CS77005-R

DIVIDED DIFFERENCE METHODS FOR GALOIS SWITCHING FUNCTIONS

T. C. Wesselkamper

revised May 1977

Department of Computer Science
Virginia Polytechnic Institute and State University

Blacksburg, Virginia 24061

The work reported herein was supported in part by National Science Foundation
Grant No. DCR74-18108

Abstract

An alternative is provided to a recently published method of Benjauthrit and Reed for calculating the coefficients of the polynomial expansion of a given function. The method herein is an adaptation to finite fields of a method of Newton. The method is exhibited for functions of one and two variables. The relative advantages and disadvantages of the two methods are discussed. Some empirical results are given for GF(9) and GF(16). It is shown that functions with "don't care" states are represented by a polynomial of minimal degree by this method.

Keywords: Divided difference methods, Reed-Muller Decomposition Theorem, finite field, Newton's Interpolation Theorem.

CR Categories: 5.30, 6.31, 5.13

MR Categories: 12C05, 41A10

I. Introduction

In a recent paper in these Transactions Benjauthrit and Reed exhibit a method for computing the coefficients f(i) in the finite field $GF(p^n)$ in the expansion:

$$F(x) = \sum_{i=0}^{k-1} f(i)x^i \qquad (1)$$

of a function F: $E(k) \to E(k)$, where $k=p^n$, and E(k) is the space $E(k) = \{0,1,\ldots,k-1\}$. The result is important since every such function F has a unique expansion over $GF(p^n)$ of this form. Further, they generalize this to the case in which F is a function of n variables, F: $E^n(k) \to E(k)$. Specifically in the case of expansion (1):

$$f(0) = F(0);$$

$$f(i) = \sum_{\gamma \neq 0} [F(0) - F(\gamma)]\, \gamma^{-i} \qquad 0 < i < k \qquad (2)$$

In the two variable case the expansion is:

$$F(x,y) = \sum_{i,j=0}^{k-1} f(i,j)x^i y^j \qquad (3)$$

and the coefficients f(i,j) are given by:

$$f(0,0) = F(0,0);$$

$$f(0,j) = \sum_{\gamma \neq 0} [F(0,0) - F(0,\gamma)]\gamma^{-j} \qquad 0 < j < k$$

$$f(i,0) = \sum_{\gamma \neq 0} [F(0,0) - F(\gamma,0)]\, \gamma^{-i} \qquad 0 < i < k$$

$$f(i,j) = \sum_{\gamma,\delta \neq 0} [F(0,0) - F(0,\delta) - F(\gamma,0) + F(\gamma,\delta)]\, \gamma^{-i}\delta^{-j} \qquad (4)$$

The coefficients of (2) and (4) are described by Benjauthrit and Reed as generalized finite differences. In the next section we present a classical approach to the same problem which makes evident the role of "difference" in the process of deriving the coefficients f(i) and f(i,j).

Throughout this paper, let p be a prime and n be a natural number. Let $k = p^n$. Let + and * denote addition and multiplication, respectively, over $GF(p^n)$. When there is no danger of confusion, we write 'ab' for 'a*b'. Finally let Σ and Π denote the extended sum and extended product in the usual way. Let $E(k) = \{0,1,\ldots,k-1\}$.

Definition 1:- For each k, let $Z_k(x) = \prod_{i=0}^{k-1}(x-i)$.

Note that for each $a \in E(k)$, we have $Z_k(a) = 0$.

II. Divided Difference Tables

The classic works on difference methods assume the domain of definition to be either the rational field or the real field [3,4]. Most often in the literature the underlying field is not specified. In this paper we show that the methods work equally over a finite field. (In fact, they work over any field, but we make use of finiteness to render life particularly simple and pleasant.)

Following the notation of [4] we state:

Definition 2:- Let p be a prime, n a natural number, and $k = p^n$. If the sequence $x_0, x_1, \cdots, x_{k-1}$ is some permutation of the elements of GF(k), then:

$$[x_i x_i] = 0;$$

$$[x_i x_j] = \frac{F(x_i) - F(x_j)}{x_i - x_j}, \text{ if } i \neq j; \tag{5}$$

$$[x_i x_{i+1} \cdots x_{i+j}] = \frac{[x_i \cdots x_{i+j-1}] - [x_{i+1} \cdots x_{i+j}]}{x_i - x_{i+j}}, \text{ if } j \geq 2. \tag{6}$$

