1. GENERAL RELATIONSHIPS AND TECHNIQUES

Whenever the functions to which the Fourier transform is applied are band-limited, or can be well approximated by band-limited functions, the discrete Fourier transform (DFT) provides a means of constructing explicit numerical solutions to the problems at hand. A great variety of investigations in physics, engineering and applied mathematics thus lead to DFT calculations, to such a degree that, at the time of writing, about 50% of all supercomputer CPU time is alleged to be spent calculating DFTs.

The straightforward use of the defining formulae for the DFT leads to calculations of size N^2 for N sample points, which become unfeasible for any but the smallest problems. Much ingenuity has therefore been exerted on the design and implementation of faster algorithms for calculating the DFT (McClellan & Rader, 1979; Nussbaumer, 1981; Blahut, 1985; Brigham, 1988). The most famous is that of Cooley & Tukey (1965) which heralded the age of digital signal processing. However, it had been preceded by the prime factor algorithm of Good (1958, 1960), which has lately been the basis of many new developments. Recent historical research (Goldstine, 1977, pp. 249–253; Heideman *et al.*, 1984) has shown that Gauss essentially knew the Cooley–Tukey algorithm as early as 1805 (before Fourier's 1807 work on harmonic analysis!); while it has long been clear that Dirichlet knew of the basis of the prime factor algorithm and used it extensively in his theory of multiplicative characters [see *e.g.* Chapter I of Ayoub (1963), and Chapters 6 and 8 of Apostol (1976)]. Thus the computation of the DFT, far from being a purely technical and rather narrow piece of specialized numerical analysis, turns out to have very rich connections with such central areas of pure mathematics as number theory (algebraic and analytic), the representation theory of certain Lie groups and coding theory – to list only a few. The interested reader may consult Auslander & Tolimieri (1979); Auslander, Feig & Winograd (1982, 1984); Auslander & Tolimieri (1985); Tolimieri (1985).

One-dimensional algorithms are examined first. The Sande mixed-radix version of the Cooley–Tukey algorithm only calls upon the additive structure of congruence classes of integers. The prime factor algorithm of Good begins to exploit some of their multiplicative structure, and the use of relatively prime factors leads to a stronger factorization than that of Sande. Fuller use of the multiplicative structure, *via* the group of units, leads to the Rader algorithm; and the factorization of short convolutions then yields the Winograd algorithms.

Multidimensional algorithms are at first built as tensor products of one-dimensional elements. The problem of factoring the DFT in several dimensions simultaneously is then examined. The section ends with a survey of attempts at formalizing the interplay between algorithm structure and computer architecture for the purpose of automating the design of optimal DFT code.

It was originally intended to incorporate into this section a survey of all the basic notions and results of abstract algebra which are called upon in the course of these developments, but time limitations have made this impossible. This material, however, is adequately covered by the first chapter of Tolimieri *et al.* (1989) in a form tailored for the same purposes. Similarly, the inclusion of numerous detailed examples of the algorithms described here has had to be postponed to a later edition, but an abundant supply of such examples may be found in the signal processing literature, for instance in the books by McClellan & Rader (1979), Blahut (1985), and Tolimieri *et al.* (1989).

1.3.3.2. *One-dimensional algorithms*

Throughout this section we will denote by $e(t)$ the expression $\exp(2\pi i t)$, $t \in \mathbb{R}$. The mapping $t \longmapsto e(t)$ has the following properties:

$$\begin{aligned} e(t_1 + t_2) &= e(t_1)e(t_2) \\ e(-t) &= \overline{e(t)} = [e(t)]^{-1} \\ e(t) &= 1 \Leftrightarrow t \in \mathbb{Z}. \end{aligned}$$

Thus e defines an isomorphism between the additive group $\mathbb{R}/\mathbb{Z}$ (the reals modulo the integers) and the multiplicative group of complex numbers of modulus 1. It follows that the mapping $\ell \longmapsto e(\ell/N)$, where $\ell \in \mathbb{Z}$ and N is a positive integer, defines an isomorphism between the one-dimensional residual lattice $\mathbb{Z}/N\mathbb{Z}$ and the multiplicative group of Nth roots of unity.

