FAST FOURIER METHODS IN COMPUTATIONAL COMPLEX ANALYSIS*

PETER HENRICI†

Abstract. In this paper we discuss the discrete Fourier transform and point out some computational problems in (mainly) complex analysis where it can be fruitfully applied. We begin by describing the elementary properties of the transform and its efficient implementation, both in the one-dimensional and in the multi-dimensional case, by the reduction formulas of Cooley, Lewis, and Welch (IBM Res. paper, 1967). The following applications are then discussed: Calculation of Fourier coefficients using attenuation factors; solution of Symm's integral equation in numerical conformal mapping; trigonometric interpolation; determination of conjugate periodic functions and their application to Theodorsen's integral equation for the conformal mapping of simply and of doubly connected regions; determination of Laurent coefficients with applications to numerical differentiation, generating functions, and the numerical inversion of Laplace transforms; determination of the "density" of the zeros of high degree polynomials. We then discuss convolution and its application to time series analysis, to the multiplication of polynomials and of large integers, and to fast Poisson solvers. The paper concludes with an account of some recent results of Brent and Kung (Carnegie-Mellon Univ., 1975, 1976) concerning fast algorithms for manipulating power series.

Fourier analysis is one of the most pervasive tools in applied mathematics. Among other places, it occurs

(i) in the modeling of time-dependent phenomena that are exactly or approximately periodic;

(ii) in the study of problems that involve a circular or rectangular geometry.

Examples for (i) include the theory of alternating currents in electrical engineering; the digital processing of information such as speech, electrocardiagrams, and electroencephalograms; the analysis of geophysical phenomena such as earthquakes and tides. Examples for (ii) occur not only in classical mathematical physics, notably in the study of vibrations of spherical, circular, or rectangular structures, but also in the processing and cleaning up of pictures, such as those beautiful photographs transmitted from remote planets.

It is one of the triumphs of mathematical abstraction, diminished only subjectively because we have become so very much used to it, that periodic phenomena in time and space can be dealt with by the same mathematical apparatus. Since the times of Fourier this apparatus has been developed to a high degree of perfection. Frequently in the modeling or analysis of real situations this theory cannot be applied directly, because the functions to be sampled are known only on a discrete set of "sampling points". It then becomes necessary to replace continuous Fourier analysis by a discretized version of it. Rather surprisingly the mathematical theory of discrete Fourier analysis enjoys an even greater symmetry than the continuous theory; moreover, it has interesting applications to problems in computation that are not a priori periodic. Originally the impact of discrete Fourier analysis was limited by the very large computational demands made by the theory in its naive form. This was changed in 1965 by the invention of a new family of algorithms, called Fast Fourier Transforms, by Cooley and Tukey, which reduced the computational work required to carry out a discrete Fourier transform by orders of magnitude.

It is my intention in this report to give an elementary introduction to these Fast Fourier Transforms, and to describe some of their applications, potential as well as

actual, to various computational problems in applied analysis, especially applied complex analysis.

1. The discrete Fourier transform

1.1. The one-dimensional transform. Let n be a natural number. We denote by Π_n the space of bilaterally infinite sequences

$$\mathbf{x} = \{x_k\}_{k=-\infty}^{\infty},$$

$x_k \in \mathbb{C}$, that are periodic with period n, i.e. that satisfy

$$x_{k+n} = x_k$$

for all k. Defining addition of two sequences $\mathbf{x} = \{x_k\}$ and $\mathbf{y} = \{y_k\} \in \Pi_n$ by

$$(\mathbf{x} + \mathbf{y})_k := x_k + y_k, \quad \forall k,$$

and multiplication by a scalar $c \in \mathbb{C}$ by

$$(c\mathbf{x})_k := cx_k, \quad \forall k,$$

it is clear that Π_n becomes a linear space. The zero element of the space is the sequence $\mathbf{0} := \{0\}$, all elements of which are zero. Any sequence $\mathbf{x} \in \Pi_n$ can be expressed as a linear combination of the n sequences

$$\mathbf{e}^{(m)} = \{e_k^{(m)}\}, \qquad m = 0, 1, \cdots, n-1,$$

where

$$e_k^{(m)} := \begin{cases} 1, & k \equiv m \pmod{n}, \\ 0, & k \not\equiv m \pmod{n}. \end{cases}$$

Because the sequences $\mathbf{e}^{(m)}$ are linearly independent, the space Π_n has dimension n.

On Π_n we define a map $\mathscr{F}_n$, called the *discrete Fourier operator*, as follows. Let

$$w_n := \exp(2\pi i/n).$$

For $\mathbf{x} \in \Pi_n$ we set $\mathscr{F}_n \mathbf{x} = \mathbf{y} = \{y_m\}_{m=-\infty}^{\infty}$ where

$$(1.1) \qquad\qquad y_m := \frac{1}{n} \sum_{k=0}^{n-1} w_n^{-mk} x_k.$$

It is clear that the map $\mathscr{F}_n$ is linear. Moreover in view of $w_n^n = 1$ there holds for any integer m

$$y_{m+n} = \frac{1}{n} \sum_{k=0}^{n-1} w_n^{-(m+n)k} x_k$$

$$= \frac{1}{n} \sum_{k=0}^{n-1} w_n^{-mk} x_k = y_m.$$

The image sequence thus is periodic with period n, that is $\mathscr{F}_n$ maps Π_n into Π_n.

It is easy to see that the mapping defined by Π_n actually is onto. By a theorem of linear algebra this will follow if the only sequence $\mathbf{x} \in \Pi_n$ that is mapped on the zero sequence is the zero sequence. Now $\mathscr{F}_n \mathbf{x} = \mathbf{0}$ means that

$$\sum_{k=0}^{n-1} w_n^{-km} x_k = 0, \qquad m = 0, 1, \cdots, n-1.$$

This represents a homogeneous system of n linear equations in the n unknowns $x_0, x_1, \cdots, x_{n-1}$. The determinant of the system is the Vandermonde determinant

formed with the numbers $1, w_n^{-1}, w_n^{-2}, \cdots, w_n^{-n+1}$ which are all distinct. The determinant thus is $\neq 0$, and the only solution is the trivial solution.

Because $\mathscr{F}_n$ is a bijective linear map of Π_n onto Π_n, the inverse map exists and is linear. To find a formula for it, we use the fact that if r is an integer,

$$(1.2) \qquad 1 + w_n^r + w_n^{2r} + \cdots + w_n^{(n-1)r} = \begin{cases} n & \text{if } r \equiv 0 \pmod{n}, \\ 0 & \text{if } r \not\equiv 0 \pmod{n}. \end{cases}$$

Thus if

$$y_m = \frac{1}{n} \sum_{k=0}^{n-1} w_n^{-km} x_k, \qquad m = 0, 1, \cdots, n-1,$$

and if r is any integer, we get by multiplying the mth equation by w_n^{rm} and adding

$$\sum_{m=0}^{n-1} w_n^{rm} y_m = \frac{1}{n} \sum_{k=0}^{n-1} x_k \sum_{m=0}^{n-1} w_n^{(r-k)m} = x_r.$$

Hence the inverse map of (1.1) is

$$x_r = \sum_{m=0}^{n-1} w_n^{rm} y_m$$

and thus closely resembles $\mathscr{F}_n$. Defining the conjugate discrete Fourier operator $\bar{\mathscr{F}}_n$ by

$$(1.3) \qquad (\bar{\mathscr{F}}_n \mathbf{y})_m := \frac{1}{n} \sum_{k=0}^{n-1} w_n^{mk} y_k$$

we in fact have

$$(1.4) \qquad \mathscr{F}_n^{-1} = n \bar{\mathscr{F}}_n.$$

Manipulations involving $\mathscr{F}_n$ can often be simplified by means of the *reversion operator R*, defined for any $\mathbf{x} \in \Pi_n$ by

$$(1.5) \qquad (R\mathbf{x})_m := x_{-m}.$$

We have

$$(R\mathscr{F}_n \mathbf{x})_m = \frac{1}{n} \sum_{k=0}^{n-1} w_n^{km} x_k = (\bar{\mathscr{F}}_n \mathbf{x})_m,$$

$$(\mathscr{F}_n R \mathbf{x})_m = \frac{1}{n} \sum_{k=0}^{n-1} x_{-k} w_n^{-km}$$

$$= \frac{1}{n} \sum_{k=0}^{n-1} x_k w_n^{km} = (\bar{\mathscr{F}}_n \mathbf{x})_m,$$

hence

$$\mathscr{F}_n R = R \mathscr{F}_n = \bar{\mathscr{F}}_n$$

and consequently

$$(1.6) \qquad \mathscr{F}_n^{-1} = n \mathscr{F}_n R = n R \mathscr{F}_n.$$

In summary, we have proved:

THEOREM 1a. *The discrete Fourier operator $\mathscr{F}_n$ is a bijective linear map from Π_n to Π_n. The inverse map is*

$$\mathscr{F}_n^{-1} = n\bar{\mathscr{F}}_n = nR\mathscr{F}_n = n\mathscr{F}_nR,$$

where $\bar{\mathscr{F}}_n$ is the conjugate Fourier operator, and where R is the reversion operator defined by (1.5).

1.2. Existence of Fast Fourier Transform algorithms. Here we consider the algorithmic problem of computing the sequence $\mathbf{y} := \mathscr{F}_n\mathbf{x}$ for a given $\mathbf{x} \in \Pi_n$. Because $\mathbf{y}$ is periodic, it suffices to calculate the elements of one full period of $\mathbf{y}$, for instance the elements $y_0, y_1, \cdots, y_{n-1}$. To compute one such element directly from the formula (1.1), assuming that the required powers of w_n have already been formed, clearly requires $n - 1$ complex multiplications (μ). To compute a full period of $\mathbf{y}$ by this method thus requires $n(n - 1)$ μ. Modern applications of discrete Fourier analysis, for instance in the analysis of time series (see § 4.2) require values of n as large as 2^{14}. Then $n(n - 1) \doteq 2.68 * 10^8$, and the time required for forming even a single Fourier transform would appear to prohibit large-scale applications of the Fourier method.

It is therefore a fundamental fact in practical Fourier analysis that for numbers n that are highly composite the computational work required to form $\mathscr{F}_n\mathbf{x}$ can be drastically reduced. The basis for this reduction is a reduction formula that can be traced back to the precomputer age, but whose importance for large-scale applications of the transform was recognized for the first time by Cooley and Tukey (1965). For $n = 2^l$ the formula results in an algorithm that permits the evaluation of $\mathscr{F}_n\mathbf{x}$ in only $\frac{1}{2}ln = \frac{1}{2}n \operatorname{Log}_2 n$ μ.

Let $n = pq$, and for a given $\mathbf{x} \in \Pi_n$ let

$$\mathbf{x}^{(j)} = \{x_k^{(j)}\} = \{x_{j+pk}\}, \qquad j = 0, 1, \cdots, p - 1.$$

($\mathbf{x}^{(j)}$ is the subsequence of those elements of $\mathbf{x}$ whose index is $\equiv j \bmod p$.) Evidently, $\mathbf{x}^{(j)} \in \Pi_q, j = 0, 1, \cdots, p - 1$. We assume that the sequences

$$\mathbf{y}^{(j)} := \mathscr{F}_q\mathbf{x}^{(j)}, \qquad j = 0, 1, \cdots, p - 1,$$

are known, and try to express $\mathbf{y} := \mathscr{F}_n\mathbf{x}$ in terms of the elements of the sequences $\mathbf{y}^{(j)}$. For all integers m

$$y_m = \frac{1}{n}\sum_{k=0}^{n-1} w_n^{-mk}x_k = \frac{1}{p}\sum_{j=0}^{p-1}\frac{1}{q}\sum_{h=0}^{q-1} w_n^{-m(j+ph)}x_{j+ph},$$

and in view of $w_n^p = w_q$

$$y_m = \frac{1}{p}\sum_{j=0}^{p-1} w_n^{-mj}\frac{1}{q}\sum_{h=0}^{q-1} w_q^{-mh}x_{j+ph}.$$

Considering

$$\frac{1}{q}\sum_{h=0}^{q-1} w_q^{-mh}x_{j+ph} = y_m^{(j)},$$

this yields

(1.7) $$y_m = \frac{1}{p}\sum_{j=0}^{p-1} w_n^{-mj}y_m^{(j)},$$

which already is a representation of the desired sort. To construct the sequence $\mathbf{y}$ from

the sequences $\mathbf{y}^{(l)}$ by means of (1.7) evidently requires $p-1$ μ for each value of m, thus a total of $(p-1)n$ μ. We ignore the multiplications that could be saved for $m=0$.

Let now the integer n be factored into l factors

(1.8) $$n = n_1 n_2 \cdots n_l;$$

it is not required that the n_i are prime factors. Formula (1.7) then may be used recursively: To compute $\mathscr{F}_n\mathbf{x}$ we require n_1 transforms of period $q = n_2 \cdots n_l$, to compute these we need $n_1 n_2$ transforms of period $q = n_3 \cdots n_l$, etc., until we arrive at $n = n_1 n_2 \cdots n_l$ transforms $\mathscr{F}_1\mathbf{x}$ which are identical with $\mathbf{x}$. The total number of multiplications will then be

$$n(n_1-1)+n_1\frac{n}{n_1}(n_2-1)+\cdots+n_1\cdots n_l\frac{n}{n_1\cdots n_l}(n_l-1) = n\sum_{i=1}^{l}(n_i-1).$$

For instance if $n = 2^l$ the required number of multiplications will be $nl = n \, \mathrm{Log}_2 \, n$. The number of additions is similar, and the divisions by n, which are mere powers of 2, may be ignored. It thus is clear that an order-of-magnitude improvement has been achieved over the naïve method of evaluating (1.1).

There is a variant of (1.7) which sometimes leads to even greater economy. Let $m = k + lq$ ($k = 0, 1, \cdots, q-1$; $l = 0, 1, \cdots, p-1$). We then have

$$w_n^{-mj} = w_n^{-(k+lq)j} = w_p^{-lj}w_n^{-kj},$$

and because the sequences $\mathbf{y}^{(j)}$ have period q there follows

(1.9) $$y_{k+lq} = \frac{1}{p}\sum_{j=0}^{p-1} w_p^{-lj}w_n^{-kj}y_k^{(j)}, \qquad k = 0, 1, \cdots, q-1; \quad l = 0, 1, \cdots, p-1.$$

The equations (1.9) are evaluated by

(A) evaluating the $(p-1)q$ products

$$z_k^{(j)} := w_n^{-kj}y_k^{(j)}, \qquad j = 1, 2, \cdots, p-1; \quad k = 0, 1, \cdots, q-1;$$

(B) evaluating the $(p-1)^2 q$ products

$$w_p^{-lj}z_k^{(j)}, \qquad l, j = 1, 2, \cdots, p-1; \quad k = 0, 1, \cdots, q-1.$$

We again have ignored the triviality of the multiplications for $k = 0$. The step from the $\mathbf{y}^{(j)}$ to $\mathbf{y}$ now requires a total of

$$(p-1)q + (p-1)^2 q = pq(p-1) \quad \mu.$$

Thus in general there is no saving in comparison to the use of (1.7). If, however, $p = 2$, then $w_p = -1$, and the multiplications of type (B) need not be counted. The step from the $\mathbf{y}^{(j)}$ to $\mathbf{y}$ then requires only q μ, and if $n = 2^l$ and (1.9) is used recursively, then the total number of μ to evaluate $\mathscr{F}_n\mathbf{x}$ is

(1.10) $$\phi(n) := \tfrac{1}{2}n \, \mathrm{Log}_2 \, n.$$

THEOREM 1b. *If n is factored in the form* (1.8), *not more than*

$$n\sum_{i=1}^{l}(n_i-1)$$

complex multiplications are required to evaluate $\mathscr{F}_n\mathbf{x}$ for a given $\mathbf{x} \in \Pi_n$. In the special case $n = 2^l$ the number of multiplications does not exceed (1.10).

We do not discuss in detail the problem of how to implement the algorithm implied by the recurrence relations (1.7) and (1.9). This may be arranged in such a manner that at no stage of the computation must more than n numbers be stored. Most implementations require the *bit inversion function* p_j which is defined as follows: If m is an integer, $0 \leq m < 2^j$, whose binary representation is

$$m = m_0 + 2m_1 + 2^2 m_2 + \cdots + 2^{j-1} m_{j-1},$$

then

$$p_j(m) = m_{j-1} + 2m_{j-2} + \cdots + 2^{j-1} m_0.$$

For an implementation that does not require the bit inversion function see Kahaner (1978).

1.3. The multi-dimensional discrete Fourier transform. Let

$$\mathbf{x} = \{x_{k_1 k_2 \cdots k_d}\}_{k_i = -\infty}^{\infty}$$

be a d-dimensional array of complex numbers which in each index is periodic with period n. The space of all such arrays is denoted by $\Pi_n^{(d)}$. The d-dimensional discrete Fourier transform is defined by

$$\mathbf{y} = \mathscr{F}_n^{(d)} \mathbf{x} = \{y_{m_1 m_2 \cdots m_d}\}$$

where

$$(1.11) \qquad y_{m_1 m_2 \cdots m_d} := \frac{1}{n^d} \sum_{k_1 = 0}^{n-1} \cdots \sum_{k_d = 0}^{n-1} w_n^{-k_1 m_1 - \cdots - k_d m_d} x_{k_1 k_2 \cdots k_d}.$$

It may be shown as in the one-dimensional case that $\mathscr{F}_n^{(d)}$ is a bijective map from $\Pi_n^{(d)}$ onto $\Pi_n^{(d)}$ whose inverse is given by

$$(1.12) \qquad \mathscr{F}_n^{(d)-1} = n^d \bar{\mathscr{F}}_n^{(d)}.$$

The d-dimensional discrete Fourier transform has applications in crystallography ($d = 3$), in the solution of the Poisson differential equation in rectangular domains ($d = 2, 3$, see § 4.4), in numerical conformal mapping ($d = 2$, see § 2.3), and in digital picture processing ($d = 2$, see Rosenfeld and Kak (1976)).

The evaluation of (1.11) may be reduced to the evaluation of one-dimensional discrete transforms. This becomes evident if (1.11) is written as follows:

$$(1.13) \quad y_{m_1 m_2 \cdots m_d} = \frac{1}{n} \sum_{k_1 = 0}^{n-1} w_n^{-k_1 m_1} \frac{1}{n} \sum_{k_2 = 0}^{n-1} w_n^{-k_2 m_2} \cdots \frac{1}{n} \sum_{k_d = 0}^{n-1} w_n^{-k_d m_d} x_{k_1 k_2 \cdots k_d}.$$

This necessitates the computation of d simple Fourier transforms, each for n^{d-1} different combinations of the indices. Thus the total number of multiplications does not exceed dn^{d-1} times that required for a single $\mathscr{F}_n$, that is

$$dn^d \sum_{i=1}^{l} (n_i - 1)$$

for the general factorization (1.8), and

$$\tfrac{1}{2} dn^d \operatorname{Log}_2 n$$

if $n = 2^l$.