In particular, we have:

$$[x_0 x_1] = \frac{F(x_0) - F(x_1)}{x_0 - x_1}, \text{ a first order difference;}$$

$$[x_0 x_1 x_2] = \frac{[x_0 x_1] - [x_1 x_2]}{x_0 - x_2}, \text{ a second order difference; and}$$

$$[x_0 x_1 \cdots x_j] = \frac{[x_0 x_1 \cdots x_{j-1}] - [x_1 x_2 \cdots x_j]}{x_0 - x_j}, \text{ a } j^{\text{th}} \text{ order difference.}$$

Note also that $[x_i x_j] = \frac{F(x_i) - F(x_j)}{x_i - x_j} = \frac{F(x_j) - F(x_i)}{x_j - x_i} = [x_j x_i]$, and in particular, $[x_0 x_1] = [x_1 x_0]$.

The relationship between these may be seen from the following table:

x_0	$F(x_0)$			
		$[x_0x_1]$		
x_1	$F(x_1)$		$[x_0x_1x_2]$	
		$[x_1x_2]$		$[x_0x_1x_2x_3]$
x_2	$F(x_2)$		$[x_1x_2x_3]$	
		$[x_2x_3]$		$[x_1x_2x_3x_4]$
x_3	$F(x_3)$		$[x_2x_3x_4]$	
		$[x_3x_4]$		$[x_2x_3x_4x_5]$
x_4	$F(x_4)$		$[x_3x_4x_5]$	
		$[x_4x_5]$		.
x_5	$F(x_5)$		.	
		.		.
.	.		.	
		.		.
.	.		.	
		.		$[x_{k-4}x_{k-3}x_{k-2}x_{k-1}]$
.	.		.	
		$[x_{k-2}x_{k-1}]$		
x_{k-1}	$F(x_{k-1})$		$[x_{k-3}x_{k-2}x_{k-1}]$	

Example 1:- Let k = 5 and construct a difference table for the function with value sequence < 13231 > .

x	F(x)				
0	1				
		2/1			
1	3		2/2		
		4/1		0	
2	2		2/2		0
		1/1		0	
3	3		2/2		
		3/1			
4	1				

Example 2:- Let k = 5 and construct a difference table for:

$$V_3(x) = \begin{cases} 1, & \text{if } x=3; \\ 0, & \text{if } x\neq 3. \end{cases}$$

x	$V_3(x)$				
0	0				
		0/1			
1	0		0/2		
		0/1		3/3=1	
2	0		1/2=3		1/4=4
		1/1		1/3=2	
3	1		3/2=4		
		4/1			
4	0				

Example 3:- Same as Example 2, but permute the order of the values of x.

x	$V_3(x)$				
0	0				
		0			
2	0		0		
		0		0	
4	0		0		2/3=4
		0		2/1	
1	0		3/4=2		
		1/2=3			
3	1				

Example 4:- Let k = 4, and construct a difference table for:

$$V_2(x) = \begin{cases} 1, & \text{if } x = 2; \\ 0, & \text{if } x \neq 2. \end{cases}$$

The field $GF(2^2)$ is given by the two tables:

+	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

.	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	3	1
3	0	3	1	2

x	$V_2(x)$			
0	0			
		0/1=0		
1	0		2/2=1	
		1/3=2		3/3=1
2	1		3/2=2	
		1/1=1		
3	0			

The above remark, that $[x_0x_1] = [x_1x_0]$, can be generalized into a pleasant and important result, contained in the corollary below. Regrettably there does not appear to be an equally pleasant proof. The following lemma proved in [2, p. 10] achieves the desired result.

Lemma 1:- Let p be a prime, n a natural number, and $k=p^n$

If $x_0, x_1, \ldots, x_j$ are distinct elements of GF(k) and

if F(x) is a polynomial of degree n ($j \leq n$), then,

$$[x_0x_1\ldots x_j] = \frac{F(x_0)}{(x_0-x_1)(x_0-x_2)\ldots(x_0-x_j)} + \frac{F(x_1)}{(x_1-x_0)(x_1-x_2)\ldots(x_1-x_j)} + \ldots . + \frac{F(x_j)}{(x_j-x_0)(x_j-x_1)\ldots(x_j-x_{j-1})}$$

Corollary:- If $x_0, x_1, \ldots, x_j$ are distinct elements of GF(k) and F(x) is a polynomial of degree n ($j \leq n$) and s is a permutation of the integers 0, 1, ..., j, then $[x_0x_1\ldots x_j] = [x_{s(0)}x_{s(1)}\ldots x_{s(j)}]$. In words, the value of $[x_0x_1\ldots x_j]$ is invariant under a permutation of its elements.

