MATHEMATICS OF COMPUTATION, VOLUME 25, NUMBER 114, APRIL, 1971

The Fast Fourier Transform in a Finite Field

By J. M. Pollard

Abstract. A transform analogous to the discrete Fourier transform may be defined in a finite field, and may be calculated efficiently by the 'fast Fourier transform' algorithm. The transform may be applied to the problem of calculating convolutions of long integer sequences by means of integer arithmetic.

1. Introduction and Basic Properties. Let GF(p^n), or F for short, denote the Galois Field (Finite Field) of p^n elements, where p is a prime and n a positive integer. Let d be a divisor of $p^n - 1$ (possibly $d = p^n - 1$), and r be a member of F of order d in the multiplicative group, F^* say, of the nonzero elements of F (which certainly exists, since this group is cyclic of order $p^n - 1$, [1, p. 125]). Then one can define the transform of a sequence (a_i) $(0 \leqq i \leqq d-1)$ of members of F to be the sequence (A_i) where

$$A_i = \sum_{i=0}^{d-1} a_i r^{ij}. \tag{1}$$

The transformed sequence (A_i) depends on the choice of r, which will be considered fixed throughout.

The inverse transform to (1), (that is, an equivalent set of equations giving the (a_i) in the terms of the (A_i)) is

$$a_i = -d' \cdot \sum_{i=0}^{d-1} A_i r^{-ij}, \tag{2}$$

where d' is the integer for which

$$d'd = p^n - 1. \tag{3}$$

The transform (1) and its inverse (2) may be calculated by the 'fast Fourier transform' (FFT) algorithm ([2], [3]), of which many versions have been described in detail (see [4] for other references). These authors were concerned with the calculation of the 'discrete Fourier transform' of sequences of complex numbers (see Section 7), but these algorithms are equally applicable to the present case; the roots of unity

$$e(k/d) = e^{2\pi ik/d}, \qquad (k \text{ an integer}),$$

are replaced by r^k throughout, and the operations of complex addition and multiplication become the corresponding operations in F. The transform (1), like the 'conventional' FFT, is simplest to perform when the integer d is highly composite (the product of many small factors, e.g. $d = 2^m$), but more complicated versions cope with any

Received February 24, 1970, revised November 18, 1970.

AMS 1970 *subject classifications.* Primary 42A68, 12C15, 10A40, 10L99.

Key words and phrases. Finite field, fast Fourier transform, residue arithmetic, primitive polynomials, Mersenne primes.

Copyright © 1971, American Mathematical Society

d ([5] and Section 8). In any case, the total number of operations (in F or in the complex field) is reduced to $O(d \cdot \log d)$ from the $O(d^2)$ operations required to calculate these transforms in the most obvious way. In the case of (1), this estimate assumes that a suitable r has already been found, and the powers r^i $(0 \leqq i \leqq d-1)$ computed (see Section 5). The principal reason for interest in the transform (1) lies in the following 'convolution property'. Suppose that three pairs of sequences, (a_i) and (A_i), (b_i) and (B_i), (c_i) and (C_i) $(0 \leqq i \leqq d-1)$ form transform pairs (i.e. are related by (1)) and that

$$C_i = A_i B_i, \qquad (0 \leqq i \leqq d-1). \tag{4}$$

Then,

$$c_i = \sum_{\substack{j=0 \\ j+k \equiv i \pmod d}}^{d-1} \sum_{k=0}^{d-1} a_j b_k, \qquad (0 \leqq i \leqq d-1). \tag{5}$$

If we extend the definition of (b_i) to all i by making the sequence periodic with period d, (5) may be written

$$c_i = \sum_{j=0}^{d-1} a_j b_{i-j}. \tag{5A}$$

Thus, the calculation of the 'cyclic convolution' of the sequences (a_i) and (b_i), as defined by (5), may be obtained by transforming the sequences, multiplying the results term-by-term as in (4), and performing the inverse transform (2). The corresponding process for sequences of complex numbers is well known (see [4]).

