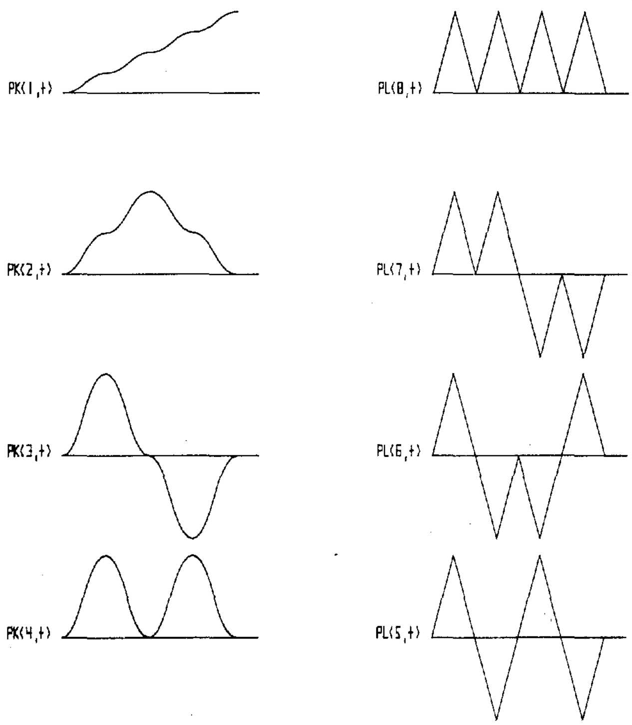

Fig. 1. PL and PK functions for $k = 3$.

$$\mathrm{PK}\,(0, t) = 1 \qquad 0 \leqslant t \leqslant T$$
$$i = 1, 2, \cdots, 2^{k-1}. \tag{2}$$

C is a normalization constant to be derived later.

The PL (i, t) functions are continuous in the interval $[0, T]$; therefore, PK (i, t) has a continuous first derivative in that interval.

We will now show that

$$\mathrm{PK}\,(i, t_\alpha) = \mathrm{PL}\,(i, t_\alpha) \qquad t_\alpha = \alpha\, T 2^{-k}$$
$$\alpha = 0, 2, 4, \cdots, 2^k$$
$$i = 1, 2, \cdots, 2^{k-1} \tag{3}$$

which is a necessary and sufficient condition for the PK and PL sets to use the same 2^{k-1} expansion coefficients in the approximation of a given function.

Consider the function

$\mathrm{Wal}^*\,(j, t)$, defined as

$\mathrm{Wal}^*\,(j, t) = \mathrm{Wal}\,(j, t)$ of subset index k,

over all odd order subintervals of $[0, T]$

$$(\alpha - 1)\, T2^{-k} \leqslant t \leqslant \alpha\, T2^{-k} \qquad \alpha = 1, 3, 5, \cdots, 2^k - 1$$

and

$$\mathrm{Wal}^*\,(j, t) = -\,\mathrm{Wal}\,(j, t)$$

over all even order subintervals of $[0, T]$

$$(\alpha - 1)\, T2^{-k} \leqslant t \leqslant \alpha\, T2^{-k} \qquad \alpha = 2, 4, 6, \cdots, 2^k.$$

It is a direct consequence of the recursive law for Walsh function generation [2] that

$$\mathrm{Wal}^*\,(2^k - 1, t) = \mathrm{Wal}\,(0, t)$$
$$\mathrm{Wal}^*\,(2^k - 1 - i, t) = \mathrm{Wal}\,(i, t). \tag{4}$$

Compare now $\mathrm{Wal}^*\,(j, t)$ and PL $(j + 1, t)$ over any subinterval pair

$$(\alpha - 2)\, T2^{-k} \leqslant t \leqslant \alpha\, T2^{-k} \qquad \alpha = 2, 4, 6, \cdots, 2^k.$$

$\mathrm{Wal}^*\,(j, t)$ is a constant b_i, where $b_i = \pm 1$, over the subinterval pair, and PL $(j + 1, t)$ is a triangle of height b_i [1].