The DFT on N points then relates vectors $\mathbf{X}$ and $\mathbf{X}^*$ in W and W^* through the linear transformations:

$$\begin{aligned} F(N): \quad & X(k) = \frac{1}{N} \sum_{k^* \in \mathbb{Z}/N\mathbb{Z}} X^*(k^*) e(-k^*k/N) \\ \bar{F}(N): \quad & X^*(k^*) = \sum_{k \in \mathbb{Z}/N\mathbb{Z}} X(k) e(k^*k/N). \end{aligned}$$

1.3.3.2.1. *The Cooley–Tukey algorithm*

The presentation of Gentleman & Sande (1966) will be followed first [see also Cochran *et al.* (1967)]. It will then be reinterpreted in geometric terms which will prepare the way for the treatment of multidimensional transforms in Section 1.3.3.3.

Suppose that the number of sample points N is composite, say $N = N_1N_2$. We may write k to the base N_1 and k^* to the base N_2 as follows:

$$\begin{aligned} k &= k_1 + N_1k_2 \quad k_1 \in \mathbb{Z}/N_1\mathbb{Z}, \quad k_2 \in \mathbb{Z}/N_2\mathbb{Z} \\ k^* &= k_2^* + k_1^*N_2 \quad k_1^* \in \mathbb{Z}/N_1\mathbb{Z}, \quad k_2^* \in \mathbb{Z}/N_2\mathbb{Z}. \end{aligned}$$

The defining relation for $\bar{F}(N)$ may then be written:

$$\begin{aligned} X^*(k_2^* + k_1^*N_2) = \sum_{k_1 \in \mathbb{Z}/N_1\mathbb{Z}} \sum_{k_2 \in \mathbb{Z}/N_2\mathbb{Z}} & X(k_1 + N_1k_2) \\ & \times e\left[\frac{(k_2^* + k_1^*N_2)(k_1 + N_1k_2)}{N_1N_2}\right]. \end{aligned}$$

The argument of $e[.]$ may be expanded as

$$\frac{k_2^*k_1}{N} + \frac{k_1^*k_1}{N_1} + \frac{k_2^*k_2}{N_2} + k_1^*k_2,$$

and the last summand, being an integer, may be dropped:

$$\begin{aligned} & X^*(k_2^* + k_1^*N_2) \\ & \quad = \sum_{k_1} \left\{ e\left(\frac{k_2^*k_1}{N}\right) \left[\sum_{k_2} X(k_1 + N_1k_2) e\left(\frac{k_2^*k_2}{N_2}\right) \right] \right\} \\ & \qquad \times e\left(\frac{k_1^*k_1}{N_1}\right). \end{aligned}$$

This computation may be decomposed into five stages, as follows:

(i) form the N_1 vectors $\mathbf{Y}_{k_1}$ of length N_2 by the prescription

$$Y_{k_1}(k_2) = X(k_1 + N_1k_2), \quad k_1 \in \mathbb{Z}/N_1\mathbb{Z}, \quad k_2 \in \mathbb{Z}/N_2\mathbb{Z};$$

(ii) calculate the N_1 transforms $\mathbf{Y}_{k_1}^*$ on N_2 points:

$$\mathbf{Y}_{k_1}^* = \bar{F}(N_2)[\mathbf{Y}_{k_1}], \quad k_1 \in \mathbb{Z}/N_1\mathbb{Z};$$

(iii) form the N_2 vectors $\mathbf{Z}_{k_2^*}$ of length N_1 by the prescription

$$\mathbf{Z}_{k_2^*}(k_1) = e\left(\frac{k_2^*k_1}{N}\right) Y_{k_1}^*(k_2^*), \quad k_1 \in \mathbb{Z}/N_1\mathbb{Z}, \quad k_2^* \in \mathbb{Z}/N_2\mathbb{Z};$$

(iv) calculate the N_2 transforms $\mathbf{Z}^*_{k_2^*}$ on N_1 points:

$$\mathbf{Z}^*_{k_2^*} = \bar{F}(N_1)[\mathbf{Z}_{k_2^*}], \quad k_2^* \in \mathbb{Z}/N_2\mathbb{Z};$$

(v) collect $X^*(k_2^* + k_1^* N_2)$ as $Z^*_{k_2^*}(k_1^*)$.