Another simple property of (1) is that

$$\sum_{j=0}^{d-1} A_j^2 = d \cdot \sum_{i=0}^{d-1} a_i a_{d-i}, \tag{6}$$

and, similarly, that

$$\sum_{i=0}^{d-1} a_i^2 = (-d') \cdot \sum_{j=0}^{d-1} A_j A_{d-j}. \tag{7}$$

The properties expressed in the equations (1) to (7) will be proved in the next section. These properties are analogous to well-known properties of the discrete Fourier transform; in fact, these properties hold, more generally, in any field F, finite or not, provided that r has finite order d in the multiplicative group F^* of F.

2. Proof of the Basic Properties. We prove first that for any integer k,

$$\sum_{i=0}^{d-1} r^{ik} = d, \quad \text{if} \quad k \equiv 0 \pmod d, \tag{8}$$

$$= 0, \quad \text{otherwise.}$$

(8) is trivially true when $k \equiv 0 \pmod d$ since each term on the left is unity; when $k \not\equiv 0 \pmod d$ we use the relation

$$\left(\sum_{i=0}^{d-1} r^{ik}\right)(1 - r^k) = 1 - r^{dk} = 0$$

from which (8) follows since $1 - r^k \neq 0$.

To show that (1) implies (2), we simplify the right side of (2) using (1), (3) and (8), as

$$-d' \cdot \sum_{i=0}^{d-1} r^{-ij} \cdot \sum_{l=0}^{d-1} a_l r^{il} = (-d') \cdot \sum_{l=0}^{d-1} a_l \left(\sum_{i=0}^{d-1} r^{i(l-j)} \right)$$
$$= -d' \cdot a_j \cdot d = a_j.$$

The proof that (2) implies (1), and the proof of the other properties expressed in (4)–(7) are equally simple deductions from (8).

3. Applications of the Transform.

(i) *Multiplication of Polynomials over* GF(p^n). Let the polynomials to be multiplied be

$$f(x) = \sum_{i=0}^{m_1} a_i x^i, \qquad g(x) = \sum_{i=0}^{m_2} b_i x^i, \quad ((a_i), (b_i) \in \mathrm{GF}(p^n)), \tag{9}$$

where $m_1 \geqq m_2$. The product is

$$h(x) = \sum_{i=0}^{m_1+m_2} e_i x^i,$$

where

$$e_k = \sum_{0 \leqq i \leqq m_1 ; 0 \leqq j \leqq m_2 ; i+j=k} a_i b_j, \qquad (0 \leqq k \leqq m_1 + m_2). \tag{10}$$

Assume first that $(p^n - 1)$ is highly composite and $m_1 + m_2 < p^n - 1$. Choose a divisor d of $(p^n - 1)$ for which $d > m_1 + m_2$ and extend the definitions of (a_i), (b_i) by setting

$$a_i = 0, \qquad (m_1 < i \leqq (d-1)), \qquad b_i = 0, \qquad (m_2 < i \leqq (d-1)).$$

The 'convolution algorithm' of Section 1 with $F = \mathrm{GF}(p)$ applied to these extended sequences gives the required (e_k); more precisely the (c_k) so obtained satisfy the equations,

$$c_k = e_k, \qquad (0 \leqq k \leqq m_1 + m_2), \qquad c_k = 0, \qquad (m_1 + m_2 < k \leqq d-1).$$

If $m_1 + m_2 \geqq p^n - 1$ or $p^n - 1$ is not highly composite it may be feasible to operate in an extension field, $F = \mathrm{GF}(p^{ln})$, say, (l a positive integer), of GF(p^n); this requires that $p^{ln} - 1$ be highly composite and $m_1 + m_2 < p^{ln} - 1$. The conditions can certainly be met for the important practical case $p = 2$, $n = 1$ (e.g. [6], [15]). Otherwise, see Section 3(ii) and Section 8.