The area under the respective curves in the ith subinterval pair is $b_i\, 2^{1-k}\, T$ for $\mathrm{Wal}^*\,(j, t)$ and $1/2\; b_i\, 2^{1-k}\, T$ for PL $(j + 1, t)$

$$\int_0^{t_\alpha} \mathrm{Wal}^*\,(j, x)\, dx = 2 \int_0^{t_\alpha} \mathrm{PL}\,(j + 1, x)\, dx$$
$$t_\alpha = \alpha\, T 2^{-k}$$
$$\alpha = 0, 2, 4, \cdots, 2^k. \tag{5}$$

If we set $j = 2^k - i$, the right side of (5) is $2/C$ PK (i, t_α), from (2) and the left side of (5) is $T2^{-k'}$ PL (i, t_α), from (4) and (1), where k' is the subset index of PL (i, t). With $C = 2^{k'+1}/T$, (3) is satisfied.

The PK functions generate a piecewise parabolic approximation of a function $f(t)$. The derivative of the approximation $\hat{f}(t)$ is 0 at $t = \alpha\, T2^{-k}$, $\alpha = 0, 2, 4, 2^k$, since PL $(2^k + 1 - i)$ is zero at these points, for $i = 1, 2, \cdots, 2^{k-1}$.

The PK (i, t) functions of index k can be integrated in turn to provide a set of 2^{k-1} basis functions, with continuous first and second derivatives; these derivatives are both 0 at $t = \alpha\, T2^{-k}$.

In general let

$$\mathrm{PK}_n\,(i, t) = C \int_0^t \mathrm{PK}_{n-1}\,(2^k + 1 - i, x)\, dx$$

$$\mathrm{PK}_n\,(0, t) = 1 \qquad i = 1, 2, \cdots, 2^{k-1}\; 0 \leqslant t \leqslant T$$

and define

$$\mathrm{PK}_0\,(i, t) = \mathrm{PL}\,(i, t)$$
$$\mathrm{PK}_1\,(i, t) = \mathrm{PK}\,(i, t).$$

The $\mathrm{PK}_n\,(i, t)$ function has n continuous derivatives in the interval $0 \leqslant t \leqslant T$. All $\mathrm{PK}_n\,(i, t)$ functions use the same set of 2^{k-1} expansion coefficients for the approximation of a given function. All continuous derivatives are 0 at $t = \alpha\, T2^{-k}$, $\alpha = 0, 2, 4, 2^k$.

REFERENCES

[1] C. R. Paul and R. W. Koch, "On piecewise linear functions and piecewise linear signal expansion," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-22, pp. 263-268, Aug. 1974.

[2] J. L. Walsh, "A closed set of normal orthogonal functions," *Amer. J. Math.*, vol. 45, pp. 5-24, 1923.

A New Principle for Fast Fourier Transformation

CHARLES M. RADER AND N. M. BRENNER

***Abstract*—An alternative form of the fast Fourier transform (FFT) is developed. The new algorithm has the peculiarity that none of the multiplying constants required are complex—most are pure imaginary.**

Manuscript received July 7, 1975; revised October 10, 1975. This work was sponsored by the Department of the Air Force.

C. M. Rader is with the Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, MA 02173.

N. M. Brenner was with the Department of Earth and Planetary Sciences, Massachusetts Institute of Technology, Cambridge, MA 02139. He is now at 3 Sumner Road, Cambridge, MA 02138.

The advantages of the new form would, therefore, seem to be most pronounced in systems for which multiplications are most costly.

Let $\{a_n\}$ be a sequence of $N = 2^m$ data, whose discrete Fourier transform (DFT) is $\{A_k\}$. Let $W = \exp(-j2\pi/N)$. Present fast Fourier transform (FFT) algorithms are derived from an equation like (1) or its dual[1]

$$A_k = \text{DFT}\,\{a_{2n}\} + W^k \times \text{DFT}\,\{a_{2n+1}\}\,. \tag{1}$$

Each of the DFT's in (1) is a DFT of a half-length data sequence, and can be expressed as two still shorter DFT's. After m such stages of simplification an algorithm is evident which requires $0(N \log_2 N)$ operations to execute.