An alternative approach to the multi-dimensional transform consists in directly generalizing the reduction formulas (1.7) and (1.9) to the multi-dimensional case. As in

the one-dimensional case this has the advantage that it is not necessary to specify n in advance. For compact notation we introduce the index vectors

$$\mathbf{k} = (k_1, k_2, \cdots, k_d), \qquad \mathbf{m} = (m_1, m_2, \cdots, m_d)$$

and denote by Q_n the period cube in the d-dimensional unit lattice, i.e., the set of all index vectors $\mathbf{k}$ where $k_i \in \{0, 1, \cdots, n-1\}$. Then (1.11) is the same as

$$y_{\mathbf{m}} = \frac{1}{n^d} \sum_{\mathbf{k} \in Q_n} w_n^{-\mathbf{k} \cdot \mathbf{m}} x_{\mathbf{k}},$$

where the dot signifies a scalar product.

Let again $n = pq$, and consider the p^d sequences

$$\mathbf{x}^{\mathbf{j}} := \{x_{\mathbf{j}+p\mathbf{h}}\}, \qquad \mathbf{j} \in Q_p.$$

These sequences are in $\Pi_q^{(d)}$. We assume that their Fourier transforms

$$\mathbf{y}^{\mathbf{j}} := \mathscr{F}_q^{(d)} \mathbf{x}^{\mathbf{j}}, \qquad \mathbf{j} \in Q_p,$$

are known. Because every vector $\mathbf{k} \in Q_n$ may be written in the form

$$\mathbf{k} = \mathbf{j} + p\mathbf{h}, \qquad \mathbf{j} \in Q_p, \qquad \mathbf{h} \in Q_q,$$

in exactly one way, we have

$$y_{\mathbf{m}} = \frac{1}{n^d} \sum_{\mathbf{j} \in Q_p} \sum_{\mathbf{h} \in Q_q} w_n^{-(\mathbf{j}+p\mathbf{h}) \cdot \mathbf{m}} x_{\mathbf{j}+p\mathbf{h}}$$

$$= \frac{1}{p^d} \sum_{\mathbf{j} \in Q_p} w_n^{-\mathbf{j} \cdot \mathbf{m}} \frac{1}{q^d} \sum_{\mathbf{h} \in Q_q} w_q^{-\mathbf{h} \cdot \mathbf{m}} x_{\mathbf{j}+p\mathbf{h}}.$$

Because

$$\frac{1}{q^d} \sum_{\mathbf{h} \in Q_q} w_q^{-\mathbf{h} \cdot \mathbf{m}} x_{\mathbf{j}+p\mathbf{h}} = y_{\mathbf{m}}^{\mathbf{j}}$$

by definition, we have

(1.14) $$y_{\mathbf{m}} = \frac{1}{p^d} \sum_{\mathbf{j} \in Q_p} w_n^{-\mathbf{j} \cdot \mathbf{m}} y_{\mathbf{m}}^{\mathbf{j}}$$

which is the multi-dimensional analog of (1.7).

To obtain the analog of (1.9), we note that every $\mathbf{m} \in Q_n$ may be represented in exactly one way also in the form

$$\mathbf{m} = \mathbf{k} + q\mathbf{l}, \qquad \mathbf{k} \in Q_q, \quad \mathbf{l} \in Q_p.$$

Because the sequences $\mathbf{y}^{\mathbf{j}}$ have period q,

$$y_{\mathbf{m}}^{\mathbf{j}} = y_{\mathbf{k}+q\mathbf{l}}^{\mathbf{j}} = y_{\mathbf{k}}^{\mathbf{j}},$$

and observing that

$$w_n^{-\mathbf{j} \cdot \mathbf{m}} = w_n^{-\mathbf{j} \cdot (\mathbf{k}+q\mathbf{l})} = w_p^{-\mathbf{l} \cdot \mathbf{j}} w_n^{-\mathbf{k} \cdot \mathbf{j}}$$

there results the desired representation

(1.15) $$y_{\mathbf{k}+q\mathbf{l}} = \frac{1}{p^d} \sum_{\mathbf{j} \in Q_p} w_p^{-\mathbf{l} \cdot \mathbf{j}} w_n^{-\mathbf{k} \cdot \mathbf{j}} y_{\mathbf{k}}^{\mathbf{j}}.$$

By use of (1.14) or (1.15), the transform $\mathbf{y} \in \Pi_n^{(d)}$ may be constructed directly from p^d transforms in $\Pi_q^{(d)}$, without iterating one-dimensional transforms. While no significant savings appear to result in the case of general factorizations (1.8), a significant reduction of the number of multiplications is possible if $n = 2^l$. In this case, $p = 2$ always and

$$w_2^{-\mathbf{l} \cdot \mathbf{j}} = \pm 1.$$

Thus the only products to be formed when stepping from $\Pi_q^{(d)}$ to $\Pi_n^{(d)}$ are

$$w_n^{-\mathbf{k} \cdot \mathbf{j}} y_{\mathbf{k}}^{\mathbf{j}}, \qquad \mathbf{j} \in Q_2, \qquad \mathbf{j} \neq 0, \qquad \mathbf{k} \in Q_q.$$

There are $(2^d - 1) q^d = (1 - 2^{-d}) n^d$ such products. When stepping from 1 to $n = 2^l$ through the powers of 2, this process is repeated l times, and we have

THEOREM 1c. *If $n = 2^l$, the number of complex multiplications required to evaluate $\mathscr{F}_n^{(d)} \mathbf{x}$ does not exceed*

$$(1 - 2^{-d}) n^d \operatorname{Log}_2 n.$$

Already if $d = 2$ this compares favorably with the $\frac{1}{2} d n^d \operatorname{Log}_2 n$ multiplications required by iterating one-dimensional transforms, and the ratio becomes more favorable as d gets larger.

Notes on § 1. § 1.1. Discrete Fourier analysis is dealt with in many numerical analysis texts; see for instance Runge and König (1924), Hamming (1973), Dahlquist and Björck (1974).

§ 1.2. The original publication of Cooley and Tukey (1965) simultaneously proves the existence of a fast Fourier transform algorithm and provides an implementation for it. The treatment based on reduction formulas given here is based on Cooley, Lewis, and Welch (1967a). For the origins of the formulas see Cooley, Lewis, and Welch (1967b). Other approaches to fast Fourier transforms are based on a factoring of the matrix representing the Fourier operator (Theilheimer (1969), Kahaner (1970), McClellan and Parks (1972)), or on determining remainders in the division of polynomials (Fiduccia (1972), Aho, Hopcroft and Ullmann (1974), Kahaner (1978)). For some European implementations of the transform see Gander and Mazzario (1972), Iselin (1971). Winograd (1978) proposes a new algorithm for computing the discrete Fourier transform which for suitable n requires only about 20% of the multiplications of the Cooley–Tukey algorithm, while the number of additions remains about the same.

§ 1.3. Multi-dimensional transforms are usually treated by iterating one-dimensional transforms. We have found no reference indicating that the direct application of the reduction formulas may result in a saving of operations.

2. Harmonic analysis.

2.1. Fourier coefficients. The most obvious application of discrete Fourier analysis consists in the numerical calculation of Fourier coefficients. Let Π denote the class of complex-valued functions defined on $\mathbb{R}$ and having period 1. If $x \in \Pi$ is integrable, its Fourier coefficients are defined by

$$(2.1) \qquad\qquad a_m := \int_0^1 x(\tau) \, e^{-2\pi i m \tau} \, d\tau,$$

$m = 0, \pm 1, \pm 2, \cdots$. With these coefficients the formal Fourier series of x,

$$\sum_{m = -\infty}^{\infty} a_m \, e^{2\pi i m \tau},$$

may be formed. Under certain conditions this series converges to $x(\tau)$ for some or all values of τ.

In many applications the integrals (2.1) cannot be evaluated in closed form. This is true, in particular, if the function x is known only empirically, or if it can be evaluated only on a discrete set of values of τ. The points τ where x can be calculated or measured are known as the *sampling points*. If the point $\tau = 0$ plays no distinguished role, it is reasonable to assume that the sampling points are equidistant, and that their distance is commensurate with a period. We thus assume the sampling points to be

$$\tau_k = kh, \qquad h := \frac{1}{n}, \qquad k \in \mathbb{Z}.$$

The values

$$x_k := x(\tau_k), \qquad k \in \mathbb{Z},$$

are called the sampling values of x. They form a sequence

$$\mathbf{x} = \{x_k\}_{k=-\infty}^{\infty}$$

which evidently belongs to Π_n.

What kind of integration rule should be used to evaluate the integrals (2.1) under these circumstances? For functions x that are not smooth a high-powered Newton–Cotes rule (even if stable) obviously brings no advantage, and one might as well use a trapezoid or midpoint rule. The same holds if x is smooth, for it may be shown by a translation argument that the trapezoid or midpoint rule furnishes a result that is accurate to the same order as the n-point Newton–Cotes formula. Thus, setting $w := \exp(2\pi i/n)$, we approximate a_m by

(2.2)
$$\hat{a}_m := \frac{1}{n} \sum_{k=0}^{n-1} x_k w^{-mk}, \qquad m \in \mathbb{Z},$$

which of course is to say that the sequence $\hat{\mathbf{a}} = \{\hat{a}_m\}$ is

$$\hat{\mathbf{a}} = \mathscr{F}_n \mathbf{x},$$

the discrete Fourier transform of $\mathbf{x}$.

Concerning the error committed in approximating a_m by $\hat{a}_m$, the classical estimate for the Newton–Cotes formula,

$$|\hat{a}_m - a_m| \le \gamma_{n+1} \max_{\tau} |(x(\tau) e^{-2\pi i m \tau})^{(n+1)}|$$

with γ_{n+1} independent of x is of no great use, because even if estimates for the derivatives of x were available, the factor $\exp(-2\pi i m \tau)$ would cause the resulting estimate to be unrealistically large for large $|m|$. Fortunately, a more explicit error estimate is available under much weaker assumptions.

THEOREM 2a. *If the Fourier series of $x \in \Pi$ is absolutely convergent, then for every $m \in \mathbb{Z}$*

(2.3)
$$\hat{a}_m - a_m = \sum_{k \ne 0} a_{m+kn}.$$

The proof is immediate if we substitute into (2.2) the values

$$x_k = x(\tau_k) = \sum_{m=-\infty}^{\infty} a_m e^{2\pi i m \tau_k} = \sum_{m=-\infty}^{\infty} a_m w^{km},$$

reverse summations, and use (1.2).

If x can be extended to a function of the complex variable $t = \tau + i\sigma$ that is analytic in the strip $-\eta \leqq \sigma \leqq \eta$, the relation

$$f(z) := x\left(\frac{1}{2\pi i} \operatorname{Log} z\right)$$

defines a single-valued analytic function in the annulus $A: e^{-2\pi\eta} \leqq |z| \leqq e^{2\pi\eta}$. The a_m then are identical with the Laurent coefficients of f for that annulus, and by the Cauchy estimate are bounded by

$$|a_m| \leqq \mu\, e^{-2\pi\eta|m|}, \qquad m \in \mathbb{Z}$$

where μ is the maximum modulus of f in A. By summing geometric series, (2.3) then yields

$$|\hat{a}_m - a_m| \leqq 2\mu \cosh(2\pi m\eta) \cdot \frac{e^{-2\pi n\eta}}{1 - e^{-2\pi n\eta}}, \qquad |m| < n,$$

and we have:

THEOREM 2b. *If m is fixed and $n \to \infty$, the error of the m-th discrete Fourier coefficient of a function $x \in \Pi$ that is analytic in the strip $|\operatorname{Im} t| \leqq \eta$ tends to zero like* $\exp(-2\pi n\eta)$.

Thus, if a Fast Fourier Transform is used with $n = 2^l$, the error will be squared with each doubling of n.

Satisfactory as this convergence behavior may be, the $\hat{a}_m$ suffer from a basic flaw when n is fixed. From the Riemann–Lebesgue lemma it is known that

$$(2.4) \qquad\qquad\qquad \lim_{m \to \pm\infty} a_m = 0.$$

The sequence $\hat{\mathbf{a}}$, on the other hand, is periodic with period n and thus cannot satisfy (2.4) unless it is the zero sequence. Fortunately, this deficiency can be corrected by making use of a simple device known as *attenuation factors*.

We continue to assume that the only values of the function $x \in \Pi$ which we know are the sampling values x_m. The idea now is to approximate the sequence $\mathbf{x}$ by a function $P\mathbf{x} \in \Pi$, and to compute the Fourier coefficients of $P\mathbf{x}$ exactly. This computation turns out to be simple if the approximation operator P has the following properties which seem very natural:

(i) P is linear;

(ii) P is translation invariant; i.e. if E denotes the shift operator defined in Π_n by $(E\mathbf{x})_k := x_{k+1}$ and in Π by $(Ex)(\tau) := x(\tau + 1/n)$, then

$$PE\mathbf{x} = EP\mathbf{x}$$

for all $\mathbf{x} \in \Pi_n$.

If $\mathbf{x}$ is the sequence to be approximated we wish to compute the exact Fourier coefficients of the function $P\mathbf{x}$, i.e., the numbers

$$(2.5) \qquad\qquad\qquad b_m := \int_0^1 (P\mathbf{x})(\tau)\, e^{-2\pi i m\tau}\, d\tau.$$

Let $\boldsymbol{\delta} = \{\delta_m\}$ denote the *delta sequence* defined by

$$(2.6) \qquad\qquad\qquad \delta_m := \begin{cases} n, & m \equiv 0 \pmod{n}, \\ 0, & m \not\equiv 0 \pmod{n}. \end{cases}$$

Then

$$\mathbf{x} = \frac{1}{n} \sum_{k=0}^{n-1} x_k E^{-k} \boldsymbol{\delta},$$

and by (i) and (ii) we have

$$P\mathbf{x} = \frac{1}{n} \sum_{k=0}^{n-1} x_k P E^{-k} \boldsymbol{\delta} = \frac{1}{n} \sum_{k=0}^{n-1} x_k E^{-k} P \boldsymbol{\delta}.$$

If $p := P\boldsymbol{\delta} \in \Pi$, it follows that

$$b_m = \frac{1}{n} \sum_{k=0}^{n-1} x_k \int_0^1 (E^{-k} p)(\tau) \, e^{-2\pi i m \tau} \, d\tau$$

$$= \frac{1}{n} \sum_{k=0}^{n-1} x_k \int_0^1 p\left(\tau - \frac{k}{n}\right) e^{-2\pi i m \tau} \, d\tau.$$

Because p is periodic with period 1,

$$\int_0^1 p\left(\tau - \frac{k}{n}\right) e^{-2\pi i m \tau} \, d\tau = \int_{-k/n}^{-k/n+1} p(\tau) \, e^{-2\pi i m (\tau + k/n)} \, d\tau$$

$$= w^{-mk} p_m,$$

where $w := \exp(2\pi i/n)$ and

$$p_m := \int_0^1 p(\tau) \, e^{-2\pi i m \tau} \, d\tau$$

is the mth Fourier coefficient of $p = P\boldsymbol{\delta}$. Hence we have the resulting expression

$$b_m = p_m \frac{1}{n} \sum_{k=0}^{n-1} x_k w^{-km} = p_m \hat{a}_m.$$

THEOREM 2c. *The Fourier coefficients of $P\mathbf{x}$ are $p_m \hat{a}_m$, where $\hat{\mathbf{a}} := \mathscr{F}_n \mathbf{x}$, and p_m is the m-th Fourier coefficient of $P\boldsymbol{\delta}$, where $\boldsymbol{\delta}$ is the delta sequence given by (2.6.).*

The numbers p_m are the Fourier coefficients of an integrable function and thus satisfy $p_m \to 0$ for $m \to \pm\infty$. They are known as the *attentuation factors* defined by the process P.

Theorem 2c delegates the responsibility for the accuracy of the Fourier coefficients entirely to the choice of the approximation operator P. This choice will be influenced by what the user knows subjectively about the function x. We consider two examples.

1. *Linear interpolation.* Here the data $\{x_k\}$ are interpolated by a piecewise linear function. It is clear that this process of approximation is linear and translation invariant. In particular, the delta sequence $\boldsymbol{\delta}$ is approximated by the function given in $(-\frac{1}{2}, \frac{1}{2})$ by

$$p(\tau) := \begin{cases} n - n^2 |\tau|, & |\tau| \leq 1/n, \\ 0, & \text{otherwise}, \end{cases}$$

and repeated periodically. For its Fourier coefficients we easily find $p_0 = 1$,

$$p_m = \left(\frac{n}{\pi m}\right)^2 \left(\sin \frac{\pi m}{n}\right)^2, \qquad m \neq 0,$$

confirming that $p_m \to 0$ for $m \to \pm\infty$.

2. *Interpolation by periodic cubic splines.* Here we interpolate the data $\mathbf{x}$ by a function $y = P\mathbf{x} \in \Pi$ satisfying the following conditions:

  (i) $y(\tau_k) = x_k$;

  (ii) in each interval $[\tau_k, \tau_{k+1}]$, y is represented by a cubic polynomial;

  (iii) y' is continuous;

  (iv) the integral

$$\int_0^1 |y''(\tau)|^2 \, d\tau$$

is as small as possible.

It can be shown that y is defined uniquely by these postulates. The resulting approximation operator P obviously is linear and translation invariant. Without going into the details of the construction of y, we mention that the attentuation factors are given by $p_0 = 1$,

$$p_m = \left(\frac{n}{\pi m}\right)^4 \left(\sin \frac{\pi m}{n}\right)^4 \frac{3}{2 + \cos (2\pi m/n)}, \qquad m \neq 0.$$

The foregoing results obviously can be extended to multivariate numerical harmonic analysis. Thus if $x(\tau, \sigma)$ is a function of two real variables which is periodic in each variable with period 1, then its Fourier coefficients

$$a_{km} := \int_0^1 \int_0^1 x(\tau, \sigma) \, e^{-2\pi i(k\tau + m\sigma)} \, d\tau \, d\sigma$$

are approximated by $\hat{a}_{km}$, where the doubly periodic sequence $\hat{a} = \{\hat{a}_{km}\}$ is the two-dimensional discrete Fourier transform

$$\hat{\mathbf{a}} := \mathscr{F}_n^{(2)} \mathbf{x}$$

of $\mathbf{x} = \{x_{km}\}$, $x_{km} := x(k/n, m/n)$, and it is fairly obvious how to extend the theory of error estimates and of attenuation factors to the multivariate case.

2.2. Numerical solution of Symm's equation. Two-dimensional Fourier analysis can be used in the numerical solution of an integral equation which arises in the numerical construction of conformal maps. Let Γ: $z = z(\tau)$, $0 \leq \tau \leq \beta$, be the boundary curve of a Jordan region D containing the point $z = 0$. We wish to determine the function f mapping D onto the unit disk $|w| < 1$ in such a manner that

$$f(0) = 0, \qquad f'(0) > 0.$$

To determine the mapping function, it is sufficient to know its values $f(z(\tau))$ on the boundary of D. Because $|f(z(\tau))| = 1$, it in turn suffices to know a continuous argument

$$\theta(\tau) = \arg (f(z(\tau))), \qquad 0 \leq \tau \leq \beta.$$

Any such continuous argument is known as an (interior) *boundary correspondence function* for the mapping f. The boundary correspondence function refers to a particular parametric representation of the boundary Γ, and thus is not determined by f alone. If Γ is analytic, then θ is likewise analytic.

