha^3 \cdot \alpha^9$	α^{12}	$\alpha^3 + \alpha^9$	α

The procedure for using Table 1 is as follows:

(i) Look for the rows in which $(p^2+q)^3 = \alpha^6$ appears in column p. (Rows 5 and 6)

(ii) Check to see if $pq+r = \alpha^8$, appearing in column S, is in the same row as α^6. (It does in row 6)

(iii) If both $(p^2+q)^3 = \alpha^6$ and $pq+r = \alpha^8$ are in the same row, there are solutions for ξ^3 and they appear in column A and in the same row as $(p^2+q)^3 = \alpha^6$ and $pq+r = \alpha^8$. In this example,

$$\xi^3 = \alpha^9 \text{ or } \alpha^{12}$$

Thus, $\xi = \mu\alpha^3$ or $\mu\alpha^4$, μ a cubic root of unity. (In practice, it is suggested that $\mu = 1$). Take $\xi = \alpha^3$. Then $\phi = \frac{p^2+q}{\xi} = \alpha^{14}$. Substituting in (6) gives

$$\beta_1 = \alpha^{14} + \alpha^{14} + \alpha^3 = \alpha^3$$
$$\beta_2 = \alpha^{14} + \alpha^9 + \alpha^8 = \alpha^5$$
$$\beta_3 = \alpha^{14} + \alpha^4 + \alpha^{13} = \alpha^{10}$$

which are the roots of the original equation. It can easily be seen that in the solution of (9), the size of the table is a major problem. For example, the table for $GF(2^4)$ requires 10 entries whereas $GF(2^6)$ would contain 210 entries.

For the solution of (9), an algorithm will now be presented to shorten the size of the table and to obtain the solution more readily.

IV. An Algorithm Over $GF(2^n)$ With a Reduced Table

We shall now present a method for reducing the table. This method applies to all $GF(2^n)$. $GF(2^4)$ will be used as an example.

(i) Find the rows in which the element 1 appears under column p. (Rows 1 and 2)

(ii) Using only these rows, factor out the term with the lowest exponent listed under B.

$$\alpha^3 + \alpha^{12} = \alpha^3(1+\alpha^9)$$
$$\alpha^6 + \alpha^9 = \alpha^6(1+\alpha^3)$$

(iii) Make a permanent correspondence between $1+\alpha^i$ and the element in the same row under S.

$$1+\alpha^9 \longleftrightarrow \alpha^{10}$$
$$1+\alpha^3 \longleftrightarrow \alpha^5$$

Denote the set of elements which corresponds to the set $\{1+\alpha^i\}$

by S_n^*. Thus, $S_4^* = \{\alpha^5, \alpha^{10}\}$.

The newly constructed tables have the obvious advantage of being much smaller than the old ones. For comparison, the old table for GF(2^n) has $\binom{\frac{2^n-1}{3}}{2}$ entries, whereas the new one has $\frac{\frac{2^n-1}{3}-1}{2} = \frac{2^n-4}{6}$. The new tables for GF($2^4$) and GF($2^6$) are listed in Table 2.

Table 2: Reduced Tables for GF(2^4) and GF(2^6)

GF(2^4)

S_4^*	$1+\alpha^i$
α^{10}	α^7
α^5	α^{14}

GF(2^6)

S_6^*	$1+\alpha^i$
α^{61}	α^{58}
α^{59}	α^{53}
α^{18}	α^9
α^{55}	α^{43}
α^{31}	α^{16}
α^{36}	α^{18}
1	α^{42}
α^{47}	α^{23}
α^9	α^{45}
α^{62}	α^{32}

V. Algorithm for Using New Table

(i) Find $\alpha^i = \frac{pq+r}{(p^2+q)^{3/2}}$. α^i must be an element in S_n^*. If $\alpha^i \notin S_n^*$, then (9) has no solutions.

(ii) Find the corresponding element to α^i in the table. Suppose that element is α^j. Find $\alpha^k = \frac{pq+r}{\alpha^j}$.