This correspondence presents an alternative to (1) which may similarly be applied iteratively to itself leading to an FFT algorithm. The new algorithm which so results has the peculiarity that none of the multiplying constants is complex. Its advantages would therefore seem to be most pronounced in systems for which multiplications are most costly.

I. Derivation

Let $\{b_n\}$ and $\{c_n\}$ be the sequences

$$\left.\begin{aligned} b_n &= a_{2n} \\ c_n &= a_{2n+1} - a_{2n-1} + Q \end{aligned}\right\} \quad n = 0, 1, \cdots, N/2 - 1 \tag{2}$$

where

$$Q = \frac{2}{N} \sum_{n=0}^{\frac{N}{2}-1} a_{2n+1}\,.$$

The $N/2$ point DFT's of these sequences are $\{B_k\}$ and $\{C_k\}$. It is helpful to consider the sequence $\{d_n\}$ and its DFT, $\{D_k\}$ defined by

$$d_n = a_{2n+1} \qquad n = 0, 1, \cdots, N/2 - 1 \tag{3}$$

so that $\{B_k\}$ and $\{D_k\}$ are the DFT's in (1).

C_k and D_k can be simply related. Since Q is a constant, it appears in C_k in only one term, C_o. Furthermore, C_o is exactly equal to D_o. For other values of k, C_k can be expressed by the circular shifting theorem for DFT.

$$\begin{aligned} C_k &= D_k(1 - W^{2k}) \qquad k = 1, 2, \cdots, \frac{N}{2} - 1 \\ &= -W^{+k} D_k (W^k - W^{-k}). \end{aligned} \tag{4}$$

Hence,

$$-W^k D_k = C_k/(W^k - W^{-k}) \qquad k \neq 0 \tag{5}$$

and we can rewrite (1)

$$\begin{aligned} A_k &= B_k - C_k/(W^k - W^{-k}) \\ &\qquad k = 1, 2, \cdots, \frac{N}{2} - 1, \frac{N}{2} + 1, \frac{N}{2} + 2, \cdots, N - 1 \end{aligned} \tag{6a}$$

$$\left.\begin{aligned} A_o &= B_o + C_o \\ A_{N/2} &= B_o - C_o \end{aligned}\right\}. \tag{6b}$$

Since W^k and W^{-k} are complex conjugates, their difference is twice the imaginary part, e.g.,

$$W^k - W^{-k} = -2j \sin 2\pi k/N$$

[1] Equation (1) is too specialized, leading to radix 2 algorithms. It portrays the essence of the derivation, however.

so that (6a) may be writter

$$A_k = B_k - \frac{1}{2} j \csc\left(\frac{2\pi}{N} k\right) C_k \qquad k \neq 0, N/2. \tag{6a}$$

Equations (6a) and (6b) are the replacements for (1) as promised. An FFT algorithm based on (6a) and (6b) requires only multiplication by pure imaginary constants.

II. Comments

$\frac{N}{2}Q$ could have been added to the output terms A_o, $A_{N/2}$ rather than to each of $\frac{N}{2}$ input terms as in (2). The algorithm resulting is more difficult to explain, although it might not be more difficult to implement. A different "Q" is needed for each substitution of (2) recursively into itself, requiring $\frac{N}{4} - 1$ extra storages in all. In hardware implementation, e.g., pipelines, this may not be a consideration.

If, in (2) we had used $a_{2n+1} + a_{2n-1}$, the minus sign would change to + in (4), (5), and (6a), thus charging $\frac{1}{2} j \csc(2\pi k/N)$ to $\frac{1}{2} \sec(2\pi k/N)$. The exceptional cases for k would then be $k = N/4$, $3N/4$ and these would involve $\pm j$, the only nonreal operation encountered. We judge that the secant form offers no advantages over the cosecant form.

Whereas the constants W^k used in (1) have unity magnitude, $\csc(2\pi k/N)$ can get very large. Therefore, small computation errors can lead to large output errors. Experience verifies that this is the case. We recommend that the method be modified if more than 8192 points are used. A conventional factoring can reduce a DFT to shorter DFT's and each of these can be computed by the cosecant method.