It was shown recently by Gaier that there exists an intimate connection between the boundary correspondence function and the integral equation of the first kind

(2.7) $$\frac{1}{2\pi} \int_0^\beta \mathrm{Log}\, |z(\sigma) - z(\tau)| \xi(\tau) \, d\tau = \mathrm{Log}\, |z(\sigma)|, \qquad 0 \leq \sigma \leq \beta,$$

for an unknown function ξ, which was first used for purposes of numerical conformal mapping by Symm. If the capacity γ of Γ satisfies $\gamma \neq 1$, then Symm's equation (2.7) has the unique solution $\xi(\tau) = \theta'(\tau)$. The condition $\gamma \neq 1$ is satisfied, in particular, if the boundary Γ lies either entirely within or entirely without the unit circle. In these cases it is thus possible to determine θ' as the unique solution of (2.7).

Because the kernel, the nonhomogeneous part and the solution of (2.7) all can be extended as periodic functions with period β, the idea does not seem far fetched to seek the solution in the form of a Fourier series. Here the difficulty is at first encountered that the Fourier series for the kernel $\text{Log}\,|z(\sigma) - z(\tau)|$ converges only very slowly due to the logarithmic singularity at $\sigma = \tau$. Assuming $\beta = 2\pi$, we thus transform the equation into one with a smooth kernel by writing it in the form

$$(2.8) \quad \frac{1}{2\pi} \int_0^{2\pi} \text{Log}\,|e^{i\sigma} - e^{i\tau}| \theta'(\tau)\,d\tau + \frac{1}{2\pi} \int_0^{2\pi} \text{Log}\,\left|\frac{z(\sigma) - z(\tau)}{e^{i\sigma} - e^{i\tau}}\right| \theta'(\tau)\,d\tau = \text{Log}\,|z(\sigma)|.$$

Assuming the solution to be

$$\theta'(\tau) = \sum_{k=-\infty}^{\infty} t_k\, e^{ik\tau}, \qquad t_0 = 1, \qquad t_{-k} = \overline{t_k},$$

the second integral can be evaluated if the kernel is expanded in a double Fourier series,

$$\text{Log}\,\left|\frac{z(\sigma) - z(\tau)}{e^{i\sigma} - e^{i\tau}}\right| = \sum_{k,m=-\infty}^{\infty} l_{km}\, e^{ik\sigma + im\tau}.$$

To deal with the first integral, we use the fact that if m is an integer and σ is any real number,

$$\frac{1}{2\pi} \int_0^{2\pi} \text{Log}\,|e^{i\sigma} - e^{i\tau}|\, e^{im\tau}\,d\tau = \begin{cases} 0, & m = 0, \\ -\dfrac{1}{2|m|}\, e^{im\sigma}, & m \neq 0. \end{cases}$$

Expanding

$$\text{Log}\,|z(\sigma)| = \sum_{m=-\infty}^{\infty} z_m\, e^{im\sigma},$$

substituting into (2.8) and considering the mth Fourier component, we obtain the linear system

$$(2.9) \qquad -\frac{1}{2|m|} t_m + \sum_{k=-\infty}^{\infty} l_{km} t_{-k} = z_m, \qquad m = \pm 1, \pm 2, \cdots.$$

No equation is obtained for t_0, but from

$$t_0 = \frac{1}{2\pi} \int_0^{2\pi} \theta'(\tau)\,d\tau = \frac{1}{2\pi}[\theta(2\pi) - \theta(0)] = 1$$

t_0 is known to have the value 1.

The structure of the system (2.9) suggests an iterative procedure for solving it. Starting with a zeroth trial solution $\{t_m^{(0)}\}$, one determines a sequence of approximate solutions $\{t_m^{(j)}\}$ by

$$(2.10) \qquad \frac{1}{2|m|} t_m^{(j+1)} = \sum_{k=-\infty}^{\infty} l_{km} t_{-k}^{(j)} - z_m.$$

If

$$(2.11) \qquad 2\left\{\sum_{k,m=-\infty}^{\infty} |kl_{km}|^2\right\}^{1/2} < 1,$$

the convergence in the space l_2 follows from standard results in functional analysis. Because all $l_{km} = 0$ if Γ is a circle, (2.11) expresses a condition of "near-circularity" of the boundary curve. On the basis of considerable numerical evidence, this condition may be interpreted rather liberally; for instance, convergence to the exact solution to 9 decimal places takes place if Γ is a square.

It is of interest also to observe the workings of the iteration process (2.10) in a situation where it can be carried through analytically. Let $0 < \varepsilon < 1$, and consider the curve

$$\Gamma: z(\tau) = e^{i\tau} + \varepsilon e^{-i\tau},$$

an ellipse with semi-axes $1 + \varepsilon$ and $1 - \varepsilon$. Here

$$\frac{z(\sigma) - z(\tau)}{e^{i\sigma} - e^{i\tau}} = 1 + \varepsilon \frac{e^{-i\sigma} - e^{-i\tau}}{e^{i\sigma} - e^{i\tau}} = 1 - \varepsilon e^{-i(\sigma+\tau)},$$

and consequently

$$\mathrm{Log} \left|\frac{z(\sigma) - z(\tau)}{e^{i\sigma} - e^{i\tau}}\right| = \mathrm{Re}\, \mathrm{Log}\, (1 - \varepsilon e^{-i(\sigma+\tau)})$$

$$= -\mathrm{Re} \sum_{n=1}^{\infty} \frac{\varepsilon^n}{n} e^{-in(\sigma+\tau)}$$

$$= -\sum_{n=1}^{\infty} \frac{\varepsilon^n}{n} \{e^{in(\sigma+\tau)} + e^{-in(\sigma+\tau)}\}.$$

Furthermore

$$\mathrm{Log}\, |z(\sigma)| = \mathrm{Log}\, |e^{i\sigma} + \varepsilon e^{-i\sigma}|$$

$$= \mathrm{Re}\, \mathrm{Log}\, (1 + \varepsilon e^{-2i\sigma})$$

$$= \sum_{n=1}^{\infty} (-1)^{n-1} \frac{\varepsilon^n}{2n} \{e^{2i\sigma} + e^{-2i\sigma}\}.$$

Thus the system (2.9) in this case becomes

$$-\frac{1}{2m} t_m - \frac{\varepsilon^m}{2m} t_m = \begin{cases} 0, & \text{if } m \text{ is odd,} \\ (-1)^{l-1} \dfrac{\varepsilon^l}{m}, & \text{if } m \text{ is even, } m = 2l, \end{cases}$$

and $t_{-m} = t_m$. Thus the successive approximations are seen to form a geometric series with ratio ε, and we find

$$t_m = \begin{cases} 0, & m \text{ odd,} \\ 2\dfrac{(-1)^l \varepsilon^l}{1 + \varepsilon^{2l}}, & m \text{ even, } m = 2l. \end{cases}$$

The boundary correspondence function for the ellipse thus turns out to be

$$(2.12) \qquad \theta(\tau) = \tau + 2\sum_{m=1}^{\infty} \frac{(-1)^m}{m} \frac{\varepsilon^m}{1 + \varepsilon^{2m}} \sin(2m\tau).$$

2.3. Trigonometric interpolation. Suppose we wish to approximate a function $x \in \Pi$ by a trigonometric polynomial of period 1, that is, by a function of the form

$$t(\tau) = \sum_m a_m e^{2\pi i m \tau},$$

where only finitely many a_m are different from zero. Because the ideal representation of a periodic function is its Fourier series, a crude way to proceed would be thus: For sampled data, we first compute approximate values of the Fourier coefficients by (2.2),

$$\hat{a}_m := \frac{1}{n} \sum_{k=0}^{n-1} x_k w^{-km},$$

and then truncate the Fourier series formed with these approximate coefficients. Because the sequence $\hat{a} = \{\hat{a}_m\}$ is periodic with period n, it makes no sense to use more than n terms in the series. If n is odd, one thus would use the polynomial

(2.13a)
$$\hat{x}(\tau) := \sum_{|m| < n/2} \hat{a}_m e^{2\pi i m \tau},$$

if n is even,

(2.13b)
$$\hat{x}(\tau) := \sideset{}{'}\sum_{m=-n/2}^{n/2} \hat{a}_m e^{2\pi i m \tau},$$

the prime indicating that the terms where $m = \pm n/2$ are to be multiplied by $\frac{1}{2}$. (If x is real, then $\hat{a}_{-m} = \overline{\hat{a}_m}$ for all m; hence $\hat{a}_{-n/2} = \hat{a}_{n/2}$ is real. The prime convention then has the effect that the real form of $\hat{x}(\tau)$ has no term in $\sin(n\pi\tau)$.)

It would seem that two errors are committed in the above approximation: (a) the error committed in replacing the exact Fourier coefficients a_m by the approximate Fourier coefficients $\hat{a}_m$; (b) the error committed in truncating the Fourier series. Very miraculously it turns out that, at least at the sampling points τ_k, these two errors just cancel each other. That is, for $\tau = \tau_k$ the function $\hat{x}(\tau)$ not only *approximates*, but actually *interpolates* x.

THEOREM 2d. *Let* $x \in \Pi$, $\mathbf{x} = \{x_k\} = \{x(\tau_k)\}$ $(\tau_k := k/n, \ k \in \mathbb{Z})$, *and let* $\hat{x}$ *be the trigonometric polynomial* (2.13) *whose coefficients* $\hat{a}_m$ *are given by* (2.2). *Then for all* $k \in \mathbb{Z}$,

(2.14)
$$\hat{x}(\tau_k) = x(\tau_k).$$

Proof. Far from being a deep result, this is a mere corollary of the inversion formula for the discrete Fourier transform. We consider the case where n is even. In view of $\exp(2\pi i m \tau_k) = w^{mk}$,

$$\hat{x}(\tau_k) = \sideset{}{'}\sum_{m=-n/2}^{n/2} \hat{a}_m w^{km}.$$

Because $\hat{\mathbf{a}} \in \Pi_n$, this in view of $w^n = 1$ equals

$$\hat{x}(\tau_k) = \sum_{m=0}^{n-1} \hat{a}_m w^{km} = (n\bar{\mathscr{F}}_n \hat{\mathbf{a}})_k = (\mathscr{F}_n^{-1} \hat{\mathbf{a}})_k = x_k = x(\tau_k).$$

What can be said about the goodness of fit of $x(\tau)$ by $\hat{x}(\tau)$ at points τ that are not sampling points? We assume that the Fourier series of x represents x and is absolutely convergent,

$$x(\tau) = \sum_{m=-\infty}^{\infty} a_m e^{2\pi i m \tau}, \qquad \sum_{m=-\infty}^{\infty} |a_m| < \infty.$$

Using Theorem 2a we have, assuming n to be odd,

$$\hat{x}(\tau) = \sum_{-n/2 < m < n/2} \hat{a}_m \, e^{2\pi i m \tau}$$

$$= \sum_{-n/2 < m < n/2} \left(\sum_{l=-\infty}^{\infty} a_{m+nl} \right) e^{2\pi i m \tau}$$

$$= \sum_{-n/2 < m < n/2} a_m \, e^{2\pi i m \tau} + \sum_{-n/2 < m < n/2} \left(\sum_{\substack{l=-\infty \\ l \neq 0}}^{\infty} a_{m+nl} \right) e^{2\pi i m \tau};$$

therefore

$$\hat{x}(\tau) - x(\tau) = \sum_{-n/2 < m < n/2} \left(\sum_{\substack{l=-\infty \\ l \neq 0}}^{\infty} a_{m+nl} \right) e^{2\pi i m \tau} - \sum_{|k| > n/2} a_k \, e^{2\pi i k \tau}$$

and thus, if n is odd,

(2.15a) $$|\hat{x}(\tau) - x(\tau)| \leq 2 \sum_{|k| > n/2} |a_k|.$$

If n is even, a similar calculation yields

(2.15b) $$|\hat{x}(\tau) - x(\tau)| \leq |a_{n/2}| + |a_{-n/2}| + 2 \sum_{|k| > n/2} |a_k|.$$

The result, due to Gaier, is summarized in

THEOREM 2e. *If $x \in \Pi$ is represented by an absolutely convergent Fourier series, then the interpolating trigonometric polynomial $\hat{x}$ defined by* (2.13) *approximates x at all real τ with an error that is bounded as indicated in* (2.15).

We conclude, for instance, that for each fixed such $x \in \Pi$ the interpolating trigonometric polynomials tend to x uniformly as $n \to \infty$. If x is analytic, it follows as in the proof of Theorem 2b that the error tends to 0 at a geometric rate.

2.4. Conjugate periodic functions. If $D \subset \mathbb{C}$ is a region and if u is a real harmonic function in D, then any harmonic function v such that $f = u + iv$ is analytic in D is called a conjugate harmonic function of u. If D is simply connected, the conjugate function of a given harmonic function u always exists and is determined (by the Cauchy–Riemann equations) up to an additive constant.

We arrive at conjugate *periodic* functions by considering the special situation where D is the unit disk. Let u be harmonic in the unit disk $E: |z| < 1$, and assume that u can be extended to a continuous function on the closure $E': |z| \leq 1$. Then the boundary values of u define the continuous 2π-periodic function

$$\alpha(\theta) := u(e^{i\theta}), \qquad \theta \in \mathbb{R}.$$

Let v be the conjugate harmonic function of u, normalized by the condition

$$v(0) = 0.$$

If v likewise can be extended to a continuous function on E', then its boundary function

$$\beta(\theta) := v(e^{i\theta}), \qquad \theta \in \mathbb{R}$$

is called the *conjugate periodic function* of α and is denoted by

$$\beta = \Re \alpha.$$

Given α, it is easy to compute β if we assume that α is represented by an absolutely convergent Fourier series,

$$(2.16) \qquad\qquad \alpha(\theta) = \sum_{m=-\infty}^{\infty} a_m e^{im\theta}$$

$(a_{-m} = \overline{a_m}$ because α is real), where

$$\sum_{m=-\infty}^{\infty} |a_m| < \infty.$$

It is clear that

$$f(z) := a_0 + 2 \sum_{m=1}^{\infty} a_m z^m$$

is an analytic function whose real part has the boundary values (2.16), and since $f(0)$ is real, we have

$$\beta(\theta) = \operatorname{Im} f(r^{i\theta}) = \sum_{m=1}^{\infty} \frac{1}{i}(a_m e^{im\theta} - \overline{a_m} e^{-im\theta}).$$

Hence the conjugate periodic function of $\alpha(\theta)$ is

$$(2.17) \qquad\qquad \beta(\theta) = \sum_{m=-\infty}^{\infty} b_m e^{im\theta},$$

where

$$(2.18) \qquad\qquad b_m = \begin{cases} -ia_m, & m > 0, \\ 0, & m = 0, \\ ia_m, & m < 0. \end{cases}$$

By means of singular integrals the operator $\Re$ can be extended to the space $L_2(0, 2\pi)$; for further details, see Gaier (1964) and Zygmund (1968).

In order to approximately determine the conjugate periodic function of a function α whose values are known only at the sampling points $\theta_k = 2\pi k/n$, one may proceed as follows. One first determines the trigonometric polynomial

$$\hat{\alpha}(\theta) = \sum_{|m| \le n/2}' \hat{a}_m e^{im\theta}$$

which interpolates $\alpha(\theta)$ at the sampling points. This requires the computation of the sequence $\hat{\mathbf{a}} := \mathscr{F}_n \alpha$, and thus if $n = 2^l$ can be done in $O(n \operatorname{Log}_2 n)$ operations. One then considers the trigonometric polynomial

$$\hat{\beta}(\theta) := \Re\hat{\alpha}(\theta) = \sum_{|m| \le n/2}' \hat{b}_m e^{im\theta}$$

whose coefficients are given by (2.18), as an approximation to $\beta := \Re\alpha$. If the values of $\hat{\beta}$ at the sampling points are desired, they may be calculated by a Fast Fourier Transform in another $O(n \operatorname{Log}_2 n)$ operations.

Concerning $\hat{\beta} - \beta$, the elementary technique used in proving Theorem 2e yields the following estimate, again due to Gaier:

THEOREM 2f. *Let α be a real 2π-periodic function whose complex Fourier series (2.16) is absolutely convergent, and let $\hat{\beta}$ be the approximation to $\beta := \Re\alpha$ determined*

above. Then for all real θ,

$$|\hat{\beta}(\theta) - \beta(\theta)| \leqq \begin{cases} 2|a_n| + 4 \sum_{k>n} |a_k|, & n \text{ even,} \\ 4 \sum_{k>n} |a_k|, & n \text{ odd.} \end{cases}$$

2.5. Solution of Theodorsen's integral equation. Conjugate periodic functions play a role in a method for the construction of conformal maps based on an integral equation due to Theodorsen. Contrary to the methods based on Symm's equation and equations related to it, Theodorsen's method determines the mapping function from the unit disk to the given domain. Furthermore, Theodorsen's equation is nonlinear, and iteration is the only practically available method for solving it.

Let the boundary curve Γ of the given region D be star-like with respect to 0, and thus be representable in polar coordinates,

$$z(\phi) = \rho(\phi) e^{i\phi}, \qquad 0 \leqq \phi \leqq 2\pi,$$

where $\rho(\phi)$ is continuous and piecewise analytic. If g maps the unit disk E onto D such that

$$g(0) = 0, \qquad g'(0) > 0,$$

we now call *boundary correspondence function* any continuous function ϕ such that

$$\phi(\theta) = \arg g(r^{i\theta}), \qquad 0 \leqq \theta \leqq 2\pi.$$

Theodorsen's method aims at determining ϕ.

It is clear that the function $w^{-1} g(w)$ is analytic in E and does not vanish. Hence it possesses a single-valued analytic logarithm

(2.19) $$h(w) := \log \frac{g(w)}{w},$$

which we define uniquely by requiring that

$$h(0) = \log g'(0) \text{ is real.}$$

Obviously h can be extended to a function that is continuous in $|w| \leqq 1$. Then

$$\alpha(\theta) = \operatorname{Re} h(e^{i\theta})$$

has the conjugate periodic function

$$\beta(\theta) = \operatorname{Im} h(e^{i\theta}).$$

But from (2.19),

$$\alpha(\theta) = \operatorname{Log} \left| \frac{g(e^{i\theta})}{e^{i\theta}} \right| = \operatorname{Log} \rho(\phi(\theta)),$$

$$\beta(\theta) = \arg \frac{g(e^{i\theta})}{e^{i\theta}} = \phi(\theta) - \theta.$$

We thus have:

THEOREM 2g. *Under the above hypotheses, the boundary correspondence function* $\phi(\theta)$ *satisfies*

(2.20) $$\phi - \theta = \Re \operatorname{Log} \rho(\phi).$$

If the operator $\mathfrak{R}$ is represented by a principal value integral as mentioned earlier, (2.20) becomes an integral equation which is generally referred to as the *Theodorsen integral equation*. If ρ satisfies an ε-condition, which is to say that there exists $\varepsilon < 1$ such that

$$(2.21) \qquad \left| \frac{\rho'(\phi)}{\rho(\phi)} \right| \leqq \varepsilon$$

for all ϕ for which ρ' exists, then it can be shown that $\phi(\theta)$ is the only solution of (2.20). Under the same condition it may be shown that the analytical iteration

$$(2.22) \qquad \phi^{(j+1)} - \theta = \mathfrak{R} \operatorname{Log} \rho(\phi^{(j)}), \qquad j = 0, 1, \cdots ,$$

converges to the true solution.