Proof:- Note that a permutation of the elements $\{x_i\}$ permutes the order of the terms in the expression in Lemma 1 but leaves the value unchanged.

III. Divided Difference Polynomials

The method which was employed in the previous section can be extended. Note that in Definition 2 we required that the points $x_0, x_1, \ldots, x_{k-1}$ be distinct. We wish to extend our definition by weakening this restriction.

Lemma 2: Let p be a prime, n a natural number, and $k = p^n$. If F(x) is a polynomial of degree n $(1 \leq n \leq k-1)$ and $x_1 \in GF(k)$, then there exists a unique polynomial q(x) of degree n-1 such that:

$$q(x) = \frac{F(x) - F(x_1)}{x - x_1}$$

Proof: If F(x) is a polynomial, then $F(x_1) \in GF(k)$ and $F(x) - F(x_1)$ is a polynomial. By the Euclidean algorithm there exists a polynomial q(x) of degree n-1 and an $r \in GF(k)$ such that $f(x) - f(x_1) = (x - x_1)q(x) + r$. Since r is a constant, setting $x = x_1$ gives $r = 0$. Since q(x) is of degree less than or equal to k-2, q(x) defines a unique function.

This lemma makes the following definition reasonable.

Definition 3: Let p be a prime, n a natural number, and $k = p^n$. Let $x_0, x_1, \ldots, x_{k-1}$ be a permutation of the elements of GF(k). Let F(x) be a polynomial of degree n, $(0 \leq n \leq k-1)$.

$$[xx_1] = \frac{F(x) - F(x_1)}{x - x_1} \tag{7}$$

$$[xx_1 \ldots x_j] = \frac{[xx_1 \ldots x_{j-1}] - [x_1 x_2 \ldots x_j]}{x - x_j} \tag{8}$$

The notation of Definition 3 is designed to lull the reader into accepting it as a simple extension of Definition 2. We need to prove that where Definitions 2 and 3 coincide syntactically they also coincide semantically.

Theorem 1: Let p be a prime, n a natural number and $k=p^n$. If the sequence $x_0, x_1, \cdots, x_{k-1}$ is a permutation of the elements of GF(k) and if F(x) is a polynomial of degree n ($0 \leq n \leq k-1$), then when $j \leq n$, $[xx_1 \cdots x_j]$ is a polynomial q(x) of degree n-j and $q(x_0) = [x_0 x_1 \cdots x_j]$; and when $n < j$, $[xx_1 \cdots x_j] = 0$.

Proof: The proof is by induction on j. If j = 1, there are two cases.

Case 1, $1 = j \leq n$. By Lemma 2 there is a q(x) of degree n-1 as required. Since

$$q(x) = \frac{F(x) - F(x_1)}{x - x_1}$$ letting $x = x_0$ gives

$$q(x_0) = \frac{F(x_0) - F(x_1)}{x_0 - x_1} = [x_0 x_1],$$ as required.

Case 2, $n < j = 1$, that is, $n = 0$. If the degree of F(x) is 0, then $F(x) = c \in GF(k)$, a constant. From the definition, $[xx_1] = \frac{c - c}{x - x_1} = 0$, and the theorem is true.

Let $2 \leq j$. As the induction hypothesis assume the theorem to be true for differences of order j-1. There are again two cases.

Case 1, $j \leq n$. Consider the expression

$$[xx_1 \cdots x_j] = \frac{[xx_1 \cdots x_{j-1}] - [x_1 x_2 \cdots x_j]}{x - x_j}$$

Since $2 \leq j \leq n$, we have $1 \leq j-1 \leq n-1$. By the induction hypothesis $[xx_1 \cdots x_{j-1}]$ is a polynomial of degree $n-j+1 \geq 1$. Let $q_1(x) = [xx_1 \cdots x_{j-1}] - [x_1 x_2 \cdots x_j]$, also of degree $n-j+1$. By the Euclidean algorithm there exists a polynomial $q(x)$ of degree $n-j$ and a constant $r \in GF(k)$ such that $q_1(x) = (x - x_j)q(x) + r$. Since r is a constant, let $x = x_j$. Then $q_1(x_j) = r$. But also, $q_1(x_j) = [x_j x_1 \cdots x_{j-1}] - [x_1 \cdots x_{j-1} x_j]$. By the corollary above the two terms on the right are equal and $r = 0$. Therefore $q(x)$ is the unique polynomial value of $[xx_1 \cdots x_j]$. Since $x_0, x_1, \ldots, x_j$ are distinct

$$q(x_0) = \frac{[x_0 x_1 \cdots x_{j-1}] - [x_1 \cdots x_j]}{x_0 - x_j} = [x_0 \cdots x_j].$$