(ii) *Multiplication of Integer Polynomials* (*or Convolution of Integer Sequences*). We require to calculate the (e_k) defined by (10) where the (a_i), (b_i) are now ordinary integers. An upper bound is needed for the members of these sequences, so suppose that

$$|a_i| \leqq M_1, \qquad (0 \leqq i \leqq m_1), \qquad |b_i| \leqq M_2, \qquad (0 \leqq i \leqq m_2), \tag{11}$$

so that, by (10)

$$|e_k| \leqq m_2 M_1 M_2 \qquad (= L, \text{ say}).$$

Let p be a prime and d a highly composite integer such that $d \mid (p-1)$, $d > (m_1 + m_2)$ and $p > 2L$; then

$$|e_k| \leqq L < p/2. \tag{12}$$

Applying the process of Section 3(i) with $F = \mathrm{GF}(p)$, we obtain a sequence (c_i) with

$$c_i \equiv e_i \pmod{p}, \qquad (0 \leqq k \leqq m_1 + m_2),$$

so that, by (12), the least absolute residue (mod p) of c_k $(0 \leqq k \leqq m_1 + m_2)$ is e_k.

It is possible to replace the one possibly very large prime p required by a number of smaller primes. Choose pairs

$$(p_i, d_i), \qquad (1 \leqq i \leqq t, \text{ say})$$

of primes p_i and highly composite integers d_i with

$$d_i > m_1 + m_2, \qquad (1 \leqq i \leqq t),$$

$$p_i \equiv 1 \pmod{d_i}, \qquad (1 \leqq i \leqq t)$$

and

$$P = p_1 p_2 \cdots p_t > 2L.$$

By applying the previous algorithm with each pair (p_i, d_i) in turn, one obtains the residues of each e_k to each modulus p_i; from these, one can find the least absolute residues of the (e_k) (mod $P = p_1 \cdots p_t$) by the 'Chinese Remainder Theorem' ([8, p. 94], [9, p. 27]) and, since by (12) $|e_k| < P/2$, these are again the required integer values.

(iii) *The Multiplication of Very Large Integers.* Suppose the integers are expressed to the base u as

$$f = \sum_{i=0}^{m_1} a_i u^i, \qquad g = \sum_{i=0}^{m_2} b_i u^i, \quad (0 \leqq a_i < u, 0 \leqq b_i < u).$$

Their product is

$$h = \sum_{i=0}^{m_1+m_2} e_i u^i,$$

where the (e_i), defined by (10), do not necessarily satisfy $0 \leqq e_i < u$. So, after obtaining the (e_i) by the method of Section 3(ii), it remains to perform some carrying to obtain h in standard form; but with a suitable choice of parameters, this is a relatively trivial operation. This algorithm is applicable to Lucas' method for testing the primality of Mersenne numbers, $M_p = 2^p - 1$, p prime ([8, p. 16 and p. 223], [10]).

As an example, I give some details of a method of multiplying two integers of 10,000 bits (binary digits) in a digital computer of word length 24 bits. I take $u = 2^{21}$ so that m_1 and m_2 are less than $2^9 = 512$. In Section 3(ii), I take $t = 3$, $d_1 = d_2 = d_3 = 2^{10} = 1024$ and so require three primes p_1, p_2, p_3 satisfying

$$2^{21} < p_1 < p_2 < p_3 < 2^{23}, \quad p_i \equiv 1 \pmod{2^{10}};$$

for example $p_1 = 6946817$, $p_2 = 7340033$, $p_3 = 7667713$ which in fact satisfy $p_i \equiv 1 \pmod{2^{16}}$. The condition $p_1 p_2 p_3 > 2L$ is seen to be easily satisfied.

Using these values, the algorithm has been programmed in machine language for the ICL 4120 computer. The multiplication of two different integers takes 45 seconds while a straightforward 'long multiplication' algorithm, using all 24 bits of each word, takes 95 seconds. For the problem of squaring a single integer (the main

part of one step of Lucas' algorithm for a prime p around 10,000), the times are 30 and 48 seconds, respectively. In this case, we perform two transforms to each prime modulus instead of three, while 'long multiplication' requires just half as many operations as before. In absolute terms, these times are slow by modern standards [10].

It is probable that the highest speeds would be obtained in a special purpose computer, taking advantage of the possibility of extensive parallel computation in the transforms. Such computers have been built for the conventional transform [4]. But, of course, it is possible to use parallel computation for the long multiplication process also.

(iv) *Division of Polynomials Over* GF(p). Let (a_i) $(0 \leqq i \leqq d-1)$ be a sequence in the finite field $F = \mathrm{GF}(p^n)$, where $d \mid p^n - 1$, and let $f(x) = \sum_{i=0}^{d-1} a_i x^i$.