To implement the iteration (2.22) numerically, we replace the exact periodic iterates $\phi^{(j)} - \theta$ by trigonometric polynomials. For this we choose a fixed number n of interpolating points which if a Fast Fourier Transform is used should be a power of 2. The functions $\tilde{\phi}^{(j)}$ approximating the $\phi^{(j)}$ are defined by

$$(2.23) \qquad \tilde{\phi}^{(j+1)}(\theta) = \theta + \mathfrak{R}[\operatorname{Log} \rho(\tilde{\phi}^{(j)}(\theta)]\hat{} ,$$

$j = 0, 1, \cdots$, where $\hat{}$ denotes trigonometric interpolation as described above, and where $\tilde{\phi}^{(0)}(\theta)$ is usually chosen as θ. Each step of the iteration requires the evaluation of $\rho(\phi)$ at n points, plus two discrete Fourier transforms. The advantages of using a Fast Fourier algorithm are obvious. It may be shown that if $\rho(\phi)$ satisfies an ε-condition where $\varepsilon < 1$, the sequence $\{\tilde{\phi}^{(j)}\}$ converges to a solution of

$$\tilde{\phi}(\theta) = \theta + \mathfrak{R}[\operatorname{Log} \rho(\tilde{\phi}(\theta))]\hat{} .$$

Both the iteration error $|\phi^{(j)}(\theta) - \phi(\theta)|$ and the discretization error $|\tilde{\phi}(\theta) - \phi(\theta)|$ may be estimated. We refer to Gaier (1964) for analytical details. It should be mentioned that according to results of Niethammer (1966) and Gutknecht (1977), the iteration may still converge if $\varepsilon \geqq 1$ if a suitable form of under-relaxation is used.

2.6. Doubly connected regions. We begin by discussing a generalization of the operator $\mathfrak{R}$ introduced in § 2.4. Let α_0 and α_1 be two real 2π-periodic functions given by their Fourier series

$$\alpha_j(\theta) = \sum_{n=-\infty}^{\infty} a_{j,n} e^{in\theta}, \qquad j = 0, 1,$$

$(a_{j,-n} = \overline{a_{j,n}})$, which we assume to be absolutely convergent. Let $0 < \mu < 1$, and let A be the annulus $\mu < |w| < 1$. We wish to carry out the following program:

   (i) To solve the Dirichlet problem of finding a function u that is harmonic in A, continuous in the closure A' of A, and that satisfies the boundary conditions

$$(2.24) \qquad \begin{aligned} u(e^{i\theta}) &= \alpha_0(\theta), \\ u(\mu e^{i\theta}) &= \alpha_1(\theta) \end{aligned}$$

for $0 \leqq \theta \leqq 2\pi$;

   (ii) To construct, if possible, a conjugate harmonic function v of u;

   (iii) To extend v continuously to A', if possible, and to express the boundary values of v directly in terms of the functions α_0 and α_1 or their Fourier constants.

Suppressing the details of the computation, we note that u can easily be expressed by a Fourier series, and that the necessary and sufficient condition for the existence of v

is

$$\int_0^{2\pi} \alpha_0(\theta)\, d\theta = \int_0^{2\pi} \alpha_1(\theta)\, d\theta$$

or $a_{1,0} = a_{0,0}$. If this condition is met, the boundary values

$$\beta_0(\theta) := v(e^{i\theta}) \quad \text{and} \quad \beta_1(\theta) := v(\mu\, e^{i\theta})$$

are found to be

(2.25)
$$\beta_0 = \mathfrak{H}_\mu \alpha_0 - \mathfrak{G}_\mu \alpha_1,$$
$$\beta_1 = \mathfrak{G}_\mu \alpha_0 - \mathfrak{H}_\mu \alpha_1,$$

where the operators $\mathfrak{H}_\mu$ and $\mathfrak{G}_\mu$ are defined as follows: If

$$\alpha(\theta) = \sum_{n=-\infty}^{\infty} a_n e^{in\theta}$$

and if the Fourier series is absolutely convergent, then

$$\mathfrak{H}_\mu \alpha(\theta) = \sum_{n=-\infty}^{\infty} b_n e^{in\theta},$$

where $b_0 = 0$,

$$b_n = -i \frac{1 + \mu^{2n}}{1 - \mu^{2n}} a_n, \qquad n \neq 0,$$

and

$$\mathfrak{G}_\mu \alpha(\theta) = \sum_{n=-\infty}^{\infty} c_n e^{in\theta},$$

where $c_0 = 0$,

$$c_n = -i \frac{2\mu^n}{1 - \mu^{2n}} a_n, \qquad n \neq 0.$$

Let now D be a doubly connected region bounded on the outside by a Jordan curve Γ_0 and on the inside by a Jordan curve Γ_1. We assume that Γ_j is represented in polar coordinates as

$$z = \rho_j(\phi)\, e^{i\phi}, \qquad 0 \leq \phi \leq 2\pi, \quad j = 0, 1,$$

Let the annulus $A: \mu < |w| < 1$ be conformally equivalent to D, and let g be a map from A to D. It is known that g possesses a continuous extension from A' to D', and thus that continuous outer and inner boundary correspondence functions

$$\phi_0(\theta) = \arg g(e^{i\theta}), \qquad \phi_1(\theta) = \arg g(\mu\, e^{i\theta})$$

may be defined. The function

$$h(w) := \log \frac{g(w)}{w}$$

may be defined as an analytic function in A. Its boundary values are

$$h(e^{i\theta}) = \text{Log } \rho_0(\phi_0(\theta)) + i[\phi_0(\theta) - \theta]$$

and, if the undetermined additive multiple of 2π in ϕ_1 is chosen correctly,

$$h(\mu\, e^{i\theta}) = \text{Log } \rho_1(\phi_1(\theta)) + i[\phi_1(\theta) - \theta].$$

Because the integral

$$\frac{1}{2\pi i} \int_{\Gamma_\rho} \frac{h(w)}{w}\, dw$$

has the same value along all circles $\Gamma_\rho \colon |w| = \rho$, it follows that

$$\frac{1}{2\pi} \int_0^{2\pi} h(e^{i\theta})\, d\theta = \frac{1}{2\pi} \int_0^{2\pi} h(\mu\, e^{i\theta})\, d\theta,$$

which on separating real and imaginary parts yields

$$(2.26) \qquad \text{Log } \mu = \frac{1}{2\pi} \int_0^{2\pi} \text{Log } \frac{\rho_1(\phi_1(\theta))}{\rho_0(\phi_0(\theta))}\, d\theta,$$

$$(2.27) \qquad \int_0^{2\pi} (\phi_0(\theta) - \theta)\, d\theta = \int_0^{2\pi} (\phi_1(\theta) - \theta)\, d\theta.$$

By rotating the annulus it can be arranged that both integrals (2.27) are zero. With the operators $\mathfrak{H}_\mu$ and $\mathfrak{G}_\mu$ defined previously the system of equations

$$(2.28) \qquad \begin{aligned} \phi_0 - \theta &= \mathfrak{H}_\mu \text{ Log } \rho_0(\phi_0) - \mathfrak{G}_\mu \text{ Log } \rho_1(\phi_1), \\ \phi_1 - \theta &= \mathfrak{G}_\mu \text{ Log } \rho_0(\phi_0) - \mathfrak{H}_\mu \text{ Log } \rho_1(\phi_1) \end{aligned}$$

holds. As in the case of a simply connected region, these equations may be used to set up an iteration for the approximate determination of the boundary correspondence functions. There is the new difficulty that the number μ is not known a priori. This may be overcome by using (2.26) to calculate an approximate value of μ at each step of the iteration. For the numerical implementation, only the values $\phi_j(\theta_k)$ $(\theta_k := 2\pi k/n)$ are used, and the functions $\text{Log } \rho_j(\phi_j(\theta))$ are replaced by interpolating trigonometric polynomials. Four discrete Fourier transforms are required at each step of the iteration.

Notes on § 2. § 2.1. On integrating periodic analytic functions see Davis (1959). Attenuation factors are discussed in a fundamental paper by Gautschi (1972) with an extensive list of references. For applications of multivariate harmonic analysis in picture processing see Rosenfeld and Kak (1976).

§ 2.2. On Symm's equation see Symm (1966) and the considerably deeper treatment by Gaier (1976). Inequalities for the capacity are found in Pólya and Szegö (1954, problems IV 97–120). The Fourier method for solving Symm's equation has been tested in a large number of examples by J.-P. Berrut (Master's thesis, Swiss Federal Institute of Technology). Another method for solving Symm's equation is discussed in Hayes, Kahaner, and Keller (1972).

§ 2.3. Trigonometric interpolation is treated in many numerical analysis texts, see the notes on § 1.1. For the error estimate (2.15) see Gaier (1974).

§ 2.4. On conjugate periodic functions see Hardy and Rogosinski (1944), Zygmund (1968) and, more in the present context, Gaier (1964). For Theorem 2f see Gaier (1974). To compute conjugate periodic functions by Fast Fourier Transforms is a proposal by Henrici (1976); for a more thorough treatment see Gutknecht (1978).

§ 2.5. For Theodorsen's integral equation see Gaier (1964). The existence of a solution of the discretized equation was established by Gutknecht (1977).

§ 2.6. Theodorsen's equation for doubly connected regions is again discussed by Gaier (1964). The method of solution using Fourier series and explicit representations of the operators $\mathfrak{H}_\mu$ and $\mathfrak{G}_\mu$ which we propose here is unpublished. It has been tested on a large number of examples by C. Lundwall (Swiss Federal Institute of Technology).

3. Cauchy theory.

3.1. The Laurent series. Let f be analytic in the annulus $A: \rho_1 < |z| < \rho_2$, where $0 \le \rho_1 < 1 < \rho_2$. (Any annulus may be reduced to an annulus of this special type by a linear change of variables.) Then f is represented in A by its Laurent series,

$$(3.1) \qquad f(z) = \sum_{m=-\infty}^{\infty} a_m z^m, \qquad z \in A,$$

where the a_m are the Laurent coefficients of f for A,

$$(3.2) \qquad a_m = \frac{1}{2\pi i} \int_{|z|=1} z^{-m-1} f(z)\, dz.$$

On setting $z = e^{i\tau}$ this becomes

$$(3.3) \qquad a_m = \frac{1}{2\pi} \int_0^{2\pi} e^{-im\tau} f(e^{i\tau})\, d\tau,$$

and we see that a_m also is the mth *Fourier* coefficient of the 2π-periodic function $\tau \mapsto f(e^{i\tau})$. Thus for instance if $n = 2^l$, $w := \exp(2\pi i/n)$, $f_m := f(w^m)$, and

$$\mathbf{f} := \{f_m\} \in \Pi_n,$$

the coefficients a_m with $|m|$ sufficiently small will be approximated by those of the sequence

$$\hat{\mathbf{a}} := \{\hat{a}_m\} = \mathscr{F}_n \mathbf{f},$$

which by a Fast Fourier Transform can be calculated in $O(n \operatorname{Log}_2 n)$ operations. The error in this approximation by Theorem 2a is given by

$$(3.4) \qquad \hat{a}_m - a_m = a_{m+n} + a_{m+2n} + \cdots + a_{m-n} + a_{m-2n} + \cdots.$$

If $\rho_1 < \rho < \rho_2$ and

$$\mu(\rho) := \max_{|z|=\rho} |f(z)|,$$

then by Cauchy's estimate

$$|a_{m+kn}| \le \frac{\mu(\rho)}{\rho^{m+kn}}.$$

Thus if $\mu^*(\rho) := \max(\mu(\rho), \mu(\rho^{-1}))$ and $\rho_1 < \rho^{-1} < \rho < \rho_2$, we obtain the following estimate for the error committed in approximating Laurent coefficients by discrete Fourier transforms:

$$(3.5) \qquad |\hat{a}_m - a_m| \le \mu^*(\rho)(\rho^m + \rho^{-m})\frac{1}{\rho^n - 1}, \qquad m \in Z.$$

We point out some applications of the implied algorithm for numerically calculating Laurent coefficients.

(a) *Numerical differentiation of analytic functions.* Let g be analytic in the disk $|z - z_0| < \sigma$. Then the function

$$f(z) := g(z_0 + \rho z)$$

satisfies the above hypotheses for every ρ such that $0 < \rho < \sigma$, and

(3.6)
$$a_m = \begin{cases} 0, & m < 0, \\ \dfrac{\rho^m}{m!} g^{(m)}(z_0), & m \geq 0. \end{cases}$$

Thus the coefficients $\hat{a}_m$ may be used to approximate the derivatives of g at z_0. If g is a polynomial of degree $< n$, (3.4) shows that $\hat{a}_m = a_m$ for $0 \leq m < n$, and the exact values of the derivatives may be obtained from (3.6). For general analytic functions, an adaptation of (3.5) shows that every fixed derivative is obtained with geometric convergence as $n \to \infty$.

The algorithm in all its simplicity highlights some of the typical differences between analytic and non analytic functions:

(i) The points where g is evaluated do not lie on a straight line, as in real numerical differentiation, but on a circle. There is no preferred direction in the complex plane. The derivative, if defined as limit of a difference quotient, is independent of the manner in which the increment tends to zero.

(ii) Even if the derivatives at z_0 are to be calculated to arbitrary precision, the points where g is evaluated do not tend to z_0. The values of g on the circle $|z - z_0| = \rho$ completely determine g.

(iii) A large number of derivatives (theoretically: all) are evaluated simultaneously. There is no question about the existence of derivatives. If the first derivative exists in a neighborhood of z_0, all derivatives exist.

Numerical analysts will also note that due to the stability of the Fast Fourier Transforms (see § 4.3) the foregoing algorithm for numerical differentiation is stable no matter how high the accuracy of the formulas. This contrasts favorably with high-order difference formulas for real numerical differentiation which tend to become unstable due to cancellation of large terms of opposite signs.

(b) *Generating functions.* If $\mathbf{a} = \{a_m\}_{m=-\infty}^{\infty}$ is a sequence of complex numbers (not assumed to be periodic), and if the series

$$f(t) = \sum_{m=-\infty}^{\infty} a_m t^m$$

converges in a suitable annulus, then f is called the *generating function* of the sequence $\mathbf{a}$. (Frequently in applications, the a_k, and hence f, depend on additional parameters.) The discrete Fourier transform may be used to generate the elements of $\mathbf{a}$ by evaluations of its generating function. We have experimented with the well-known example

$$f(t, x) = e^{(x/2)(t-t^{-1})} = \sum_{m=-\infty}^{\infty} J_m(x) t^m$$

(J_m = Bessel function of order m) and have found that even in this simple case the method is competitive with the well-known trick of using the three-term recurrence relation backwards. In an example like

$$f(t, x) = \frac{t e^{xt}}{e^t - 1} = \sum_{m=0}^{\infty} \frac{B_m(x)}{m!} t^m$$

$(B_m(x) = $ Bernoulli polynomial) the usual recurrence relation requires $O(n^2)$ operations to generate $B_0, \cdots, B_n$, whereas a Fast Fourier Transform requires only $O(n \operatorname{Log} n)$ operations to generate approximations to the same numbers.

(c) *Numerical inversion of Laplace transforms.* The problem of numerically inverting the Laplace transform, i.e., of computing values of an original function $F(\tau)$ from its image function

$$f(s) := \int_0^\infty F(\tau) e^{-s\tau} d\tau,$$

is much discussed in the engineering literature. Fast Fourier Transforms are essential for the numerical implementation of a technique which makes use of the following function-theoretic facts satisfied by any Laplace transform f:

(i) f is analytic in a half-plane Re $s > \gamma_F$, where γ_F is the growth indicator of F.

(ii) $f(s) \to 0$ if $s \to \infty$ in the half-plane just described.

By considering $e^{-\gamma_F \tau} F(\tau)$ in place of F, we may assume that $\gamma_F = 0$. Every f analytic in Re $s > 0$ can be represented as a series of powers of

$$z := \frac{s - \alpha}{s + \alpha}$$

where $\alpha > 0$, because for any such choice of α, $|z| < 1$ precisely if Re $s > 0$. It thus seems natural to expand $f \bullet\!\!-\!\!\circ F$ in powers of z and thus to obtain an expansion of F in terms of the original functions of z^m. However, there are no such original functions, because z^m does not satisfy (ii). On the other hand, the functions

$$l_m(s) = \frac{z^m}{s + \alpha} = \frac{(s - \alpha)^m}{(s + \alpha)^{m+1}}$$

are image functions, and their original functions are readily expressed in terms of the Laguerre polynomials

$$L_m(\tau) := \frac{1}{m!} e^\tau \frac{d^m}{d\tau^m}(e^{-\tau} \tau^m).$$

The precise correspondence is

$$l_m(s) \bullet\!\!-\!\!\circ e^{-\alpha\tau} L_m(2\alpha\tau).$$

Thus the problem is reduced to determining the coefficients a_m in the expansion

$$f(s) = \sum_{m=0}^\infty a_m \frac{(s - \alpha)^m}{(s + \alpha)^{m+1}}$$

which also may be written

$$(3.7) \qquad\qquad\qquad g(z) = \sum_{m=0}^\infty a_m z^m$$

where

$$(3.8) \qquad\qquad g(z) := (s + \alpha) f(s) = \frac{2\alpha}{1 - z} f\left(\alpha \frac{1 + z}{1 - z}\right).$$

If the a_m have been determined, then at least formally

$$(3.9) \qquad\qquad F(\tau) = e^{-\alpha\tau} \sum_{m=0}^\infty a_m L_m(2\alpha\tau);$$

for conditions of validity of this expansion see Doetsch (1950, p. 301). The values of $L_m(2\alpha\tau)$ required for the numerical evaluation of the series (3.9) may be generated rapidly by the forward recurrence relation satisfied by the Laguerre polynomials which turns out to be stable.

What makes this method numerically feasible is the fact that the Taylor coefficients a_m can be evaluated efficiently by the technique which we have discussed at the outset of this section. Let $n = 2^l$ be a (large) integer, $w := \exp(2\pi i/n)$, select ρ, $0 < \rho < 1$, and define $\mathbf{f} = \{f_m\}$ where

$$f_m := \frac{2\alpha}{1 - \rho w^m} f\left(\alpha \frac{1 + \rho w^m}{1 - \rho w^m}\right).$$

The sequence

$$\mathbf{b} = \{b_m\} = \mathscr{F}_n \mathbf{f}$$

can then be evaluated in $O(n \operatorname{Log} n)$ operations, and the desired coefficients a_m are approximated by $\rho^{-m} b_m$ with an error that tends to zero geometrically as $n \to \infty$.