If the intermediate transforms in stages (ii) and (iv) are performed *in place*, *i.e.* with the results overwriting the data, then at stage (v) the result $X^*(k_2^* + k_1^* N_2)$ will be found at address $k_1^* + N_1 k_2^*$. This phenomenon is called *scrambling* by 'digit reversal', and stage (v) is accordingly known as *unscrambling*.

The initial *N-point* transform $\bar{F}(N)$ has thus been performed as N_1 transforms $\bar{F}(N_2)$ on N_2 points, followed by N_2 transforms $\bar{F}(N_1)$ on N_1 points, thereby reducing the arithmetic cost from $(N_1 N_2)^2$ to $N_1 N_2 (N_1 + N_2)$. The phase shifts applied at stage (iii) are traditionally called 'twiddle factors', and the transposition between k_1 and k_2^* can be performed by the fast recursive technique of Eklundh (1972). Clearly, this procedure can be applied recursively if N_1 and N_2 are themselves composite, leading to an overall arithmetic cost of order $N \log N$ if N has no large prime factors.

The Cooley–Tukey factorization may also be derived from a geometric rather than arithmetic argument. The decomposition $k = k_1 + N_1 k_2$ is associated to a geometric partition of the residual lattice $\mathbb{Z}/N\mathbb{Z}$ into N_1 copies of $\mathbb{Z}/N_2\mathbb{Z}$, each translated by $k_1 \in \mathbb{Z}/N_1\mathbb{Z}$ and 'blown up' by a factor N_1. This partition in turn induces a (direct sum) decomposition of $\mathbf{X}$ as

$$\mathbf{X} = \sum_{k_1} \mathbf{X}_{k_1},$$

where

$$\begin{aligned} X_{k_1}(k) &= X(k) \quad \text{if } k \equiv k_1 \bmod N_1, \\ &= 0 \qquad \text{otherwise.} \end{aligned}$$

According to (i), $\mathbf{X}_{k_1}$ is related to $\mathbf{Y}_{k_1}$ by *decimation by* N_1 and *offset by* k_1. By Section 1.3.2.7.2, $\bar{F}(N)[\mathbf{X}_{k_1}]$ is related to $\bar{F}(N_2)[\mathbf{Y}_{k_1}]$ by *periodization by* N_2 and *phase shift by* $e(k^* k_1/N)$, so that

$$X^*(k^*) = \sum_{k_1} e\left(\frac{k^* k_1}{N}\right) Y^*_{k_1}(k_2^*),$$

the periodization by N_2 being reflected by the fact that $Y^*_{k_1}$ does not depend on k_1^*. Writing $k^* = k_2^* + k_1^* N_2$ and expanding $k^* k_1$ shows that the phase shift contains both the twiddle factor $e(k_2^* k_1/N)$ and the kernel $e(k_1^* k_1/N_1)$ of $\bar{F}(N_1)$. The Cooley–Tukey algorithm is thus naturally associated to the coset decomposition of a lattice modulo a sublattice (Section 1.3.2.7.2).

It is readily seen that essentially the same factorization can be obtained for $F(N)$, up to the complex conjugation of the twiddle factors. The normalizing constant $1/N$ arises from the normalizing constants $1/N_1$ and $1/N_2$ in $F(N_1)$ and $F(N_2)$, respectively.

Factors of 2 are particularly simple to deal with and give rise to a characteristic computational structure called a 'butterfly loop'. If $N = 2M$, then two options exist:

(*a*) using $N_1 = 2$ and $N_2 = M$ leads to collecting the even-numbered coordinates of $\mathbf{X}$ into $\mathbf{Y}_0$ and the odd-numbered coordinates into $\mathbf{Y}_1$

$$\begin{aligned} Y_0(k_2) &= X(2k_2), \qquad k_2 = 0, \ldots, M-1, \\ Y_1(k_2) &= X(2k_2+1), \qquad k_2 = 0, \ldots, M-1, \end{aligned}$$

and writing:

$$\begin{aligned} X^*(k_2^*) &= Y_0^*(k_2^*) + e(k_2^*/N) Y_1^*(k_2^*), \\ &\qquad k_2^* = 0, \ldots, M-1; \\ X^*(k_2^* + M) &= Y_0^*(k_2^*) - e(k_2^*/N) Y_1^*(k_2^*), \\ &\qquad k_2^* = 0, \ldots, M-1. \end{aligned}$$

This is the original version of Cooley & Tukey, and the process of formation of $\mathbf{Y}_0$ and $\mathbf{Y}_1$ is referred to as 'decimation in time' (*i.e.* decimation along the *data* index $\mathbf{k}$).