Then, Eq. (1) could be written

$$A_j = f(r^j), \qquad (0 \leqq j \leqq d-1).$$

So, what I have termed a fast Fourier transform in F might be regarded simply as 'fast evaluation of a polynomial over F'.

If $d = p^n - 1$, so that every nonzero element of F is of the form r^j $(0 \leqq j \leqq d-1)$, it follows that the condition that the transformed sequence (A_j) contains no zeros is just that $f(x)$ has no roots in F, except perhaps for $x = 0$; while if $d < p^n - 1$ then this condition is sufficient but not necessary.

This remark is used in the following division algorithm. Let

$$f(x) = \sum_{i=0}^{m_1} a_i x^i \quad \text{and} \quad h(x) = \sum_{i=0}^{m_1+m_2} e_i x^i, \qquad (m_1 > 1,\ m_2 \geqq 1),$$

be polynomials over $F = \mathrm{GF}(p)$, $f(x)$ being given to be irreducible. We wish to find whether

$$f(x) \mid h(x), \tag{13}$$

and if so, to find the quotient, say

$$g(x) = \sum_{i=0}^{m_2} b_i x^i. \tag{14}$$

Choose integers $n \geqq 1$ and $d \geqq 1$ (and highly composite) such that

$$d > m_1 + m_2, \qquad d \mid p^n - 1, \qquad m_1 \nmid n,$$

extend the sequences (a_i), (e_i) to d terms (i.e., $0 \leqq i \leqq d-1$) by the addition to zeros (as in Section 3(i)) and apply the transform (1) in the field $F' = \mathrm{GF}(p^n)$ to obtain sequences (A_i), (E_i), $(0 \leqq i \leqq d-1)$. Since $m_1 \nmid n$, the field F' does not contain $\mathrm{GF}(p^{m_1})$, the splitting field of $f(x)$ (see [1, Chapter IV]), so F' contains no roots of $f(x)$ and so $A_i \neq 0$, $(0 \leqq i \leqq d-1)$.

Define (B_i) by

$$B_i = E_i / A_i, \qquad (0 \leqq i \leqq d-1) \tag{15}$$

and perform the inverse transform (2) in F' to obtain (b_i), $(0 \leqq i \leqq d-1)$. If

$$b_i = 0, \qquad (m_2 < i \leqq d-1), \tag{16}$$

then (13) holds, (14) gives the quotient and $b_i \in \mathrm{GF}(p)$ $(0 \leqq i \leqq m_2)$; otherwise (13) is false.

Proof. Suppose first that (13) holds, with quotient (14); extend the sequence (b_i) to d terms by the addition of zeros. By the multiplication algorithm of Section 3(i), the transform (B_i) satisfies

$$A_iB_i = E_i, \qquad (0 \leqq i \leqq d-1).$$

Hence the division algorithm leads to a sequence (b_i) with the properties stated.

Now suppose that (16) holds; by defining $g(x)$ by (14), it follows from (15) that

$$f(x)\cdot g(x) = h(x), \qquad (\text{over } F'). \tag{17}$$

Suppose that some of (b_i) $(0 \leqq i \leqq m_2)$ are not in F. If b_K is the first such b, then by taking coefficients of x^{K+m_1} in (17)

$$b_Ka_{m_1} + b_{K+1}a_{m_1-1} + \cdots + b_{K+m}a_{m_1-m} = e_{K+m_1}, \qquad (m = \min(m_1, m_2 - K))$$

in which every element except b_K is in F and $a_{m_1} \neq 0$. This is a contradiction so $b_i \in F$ $(0 \leqq i \leqq m_2)$ as asserted.

4. The Transform in the Ring of Integers Modulo m. A version of the transform (1) can also be given in a particular ring which is not a field, namely the ring R_m of integers modulo a composite integer m. Suppose first that m is prime power, say $m = p^n$, $p > 2$, and that d is a divisor of $(p-1)$. The multiplicative group, R^* say, of the members of R_m prime to m is still cyclic and of order $p^{n-1}(p-1)$ divisible by d, and so it contains a number r of order d. The transform of a sequence of members of R_m can still be defined by (1) and the results of Section 1 still hold. To prove this it suffices to show that (8) still holds (as a congruence mod m).