3.2. Incomplete factoring of polynomials. Here we consider the following problem: We are given a polynomial p, a complex number z_0, and a real number $\rho > 0$. Assuming that p has no zeros on the circle $|z - z_0| = \rho$, we wish to construct the polynomial p_1 whose zeros are precisely the zeros of p satisfying $|z - z_0| < \rho$.

If this problem could be solved efficiently, a new approach to the problem of determining the zeros of high degree polynomials might be feasible. Instead of determining the zeros one by one and deflating, as in the conventional approach, or instead of trying to isolate the zeros by a Weyl type exclusion algorithm, one would, by covering the plane by disks and determining the polynomial corresponding to the zeros in each disk, immediately break down the problem into smaller subproblems. If necessary, the process could be iterated, much like the well-known Lehmer method; on the other hand, even if not carried to its conclusion where each disk contains at most one zero, the algorithm would still permit conclusions about the distribution or "density" of zeros.

By a shift of variable, it suffices to consider our problem in the special case where $z_0 = 0$, $\rho = 1$. Thus let p be our polynomial, and let the annulus $A: \gamma < |z| < \gamma^{-1}$ be free of zeros of p. By the algorithm described in § 3.1, a Fast Fourier Transform may be used to generate approximate values $\hat{a}_m$ of the coefficients a_m of the Laurent series of the function

$$f(z) := \frac{p'(z)}{p(z)}$$

in A,

$$f(z) = \sum_{m=-\infty}^{\infty} a_m z^m, \qquad z \in A.$$

On the other hand, these coefficients may be calculated explicitly. If the zeros of p are $z_1, \cdots, z_d$, we have, as is well known,

$$f(z) = \frac{p'(z)}{p(z)} = \sum_{i=1}^{d} \frac{1}{z - z_i},$$

and if $|z_i| \leqq \gamma$, $i = 1, 2, \cdots, k$, and $|z_i| \geqq \gamma^{-1}$, $i > k$, then

$$
\sum_{i=1}^{d} \frac{1}{z - z_i} = \sum_{i=1}^{k} \frac{1}{z - z_i} - \sum_{i=k+1}^{d} \frac{1}{z_i - z}
$$

$$
= \sum_{i=1}^{k} \frac{1}{z} \frac{1}{1 - (z_i/z)} - \sum_{i=k+1}^{d} \frac{1}{z_i} \frac{1}{1 - (z/z_i)}
$$

$$
= \sum_{m=1}^{\infty} z^{-m} \sum_{i=1}^{k} z_i^{m-1} - \sum_{m=0}^{\infty} z^m \sum_{i=k+1}^{d} z_i^{-m-1}.
$$

Thus if we define, for arbitrary integers m, interior power sums s_m and exterior power sums t_m by

$$
s_m := \sum_{i=1}^{k} z_i^m, \qquad t_m := \sum_{i=k+1}^{d} z_i^m,
$$

we see that

$$
\begin{aligned}
a_{-m} &= s_{m-1}, & m &= 1, 2, \cdots, \\
a_m &= -t_{-m-1}, & m &= 0, 1, 2, \cdots.
\end{aligned} \tag{3.10}
$$

In particular,

$$
a_{-1} = s_0 = k,
$$

the number of zeros of p inside $|z| = 1$; furthermore, if there is only one such zero,

$$
a_{-2} = s_1 = z_1.
$$

In general, if the interior power sums are known to sufficient accuracy, the polynomial

$$
p_1(z) = \prod_{i=1}^{k} (z - z_i) =: z^k + b_1 z^{k-1} + \cdots + b_k
$$

having the zeros $z_1, \cdots, z_k$ may be constructed as follows: We consider the reciprocal polynomial

$$
q_1(z) := z^k p(z^{-1}) = \prod_{i=1}^{k} (1 - z z_i)
$$

$$
= 1 + b_1 z + \cdots + b_k z^k,
$$

which satisfies

$$
\frac{q_1'(z)}{q_1(z)} = - \sum_{m=1}^{\infty} s_m z^{m-1}.
$$

If the s_m are known, the b_j may be determined by comparing coefficients in

$$
\sum_{j=1}^{k} j b_j z^{j-1} = - \sum_{m=1}^{\infty} s_m z^{m-1} \sum_{j=0}^{k} b_j z^j
$$

$(b_0 := 1)$, which yields the recurrence relations

$$
(3.11) \qquad b_m = -\frac{1}{m}(s_m b_0 + s_{m-1} b_1 + \cdots + s_1 b_{m-1}), \qquad m = 1, 2, \cdots.
$$

The accuracy of the approximations $\hat{a}_{-m}$ to s_{m-1} may be assessed by Theorem 2a. In view of (3.10) we have for $1 \leqq m \leqq n$

$$\hat{a}_{-m} - s_{m-1} = s_{m+n-1} + s_{m+2n-1} + \cdots + t_{m-n-1} + t_{m-2n-1} + \cdots,$$

hence

$$|\hat{a}_{-m} - s_{m-1}| \leqq \frac{\gamma^n}{1 - \gamma^n} \{k\gamma^{m-1} + (d-k)\gamma^{-m+1}\}.$$

In addition, the correct choice of n is facilitated by the following considerations:

(i) Because $s_0 = k$, the approximate value $\hat{a}_{-1}$ should be, within the permitted tolerance, an integer.

(ii) For the integer k approximately determined by $\hat{a}_{-1}$, the coefficients b_j determined by (3.11) should automatically become zero for $j > k$.

The computation of the Laurent coefficients by a discrete Fourier transform requires the evaluation of the function $f = p'/p$ at the points $z = w^m$, $m = 0, 1, \cdots, n-1$. By separately evaluating numerator and denominator, this task itself may be carried out by Fast Fourier Transforms.

To determine $b_1, \cdots, b_k$ from (3.11) requires $O(k^2)$ operations which may be unpleasant if k is large. In § 5.4 we discuss an algorithm for accomplishing the same purpose which requires only $O(k \operatorname{Log} k)$ operations.

Notes on § 3. § 3.1. The idea of using Cauchy's formula for numerical differentiation seems to have been first suggested by Lyness and Moler (1967). Experiments on generating sequences from generating functions by Fast Fourier Transforms have been performed by P. Geiger (unpublished). For the theoretical background of the method of inverting the Laplace transform see Henrici (1977, § 10.5). For an implementation see Wing (1967).

§ 3.2. The algorithm described here has been implemented successfully by P. Geiger (to be published).

4. Convolution. In this section we discuss two kinds of multiplication in Π_n, called *Hadamard multiplication* and *convolution*. We study their behavior under discrete Fourier transformation and describe some of their applications.

4.1. Multiplication in Π_n. Let $\mathbf{x} = \{x_k\}$ and $\mathbf{y} = \{y_k\}$ be sequences in π_n. We have already agreed to define scalar multiplication in Π_n by

$$c\mathbf{x} := \{cx_k\}$$

for any $c \in \mathbb{C}$. We next define a product of two sequences called *Hadamard product* by

(4.1) $$\mathbf{x} \cdot \mathbf{y} := \{x_k y_k\}.$$

(We always write the dot for clarity.) It is clear that the Hadamard product is commutative, associative, and (with respect to the addition defined earlier) distributive. Under addition and Hadamard multiplication the space Π_n becomes a commutative ring. For $n > 1$ this ring has divisors of zero.

We discover a second, mathematically more interesting kind of multiplication by relating the Fourier transform of $\mathbf{x} \cdot \mathbf{y}$ to the transforms of $\mathbf{x}$ and of $\mathbf{y}$. Let $\mathbf{x}, \mathbf{y} \in \Pi_n$, and

$$\mathbf{u} := \mathscr{F}_n \mathbf{x}, \qquad \mathbf{v} := \mathscr{F}_n \mathbf{y}.$$

By definition,

$$(\mathscr{F}_n(\mathbf{x} \cdot \mathbf{y}))_m = \frac{1}{n} \sum_{k=0}^{n-1} x_k y_k w^{-km}.$$

Replacing y_k by $(\mathscr{F}_n^{-1}\mathbf{v})_k$ this in view of the discrete inversion formula becomes

$$\frac{1}{n}\sum_{k=0}^{n-1}x_k\left(\sum_{l=0}^{n-1}v_l w^{lk}\right)w^{-km} = \sum_{l=0}^{n-1}v_l\left(\frac{1}{n}\sum_{k=0}^{n-1}x_k w^{-(m-l)k}\right).$$

The inner sum equals $(\mathscr{F}_n\mathbf{x})_{m-l} = u_{m-l}$, and we thus have

$$(4.2) \qquad (\mathscr{F}_n(\mathbf{x}\cdot\mathbf{y}))_m = \sum_{l=0}^{n-1}v_l u_{m-l}.$$

For any two sequences $\mathbf{u} = \{u_k\}$ and $\mathbf{v} = \{v_k\}$ in Π_n, the sequence $\mathbf{c} = \{c_m\}$ defined by

$$(4.3) \qquad c_m = \sum_{k=0}^{n-1}u_k v_{m-k} = \sum_{k=0}^{n-1}v_k u_{m-k}$$

is denoted by

$$\mathbf{c} = \mathbf{u} * \mathbf{v}$$

and is called the *convolution* of $\mathbf{u}$ and $\mathbf{v}$. All sums in (4.3) have the same number of terms. (In order to distinguish $\mathbf{u} * \mathbf{v}$ from other kinds of convolution, this is also called *wrapped-around convolution* or *convolution on the circle*.) In terms of the convolution product,

$$(4.4) \qquad \mathscr{F}_n(\mathbf{x}\cdot\mathbf{y}) = \mathscr{F}_n\mathbf{x} * \mathscr{F}_n\mathbf{y}.$$

Any sequence $\mathbf{x}\in\Pi_n$ is the image under $\mathscr{F}_n$ of some sequence in Π_n. Thus, writing $\mathbf{x}$, $\mathbf{y}$ for $\mathscr{F}_n\mathbf{x}$ and $\mathscr{F}_n\mathbf{y}$, we also have, using properties of the reversion operator R established earlier,

$$\mathbf{x} * \mathbf{y} = \mathscr{F}_n(\mathscr{F}_n^{-1}\mathbf{x}\cdot\mathscr{F}_n^{-1}\mathbf{y})$$
$$= n^2\mathscr{F}_n(R\mathscr{F}_n\mathbf{x}\cdot R\mathscr{F}_n\mathbf{y})$$
$$= n^2\mathscr{F}_n R(\mathscr{F}_n\mathbf{x}\cdot\mathscr{F}_n\mathbf{y})$$
$$= n^2 R\mathscr{F}_n(\mathscr{F}_n\mathbf{x}\cdot\mathscr{F}_n\mathbf{y}),$$

hence in view of $\mathscr{F}_n R = (1/n)\mathscr{F}_n^{-1}$

$$\mathscr{F}_n(\mathbf{x} * \mathbf{y}) = n\mathscr{F}_n\mathbf{x}\cdot\mathscr{F}_n\mathbf{y}.$$

These results are summarized in the *convolution theorem*:

THEOREM 4a. *For arbitrary sequences* $\mathbf{x}, \mathbf{y}\in\Pi_n$,

$$(4.5) \qquad \begin{array}{l}\mathscr{F}_n(\mathbf{x}\cdot\mathbf{y}) = \mathscr{F}_n\mathbf{x} * \mathscr{F}_n\mathbf{y}, \\[2mm] \mathscr{F}_n(\mathbf{x} * \mathbf{y}) = n\mathscr{F}_n\mathbf{x}\cdot\mathscr{F}_n\mathbf{y}.\end{array}$$

The convolution of periodic sequences is important not only in connection with Fourier analysis, but will be shown to also have many other significant applications. If performed in a straightforward way by direct evaluations of the sums (4.3), the convolution of two sequences in Π_n evidently requires n^2 μ and a similar number of additions (α). If n is large, this is prohibitively expensive. We thus call particular attention to the relation

$$(4.6) \qquad \mathbf{x} * \mathbf{y} = n^2 R\mathscr{F}_n(\mathscr{F}_n\mathbf{x}\cdot\mathscr{F}_n\mathbf{y})$$

established in the course of the proof of the convolution theorem. It shows that the convolution can be computed by taking three discrete Fourier transforms and forming

one Hadamard product. Using Fast Fourier Transforms, this can reduce the time required by orders of magnitude. In view of Theorem 1b we have, for instance,

THEOREM 4b. *If $n = 2^l$, the convolution of two sequences in Π_n, if performed via (4.6) using the Fast Fourier Transform, requires no more than*

$$\tfrac{3}{2} n \, Log_2 \, (2n)$$

complex multiplications.

We next present several applications of the convolution operation, and of the fact that it can be computed cheaply.

4.2. Time series analysis. Historically, one of the moving forces in the development of Fast Fourier Transform methods was the need to economize numerical operations in time series analysis. Although they are not directly connected with complex analysis, time series are of such paramount importance in all of applied mathematics that it would be a serious omission not to discuss them at least briefly in this survey article.

Let $x_0, x_1, \cdots, x_{n-1}$ be a finite sequence of real numbers. In applications, x_k may be the result of sampling a physical quantity x at time $\tau_k = \tau_0 + k \, \Delta\tau$. The quantity x may be a brain current (as recorded in an electroencephalogram), the displacement of a seismograph during an earthquake, or the evaluation of an oceanic tide. In electroencephalograms, $\Delta\tau$ is of the order of $\frac{1}{100}$ of a second. Any such sequence $\{x_k\}$ obtained by sampling at (equidistant) time intervals is called a *time series*. The length n of a time series in electroencephalic applications can be of the order of 2^{11} or 2^{12}.

By analyzing a time series, important conclusions on the nature of the underlying physical process can often be drawn. For instance in electroencephalography, epilepsy may be discovered. The analysis of time series usually requires the calculation of the following new sequences from the given time series $\{x_k\}$:

(a) the *covariance function*, i.e., the sequence **r** with tth element

$$(4.7) \qquad r_t = \sum_{k=0}^{n-t-1} x_k x_{k+t}, \qquad t = 0, 1, \cdots ;$$

(b) the *power spectrum*, i.e., the sequence **f** with qth element

$$(4.8) \qquad f_q := \frac{1}{2n} \sum_{t=0}^{n-1} r_t \, e^{-(2\pi i/(2n))tq}, \qquad q = 0, 1, \cdots ;$$

(c) the *smoothed power spectrum* **g** defined by

$$(4.9) \qquad g_q = \sum_{j=-h}^{h} c_j f_{q-j}, \qquad q = 0, 1, \cdots ,$$

where h is a (small) positive integer, and the c_j are constants satisfying

$$(4.10) \qquad c_j \geqq 0, \qquad |j| \leqq h; \qquad \sum_{j=-h}^{h} c_j = 1.$$

It was recognized independently by Tukey and Bartlett in 1944/45 that smoothing of the power spectrum was indispensible in order to obtain meaningful physical interpretations. It soon became clear that discrete Fourier analysis provides a common basis for the numerical analysis of noisy data.

First of all, discrete Fourier analysis permits us to express the above operations concisely. Let the given time series be followed by n zeros, and let **x** denote the

sequence in Π_{2n} obtained by repeating these $2n$ elements periodically. The other sequences $\mathbf{r}$, $\mathbf{f}$, $\mathbf{g}$ defined above will likewise be embedded in Π_{2n}.

(a) For $t < 0$, $t = -s$ we extend the definition (4.7) by setting

$$r_{-s} := \sum_{k=0}^{n-1+s} x_k x_{k-s}.$$

Then evidently

$$r_{-s} = \sum_{k=s}^{n-1} x_k x_{k-s} = \sum_{m=0}^{n-1-s} x_{m+s} x_m = r_s,$$

and the sequence $\mathbf{r}$ becomes symmetric, $\mathbf{r} = R\mathbf{r}$. Setting $\mathbf{y} := R\mathbf{x}$ we have

$$r_t = r_{-t} = \sum_{k=0}^{n-1} x_k x_{k-t} = \sum_{k=0}^{n-1} x_k y_{t-k} = (\mathbf{x} * \mathbf{y})_t,$$

and we see that

(4.11) $\mathbf{r} = \mathbf{x} * R\mathbf{x}.$

(b) If $\mathscr{F} := \mathscr{F}_{2n}$, then the power spectrum is

(4.12) $\mathbf{f} = \mathscr{F}\mathbf{r} = \mathscr{F}(\mathbf{x} * R\mathbf{x}).$

(c) The smoothing operation can likewise be expressed by a convolution. Denoting by $\mathbf{c}$ the sequence with elements c_j in the positions $\equiv j \pmod{2n}$ and with zeros elsewhere, we have

(4.13) $\mathbf{g} = \mathbf{c} * \mathbf{f}.$

It was observed already before the advent of Fast Fourier Transforms that the calculation of $\mathbf{g}$ could be simplified by performing the smoothing in the time domain. Let

(4.14) $\mathbf{s} := \mathscr{F}^{-1}\mathbf{c}.$

Because $\mathbf{f} = \mathscr{F}\mathbf{r}$ we have in view of the convolution theorem (Theorem 4a)

$$\mathbf{c} * \mathbf{f} = \mathscr{F}(\mathscr{F}^{-1}\mathbf{c} \cdot \mathscr{F}^{-1}\mathbf{f}) = \mathscr{F}(\mathbf{s} \cdot \mathbf{r}).$$

Thus (4.13) may be replaced by

(4.15) $\mathbf{g} = \mathscr{F}(\mathbf{s} \cdot \mathbf{r}).$

The sequence $\mathbf{s}$ is called the *time window* of the smoothing process defined by the constants c_j. If the smoothing process is simple enough, its time window may be calculated analytically. By performing the smoothing operation in the time domain, it requires n multiplications only compared to the $(2h+1)2n$ multiplications that are required by smoothing in the frequency domain.

However, there still remain the roughly $\frac{1}{2}n^2$ multiplications required for forming $\mathbf{r}$. Here much greater savings are achieved by Fast Fourier Transforms. Let $\mathbf{a} := \mathscr{F}^{-1}\mathbf{x}$. Then $\mathbf{y} = R\mathbf{x} = \mathscr{F}R\mathbf{a}$, and the determination of $\mathbf{r}$ requires forming

$$\mathbf{x} * \mathbf{y} = \mathscr{F}\mathbf{a} * \mathscr{F}R\mathbf{a}.$$

By the convolution theorem,

$$\mathbf{x} * \mathbf{y} = \mathscr{F}(\mathbf{a} \cdot R\mathbf{a}).$$

Because the sequence $\mathbf{x}$ is real, $R\mathbf{a} = \bar{\mathbf{a}}$, hence

$$(4.16) \qquad\qquad \mathbf{r} = \mathbf{x} * \mathbf{y} = \mathscr{F}(\mathbf{a} \cdot \bar{\mathbf{a}}).$$

In view of $\mathscr{F}^2 = R\mathscr{F}R\mathscr{F} = (1/(2n))R$, it further follows that

$$(4.17) \qquad\qquad \mathbf{f} = \frac{1}{2n} R(\mathbf{a} \cdot \bar{\mathbf{a}}) = \frac{1}{2n}\mathbf{a} \cdot \bar{\mathbf{a}}.$$

Thus in order to find $\mathbf{r}$ and $\mathbf{f}$, it is best to first compute $\mathbf{a} = \mathscr{F}^{-1}\mathbf{x}$ by one application of FFT; $\mathbf{f}$ is then obtained trivially from (4.17), and $\mathbf{r}$ from (4.16) by one further application of FFT. The smoothed power spectrum can now be obtained from (4.9) (if h is small enough) or, if the time window is known, from (4.15).