Substantial savings in multiplications can be made in the conventional FFT by deriving the algorithm in a higher radix. The method proposed here can also be developed for radices other than two.

III. Program

A program has been written based on these ideas. Since the program has been run only on a large general purpose computer, it offers no speed advantage relative to other FFT programs—the multiplication time of the large machine is too small to be worth saving. A copy of the program is available from the authors.

For some FFT algorithms it is a simple matter to arrange the sequence in which the constants W^{nk} are used so that each constant needs to be looked up or computed only once. The cosecant algorithm can be similarly organized. This has been shown useful for very simple computers, for which multiplication is bit by bit using subroutines.

IV. Summary

FFT's can be derived by recursive substitution of an equation into itself. The equation involves constants which are pure imaginary so the resulting FFT algorithm requires no complex multiplications. Its noise properties are poor, but it is thought likely that it will have advantages when the cost of multiplication forms the dominant part of the FFT computational cost.

References

[1] A. V. Oppenheim and R. Schafer, *Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975, pp. 290–291.

[2] L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975, pp. 357–360.

[3] B. Gold and C. Rader, *Digital Processing of Signals*. New York: McGraw-Hill, 1969, pp. 165–182.

[4] R. C. Singleton, "An algorithm for computing the mixed radix fast fourier trasnform," *IEEE Trans. Audio Electroacoust. (Special Issue on Fast Fourier Transform)*, vol. AU-17, pp. 93–103, June 1969; also in *Digital Signal Processing*. New York: IEEE Press, 1972, pp. 294–304.

Transformation Matrices for Bilinear Transformation of Multivariable Polynomials

V. V. BAPESWARA RAO AND V. K. AATRE

Abstract—The Q-matrix technique of bilinear transformation of a single-variable polynomial is extended to multivariable polynomials. A computer program for the transformation is included in the Appendix.

I. INTRODUCTION

The transformation matrix technique for bilinear transformation of a single-variable polynomial is well known [1]. Recently the same technique has been extended to transform multivariable polynomials by Bose and Jury [2]. In their procedure each variable is transformed separately. Thus, it involves matrix multiplication in which the elements of one of the matrices are not numbers but polynomials. In this correspondence a procedure for the generation of the transformation matrix for bilinear transformation of a multivariable polynomial is outlined, in which the elements of the matrices are numbers.

Let $G = g(z_1, z_2, \cdots, z_n) = 0$ be the given multivariable polynomial to be transformed into $F = f(s_1, s_2, \cdots, s_n) = 0$ by the bilinear transformation $z_k = (s_k + 1)/(s_k - 1)$ for $k = 1, 2, \cdots, n$. Let $[a]$ be a vector whose elements are the coefficients of G and $[b]$ a vector whose elements are the coefficients of F. Let

$$[b] = [Q]\,[a] \tag{1}$$

where Q is a transformation matrix. The methods of construction of these matrices are outlined. Justification of the technique proposed is not included to restrict the length of this correspondence.

II. THE COEFFICIENT ARRAYS

Let d_k denote the highest degree of z_k in G. The number of coefficients N_c of G (and F) is given by

$$N_c = \prod_{i=1}^{n} (d_i + 1). \tag{2}$$

The terms of G are arranged as follows.

$$\begin{aligned} G = {} & z_1^{d_1}[g_1(z_2, z_3, \cdots, z_n)] + z_1^{d_1-1}[g_2(z_2, z_3, \cdots, z_n] \\ & + \cdots + z_1^{d_1-k+1}[g_k(z_2, z_3, \cdots, z_n] + \cdots, \\ & + z_1^0[g_{d_1+1}(z_2, z_3, \cdots, z_n)]. \end{aligned}$$

The terms $g_k(z_2, z_3, \cdots, z_n)$ for $k = 1, 2, \cdots, (d_1 + 1)$ are then arranged likewise, i.e.,

$$\begin{aligned} g_k(z_2, z_3, \cdots, z_n) = {} & z_2^{d_2}[g_{k,1}(z_3, z_4, \cdots, z_n)] \\ & + z_2^{d_2-1}[g_{k,2}(z_3, z_4, \cdots, z_n)] \\ & + \cdots + z_2^{d_2-l+1}[g_{k,l}(z_3, z_4, \cdots, z_n)] \\ & + \cdots + z_2^0[g_{k,d_2+1}(z_3, z_4, \cdots, z_n). \end{aligned}$$

The process is repeated over all the variables, except z_n (see the example in Section IV).