In the nontrivial case $K \not\equiv 0 \pmod d$, we have now

$$\left(\sum_{i=0}^{d-1} r^{iK}\right)(1 - r^K) \equiv 1 - r^{dK} \equiv 0, \qquad (\text{mod } m = p^n),$$

and need to show that $r^K \not\equiv 1 \pmod p$.

The numbers r^i $(0 \leqq i \leqq d-1)$ form a subgroup of R^* of order d, and the members of R_m congruent to 1 (mod p) form a subgroup of order p^{n-1}; the orders of these subgroups being coprime, they can have only the unit element in common, which proves the assertion.

More generally, we can take $m = p_1^{a_1} \cdots p_t^{a_t}$, where $d \mid (p_i - 1)$ $(1 \leqq i \leqq t)$ and where r has order d (mod $p_i^{a_i}$) for each i $(1 \leqq i \leqq t)$ and hence also order d (mod m).

5. Practical Methods of Performing the Transform.

(i) *The Case F = GF(p).* The obvious method is to represent the elements of F by the integers $0, 1, \cdots, (p-1)$, and perform addition and multiplication followed by reduction (mod p). Alternatively, if p is not too large, one can replace the multiplications (mod p) by additions (mod $(p-1)$) and 'table look-up' operations by means of previously constructed tables of powers of a primitive root w (mod p), [11, p. 63]. This is well known in connection with 'residue-system' computers [9].

A suitable choice for r (of order d in F^*) is $r \equiv w^{d'} \pmod p$, where, as in Section 1, $d'd = p - 1$. One method to find a primitive root w is to test successively the positive integers by the following trivial theorem, which applies also to the case F = GF(p^n).

THEOREM. *The element w in $F = \mathrm{GF}(p^n)$ generates F^*, the multiplicative group of F, if and only if for each prime factor q of $p^n - 1$ (with $q < p^n - 1$) we have $w^{((p^n-1)/q)} \neq 1$.*

The application of the theorem can be made by a device discussed in [12, Section 3].

(ii) *The Case $F = \mathrm{GF}(p^n)$, where $n > 1$.* The following table 'look-up' method [7, pp. 37–50] is probably best, when possible, that is, when p^n is not too large. We take a generator w of F^* and represent the nonzero elements of F by their exponents when written as powers of w, with a special symbol for the zero element (say, $0 = w^*$). Multiplication in F is merely addition of exponents (mod $(p^n - 1)$); for the addition of nonzero elements w^a, w^b $(a \leqq b)$ we write

$$w^a + w^b = w^a(1 + w^{b-a}),$$

and then consult a table giving the solution x of

$$w^x = 1 + w^y,$$

for given y (where possibly $x = *$).

The use of a table of p^n entries can be avoided by the following less efficient method. Let $F[x]$ be the ring of polynomials in a single variable x over the field $\mathrm{GF}(p)$ and let $h(x) \in F[x]$ be irreducible and of degree n. The members of $F = \mathrm{GF}(p^n)$ can be represented by the residue classes (mod $h(x)$) of $F[x]$, and so can be written as

$$f(x) = \sum_{i=0}^{h-1} a_i x^i \qquad (a_i \in \mathrm{GF}(p)). \tag{18}$$

Addition in F is now straightforward; but multiplication necessitates forming the product in $F[x]$ of two polynomials of form (18) and then division by $h(x)$ to give a remainder of this form again.

It is possible to choose $h(x)$ in such a way that the element $w(x) = x$ is itself a generator of F^*, and so take

$$r(x) = \{w(x)\}^{d'} = x^{d'},$$

as the required element of order d. The suitable choices for $h(x)$, the 'primitive polynomials over F', are the irreducible factors in $F[x]$ of $\Phi_{p^n-1}(x)$, where $\Phi_m(x)$ denotes the cyclotomic polynomial

$$\Phi_m(x) = \prod_{k \mid m} (x^k - 1)^{\mu(m/k)},$$

[1, p. 132]. These polynomials are of interest in connection with 'shift register sequences', [6], [13]. An efficient test for a given $h(x) \in P[x]$, of degree n but not known to be irreducible is provided by the following theorem, which generalises that of [13].