4.3. Multiplication of polynomials and of large integers.

Convolution performed via FFT yields a fast algorithm for the multiplication of two polynomials. Let

$$p(x) = p_0 + p_1 x + \cdots + p_{n-1}x^{n-1},$$
$$q(x) = q_0 + q_1 x + \cdots + q_{n-1}x^{n-1}$$

be two polynomials of degree $n-1$. We define two sequences in Π_{2n},

$$\mathbf{p} := \|: p_0, p_1, \cdots, p_{n-1}, 0, \cdots, 0 :\|,$$

$$\mathbf{q} := \|: q_0, q_1, \cdots, q_{n-1}, 0, \cdots, 0 :\|,$$

where each line has n zeros, and where the symbol $\|: \cdot \cdot :\|$ indicates periodic repetition. If

$$\mathbf{r} = \{r_k\} := \mathbf{p} * \mathbf{q},$$

then clearly

$$(4.18) \qquad\qquad r(x) := p(x)q(x) = \sum_{k=0}^{2n-2} r_k x^k.$$

In view of Theorem 4b this implies

THEOREM 4c. *If $n = 2^l$, the multiplication of two polynomials of degree $<n$ requires no more than*

$$(4.19) \qquad\qquad \phi(n) := 3n \operatorname{Log}_2 (4n)$$

complex multiplications.

The function $\phi(n)$ occurring here will be used repeatedly.

The foregoing result has an immediate application in the multiplication of large integers. Let $b > 0$ be a (small) integer, and let p and q be (large) integers, represented in the number system with base b as

$$p = \sum_{j=0}^{n-1} p_j b^j, \qquad q = \sum_{j=0}^{n-1} q_j b^j,$$

where the p_j and q_j are integers, $0 \leqq p_j, q_j < b$. Then clearly

$$(4.20) \qquad\qquad r := pq = \sum_{j=0}^{2n-2} r_j b^j,$$

where the r_j are defined by (4.18). Although the definition implies that the r_j are integers, they are not necessarily the correct digits in the representation of r in the base

b, because they need not satisfy the inequality $0 \leqq r_j < b$. However, they do satisfy

$$0 \leqq r_j < nb^2$$

and thus possess representations in the base b,

(4.21) $$r_j = \sum_{l=0}^{m} r_{j,l} b^l,$$

where

$$m := [\mathrm{Log}_b \, n] + 2$$

is, in general, much smaller than n. The correct representation of r in the base b is then easily obtained from (4.21).

It is evident that by calculating the convoluted sequence $\mathbf{r}$ by the Fast Fourier Transform, the product pq can be computed much faster than by the conventional method, which requires n^2 multiplications of integers $<b$. However, to conclude from Theorem 4c that pq can be computed in $\phi(n) \, \mu$ would be incorrect, because the multiplications required to form the convolution by FFT are not multiplications by integers, but multiplications of complex numbers whose real and imaginary parts are arbitrary real numbers. In that sense, forming the product pq requires only 1 μ.

For a correct appraisal of the FFT method we note that the r_j are known to be integers. Thus there is no need to compute them with high precision. Rather, it is sufficient to compute them with errors $\leqq \frac{1}{4}$, say. To achieve this accuracy, we apply (4.6) in the form

(4.22) $$\mathbf{r} = \overline{\mathscr{F}_{2n}} \, (2n\mathscr{F}_{2n}\mathbf{p} \cdot 2n\mathscr{F}_{2n}\mathbf{q}).$$

The following facts on the operation $\mathscr{F}_n$ are easily established.

(i) If the elements of a sequence $\mathbf{x} \in \Pi_n$ satisfy $|x_i| \leqq \alpha$, then the elements of the sequence $\mathbf{y} := \mathscr{F}_n \mathbf{x}$ satisfy $|y_i| \leqq \alpha$, and those of the sequence $\mathbf{z} := n\mathscr{F}_n\mathbf{x}$ satisfy $|z_i| \leqq n\alpha$.

(ii) Let $n = 2^l$, and let the operation $\mathscr{F}_n$ be implemented by a Fast Fourier Transform using the factorization $n = 2 \cdot 2 \cdot \cdots \cdot 2$. If the elements of $\mathbf{x}$ have errors $\leqq \eta$, and if at each step of the algorithm a local error $\leqq \varepsilon$ is tolerated, then the elements of the sequence $\mathscr{F}_n\mathbf{x}$ are in error by at most $l\varepsilon + \eta$.

(iii) Let again $n = 2^l$, and let the operation $n\mathscr{F}_n$ be implemented by a Fast Fourier Transform where the factor $\frac{1}{2}$ is omitted at each step. If the elements of $\mathbf{x}$ have errors $\leqq \eta$, and if at the jth step of the algorithm a local error $\leqq 2^{j+1}\varepsilon$ is tolerated (the natural assumption for floating point arithmetic in view of the growth of the intermediate arrays), then the elements of the sequence $n\mathscr{F}_n\mathbf{x}$ are in error by at most $n \, \mathrm{Log}_2 \, n \cdot \varepsilon + n\eta$.

To apply these results, we assume that the machine works in floating point arithmetic, representing the mantissa as a k-digit number in the base b. The elements of $\mathbf{p}$ and $\mathbf{q}$ are then bounded by b, and are known exactly. Thus (iii) may be applied with $\varepsilon = b^{-k+1}$, $\eta = 0$. We find that the elements of $2n\mathscr{F}_{2n}\mathbf{p}$ and $2n\mathscr{F}_{2n}\mathbf{q}$ are bounded by $2nb$ and are in error by at most $2n \, \mathrm{Log}_2 \, (2n) \cdot b^{-k+1}$. The elements of the product $\mathbf{z} := 2n\mathscr{F}_{2n}\mathbf{p} \cdot 2n\mathscr{F}_{2n}\mathbf{q}$ are bounded by $(2nb)^2$, and will be in error by no more than

$$\eta := 4nb \cdot 2n \, \mathrm{Log}_2 \, (2n) \cdot b^{-k+1} = 8n^2 \, \mathrm{Log}_2 \, (2n) b^{-k+2}.$$

The errors in the calculation of $\mathbf{r} = \overline{\mathscr{F}_{2n}}\mathbf{z}$ may be appraised by (ii). The local errors being bounded by $4n^2 b^2 b^{-k}$, the errors in $\mathbf{r}$ will not exceed

$$8n^2 \, \mathrm{Log}_2 \, (2n) b^{-k+2} + 4n^2 b^2 b^{-k} \, \mathrm{Log}_2 \, (2n) = 12n^2 \, \mathrm{Log}_2 \, (2n) b^{-k+2}.$$

For this to be $\leqq \frac{1}{4}$, we must have

$$(4.23) \qquad\qquad b^{2-k} \leqq \frac{1}{48 n^2 \operatorname{Log}_2 (2n)}.$$

If $b = 10$, $n = 2^{10} \sim 10^3$, this will be satisfied if $k \geqq 11$, which is a wholly manageable requirement.

4.4. Fast Poisson solvers. The convolution theorem also holds for multidimensional sequences. If

$$\mathbf{x} = \{x_{\mathbf{k}}\} \in \Pi_n^{(d)}, \qquad \mathbf{y} = \{y_{\mathbf{k}}\} \in \Pi_n^{(d)},$$

where $\mathbf{k} = (k_1, k_2, \cdots, k_d)$ is the index vector, we again define the *Hadamard product* by

$$\mathbf{x} \cdot \mathbf{y} := \{x_{\mathbf{k}}, y_{\mathbf{k}}\},$$

and the *convolution product* by

$$\mathbf{x} * \mathbf{y} := \{z_{\mathbf{k}}\},$$

where

$$z_{\mathbf{k}} := \sum_{\mathbf{m} \in Q_n} x_{\mathbf{m}} y_{\mathbf{k}-\mathbf{m}},$$

Q_n being the d-dimensional lattice cube defined in § 1.3. Exactly as in the one-dimensional case one may prove

THEOREM 4d. *For arbitrary sequences* $\mathbf{x}, \mathbf{y} \in \Pi_n^{(d)}$,

$$(4.24) \qquad \begin{aligned} \mathscr{F}_n^{(d)} (\mathbf{x} \cdot \mathbf{y}) &= \mathscr{F}_n^{(d)} \mathbf{x} * \mathscr{F}_n^{(d)} \mathbf{y}, \\ \mathscr{F}_n^{(d)} (\mathbf{x} * \mathbf{y}) &= n^d \mathscr{F}_n^{(d)} \mathbf{x} \cdot \mathscr{F}_n^{(d)} \mathbf{y}. \end{aligned}$$

These results have an interesting algorithmic application to the problem of solving Poisson's differential equation in rectangular domains. Although our approach applies to any number of dimensions, we consider only the two-dimensional case and assume, furthermore, that the domain S where the equation is to be solved is the unit square,

$$S: 0 \leqq \xi \leqq 1, \qquad 0 \leqq \eta \leqq 1,$$

in the (ξ, η)-plane. For a given function f defined on S, we wish to determine the solution u of

$$(4.25) \qquad\qquad -\Delta u = f(\xi, \eta)$$

that continuously assumes the values 0 on the boundary of S. This problem arises, for instance, in the determination of the motion of electrostatically interacting particles in two dimensions (plasma flows). If the charge distribution f at time τ_k is known, the resulting potential u can be found by solving Poisson's equation as given above. The potential then determines the forces acting on the charges, and thus the acceleration and the position of the charges at time $\tau_{k+1} = \tau_k + \Delta \tau$. Since Poisson's equation must be solved at each time step, it is essential to find the solution as economically as possible.

To solve the problem by discretization, we choose a grid constant $h = 1/n$, let

$$\xi_i = ih, \qquad \eta_j = jh,$$

$(i, j = 0, 1, \cdots, n)$, and obtain approximate values u_{ij} of $u(\xi_i, \eta_j)$ by solving the linear

system

(4.26) $$4u_{ij} - u_{i+1,j} - u_{i,j+1} - u_{i-1,j} - u_{i,j-1} = h^2 f_{ij}$$

($f_{ij} := f(\xi_i, \eta_j)$, $u_{0j} = u_{nj} = u_{i0} = u_{in} = 0$) obtained by discretizing the Laplacian $-\Delta$ in the simplest possible manner.

The solution of (4.26) may be accomplished most readily by discrete Fourier transforms. To this end we continue both u_{ij} and f_{ij} as odd periodic sequences of period $2n$. (This requires that $u_{ij} = f_{ij} = 0$ for i or $j \equiv 0 \pmod{n}$.) For u_{ij} this condition is satisfied because of the boundary condition; for f_{ij} it can be satisfied because the values of f on the boundary of S are irrelevant. The relations (4.26) are still satisfied for the continued sequences and may be regarded as a relation between two elements $\mathbf{u}$ and $\mathbf{f}$ of the space $\mathring{\Pi}_{2n}^{(2)}$ of odd sequences in $\Pi_{2n}^{(2)}$. The crucial fact now is that (4.26) is a convolution. Indeed if we define the sequence $\mathbf{d} \in \Pi_{2n}^{(2)}$ to have zero elements except

$$d_{00} = 4, \qquad d_{1,0} = d_{0,1} = d_{-1,0} = d_{0,-1} = -1,$$

then (4.26) is nothing but

(4.27) $$\mathbf{d} * \mathbf{u} = h^2 \mathbf{f}.$$

Applying $\mathscr{F}_{2n}^{(2)}$ to either side of this equation and using the symbol $\hat{\cdot}$ to denote transforms, we obtain in view of the convolution theorem

(4.28) $$4n^2 \hat{\mathbf{d}} \cdot \hat{\mathbf{u}} = h^2 \hat{\mathbf{f}}.$$

It is readily verified that $\hat{\mathbf{d}} = \{\hat{d}_{km}\}$ where

(4.29) $$\hat{d}_{km} = \left(1 - \tfrac{1}{2}\cos\frac{k\pi}{n} - \tfrac{1}{2}\cos\frac{m\pi}{n}\right)h^2.$$

This is $\neq 0$ for all $(k, m) \not\equiv (0, 0)$. Thus (4.28) may be solved for $\hat{u}$,

$$\hat{\mathbf{u}} = \frac{h^2}{4n^2}\hat{\mathbf{d}}^{-1} \cdot \hat{\mathbf{f}}$$

(the inverse to be taken in the sense of Hadamard multiplication), and we find

(4.30) $$\mathbf{u} = \frac{h^2}{4n^2}\mathscr{F}_{2n}^{(2)-1}(\hat{\mathbf{d}}^{-1} \cdot \hat{\mathbf{f}})$$

The desired solution thus may be found in the following steps:
(i)   calculate $\hat{\mathbf{f}} := \mathscr{F}_{2n}^{(2)}\mathbf{f}$;
(ii) using the explicit values of $\hat{d}_{km}$ given by (4.29), calculate $\hat{\mathbf{d}}^{-1} \cdot \hat{\mathbf{f}}$;
(iii) obtain $\mathbf{u}$ from (4.30). In view of

$$\frac{1}{4n^2}\mathscr{F}_{2n}^{(2)-1} = \bar{\mathscr{F}}_{2n}^{(2)},$$

(4.30) may be replaced by

(4.31) $$\mathbf{u} = h^2 \bar{\mathscr{F}}_{2n}^{(2)}(\hat{\mathbf{d}}^{-1} \cdot \hat{\mathbf{f}}).$$

The entire process requires only taking two transforms $\mathscr{F}_{2n}^{(2)}$ and n^2 scalar multiplications. Thus by Theorem 1c, if $n = 2^l$, it may be accomplished in $n^2[6 \operatorname{Log}_2 n + 7]\,\mu$. By taking advantage of the fact that the sequences involved are real and odd, this number may be reduced even further.

The method is not restricted to the simplest finite difference formula (4.26). For instance, in the more accurate nine-point formula

(4.32) $$\begin{aligned}&\tfrac{1}{6}\{20u_{ij} - 4(u_{i+1,j} + u_{i,j+1} + u_{i-1,j} + u_{i,j-1})\\&- (u_{i+1,j+1} + u_{i-1,j+1} + u_{i-1,j-1} + u_{i+1,j-1})\} = h^2 f_{ij}\end{aligned}$$

the expression on the left is the convolution of $\mathbf{u}$ with the sequence $\mathbf{d} = \{d_{km}\}$ where all elements are zero except

(4.33)
$$d_{00} = \tfrac{20}{6}, \qquad d_{1,0} = d_{0,1} = d_{-1,0} = d_{0,-1} = -\tfrac{2}{3},$$
$$d_{1,1} = d_{-1,1} = d_{1,-1} = d_{-1,-1} = -\tfrac{1}{6}.$$

It is easily verified that

$$\mathscr{F}_{2n}^{(2)} \mathbf{d} = \hat{\mathbf{d}} = \{\hat{d}_{km}\}$$

where

(4.34)
$$\hat{d}_{km} = \frac{h^2}{6}\left(5 - 2\cos\frac{k\pi}{n} - 2\cos\frac{m\pi}{n} - \cos\frac{k\pi}{n}\cos\frac{m\pi}{n}\right).$$

With this definition of $\hat{\mathbf{d}}$, the formulas (4.30) and (4.31) yield the solution $\mathbf{u}$ for the nine-point operator.

We can also treat the Mehrstellenverfahren of Collatz which furnishes solutions with smaller discretization errors. For instance if the term $h^2 f_{ij}$ in (4.32) is replaced by

$$\frac{h^2}{12}\{8f_{ij} + f_{i+1,j} + f_{i,j+1} + f_{i-1,j} + f_{i,j-1}\},$$

then for sufficiently smooth f the solution u_{ij} of (4.32) has a global error of only $O(h^4)$. Now the foregoing expression clearly is the convolution of $\mathbf{f}$ with a smoothing sequence $\mathbf{c} = \{c_{km}\}$ with all elements zero except

$$c_{00} = \tfrac{8}{12}, \qquad c_{1,0} = c_{0,1} = c_{-1,0} = c_{0,-1} = \tfrac{1}{12}.$$

Hence the equation (4.32) becomes

$$\mathbf{d} * \mathbf{u} = h^2 \mathbf{c} * \mathbf{f}$$

and on taking Fourier transforms

(4.35)
$$\hat{\mathbf{d}} \cdot \hat{\mathbf{u}} = h^2 \hat{\mathbf{c}} \cdot \hat{\mathbf{f}},$$

where $\hat{\mathbf{d}}$ is defined by (4.34) and $\hat{\mathbf{c}} = \{\hat{c}_{km}\}$ is readily calculated to be

(4.36)
$$\hat{c}_{km} = \frac{h^2}{24}\left(4 + \cos\frac{k\pi}{n} + \cos\frac{m\pi}{n}\right).$$

Because $\hat{d}_{km} \neq 0$ for all $(k, m) \neq (0, 0)$, (4.35) may be solved for

$$\hat{\mathbf{u}} = h^2 \hat{\mathbf{d}}^{-1} \cdot \hat{\mathbf{c}} \cdot \hat{\mathbf{f}},$$

and on transforming back we obtain

(4.37)
$$\mathbf{u} = h^2 \mathscr{F}_{2n}^{(2)^{-1}} (\hat{\mathbf{d}}^{-1} \cdot \hat{\mathbf{c}} \cdot \hat{\mathbf{f}}).$$

We emphasize that the foregoing method applies to any difference approximation or Mehrstellenverfahren for solving Poisson's equation in rectangular domains, and indeed to any difference equation that can be expressed as a convolution. In applying the method to a Mehrstellenverfahren for solving $-\Delta u = f$, it is not necessary to know or work out the actual form of the difference equations in the physical domain, which often are rather complicated. All that is needed is a sequence

$$\mathbf{r} = \{r_{km}\} \in \Pi_{2n}^{(2)}$$

where r_{km} is a nonvanishing rational function of $\cos(k\pi/n)$ and $\cos(m\pi/n)$ such that

$n^2 r_{km}$ approximates the eigenvalues of $-\Delta$,

$$\lambda_{km} = (k\pi)^2 + (m\pi)^2,$$

as well as possible for $n \to \infty$. If

$$n^2 r_{km} - \lambda_{km} = O(n^{-2p})$$

and if f is sufficiently smooth, then

$$\mathbf{u} := h^2 \mathscr{F}_{2n}^{(2)^{-1}} (\mathbf{r}^{-1} \cdot \hat{\mathbf{f}})$$

defines an approximate solution of $-\Delta u = f$ which is in error by $O(h^{2p})$. For the standard five-point operator,

$$r_{km} = 4 - 2 \cos \frac{k\pi}{n} - 2 \cos \frac{m\pi}{n},$$

for the Mehrstellenverfahren involving the nine point operator,

$$r_{km} = \frac{20 - 8 \cos (k\pi/n) - 8 \cos (m\pi/n) - 4 \cos (m\pi/n)}{4 + \cos (k\pi/n) + \cos (m\pi/n)}.$$

Better functions r_{km} can be obtained systematically as certain two-dimensional Padé approximants to λ_{km}.