The coefficient of $z_1^{\alpha_1} z_2^{\alpha_2} \cdots z_n^{\alpha_n}$ (where $\alpha_m \leqslant d_m$ for $m = 1, 2, \cdots, n$) is the kth element of $[a]$ where

$$\begin{aligned} k = {} & (d_1 - \alpha_1)(d_2 + 1)(d_3 + 1) \cdots (d_n + 1) \\ & + (d_2 - \alpha_2)(d_3 + 1)(d_4 + 1) \cdots (d_n + 1) + \cdots \\ & + (d_{n-2} - \alpha_{n-2})(d_{n-1} + 1)(d_n + 1) \\ & + (d_{n-1} - \alpha_{n-1})(d_n + 1) + (d_n - \alpha_n) + 1. \end{aligned} \tag{3}$$

III. THE TRANSFORMATION MATRIX Q

Let Q_k represent the transformation matrix which transforms the coefficients of a single-variable polynomial of order k. The rules for the construction of Q_k are well known [1]. The transformation matrix Q is obtained as

$$Q = Q_{d_1} \times Q_{d_2} \times Q_{d_3} \times \cdots \times Q_{d_n}$$

where $\times$ denotes the Kronecker product [3]. The "multiplication" is carried out from the right end of the expression, i.e.,

$$\begin{aligned} & Q_1 \times Q_2 \times Q_3 \times Q_4 \times Q_5 \\ & \quad = Q_1 \times [Q_2 \times \{Q_3 \times (Q_4 \times Q_5)\}]. \end{aligned}$$

IV. EXAMPLE

Let $g(z_1, z_2, z_3) = (4z_2z_3 + z_2 - z_3)z_1^2 + z_2z_3z_1 + z_3$. The example is taken from [2]. For this example $d_1 = 2$, $d_2 = 1$, and $d_3 = 1$.

$$\begin{aligned} g(z_1, z_2, z_3) = {} & 4z_1^2z_2^1z_3^1 + 1z_1^2z_2^1z_3^0 + (-1)z_1^2z_2^0z_3^1 + 0z_1^2z_2^0z_3^0 \\ & + 1z_1^1z_2^1z_3^1 + 0z_1^1z_2^1z_3^0 + 0z_1^1z_2^0z_3^1 + 0z_1^1z_2^0z_3^0 \\ & + 0z_1^0z_2^1z_3^1 + 0z_1^0z_2^1z_3^0 + 1z_1^0z_2^0z_3^1 + 0z_1^0z_2^0z_3^0. \end{aligned}$$

Thus,

$$[a]^t = [4 \;\; 1 \;\; -1 \;\; 0 \;\; 1 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 1 \;\; 0]$$

and

$$Q = Q_{d_1} \times Q_{d_2} \times Q_{d_3} = Q_2 \times Q_1 \times Q_1$$

$$Q_{d_1} = Q_2 = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 0 & -2 \\ 1 & -1 & 1 \end{bmatrix}$$

$$Q_{d_2} = Q_{d_3} = Q_1 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

The matrix $Q_{d_2} \times Q_{d_3}$ is obtained by replacing each element y of Q_{d_2} by $y\,[Q_{d_3}]$. Thus,

$$Q_{d_2} \times Q_{d_3} = \left[\begin{array}{cc|cc} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ \hline 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{array}\right].$$

The Q matrix is obtained by replacing each element y of Q_{d_1} by $y\,[Q_{d_2} \times Q_{d_3}]$. Thus,

Manuscript received May 18, 1975; revised September 5, 1975. This work was supported by the National Research Council of Canada under Grant A-7237.

The authors are with the Department of Electrical Engineering, Nova Scotia Technical College, Halifax, N.S., Canada.