Case 2, $n < j$. If $n = j-1$, then by the induction hypothesis $[xx_1 \cdots x_{j-1}]$ is a polynomial of degree $n-j+1 = j-1-j+1 = 0$, that is, a constant, say $[xx_1 \cdots x_j] = c$. If $x = x_j$, then $[x_j x_1 \cdots x_{j-1}] = c$, and by the Corollary, $[x_1 \cdots x_{j-1} x_j] = c$.

$$\text{So } [xx_1 \cdots x_j] = \frac{[xx_1 \cdots x_{j-1}] - [x_1 \cdots x_j]}{x - x_j}$$

$$= \frac{c - c}{x - x_j} = 0.$$

If $n > j-1$, then by the induction hypothesis the $(j-1)$th order differences are 0 and, as above with $c = 0$, $[xx_1 \cdots x_j] = 0$.

We know that for any finite field $GF(k)$, $k = p^n$, each function $F: E(k) \to E(k)$ may be represented by a polynomial over $GF(k)$ of degree less than or equal to $k-1$. We know that for each polynomial

of degree n the n+1 order differences and all higher order differences are 0. Hence, we know that for the polynomial representation of each function F, the k order differences are 0. (Differences of order higher than k are not defined.)

We can now prove the main result:

Theorem 2 (Newton's Interpolation Formula with Divided Differences): Let p be a prime, n a natural number, and $k=p^n$. If GF(k) is a finite field of k elements and if F: E(k)→ E(k) is a function, and if $x_1, x_2, \ldots, x_k$ is a permutation of the elements of GF(k), then F is given by the polynomial:

$$F(x) = F(x_1) + \sum_{i=1}^{k-1}(x - x_1)(x - x_2)\ldots(x - x_i)[x_1x_2\ldots x_{i+1}]$$

Proof: Reversing the order of the terms on the right side of Definition 3, we obtain:

$$[xx_1\ldots x_k] = -\frac{[x_1x_2\ldots x_k]}{x - x_k} + \frac{[xx_1\ldots x_{k-1}]}{x - x_k}$$

$$[xx_1\ldots x_{k-1}] = -\frac{[x_1x_2\ldots x_{k-1}]}{x - x_{k-1}} + \frac{[xx_1\ldots x_{k-2}]}{x - x_{k-1}}$$

$$[xx_1\ldots x_{k-2}] = -\frac{[x_1x_2\ldots x_{k-2}]}{x - x_{k-2}} + \frac{[xx_1\ldots x_{k-3}]}{x - x_{k-2}}$$

.

.

.

$$[xx_1x_2] = -\frac{[x_1x_2]}{x - x_2} + \frac{[xx_1]}{x - x_1}$$

$$[xx_1] = -\frac{F(x_1)}{x - x_1} + \frac{F(x)}{x - x_1}$$

Note that in each line except the last the numerator of the second term on the right is the left side of the succeeding line. Repeated substitutions yield:

$$[xx_1x_2\ldots x_k] = -\frac{[x_1x_2\ldots x_k]}{x - x_k} - \frac{[x_1x_2\ldots x_{k-1}]}{(x - x_k)(x - x_{k-1})}$$

$$- \frac{[x_1x_2\ldots x_{k-2}]}{(x - x_k)\ (x - x_{k-1})\ (x - x_{k-2})}$$

$$- \ldots$$

$$- \frac{[x_1x_2]}{(x - x_k)\ldots(x - x_2)}$$

$$- \frac{F(x_1)}{(x - x_k)\ldots(x - x_1)}$$

$$+ \frac{F(x)}{(x - x_k)\ \ldots(x - x_1)}$$

This may be rewritten:

$$F(x) = F(x_1) + (x-x_1)\ [x_1x_2] + (x-x_1)\ (x-x_2)\ [x_1x_2x_3]$$

$$+ \ldots$$

$$+ (x-x_1)(x-x_2)\ldots(x-x_{k-1})\ [x_1x_2\ldots x_k]$$

$$+ (x-x_1)(x-x_2)\ldots(x-x_k)\ [xx_1x_2\ldots x_k]$$

Inspecting the last term, we note that the product $(x-x_1)(x-x_2)\ldots(x-x_k)$ is a permutation of the product $Z_k(x)$ of Definition 1. $Z_k(x) = 0$ for all x in GF(k). Further, $[xx_1 \ldots x_k]$ is a k order difference of F, a polynomial of degree at most k-1. Hence $[xx_1\ldots x_k]$ is 0, that is, the last term is 0; the theorem is true.