Notes on § 4. § 4.1. On convolution of sequences see Cooley, Lewis and Welch (1967), Aho, Hopcroft and Ullmann (1974). We have coined the name "Hadamard multiplication" in analogy to a similar product in the theory of power series, because a name seemed to be required.

§ 4.2. Koopmans (1974) and Bloomfield (1976) are standard references on time series. These books discuss geophysical applications, as does Claerbout (1976). Blackman and Tukey (1959) is of historical interest. Anderson and Bloomfield (1974a, b) are some key references on the analysis of noisy data.

§ 4.3. A pre-FFT treatment of the multiplication of large numbers is given by Karatsuba and Ofman (1962). Schönhage and Strassen (1971) have more sophisticated algorithms than those given here and count operations precisely. Ramos (1971) treats the stability of Fast Fourier Transforms.

§ 4.4. Hockney (1965) first considered the solution of Poisson's equation by Fast Fourier Methods and later described numerous applications (Hockney (1970), (1972), (1972a), Hockney, Warriner and Reiser (1974), Hockney and Brownrigg (1974), Hockney and Brown (1975), Hockney and Goel (1975)). For further developments on Fast Poisson Solvers see Bunemann (1969), Buzbee, Golub and Nielson (1970), Dorr (1970), Buzbee, Dorr, George and Golub (1971), Concus and Golub (1973), Fischer, Golub, Hald, Leiva and Widlund (1974), Proskurowski and Widlund (1976). Most of these papers are concerned only with the standard five-point operator. Pickering (1977) extends Hockney's original method to the nine-point operator. On the Mehrstellen-verfahren see Collatz (1955, p. 360). The observation that the convolution theorem provides an easy access to fast Poisson solvers for difference operators of arbitrarily high order, and also to the Mehrstellenverfahren, is possibly new.

5. Fast algorithms for power series. In this section we consider formal power series (fps) in one indeterminate with complex coefficients. The reader will recall the various algebraic operations that can be carried out with fps without regard to

convergence. Here we review some of these operations from the point of view of economizing arithmetic operations.

If $P = a_0 + a_1 x + a_2 x^2 + \cdots$ is a fps, we denote, for $n = 1, 2, \cdots$, by

$$P_n(x) := a_0 + a_1 x + \cdots + a_{n-1} x^{n-1}$$

its partial sum of degree $n - 1$, consisting of the first n terms. (This notation is at variance with that ordinarily used.) We call P_n a *polynomial of length n*. If P is any fps such that $P_n = 0$, we write

$$P = O(x^n).$$

5.1. Multiplication. While nothing needs to be said about addition or subtraction, the Fast Fourier Transform immediately furnishes an important result on the multiplication of two fps. For any fps P and Q,

$$(PQ)_n = (P_n Q_n)_n.$$

By Theorem 4c, if $n = 2^l$, the multiplication of the two polynomials on the right can be accomplished in

$$\phi(n) := 3n \, \mathrm{Log}_2 (4n)$$

multiplications. We restate the result for reference.

THEOREM 5a. *If $n = 2^l$ and P, Q are fps, the computation of $(PQ)_n$ by FFT requires no more than $\phi(n)$ complex multiplications.*

If performed in the conventional manner, the evaluation of $(PQ)_n$ would require $\frac{1}{2}n^2$ multiplications. The smallest power of 2 for which $\phi(n) < \frac{1}{2}n^2$ is $n = 64$.

5.2. Newton's method for formal power series. Many fast algorithms for formal power series can be based on an extension of Newton's method to nonlinear equations in formal power series. We shall see that in this formal context Newton's method always produces the exact solution in a finite number of steps.

Let $\mathcal{P}$ be the integral domain of formal power series, let $P = a_0 + a_1 x + a_2 x^2 + \cdots \in \mathcal{P}$, and let $Q = b_1 x + b_2 x^2 + \cdots$ be a nonunit in $\mathcal{P}$. We recall that the composition of P with Q is defined by substituting Q for x in P,

$$(5.1) \qquad P \circ Q = a_0 + a_1 Q + a_2 Q^2 + \cdots,$$

and collecting coefficients of equal powers. Because Q has constant coefficient zero, only finitely many terms can arise for each power, and the operation of composition is algebraically well defined. We further recall that the *almost units* in $\mathcal{P}$, (i.e., the series $P = a_1 x + a_2 x^2 + \cdots$ where $a_1 \neq 0$) form a group under composition, the unit element being $X = 1x + 0x^2 + \cdots$. The inverse $P^{[-1]}$ of an almost unit P is called the *reversion* of P.

Here we consider the equation

$$(5.2) \qquad Q \circ W - R = 0,$$

where Q and R are given almost units, and where W is sought. The solution is clearly given by the formula

$$W = Q^{[-1]} \circ R,$$

but the question remains how to construct W.

Suppose W_k is an approximate solution to (5.2) in the sense that

$$(5.3) \qquad W = W_k + O(x^k),$$

where $k > 0$. This implies that W_k is a nonunit. Applying, in a purely formal sense, Newton's method to (5.2) we would expect to improve the approximation by forming

$$W^+ := W_k - \frac{Q \circ W_k - R}{Q' \circ W_k}.$$

Here the quotient is well defined, for because Q is an almost unit, Q' is a unit, and so is $Q' \circ W_k$. That W^+ is a better approximation to W than W_k is easily confirmed as follows. By (5.3) there exists $W^* \in \mathscr{P}$ such that

$$W_k = W + x^k W^*.$$

Thus by the formal analog of Taylor's formula,

$$Q \circ W_k = Q \circ (W + x^k W^*)$$
$$= Q \circ W + (Q' \circ W)x^k W^* + O(x^{2k})$$

and also

$$Q' \circ W_k = Q' \circ W + O(x^k).$$

Because $Q \circ W = R$, we get

$$W^+ = W + x^k W^* - \frac{(Q' \circ W)x^k W^* + O(x^{2k})}{Q' \circ W + O(x^k)}$$

and because $Q' \circ W$ is a unit this simplifies to

$$W^+ = W + x^k W^* - x^k W^*(1 + O(x^k))$$
$$= W + O(x^{2k}).$$

Thus in passing from W_k to W^+, the number of correct coefficients in the solution has been doubled.

We thus may construct a sequence of approximations $\{W^{(m)}\}$ to W by the following algorithm: Let

(5.4a) $W^{(0)} := 0$

and for $m = 0, 1, \cdots$

(5.4b) $W^{(m+1)} := W^{(m)} - \left(\dfrac{Q \circ W^{(m)} - R}{Q' \circ W^{(m)}}\right)_{2n}$

where $n := 2^m$. Because $W^{(0)}$ satisfies (5.3) for $k = 1$, we obtain the following lemma:

LEMMA 5b. *For* $m = 0, 1, 2, \cdots$,

$$W^{(m)} = (Q^{[-1]} \circ R)_{2^m}.$$

If $\omega(n)$ denotes the number of multiplications required to compute W_n and $\nu(n)$ the number of multiplications required to compute the Newton correction

$$\left(\frac{Q \circ W^{(m)} - R}{Q' \circ W^{(m)}}\right)_{2n},$$

then clearly for $n = 2^m$

$$\omega(2n) \leqq \omega(n) + \nu(n);$$

hence we have

COROLLARY 5c. *For $l = 1, 2, \cdots$*,

(5.5) $$\omega(2^l) = \nu(2^{l-1}) + \nu(2^{l-2}) + \cdots + \nu(2) + \nu(1).$$

The Newton algorithm will now be applied to the solution of several basic problems in power series manipulation.

5.3. Division. To begin with, we discuss the determination of

$$P^{-1} = c_0 + c_1 x + c_2 x^2 + \cdots,$$

where

$$P = a_0 + a_1 x + a_2 x^2 + \cdots$$

is a *unit* in $\mathscr{P}$ ($a_0 \neq 0$). The conventional algorithm for determining the c_i is based on comparing coefficients in the identity $P^{-1}P = 1$, i.e.

$$(c_0 + c_1 x + c_2 x^2 + \cdots)(a_0 + a_1 x + a_2 x^2 + \cdots) = 1,$$

which yields the recurrence relation

$$c_0 = a_0^{-1}, \qquad c_n = -a_0^{-1}(a_1 c_{n-1} + a_2 c_{n-2} + \cdots + a_n c_0).$$

To compute the first n coefficients c_i by this method, n $\quad \delta$ and $1 + 2 + \cdots + (n-1) = \frac{1}{2}(n-1)n$ $\quad \mu$ are required.

To compute P^{-1} by Newton's method, we recall that in elementary numerical analysis the reciprocal of a real number $c \neq 0$ may be found by applying Newton's method to the equation

$$\frac{1}{y} - c = 0.$$

This leads to an algorithm that does not require divisions. Proceeding by analogy, we seek $Y := P^{-1}$ as the solution of

(5.6) $$Y^{-1} - P = 0$$

This is of the form (5.2) if we set

$$P = a_0 + R, \qquad Y = a_0^{-1} + W$$

and

$$Q = \frac{1}{a_0^{-1} + x} - a_0 = -\frac{a_0^2 x}{1 + a_0 x}.$$

We thus could apply Newton's method to the resulting equation

$$\frac{a_0 W}{1 + a_0 W} + R = 0;$$

however, simpler formulas are obtained by expressing the Newton iteration (5.2) directly in terms of

$$Y^{(m)} := a_0^{-1} + W^{(m)}.$$

One readily finds

(5.7a) $$Y^{(0)} := a_0^{-1},$$

(5.7b) $$Y^{(m+1)} := [Y^{(m)}(2 - PY^{(m)})]_{2n},$$

$m = 0, 1, 2, \cdots$. As a special case of Lemma 5b we have

THEOREM 5d. *For $m = 0, 1, 2, \cdots$,*

$$(5.8) \qquad\qquad Y^{(m)} = (P^{-1})_{2^m}.$$

Thus $Y^{(m)}$ is a polynomial of length 2^m whose coefficients agree with the first 2^m coefficients of P^{-1}. Each step of the iteration doubles the number of correct coefficients.

To appraise the cost of division by Corollary 5c, we count the number of multiplications required to carry out one step of the recurrence (5.7b). Because the final result is truncated to length $2n$, all intermediate results may be truncated likewise. Thus in actual calculation (5.7b) is replaced by

$$Y^{(m+1)} = \{Y^{(m)}[2 - (P_{2n}Y^{(m)})_{2n}]\}_{2n}$$

$(n := 2^m)$. This can be evaluated by forming two products of polynomials of length $2n$, which requires $2\phi(2n)\ \mu$. (This crude operations count could be somewhat refined by taking advantage of coefficients that are a priori known to be zero.) In the notation of Corollary 5c we thus have $\nu(n) = 2\phi(2n) = 12n \operatorname{Log}_2 (8n)$. Using the formula

$$\sum_{k=1}^{l-1} 2^k k = 2^l(l-2) + 2,$$

the sum (5.5) is easily evaluated and yields

$$\omega(n) \leqq 12n \operatorname{Log}_2 (2n) \leqq 4\phi(n).$$

If Q is any fps and $(Q/P)_n$ is required, this may be computed as $(Q_n P_n^{-1})_n$ which requires another $\phi(n)$ multiplications. Thus in toto we have

THEOREM 5e. · *To compute $(Q/P)_n$ where P is a unit in $\mathscr{P}$ and Q is arbitrary requires for $n = 2^l$ no more than $5\phi(n)$ multiplications.*

5.4. Composition: Some special cases. Before discussing the composition problem for general fps, we consider some special cases that can be treated by exploiting special functional relationships.

Consider, for example, the logarithmic series,

$$L = \operatorname{Log} (1 + x)$$
$$= x - \tfrac{1}{2}x^2 + \tfrac{1}{3}x^3 - \cdots.$$

If Q is any nonunit, then $Y := L \circ Q$ satisfies

$$Y' = \frac{Q'}{1 + Q}$$

and therefore, if n is any integer > 1,

$$(5.8) \qquad\qquad (Y')_{n-1} = \left(\frac{(Q')_{n-1}}{1 + Q_{n-1}}\right)_{n-1}.$$

To compute $(Q')_{n-1}$ for a given Q requires $n - 2\ \mu$. By Theorem 5e the computation of the quotient in (5.9) requires less than $5\phi(n)\ \mu$ if $n = 2^l$. Y_n can be recovered unambiguously from $(Y')_{n-1}$ because the zeroth coefficient of Y is known to be zero. This requires another $n - 2$ multiplications. Taking into account the margin by which $5\phi(n)$ overestimates the actual number of multiplications as given in Corollary 5c, we find

THEOREM 5f. *If $n = 2^l$, to compute $(L \circ Q)_n$ for any nonunit Q requires no more than $5\phi(n)$ multiplications.*

We next turn to the problem of computing the exponential of a given nonunit Q,

$$(5.9) \qquad E \circ Q = 1 + \frac{1}{1!}Q + \frac{1}{2!}Q^2 + \cdots.$$

If $W := E \circ Q - 1$, then

$$(5.10) \qquad L \circ W - Q = 0$$

which is an equation of the form (5.2) to which Newton's method can be applied. The algorithm (5.7) in this case yields

$$(5.11) \qquad \begin{aligned} W^{(0)} &:= 0, \\ W^{(m+1)} &:= W^{(m)} - \{(1 + W^{(m)})(L \circ W^{(m)} - Q)\}_{2n} \end{aligned}$$

$n := 2^m$, $m = 0, 1, 2, \cdots$. One step of the algorithm requires computing a logarithm to precision $O(x^{2n})$, and of a product to the same precision. By the Theorems 5f and 5a we thus have

$$\nu(n) = 6\phi(2n).$$

Evaluating the sum (5.5) we get

THEOREM 5g. *If $n = 2^l$, if Q is any nonunit and if E is the exponential series, the evaluation of $(E \circ Q)_n$ by the algorithm (5.11) requires no more than $12\phi(n)$ multiplications.*

As an immediate application, we consider the computation of $(1 + Q)^\alpha$ for a given nonunit Q, where α is an arbitrary complex number. In view of

$$(1 + Q)^\alpha = E \circ (\alpha L \circ Q)$$

this is reduced to forming $L \circ Q$, scalar multiplication, and exponentiation. By the foregoing results we have

THEOREM 5h. *To compute $[(1 + Q)^\alpha]_n$, where $n = 2^l$, requires no more than $17\phi(n)$ multiplications.*

This result is asymptotically much better than the $\frac{1}{2}n^2$ μ that are required by Euler's already ingenious algorithm (called "J. C. P. Miller algorithm" by Henrici (1974)). Even if α is a large integer the result is more favorable than what would be obtained, say, by successive squaring and using the binary decomposition of α. However, for special values of α such as $\alpha = \pm\frac{1}{2}$ direct application of Newton's method will produce a yet more favorable $O(n \operatorname{Log}_2 n)$ result.

As a second application, we return to the problem (see § 3.2) of determining the coefficients of the polynomial

$$p(z) = z^n + a_1 z^{n-1} + a_2 z^{n-2} + \cdots + a_{n-1}z + a_n$$

$$= \prod_{i=1}^{n} (z - z_i)$$

from the power sums of its zeros,

$$s_k := \sum_{i=1}^{n} z_i^k, \qquad k = 1, 2, \cdots, n,$$

where we do not assume that all $z_i \neq 0$. The polynomial

$$q(z) := z^n p\left(\frac{1}{z}\right) = 1 + a_1 z + a_2 z^2 + \cdots + a_n z^n$$

has the zeros z_i^{-1}. Hence

$$\frac{q'(z)}{q(z)} = \sum_{i=1}^{n} \frac{1}{z - z_i^{-1}}$$

and

$$-\frac{q'(z)}{q(z)} = \sum_{i=1}^{n} \frac{z_i}{1 - zz_i}$$

$$= \sum_{k=1}^{\infty} s_k z^{k-1}.$$

Thus we have (analytically if $|z|$ is sufficiently small, and in any case formally)

$$-\text{Log } q(z) = \sum_{k=1}^{\infty} \frac{s_k}{k} z^k,$$

hence

(5.12) $$q(z) = \exp\left(-\sum_{k=1}^{\infty} \frac{s_k}{k} z^k\right).$$

In exact computation, the last series will terminate automatically after n terms; in numerical computation we may use this fact as a check. To solve the problem stated for $n = 2^l$ thus requires only forming the n δ to compute s_k/k plus the $12\phi(n)$ μ to carry out the exponentiation (5.12).

5.5. Composition: The general case. Let $P = a_0 + a_1 x + \cdots \in \mathscr{P}$, and let $Q = b_1 x + b_2 x^2 + \cdots$ be a nonunit in $\mathscr{P}$. To compute, for a given integer $n = 2^l$, $(P \circ Q)_n$ directly from the definition of composition would require, first of all, building up the powers

$$(Q^k)_n = \sum_{m=k}^{n-1} b_m^{(k)} x^m, \qquad k = 2, 3, \cdots, n.$$

If done by fast multiplication, this requires $(n-1)\phi(n) = O(n^2 \text{ Log}_2 n)$ μ. To this number there are to be added the multiplications by the coefficients a_k, which requires another $\frac{1}{2}n^2$ μ. The grand total for this method thus is $O(n^2 \text{ Log}_2 n)$.

Another possibility for computing the composition is to use Horner's scheme. Although more elegant, the resulting algorithm still requires $O(n^2 \text{ Log}_2 n)$ μ.

Here we present an algorithm due to Brent and Kung (1975) which achieves the same result in only $O((n \text{ Log}_{2 n})^{3/2})$ μ. Some preliminary results are required. For simplicity of presentation, it will be assumed that Q is an almost unit in $\mathscr{P}$, that is, $b_1 \neq 0$.

LEMMA 5i. *For P and Q as above, let $C := P \circ Q$, $D := P' \circ Q$. If $n = 2^l$, and if C_n is known, the computation of D_{n-1} requires no more than $6\phi(n)$ multiplications.*

Proof. By the chain rule for formal power series,

$$C' = (P' \circ Q)Q' = DQ'.$$

Because Q' is a unit,

$$D = \frac{C'}{Q'}, \qquad D_{n-1} = \left(\frac{C'_{n-1}}{D'_{n-1}}\right)_{n-1}.$$

The computation of C'_{n-1} and Q'_{n-1} each requires $n - 1$ μ. (Strictly speaking, since the multipliers are integers, these multiplications could be reduced to additions and thus

would not have to be counted.) The division in view of Theorem 5e requires another $5\phi(n)$ μ. The conclusion follows since $\phi(n) > 3n$.