(*b*) using $N_1 = M$ and $N_2 = 2$ leads to forming

$$\begin{aligned} Z_0(k_1) &= X(k_1) + X(k_1 + M), \qquad k_1 = 0, \ldots, M-1, \\ Z_1(k_1) &= [X(k_1) - X(k_1 + M)] e\left(\frac{k_1}{N}\right), \qquad k_1 = 0, \ldots, M-1, \end{aligned}$$

then obtaining separately the even-numbered and odd-numbered components of $\mathbf{X}^*$ by transforming $\mathbf{Z}_0$ and $\mathbf{Z}_1$:

$$\begin{aligned} X^*(2k_1^*) &= Z_0^*(k_1^*), \quad k_1^* = 0, \ldots, M-1; \\ X^*(2k_1^* + 1) &= Z_1^*(k_1^*), \quad k_1^* = 0, \ldots, M-1. \end{aligned}$$

This version is due to Sande (Gentleman & Sande, 1966), and the process of separately obtaining even-numbered and odd-numbered results has led to its being referred to as 'decimation in frequency' (*i.e.* decimation along the *result* index k^*).

By repeated factoring of the number N of sample points, the calculation of $F(N)$ and $\bar{F}(N)$ can be reduced to a succession of stages, the smallest of which operate on single prime factors of N. The reader is referred to Gentleman & Sande (1966) for a particularly lucid analysis of the programming considerations which help implement this factorization efficiently; see also Singleton (1969). Powers of two are often grouped together into factors of 4 or 8, which are advantageous in that they require fewer complex multiplications than the repeated use of factors of 2. In this approach, large prime factors P are detrimental, since they require a full P^2-size computation according to the defining formula.

1.3.3.2.2. *The Good (or prime factor) algorithm*

1.3.3.2.2.1. *Ring structure on* $\mathbb{Z}/N\mathbb{Z}$

The set $\mathbb{Z}/N\mathbb{Z}$ of congruence classes of integers modulo an integer N [see *e.g.* Apostol (1976), Chapter 5] inherits from $\mathbb{Z}$ not only the additive structure used in deriving the Cooley–Tukey factorization, but also a *multiplicative* structure in which the product of two congruence classes mod N is uniquely defined as the class of the ordinary product (in $\mathbb{Z}$) of representatives of each class. The multiplication can be distributed over addition in the usual way, endowing $\mathbb{Z}/N\mathbb{Z}$ with the structure of a *commutative ring*.

If N is composite, the ring $\mathbb{Z}/N\mathbb{Z}$ has *zero divisors*. For example, let $N = N_1 N_2$, let $n_1 \equiv N_1 \bmod N$, and let $n_2 \equiv N_2 \bmod N$: then $n_1 n_2 \equiv 0 \bmod N$. In the general case, a product of non-zero elements will be zero whenever these elements collect together all the factors of N. These circumstances give rise to a fundamental theorem in the theory of commutative rings, the *Chinese Remainder Theorem* (CRT), which will now be stated and proved [see Apostol (1976), Chapter 5; Schroeder (1986), Chapter 16].

1.3.3.2.2.2. *The Chinese remainder theorem*

Let $N = N_1 N_2 \ldots N_d$ be factored into a product of pairwise coprime integers, so that g.c.d. $(N_i, N_j) = 1$ for $i \neq j$. Then the system of congruence equations

$$\ell \equiv \ell_j \bmod N_j, \qquad j = 1, \ldots, d,$$

has a unique solution $\ell \bmod N$. In other words, each $\ell \in \mathbb{Z}/N\mathbb{Z}$ is

references