Example 1:- Let k=5 and let F and its differences be given in the table:

x	F(x)				
0	1				
		2			
1	3		1		
		4		0	
2	2		1		0
		1		0	
3	3		1		
		3			
4	1				

Then $F(x) = 1 + 2x + x(x-1) = 1 + 2x + x(x+4) = 1 + x + x^2$

Example 2:- Let k=5 and let V_3 and its differences be given in the table:

x	$V_3(x)$				
0	0				
		0			
1	0		0		
		0		1	
2	0		0		4
		1		2	
3	1		4		
		4			
4	0				

Then $V_3(x) = x(x-1)(x-2) + 4x(x-1)(x-2)(x-3)$

$= 4x^4 + 2x^3 + x^2 + 3x$

Example 3:- The same as Example 2, but permute the order of the values of x.

x	$V_3(x)$				
0	0				
		0			
2	0		0		
		0		0	
4	0		0		4
		0		1	
1	0		2		
		3			
3	1				

Then $V_3(x) = 4x(x-2)(x-4)(x-1) = 4x(x+1)(x+3)(x+4)$

$= 4x^4 + 2x^3 + x^2 + 3x.$

Example 4:- Let $k = 4$ and let $V_2(x)$ and its differences be given in the table:

x	$V_2(x)$			
0	0			
		0		
1	0		1	
		2		1
2	1		2	
		1		
3	0			

Then $V_2(x) = x(x-1) + x(x-1)(x-2) = x(x+1) + x(x+1)(x+2)$

$= x(x+1)(1+x+2)$

$= x(x+1)(x+3)$

$= x^3 + 2x^2 + 3x$

IV. Functions of Two Variables

The notation of Definitions 2 and 3 does not lend itself to generalization to the case of a function of more than one variable. We introduce here a notation closer to that which has developed for the differential calculus.

Let $F: E^2(k) \to E(k)$ and let $x_0, x_1, \ldots, x_{k-1}$ and $y_0, y_1, \ldots, y_{k-1}$ be two (not necessarily distinct) permutations of $E(k)$.

Definition 4:-

$$D_x^1 F(x_i, y_j) = \frac{F(x_i, y_j) - F(x_{i+1}, y_j)}{x_i - x_{i+1}}$$

$$D_y^1 F(x_i, y_j) = \frac{F(x_i, y_j) - F(x_i, y_{j+1})}{y_j - y_{j+1}}$$

$$D_x^p F(x_i, y_j) = \frac{D_x^{p-1} F(x_i, y_j) - D_x^{p-1} F(x_{i+1}, y_j)}{x_i - x_{i+p}}$$

$$D_y^p F(x_i, y_j) = \frac{D_y^{p-1} F(x_i, y_j) - D_y^{p-1} F(x_i, y_{j+1})}{y_j - y_{j+p}}$$

By analogy, we have in the one variable case:

$$D_x^1 F(x_i) = [x_i x_{i+1}], \text{ and}$$

$$D_x^p F(x_i) = [x_i x_{i+1} \cdots x_{i+p}].$$

By analogy to the differential calculus, one may prove that $D_x^p D_y^q F(x_i, y_j) = D_y^q D_x^p F(x_i, y_j)$ and also develop formulae evocative of the sum, product, and quotient formulae of the differential calculus.

Finally, we may reformulate Newton's Theorem in terms of a function of two variables:

Theorem 3:- Let p be a prime, n a natural number, and let $k=p^n$.

If GF(k) is a finite field of k elements and if $F: E^2(k) \to E(k)$ is a function, and if $x_1, x_2, \ldots, x_k$ and $y_1, y_2, \ldots, y_k$ are two permutations of E(k), then F is given by the polynomial:

$$F(x,y) = F(x_1,y_1) +$$

$$\sum_{i=1}^{k-1} D_x^i F(x_1,y_1)(x-x_1)\ldots(x-x_i) +$$

$$\sum_{i=1}^{k-1} D_y^i F(x_1,y_1)(y-y_i)\ldots(y-y_i) +$$

$$\sum_{i,j=1}^{k-1} D_x^i D_y^j F(x_1,y_1)(x-x_1)\ldots(x-x_i)(y-y_1)\ldots(y-y_j).$$