LEMMA 5j. *Let* $n = 2^l$, *and let*

$$P = p_0 + p_1 x + \cdots + p_{j-1} x^{j-1},$$

$$Q = q_1 x + \cdots + q_{m-1} x^{m-1},$$

where j, $m \leq n$. *Then the computation of*

$$C_n := (P \circ Q)_n$$

requires no more than $18mj\{\text{Log}_2(4n)\}^2$ *multiplications.*

Proof. To begin with, let j and m be powers of 2, $j \geq 2$. We reduce the composition of a polynomial of length j with Q to two compositions of polynomials of length $j/2$ with Q by using the decomposition

$$P = P_{j/2} + x^{j/2} P^*,$$

where P^* is a suitable polynomial of length $j/2$. This yields

$$(5.13) \qquad P \circ Q = P_{j/2} \circ Q + Q^{j/2}(P^* \circ Q)$$

where in all operations terms of degree $\geq n$ are truncated. If, for fixed m and $n \geq m$, $\tau(j)$ denotes the number of multiplications required to compute the composition of a polynomial of length j with a polynomial of length m to n terms, the forming of each series $(P_{j/2} \circ Q)_n$ and $(P^* \circ Q)_n$ requires $\tau(j/2)$ μ. Assuming inductively that $Q^{j/4}$ is known, the computation of $Q^{j/2}$ costs $\phi(\frac{1}{2}jm)$ μ, and the remaining multiplication in (5.13) costs another $\phi(\frac{1}{2}jm)$ μ. Thus

$$\tau(j) \leq 2\tau(j/2) + 2\phi(\tfrac{1}{2}jm),$$

and in view of $\tau(1) = 0$ it follows that

$$\tau(2^h) \leq 2\phi(2^{h-1}m) + 4\phi(2^{h-2}m) + 8\phi(2^{h-3}m) + \cdots$$

$$\leq 3m2^h\{h \, \text{Log}_2 \, m + \tfrac{1}{2}(h+1)(h+2)\}.$$

Hence if $j = 2^h \leq n$ we get

$$\tau(j) \leq 3mj \cdot \tfrac{3}{2}\{\text{Log}_2(4n)\}^2.$$

If m and j are not powers of 2, we may replace them by the next higher power of 2 by filling up P and Q with zero coefficients. This yields the estimate of the lemma.

After these preparations, the Brent–Kung algorithm for constructing $(P \circ Q)_n$ (Q an almost unit) may be described as follows.

For the n given, let

$$m := \left[\sqrt{\frac{n}{\text{Log}_2(4n)}}\right].$$

We set $Q = Q_m + x^m Q^*$. By the formal analog of Taylor's theorem,

$$P \circ Q = P \circ (Q_m + x^m Q^*)$$

$$= P \circ Q_m + \frac{1}{1!}(P' \circ Q_m)x^m Q^* + \frac{1}{2!}(P'' \circ Q_m)x^{2m}Q^{*2} + \cdots.$$

If $j > k := [n/m]$, then $x^{mj} = O(x^n)$, and so

$$(5.14) \qquad P \circ Q = P \circ Q_m + \frac{1}{1!}(P' \circ Q_m)x^m Q^* + \cdots + \frac{1}{k!}(P^{(k)} \circ Q_m)x^{km}Q^{*k} + O(x^n).$$

This formula is implemented in the following five steps:

Step 1. Compute R_n, where $R := P \circ Q_m$. By Lemma 5j, this requires no more than

$$18mn\{\text{Log}_2\,(4n)\}^2 = 18\{n\,\text{Log}_2\,(4n)\}^{3/2}$$

multiplications.

Step 2. For $j = 1, 2, \cdots, k$, compute

$$(P^{(j)} \circ Q_m)_{n-j}.$$

By Lemma 5i, this takes at most $5\phi(n) = 15n\,\text{Log}_2\,(4n)$ μ for each series. The whole step thus takes no more than

$$15kn\,\text{Log}_2\,(4n) \leq 15\frac{n^2}{m}\,\text{Log}_2\,(4n) \leq 30\{n\,\text{Log}_2\,(4n)\}^{3/2}$$

multiplications. Here we have assumed that $\sqrt{n} \geq 2\sqrt{\text{Log}_2\,(4n)}$, that is, $n \geq 64$.

Step 3. For $j = 2, 3, \cdots, k$, compute

$$((x^m Q^*)^j)_n.$$

By Theorem 5a, this takes no more than

$$(k-1)\phi(n) \leq \frac{n}{m}\phi(n) \leq 6\{n\,\text{Log}_2\,(4n)\}^{3/2}$$

multiplications. Again we have assumed that $n \geq 64$ in order to ensure $\sqrt{n} \geq 2\sqrt{\text{Log}_2\,(4n)}$.

Step 4. For $j = 1, 2, \cdots, k$, compute

$$\frac{1}{j!}[(P^{(j)} \circ Q_m)(x^m Q^*)^j]_n.$$

This requires forming k products of length n, and thus again at most

$$6\{n\,\text{Log}_2\,(4n)\}^{3/2}$$

μ. To this there must be added the multiplication of k series of length n by the appropriate factorials, which yields another $kn \leq 2n^{3/2}\{\text{Log}_2\,(4n)\}^{1/2}$ μ, plus $k \leq 2\{n\,\text{Log}_2\,(4n)\}^{1/2}$ μ for building up the factorials. Altogether Step 4 may be performed in

$$8\{n\,\text{Log}_2\,(4n)\}^{3/2}$$

multiplications.

Step 5. Sum the series obtained in Step 4. This requires no multiplications.

Altogether we have obtained:

THEOREM 5k. *If $n = 2^l \geq 64$, if $P \in \mathscr{P}$ and if Q is an almost unit in $\mathscr{P}$, the composition $(P \circ Q)_n$ can be computed using no more than $50\{n\,\text{Log}_2\,(4n)\}^{3/2}$ multiplications.*

5.6. Reversion: The general case. The problem of computing $W := Q^{[-1]}$ for a given almost unit Q can, in the general case, again be solved by Newton's algorithm,

because W by definition satisfies the equation

$$(5.15) \qquad\qquad Q \circ W - X = 0$$

which is the special case $R = X = x$ of (5.2). Newton's algorithm (5.6) here takes the form

$$W^{(0)} := 0,$$

$$(5.16) \qquad W^{(m+1)} := W^{(m)} - \left(\frac{Q \circ W^{(m)} - X}{Q' \circ W^{(m)}}\right)_{2h},$$

where $h := 2^m$, $m = 0, 1, \cdots$. One Newton step requires

(a) forming $(Q \circ W^{(m)})_{2h}$ (cost by Theorem 5k if $h \geq 32$: $50\{2h \operatorname{Log}_2 (8h)\}^{3/2} \quad \mu$);

(b) forming $(Q' \circ W^{(m)})_h$ (cost by Lemma 5i: $6\phi(h) \quad \mu$);

(c) forming the quotient to precision $O(x^{2h})$. Because the numerator is $O(x^h)$, the quotient after factoring out x^h must be formed only to precision $O(x^h)$, which by Theorem 5e costs $5\phi(h)$ multiplications.

The total number of multiplications to compute W_n where $n = 2^l$ is given by the sum of the foregoing for $h = 2^{l-1}, 2^{l-2}, \cdots, 1$. Values of (a) where $h \leq 32$ are replaced by the values for $h = 32$. Crude estimates yield

THEOREM 5l. *If n is a sufficiently high power of 2 and Q is an almost unit, the computation of $(Q^{[-1]})_n$ by the algorithm (5.16) requires no more than $150\{n \operatorname{Log}_2 (8n)\}^{3/2}$ multiplications.*

Notes on § 5. § 5.2. The fundamental role of Newton's method in power series manipulation is implicit in Brent (1976).

§ 5.3. Algorithm (5.7) (in a different notation) is due to Sieveking (1972). Kung (1974) showed that this is just Newton's method, and studied other rootfinding methods to construct reciprocals.

§ 5.4. See Brent (1976).

§ 5.5 and § 5.6. See Brent and Kung (1975), (1976). The latter paper also proposes $O(n^2)$ algorithms for reversion and composition that should be attractive for moderate n. Multivariate extensions of these results are reported in Brent and Kung (1977).

REFERENCES

A. V. AHO, J. E. HOPCROFT AND J. D. ULLMANN (1974), *The Design and Analysis of Computer Algorithms*, Addison-Wesley, Reading, MA.

R. S. ANDERSSON AND P. BLOOMFIELD (1974a), *A time series approach to numerical differentiation*, Technometrics, 16, pp. 69–75.

——— (1974b), *Numerical differentiation procedures for non-exact data*, Numer. Math., 22, pp. 157–182.

R. B. BLACKMAN AND J. W. TUKEY (1959), *The Measurements of Power Spectra*, Dover, New York.

P. BLOOMFIELD (1976), *Fourier Analysis of Time Series*, Wiley, New York.

R. P. BRENT (1976), *Fast multiple-precision evaluation of elementary functions*, J. Assoc. Comput. Mach., 23, pp. 242–251.

R. P. BRENT AND H. T. KUNG (1975), *$O((n \log n)^{3/2})$ algorithms for composition and reversion of power series*, preprint, Computer Sci. Dept., Carnegie-Mellon University, Pittsburgh.

——— (1976), *Fast algorithms for manipulating formal power series*, preprint, Computer Sci. Dept., Carnegie-Mellon University, Pittsburgh. To appear in J. Assoc. Comput. Mach., 1978.

——— (1977), *Fast algorithms for composition and reversion of multivariate formal power series*, Proc. Conf. Theoretical Comp. Sci. (University of Waterloo, Waterloo, Ontario), to appear.

O. BUNEMANN (1969), *A compact non-iterative Poisson solver*, Rep. SUIPR-294, Institute for Plasma Research, Stanford University, Stanford, CA.

B. L. BUZBEE, G. H. GOLUB AND C. W. NIELSON (1970), *On direct methods for solving Poisson's equation*, SIAM J. Numer. Anal., 7, pp. 627–656.

B. L. BUZBEE, F. W. DORR, J. A. GEORGE AND G. H. GOLUB (1971), *The direct solution of the discrete Poisson equation*, Ibid., 8, pp. 722–736.

B. L. BUZBEE AND F. W. DORR (1974), *The direct solution of the biharmonic equation on rectangular regions and the Poisson equation on irregular regions*, Ibid., 11, pp. 753–763.

J. F. CLAERBOUT (1976), *Fundamentals of Geophysical Data Processing, with Application to Petroleum Prospecting*, McGraw-Hill, New York.

L. COLLATZ (1955), *Numerische Behandlung von Differentialgleichungen*, Springer-Verlag, Berlin.

P. CONCUS AND G. H. GOLUB (1973), *Use of fast direct methods for efficient numerical solution of nonseparable elliptic equations*, SIAM J. Numer. Anal. 10, pp. 1103–1120.

J. W. COOLEY AND J. W. TUKEY (1965), *An algorithm for the machine calculation of complex Fourier series*, Math. Comput., 19, pp. 297–301.

J. W. COOLEY, P. A. W. LEWIS AND P. D. WELCH (1967a), *The Fast Fourier Transform algorithm and its applications*, Research paper RC-1743, IBM Research.

―――― (1967b), *Historical notes on the Fast Fourier Transform*, Proc. IEEE, 55, pp. 1674–1677.

G. DAHLQUIST AND A. BJÖRCK (1974), *Numerical Methods*, Prentice-Hall, Englewood Cliffs, NJ.

P. J. DAVIS (1959), *On the numerical integration of periodic analytic functions*, On Numerical Approximation, R. E. Langer, ed., University of Wisconsin Press, Madison, pp. 44–59.

G. DOETSCH (1950), *Handbuch der Laplace-Transformation*, vol. 1. Birkhäuser, Basel.

F. W. DORR (1970), *The direct solution of the discrete Poisson equation on a rectangle*, this Review, 12, pp. 248–263.

C. M. FIDUCCIA (1972), *Polynomial evaluation via the division algorithm; The Fast Fourier Transform revisited*, Proc. 4th Annual ACM Symposium on Theory of Computing.

D. FISCHER, G. H. GOLUB, O. HALD, C. LEIVA AND O. WIDLUND (1974), *On Fourier–Toeplitz methods for separable elliptic problems*, Math. Comput., 28, pp. 349–368.

D. GAIER (1964), *Konstruktive Methoden der konformen Abbildung*, Springer Tracts in Natural Philosophy, vol. 3, Springer-Verlag, Berlin.

―――― (1974), *Ableitungsfreie Abschätzungen bei trigonometrischer Interpolation und Konjugiertenbestimmung*, Computing, 12, pp. 145–148.

―――― (1976), *Intergralgleichungen erster Art und konforme Abbildung*, Math. Z., 147, pp. 113–129.

W. GANDER AND A. MAZZARIO (1972), *Numerische Prozeduren I* (in memoriam Heinz Rutischauser), Berichte der Fachgruppe für Computerwissenschaften 4, Eidgenössische Technische Hochschule, Zürich.

W. GAUTSCHI (1972), *Attenuation factors in practical Fourier analysis*, Numer. Math., 18, pp. 373–400.

E. GEKELER (1969), *Über Iterationsverfahren bei einer Klasse nichtlinearer Gleichungssysteme*, Dissertation, Bochum.

M. GUTKNECHT (1977), *Existence of a solution of the discrete Theodorsen equation for conformal mappings*, Math. Comput., 31, pp. 478–480.

―――― (1978), *Fast algorithms for the conjugate periodic function*, Research Report No. 78-05, Eidgenössische Technische Hochschule, Zürich.

R. W. HAMMING (1973), *Numerical Methods for Scientists and Engineers*, 2nd ed., McGraw-Hill, New York.

G. H. HARDY AND W. W. ROGOSINSKI (1944), *Fourier Series*, University Press, Cambridge.

J. K. HAYES, D. K. KAHANER AND R. KELLER (1972), *An improved method for numerical conformal mapping*, Math. Comput., 26, pp. 327–334.

P. HENRICI (1974), (1977), *Applied and Computational Complex Analysis*, vols. 1 and 2, John Wiley, New York.

―――― (1976), *Einige Anwendungen der schnellen Fouriertransformation*, Internationale Schriftenreihe zur numerischen Mathematik, 32, pp. 111–124.

R. HOCKNEY (1965), *A fast direct solution of Poisson's equation using Fourier analysis*, J. Assoc. Comput. Mach., 12, pp. 95–113.

―――― (1970), *The potential calculation and some applications*, Methods Computational Phys., 9, pp. 135–211.

―――― (1972), *Plasma, gravitational, and vortex simulation*, Computing as a Language of Physics, International Atomic Energy Agency, Vienna, pp. 95–107.

―――― (1972a), *The solution of Poisson's equation*, Computing as a Language of Physics, International Atomic Energy Agency, Vienna, pp. 119–127.

R. HOCKNEY AND T. R. BROWN (1975), *A lambda transition in classical electron film*, J. Phys. C: Solid State Phys., 8, pp. 1813–1822.

R. HOCKNEY AND D. R. K. BROWNRIGG (1974), *Effect of population II stars and three-dimensional motion on spiral structure*, Mon. Not. Roy. Astr. Soc., 167, pp. 351–357.

R. HOCKNEY AND S. P. GOEL (1975), *Phase transitions of two-dimensional potassium chloride*, Chem. Phys. Lett., 35, pp. 500–507.

R. HOCKNEY, R. A. WARRINER AND M. REISER (1974), *Two-dimensional particle models in semiconductor-device analysis*, Electron. Lett. 10, No. 23.

CH. ISELIN (1971), *An approach to fast Fourier transform*, Report 71-19, CERN, Geneva.

D. KAHANER (1970), *Matrix description of the Fast Fourier Transform*, IEEE Trans. Audio and Electroacoustics, AU-18, pp. 442–450.

——— (1978), *The Fast Fourier Transform by polynomial evaluation*, Z. Angew. Math. Phys., 29, pp. 387–394.

A. KARATSUBA AND Y. OFMAN (1962), *Multiplication of multidigit numbers on automata*, Dokl. Akad. Nauk SSSR, 145, pp. 293–294.

L. H. KOOPMANS (1974), *The Spectral Analysis of Time Series*, Academic Press, New York.

H. T. KUNG (1974), *On computing reciprocals of power series*, Numer. Math., 22, pp. 341–348.

J. N. LYNESS AND C. B. MOLER (1967), *Numerical differentiation of analytic functions*, SIAM J. Numer. Anal., 4, pp. 202–210.

J. H. MCCLELLAN AND T. W. PARKS (1972), *Eigenvalue and eigenvector decomposition of the discrete Fourier transform*, IEEE Trans. Audio Electroacoustics, AU-20, pp. 66–74.

W. NIETHAMMER (1966), *Iterationsverfahren bei der konformen Abbildung*, Computing, 1, pp. 146–153.

W. M. PICKERING (1977), *Some comments on the solution of Poisson's equation using Bickley's formula and Fast Fourier Transforms*, J. Inst. Math. Appl., 19, pp. 337–338.

G. POLYA AND G. SZEGÖ (1954), *Aufgaben und Lehrsätze der Analysis*, 2nd ed., Springer-Verlag, Berlin.

W. PROSKUROWSKI AND O. WIDLUND (1976), *On the numerical solution of Helmholtz's equation by the capacitance matrix method*, Math. Comput., 30, pp. 433–468.

G. U. RAMOS (1971), *Roundoff error analysis of the Fast Fourier Transform*, Ibid., 25, pp. 757–768.

A. ROSENFELD AND A. C. KAK (1976), *Digital Picture Processing*, Academic Press, New York.

C. RUNGE AND H. KÖNIG (1924), *Vorlesungen über Numerisches Rechnen*, Springer, Berlin.

A. SCHÖNHAGE AND V. STRASSEN (1971), *Schnelle Multiplikation grosser Zahlen*, Computing, 7, pp. 281–292.

M. SIEVEKING (1972), *An algorithm for the division of power series*, Ibid., 10, pp. 153–156.

G. T. SYMM (1966), *An integral equation method in conformal mapping*, Numer. Math., 9, pp. 250–258.

F. THEILHEIMER (1969), *A matrix version of the Fast Fourier Transform*, IEEE Trans. Audio and Electroacoustics, AU-17, pp. 158–161.

J. W. TUKEY (1967), *An introduction to the calculation of numerical spectrum analysis*, The Spectral Analysis of Time Series, B. Harris, ed., John Wiley, New York.

O. WING (1967), *An efficient method of numerical inversion of the Laplace transforms*, Computing, 2, pp. 11–17.

S. WINOGRAD (1976), *On computing the discrete Fourier Transform*, Proc. Nat. Acad. Sci. U.S.A., 73, pp. 1005–1006.

——— (1978), *On computing the discrete Fourier transform*, Math. Comput., 32, pp. 175–199.

A. ZYGMUND (1968), *Trigonometric Series*, vol. 1, 2nd ed., Cambridge University Press, London.