α^k will be one of the solutions for ξ^3. The other is $\frac{(p^2+q)^3}{\alpha^k}$.

VI. Another Example

$(x+\alpha)(x+\alpha^2)(x+\alpha^5) = x^3+\alpha^{17}x^2 + \alpha^{48}x + \alpha^8 = 0$ over $GF(2^6)$. Here,

$$pq+r = \alpha^3 \quad , \quad (p^2+q)^3 = (\alpha^{23})^3 = \alpha^6$$

(9) becomes $(\xi^3)^2 + \alpha^3\xi^3 + \alpha^6 = 0$. Applying algorithm, find

$$\alpha^i = \frac{\alpha^3}{\alpha^3} = 1 \,\varepsilon\, S_6^* .$$

From table, $1 \longleftrightarrow \alpha^{42}$. Thus,

$$\alpha^k = \frac{\alpha^3}{\alpha^{43}} = \alpha^{24} \text{ and } \frac{(p^2+q)^3}{\alpha^k} = \frac{\alpha^6}{\alpha^{24}} = \alpha^{45}$$

Take $\xi = \alpha^8$. Then $\phi = \frac{\alpha^{23}}{\alpha^8} = \alpha^{15}$

$$\beta_1 = \alpha^{17} + \alpha^{15} + \alpha^8 = \alpha$$

$$\beta_2 = \alpha^{17} + \alpha^{57} + \alpha^{29} = \alpha^5$$

$$\beta_3 = \alpha^{17} + \alpha^{36} + \alpha^{50} = \alpha^2$$

VII. The Case Where n is Odd

Thus far, the field has been restricted to $GF(2^n)$, n an even integer.

This is done because of the need for a cubic root of unity different from 1. To make the theory adaptable for the cases when n is odd, it is noted that $GF(2^n)$ is a subfield of $GF(2^m)$ if and only if $n|m$. Thus, if n is odd, transform the existing equation in $GF(2^n)$ into an equation in $GF(2^{2n})$.

In doing so, the disadvantages are the additional time spent, increase in the size of tables, and the need to find the correct mapping.

<u>References</u>

[1]Peterson, W. W. & E. J. Weldon, "Error Correcting Codes," M.I.T. Press, 1970.

[2]Berlekamp, E. R., H. Ramsey, and G. Solomon, "On the Solution of Algebraic Equations over Finite Fields," Information and Control, <u>10</u>, pp. 553-564 (1967).

[3]Berlekamp, E. R., <u>Algebraic Coding Theory</u>, McGraw-Hill, 1968.

[4]Chien, R. T., B. E. Cunningham, and I. B. Oldham, "Hybrid Methods for Finding Roots of a Polynomial - with Application to BCH Decoding," IEEE Transactions on Information Theory, <u>15</u>, pp. 329-335 (March 1969).

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)* Coordinated Science Laboratory University of Illinois Urbana, Illinois 61801	2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP	
3. REPORT TITLE POLYNOMIAL ROOT COMPUTATION WITH A STORED TABLE		
4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*		
5. AUTHOR(S) *(First name, middle initial, last name)* R. T. Chien and A. Moy		
6. REPORT DATE June, 1971	7a. TOTAL NO. OF PAGES 9	7b. NO. OF REFS 4
8a. CONTRACT OR GRANT NO. DAAB-07-67-C-0199; NSF GK-2339 and b. PROJECT NO. NSF GK-24879 c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) R-516 9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)* UILU-ENG 71-2219	
10. DISTRIBUTION STATEMENT This document has been approved for public release and sale; its distribution is unlimited.		
11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Joint Services Electronics Program through U. S. Army Electronics Command	

13. ABSTRACT

A method of finding the roots of a polynomial over a finite field is presented. The proposed method uses a small table to help reduce the computational complexity.

This method is applicable to algebraic decoding techniques, particularly toward the computation of error locations. The stored table approach is attractive due to its high speed.

DD FORM 1 NOV 65 1473

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Algebraic Decoding Techniques						
Error Locations						
Stored Table						