Multiplication of the terms yields the form of (3), namely:

$$F(x,y) = \sum_{i,j=0}^{k-1} f(i,j)x^i y^j$$

For a function of n variables, say $z_1, z_2, \ldots, z_n$, one defines n difference operators, $D^i_{z_1}, D^i_{z_2}, \ldots, D^i_{z_n}$. The general case of Newton's Formula consists of s constant followed by n summations, each involving one variable and its associated difference operator, followed by n(n-1)/2 double summations each involving two variables and their associated difference operators, and continuing to the n-fold summation involving all n variables and the n associated difference operators.

The organization on paper of difference tables is not as easy in the two variable situation, but the tables may be conveniently implemented in a high level computer language with the capacity for defining recursive functions.* In order to be reasonably efficient one needs to save the values of differences as they are computed.

Each of five functions was defined by a polynomial over GF(9) and over GF(16). The last function is defined by:

$$\text{Order}(x,y) = \begin{cases} 1, & \text{if } x < y; \\ 0, & \text{if } x = y; \\ -1, & \text{if } x > y. \end{cases}$$

*The Appendix to this paper contains the source text of PL/I procedures which demonstrate this fact.

(Here, as usual, the high integers are taken to be negative.) The table below indicated the number of terms in each polynomial definition. The columns labelled UE are the number of terms in the polynomial in its unexpanded form, that is, in the form given by Theorem 3; the columns marked E are the number of terms in the polynomial in the form (3). The reader should note that a polynomial over GF(9) might have 81 terms while a polynomial over GF(16) might have 256 terms.

In the expanded form (3) the representation of the function is unique. In the form given in Theorem 3, the representation depends upon the particular permutation of the elements of E(k) chosen for $x_1, \ldots, x_k$ and $y_1, \ldots, y_k$.

Table

Function	GF(9)		GF(16)	
	UE	E	UE	E
x+y (mod k)	17	18	124	124
x*y (mod k)	17	25	129	174
x+y (mod k-1)	42	69	134	233
x*y (mod k-1)	50	48	161	206
Order (x,y)	57	55	184	163

V. Comparison of the Methods

Faced with two different methods for computing the coefficients of the polynomial expansion of a function, it is reasonable to wonder whether there is a clear preference between the two from the point of view of the amount of computation involved. We consider here the case of a function F(x) of one variable.

The method of Benjauthrit and Reed (hereinafter called the "B-R method") leads directly to the form $\Sigma f(i)x^i$, while Newton's method arrives there via the expansion of $\Sigma b_i(x-x_1)(x-x_2)\ldots(x-x_i)$.

In the case of the B-R method, for the calculation of each f(i) one needs the set $\{\gamma^i \mid \gamma \neq 0, \gamma \varepsilon GF(k)\}$. For each $\gamma \neq 1$ the development of the set $\{\gamma^2, \gamma^3, \ldots, \gamma^{k-1}\}$ requires k-2 multiplications and so the development of the whole table of powers requires $(k-2)^2$ operations.

Armed with such a table of powers each coefficient f(i) is the sum of (k-1) terms, each term has two operations: a subtraction and a division, and there are (k-1) such terms. Thus the B-R method requires $(k-2)^2 + (k-1)\ (2(k-1) + k-2)$ operations, that is, $4k^2 - 11k + 8$ operations.

In the case of Newton's method, one first develops the difference table. The table contains k(k-1)/2 entries, each of which involves three operations. This form, requiring 3k(k-1)/2 operations is the form in which one would probably choose to implement a circuit, rather than modifying it to obtain the normal form. Should one wish the normal form, one would expand the polynomial:

$$b_0 +$$

$$b_1(x-x_1) +$$

$$b_2(x-x_1)(x-x_2) +$$

$$\ldots +$$

$$b_{k-1}(x-x_1)(x-x_2)\ldots(x-x_{k-1}).$$

Obtaining the product $(x-x_1)(x-x_2) = (x^2 - (x_1+x_2)x + x_1x_2)$, requires two operations. Multiplication of that product by $(x-x_3)$ requires four operations, and in general, development of the $(i+1)^{th}$ row from the i^{th} row requires $2i$ operations. Thus the expansion of the above polynomial requires $2 + 4 + \ldots + 2(k-2) = (k-1)(k-2)$ operations. The i^{th} row then has the form: $b_i(x^i + c_{i-1}x^{i-1} + \ldots + c_1x + c_0)$. This row expansion requires i operations, and therefore the expansion of the whole polynomial requires $1 + 2 + \ldots + (k-1) = k(k-1)/2$ operations. This results in the form:

$$d_{00} +$$

$$d_{10} + d_{11}x +$$

$$d_{20} + d_{21}x + d_{22}x^2 +$$

$$\ldots$$

$$d_{(k-1)0} + d_{(k-1)1}x + \ldots + d_{(k-1)(k-1)}x^{k-1}$$

Summation of these columns, to obtain the final coefficients, requires $(k-1) + (k-2) + \ldots + 2 + 1 = k(k-1)/2$ operations. Thus the expansion of the polynomial requires $(k-1)(k-2) + k(k-1) = 2(k-1)^2$ operations, and so Newton's method requires $3k(k-1)/2 + 2(k-1)^2 = (7k^2 - 11k + 4)/2$ operations.

Thus, for the B-R method, the coefficient of the k^2 term is 4, whereas for Newton's method the coefficient of the k^2 term is 7/2 if one normalizes the polynomial and 3/2 if one does not. This gives an advantage to Newton's method.

There is a practical situation in which Newton's method has a marked advantage. It is the situation in which one wishes to represent as a polynomial a function which is undefined at certain points, that is, for which at certain points we have "don't care". In the B-R method one could only assign some arbitrary value at these points and calculate the coefficients. In the case of Newton's method one merely uses the points at which the function is defined to develop a partial difference table, assumes that all further differences are zero, and produces a polynomial of minimal degree which fits the function at the defined points.

Suppose that S is the q-subset of E(k) upon which a partial function $F: E(k) \to E(k)$ is defined. Let $x_1, x_2, \ldots, x_q$ be the points of S and let $x_{q+1}, \ldots, x_k$ be the points of E(k) - S. Assume all undefined differences to be 0. By this F is represented as a polynomial of degree q-1 over E(k).

Example: Suppose that over GF(7),

$$F(x) = \begin{cases} 1, & \text{if } x = 1, 3; \\ 6, & \text{if } x = 4, 5. \end{cases}$$

Elsewhere F(x) is undefined. The table of differences is:

x	F(x)	[ab]	[abc]	[abcd]
1	1			
		0/2=0		
3	1		5/3=4	
		5/1=5		4/4=1
4	6		2/2=1	
		0/1=0		
5	6			

The function is defined as $F(x) = 1 + 4(x-1)(x-3) + (x-1)(x-3)(x-4)$ $= 1 + 4(x=6)(x+4) + (x+6)(x+4)(x+3)$.

This technique can be coupled with a decomposition technique for functions of several variables. Specifically, if $S \subset E^2(k)$ and F is a partial function on E(k), $F: S \to E(k)$, then let $S_x = \{x: (x,y) \in S\}$ and let $x_1, x_2, \ldots, x_q$ be some permutation of the points of S_x. Define the q partial functions

$$g_i(y) = F(x_i, y).$$

Each of these is a partial function of y defined upon some subset of E(k). By the technique sketched above, each g_i may be expressed as a polynomial in y. These form the difference table:

x	F(x)	[ab]	[abc]	...
x_1	$g_1(y)$			
		$[g_1g_2]$		
x_2	$g_2(y)$		$[g_1g_2g_3]$	
		$[g_2g_3]$		
x_3	$g_3(y)$			
.	.			
.	.			
.	.			

By Theorem Two we have:

$$F(x,y) = g_1(y) + [g_1g_2](x-x_1) + [g_1g_2g_3](x-x_1)(x-x_2) + \ldots$$

$$[g_1g_2\cdots g_q]\ (x-x_1)(x-x_2)\ldots(x-x_{q-1}).$$

Example: Suppose that over GF(4) max is given by the usual table:

	0	1	2	3
0	0	1	2	3
1	1	1	2	3
2	2	2	2	3
3	3	3	3	3

and suppose one wishes to make use of the symmetry in the table to simplify the resulting polynomial. By substituting the simple symmetric functions $z = x+y$ and $w = xy$, one obtains the table:

		w			
		0	1	2	3
	0	0	1	3	2
z	1	1	3		
	2	2			3
	3	3		2	

wherein the blank entries are not defined.

Except for the zero row and the zero column, each row and each column can be represented as a linear function, since each contains only two entries. The zero column is clearly also linear.