THEOREM. *Let $h(x)$ be a polynomial of degree n $(\geqq 1)$ in $F[x]$. Then*

$$h(x) \mid \Phi_{p^n-1}(x), \quad (\textit{in } F[x])$$

if and only if

(a) $x^{p^n-1} \equiv 1 \pmod{h(x)}$, *in $F[x]$, and*

(b) *for each prime factor q of $p^n - 1$ with $q < p^n - 1$ (if any exist), the polynomials $h(x)$, $x^m - 1$, where $m = (p^n - 1)/q$, are relatively prime in $F[x]$.*

Proof. The necessity of the conditions (a) and (b) is trivial. Suppose these conditions hold and that $h_1(x)$ is an irreducible factor of $h(x)$ in $F[x]$. Then $h_1(x) \mid \Phi_{p^n-1}(x)$

which implies that $h_1(x)$ has degree n, and so $h(x) = k \cdot h_1(x)$ ($k \in F$, $k \neq 0$) and $h(x) \mid \Phi_{p^n-1}(x)$.

To apply (a), calculate the residue (mod $h(x)$) of $x^{(p^n-1)} - 1$ by the device mentioned in Section 5(i). For (b), first find the residue of $x^m - 1$ by the same means, then the greatest common divisor of this polynomial and $h(x)$ by Euclid's algorithm (for polynomials over a field, analogous to [12, Section 2]); the result should be a nonzero constant, i.e. member of F.

(iii) *Division in* GF(p^n), $n \geqq 1$. The division algorithm of Section 3(iv) requires the operation of dividing elements of $F = \mathrm{GF}(p^n)$, (Eq. (15)). By writing

$$a/b = a \cdot b^{-1}, \qquad (a, b \in F, b \neq 0),$$

the problem becomes that of computing inverses in F, discussed in [12], (this is for GF(p), but extends to any GF(p^n)); one method is to use

$$b^{-1} = b^{p^n-2} \qquad (b \in F, b \neq 0).$$

(iv) *Program Layout.* As with the conventional fast Fourier transform, it is simplest to divide the calculation of the transform (1) (or (2)) into two separate stages, a permutation and a calculation stage, which may be performed in either order [17]. But for most applications, including those of Section 3, it is sufficient to obtain the transformed sequence in 'digit-reversed' order and the permutation stage can therefore be omitted from the transforms, as has been done in convolution algorithms for real numbers.

6. Generalisation to k Dimensions. The transform (1) which has been considered up to this point, whether in GF(p) or GF(p^n) with $n > 1$, is to be regarded as 1-dimensional in that it acts on a 1-dimensional sequence (a_i) of elements of F. A generalisation to higher dimensions is possible.

Let $F = \mathrm{GF}(p^n)$ ($n \geqq 1$) be a finite field and $d_1, d_2, \cdots, d_k$ any series of divisors of $p^n - 1$. Let w be a generator of F^* and put

$$d_i' d_i = p^n - 1 \qquad (1 \leqq i \leqq k),$$

so that the elements $w^{d_i'}$ have order d_i in F^*. Then the transform of the k-dimensional matrix $(a_{i_1,\cdots,i_k})$ ($0 \leqq i_l \leqq d_l - 1$ for $l = 1, \cdots, k$) may be defined to be

$$A_{i_1,\cdots,i_k} = \sum_{j_1=0}^{d_1-1} \cdots \sum_{j_k=0}^{d_k-1} a_{j_1,\cdots,j_k} w^{(d_1' i_1 j_1 + \cdots + d_k' i_k j_k)}. \tag{19}$$

The inverse transform is

$$a_{j_1,\cdots,j_k} = \prod_{i=1}^{k} (-d_i') \cdot \sum_{i_1=0}^{d_1-1} \cdots \sum_{i_k=0}^{d_k-1} A_{i_1,\cdots,i_k} w^{-(d_1' i_1 j_1 + \cdots + d_k' i_k j_k)}, \tag{20}$$

$$(0 \leqq j_l \leqq d_l - 1 \text{ for } l = 1, \cdots, k).$$

The properties expressed by Eqs. (4), (5), (6), (7) have simple generalisations and permit the extension of the algorithms of Section 3 to 'k-dimensional convolutions', or multiplication of polynomials in k variables. Thus, the equations analogous to (4) and (5A) are

$$C_{i_1,\cdots,i_k} = A_{i_1,\cdots,i_k} B_{i_1,\cdots,i_k} \qquad (0 \leqq j_l \leqq d_l - 1 \text{ for } l = 1, \cdots, k) \tag{21}$$

and

$$c_{i_1,\cdots,i_k} = \sum_{j_1=0}^{d_1-1} \cdots \sum_{j_k=0}^{d_k-1} a_{j_1,\cdots,j_k} b_{i_1-j_1,\cdots,i_k-j_k}. \tag{22}$$

7. Comparison with the Conventional Discrete Fourier Transform. The k-dimensional form of this transform may be written

$$A_{i_1,\cdots,i_k} = \sum_{j_1=0}^{d_1-1} \cdots \sum_{j_k=0}^{d_k-1} a_{j_1,\cdots,j_k} e\left(\frac{i_1 j_1}{d_1} + \cdots + \frac{i_k j_k}{d_k}\right), \tag{23}$$

where $e(x) = e^{2\pi i x}$, and the (d_l) $(1 \leqq l \leqq k)$ are now any series of positive integers. The inverse transform is

$$a_{i_1,\cdots,i_k} = \left(\prod_{l=1}^{k} d_l\right)^{-1} \cdot \sum_{i_1=0}^{d_1-1} \cdots \sum_{i_k=0}^{d_k-1} A_{i_1,\cdots,i_k} \cdot e\left(-\frac{i_1 j_1}{d_1} - \cdots - \frac{i_k j_k}{d_k}\right). \tag{24}$$

The convolution property is again expressed by Eqs. (21), (22), where these equations are now, of course, ordinary equalities between complex numbers. Suppose this transform is applied to the problem of calculating convolutions of integer sequences exactly. In general, this is most simply done by conventional 'real arithmetic', in which the real and imaginary parts of the complex quantities are stored and handled to only a finite accuracy. This accuracy must be such that when the results of the convolution process are rounded to the nearest integer the true sums are obtained. However, in special cases this difficulty does not exist. Consider first the case

$$d_1 = \cdots = d_k = 2. \tag{25}$$

The exponential factors in (23) and (24) now both become

$$(-1)^{i_1 j_1 + \cdots + i_k j_k}.$$

The corresponding factors in (19) and (20) also reduce to this expression since the unique member of $F = \mathrm{GF}(p^n)$ of order 2 in F^* is (-1) (the choice (25) implies that p is odd to satisfy $d_i = 2 \mid (p^n - 1)$).

The transforms now require only additions and subtractions for their calculation, and so may be calculated faster and more simply than the general case. This is true for (23) even when the data is not integral (see [14]). But when it is integral then (23) and (24) are obtained by integer arithmetic and provide a simple means of calculating the convolution (22) exactly; so that the advantages of the finite field transform (19) vanish in this case (see [15, p. 29–32]). It is perhaps worth pointing out that the conventional transform (23) with integral data could in principle always be calculated by integer operations; suppose for simplicity that $d_1 = \cdots = d_k = d$, and put $\theta = e(1/d) = e^{2\pi i/d}$. θ is an algebraic number of degree $\phi = \phi(d)$ ([16, Chapter IV]), whose minimal polynomial is $\Phi_d(x)$ (see Section 5(ii)). The quantities to be manipulated during the evaluation of (23) and (24) are integer polynomials in θ, which may be written in the form

$$Z = \sum_{i=0}^{\phi-1} b_i \theta^i.$$

Multiplication of numbers in this form, however, consists of polynomial multiplication, followed by division by the monic polynomial $\Phi_d(\theta)$. This is a relatively

complicated process and renders the algorithm impracticable for all but small values of d. (Compare with Section 5(ii), where the polynomials have degree only n; here their order may be near p^n.)