Representing reach column as a polynomial in z, we have the difference table:

w	F(w)	[ab]	[abc]	[abcd]
0	z			
		3z+1		
1	2z+1		(3z+2)/2 = 2z+1	
		2/3=3		z/3=2z
2	2z+3		(z+2)/2 = 3z+1	
		z+1		
3	3z+2			

Hence, $F(z,w) = z + (3z+1)w + (2z+1)w(w+1) + 2zw(w+1)(w+2)$

$= 2zw(w^2+2w+1) + w^2 + z.$

Example: Suppose that over GF(3) we have the function of four variables: F(x,y,z,w) = y if x = 0; = z if x = 1; = w if x = 2.

x	F(x)	[ab]	[abc]
0	y		
		z-y=z+2y	
1	z		(w+z+y)/2
		w-z=w+2z	
2	w		

and so, F(x,y,z,w) = y + x(2y+z) + 2x(x+2)(y+z+w).

Other decomposition techniques can be used in conjunction with these divided difference methods.

Appendix

The procedures below, coded in PL/1, are examples of procedures to calculate differences with respect to a variable x of a function F(x,y) and the mixed differences of F(x,y). The procedure Dx has three parameters: the order of the difference, the x coordinate of F, and the y coordinate of F. The procedure calculates the difference with respect to x, y being fixed, and places the difference into the proper position in a global table X-diff. The procedure makes use of the following global identifiers:

X-diff; a three dimensional array in which differences calculated are stored; the entries of this array are initialized at compile time to -1.

Quot; a procedure which performs division over the Galois Field being used.

Diff; a procedure which performs subtraction over the Galois Field being used.

X: a linear array containing a permutation of the elements of the space being used.

Y: a linear array containing a permutation of the elements of the space being used.

```
Dx:  procedure (order, X-ind, Y-ind) returns (fixed binary) recursive;
     dcl( order, X-ind, Y-ind) fixed binary;
     If X-diff(order, X-ind, Y-ind) ¬= -1
              then return (X-diff(order, X-ind, Y-ind));
     If order = 1 then do·
                         X-diff(order, X-ind, Y-ind) =
                              Quot(Diff(F(X(X-ind), Y(Y-ind)),
                                        F(X(X-ind + 1), Y(Y-ind))),
                                   Diff(X(X-ind), X(X-ind)));
                         return(X-diff(order, X-ind, Y-ind));
                         end;
                    else X-diff(order, X-ind, Y-ind) =
                              Quot(Diff(Dx(order - 1, X-ind, Y-ind),
                                        Dx(order - 1, X-ind + 1, Y-ind)),
                                   Diff(X(X-ind), X(X-ind + order)));
                    return(X-diff(order,  X-ind, Y-ind));
end Dx;
```

A procedure Dy to calculate differences with respect to y is the exact analog of the above procedure.

The procedure Dxy below calculates the mixed differences of a function $f(x,y)$. This procedure invokes both the procedures Dx and Dy and it stores the differences in a table XY-diff.

```
Dxy: procedure (X-ord, Y-ord, X-ind, Y-ind)
                returns(fixed binary) recursive;
     dcl (X-ord, Y-ord, X-ind, Y-ind) fixed binary;
     If XY-diff(X-ord, Y-ord, X-ind, Y-ind) ¬= -1
                & X-ord ¬= 0 & Y-ord ¬= 0
                then return (XY-diff(X-ord, Y-ord, X-ind, Y-ind));
     If Y-ord = 0 then return (Dx(X-ord, X-ind, Y-ind));
     If X-ord = 0 then return (Dy(Y-ord, X-ind, Y-ind));
     XY-diff(X-ord, Y-ord, X-ind, Y-ind) =
            Quot(Diff(Dxy(X-ord, Y-ord - 1, X-ind, Y-ind),
                      Dxy(X-ord, Y-ord - 1, X-ind, Y-ind + 1)),
                 Diff(Y(Y-ind), Y(Y-ind + Y-ord)));
     return (XY-diff(X-ord, Y-ord, X-ind, Y-ind));
end Dxy;
```

References

1.) Boonsieng Benjauthrit and Irving S. Reed, "Galois Switching Functions and Their Applications", IEEETC, C-25, no. 1 (January 1976) pp. 78-86.

2.) Tomlinson Fort, Finite Differences, (Oxford: Clarendon Press, 1948).

3.) Charles Jordan, Calculus of Finite Differences, (New York: Chelsa Publishing Company, 1947).

4.) L.M. Milne-Thompson, The Calculus of Finite Differences, (London: Macmillan and Sons, 1961).