8. Calculation of (1) When d is not Highly Composite. The device given in [5] for the discrete Fourier transform can be applied also to the case $F = \mathrm{GF}(p)$ of the transform (1). Considering for simplicity the 1-dimensional case (with the notation of Section 1), we must make the condition $2d \mid p - 1$, instead of $d \mid p - 1$ as previously. Then we can find $s \in F$ such that $s^2 = r$ and write (1) as

$$A_i = \sum_{j=0}^{d-1} a_j s^{2ij} = s^{-i^2} \cdot \sum_{j=0}^{d-1} a_j s^{-j^2} \cdot s^{(i+j)^2}.$$

This sum has the form of a noncyclic convolution on the sequences

$$a_k s^{-k^2}, \quad (0 \leqq k \leqq d-1), \qquad s^{k^2}, \quad (0 \leqq k \leqq 2d-1).$$

By representing the elements of F by the integers $0, 1, \cdots, (p-1)$, this convolution can be performed by the method of Section 3(ii).

Mathematics Laboratory
Plessey Telecommunications Research Ltd.
Taplow Court, Taplow, Nr. Maidenhead
Berkshire, England.

1. I. T. ADAMSON, *Introduction to Field Theory,* Oliver & Boyd, London; Interscience, New York, 1964. MR **33** #7325.
2. J. W. COOLEY & J. W. TUKEY, "An algorithm for the machine calculation of complex Fourier series," *Math. Comp.*, v. 19, 1965, pp. 297–301. MR **31** #2843.
3. W. M. GENTLEMAN, "Matrix multiplication and fast Fourier transformations," *Bell System Tech J.*, v. 47, 1968, pp. 1099–1102.
4. G. D. BERGLAND, "A guided tour of the fast Fourier transform," *IEEE Spectrum,* v. 6, no. 7, 1969, pp. 41–53.
5. L. I. BLUESTEIN, *A Linear Filtering Approach to the Computation of the Discrete Fourier Transform,* Northeast Electronics Research and Engineering Meeting Record, v. 10, 1968, pp. 218–219.
6. S. W. GOLOMB, *Shift Register Sequences,* Holden-Day, San Francisco, Calif., 1967. MR **39** #3906.
7. R. F. CHURCHOUSE & J. C. HERZ (Editors), *Computers in Mathematical Research,* North-Holland, Amsterdam, 1968. MR **38** #1972.
8. G. H. HARDY & E. M. WRIGHT, *An Introduction to the Theory of Numbers,* Clarendon Press, Oxford, 1938.
9. N. S. SZABO & R. I. TANABA, *Residue Arithmetic and its Applications to Computer Technology,* McGraw-Hill, New York, 1967.
10. D. B. GILLIES, "Three new Mersenne primes and a statistical theory," *Math. Comp.*, v. 18, 1964, pp. 93–97. MR **28** #2990.
11. H. DAVENPORT, *The Higher Arithmetic. An Introduction to the Theory of Numbers,* Hutchinson, London; Longmens, Green, New York, 1952. MR **14,** 352.
12. G. E. COLLINS, "Computing multiplicative inverses in GF(p)," *Math. Comp.*, v. 23, 1969, pp. 197–200. MR **39** #3676.
13. E. P. RODEMICH & H. RUMSEY, "Primitive trinomials of high degree," *Math. Comp.*, v. 22, 1968, pp. 863–865. MR **39** #177.
14. W. K. PRATT, J. KANE & H. C. ANDREWS, "Hadamard transform image coding," *Proc. IEEE,* v. 57, 1969, pp. 58–68.
15. H. B. MANN (Editor), *Error Correcting Codes,* Proc. Sympos. Math. Res. Center (University of Wisconsin, Madison, Wis., 1968), Wiley, New York, 1968. MR **38** #3071.
16. H. POLLARD, *The Theory of Algebraic Numbers,* Carus Math. Monographs, no. 9, Math. Assoc. Amer.; Distributed by Wiley, New York, 1950. MR **12,** 243.
17. W. T. COCHRAN ET AL., "What is the fast Fourier transform?" *Proc. IEEE,* v. 55, 1967, pp. 1664–1674.