The Fast Hartley Transform

RONALD N. BRACEWELL, FELLOW, IEEE

A fast algorithm has been worked out for performing the Discrete Hartley Transform (DHT) of a data sequence of N elements in a time proportional to $N\log_2 N$. The Fast Hartley Transform (FHT) is as fast as or faster than the Fast Fourier Transform (FFT) and serves for all the uses such as spectral analysis, digital processing, and convolution to which the FFT is at present applied. A new timing diagram (stripe diagram) is presented to illustrate the overall dependence of running time on the subroutines composing one implementation; this mode of presentation supplements the simple counting of multiplies and adds. One may view the Fast Hartley procedure as a sequence of matrix operations on the data and thus as constituting a new factorization of the DFT matrix operator; this factorization is presented. The FHT computes convolutions and power spectra distinctly faster than the FFT.

I. Introduction

A sequence of N real numbers possesses a Discrete Hartley Transform (DHT) that is a sequence of the same length and is also real valued [1]. From the DHT one can return to the original sequence by applying the same transformation formula a second time. The convenience of not having to manage the real and imaginary parts either in separate arrays, or interleaved in one array of double length, or in other ingenious ways that have been adopted in various embodiments [2] of the Fourier transform commends the DHT for consideration in applications to numerical spectral analysis and convolution. Not having to allow for an inverse transformation that is different and differs among authors is also helpful.

The DHT is a suitable *substitute* for the Discrete Fourier Transform (DFT) for some purposes; however, if the real and imaginary parts of the DFT are expressly required then they are directly obtainable as the even and odd parts of the DHT. When the power spectrum is the desired goal, it may be obtained directly from the DHT without first calculating the real and imaginary parts of the DFT as in the usual way of calculating power spectra.

A fast algorithm has been developed for computing the DHT and is presented in this paper. By analogy with the Fast Fourier Transform (FFT) this algorithm will be referred to as the Fast Hartley Transform (FHT). Because of the general interest of the subject and in order to make the material as accessible as possible to nonspecialists and personal-computer users, computer programs in this paper have been presented in BASIC. Programs that reuse the arrays and programs in other languages including assembler language have also been tested.

In addition to the convenient features mentioned, the FHT is as fast as or faster than the FFT, by a factor that depends on circumstances discussed below. Convolution, as performed for digital filtering is, in most applications, distinctly faster. There are fast programs related to the FFT which are available for use on real data and do not compute the coefficients that are complex conjugates of other coefficients. To invert the output requires storage of an additional program that accepts complex data. For a fast inversion program see Parson [3].

A real, strictly reciprocal, integral transform [4], [5] which forms the basis for the present discrete formulation was presented in Proceedings of the IRE in 1942 by Ralph V. L. Hartley [6]. For this reason it is appropriate to name the DHT and FHT, which derive from his original idea, in his honor. Hartley was a Rhodes scholar, a Fellow of the IRE, and rose to be in charge of telephone line research at the Bell Telephone Laboratories (1918–1929). To generations of electrical engineers his name was well known through the Hartley oscillator [7], which was once the standard textbook example of an electronic sine-wave source. A simple system to analyze, it comprised a single triode amplifier with feedback derived from a tapped inductance.

II. Definition

Given a real function $f(\tau)$ for $\tau = 0,1,\cdots,N-1$ one defines the DHT $H(\nu)$ as the sum of the cosine and sine transforms

$$H(\nu) = N^{-1}\sum_{\tau=0}^{N-1} f(\tau)\operatorname{cas}(2\pi\nu\tau/N), \qquad \nu = 0,1,\cdots,N-1$$

where $\operatorname{cas}\theta \equiv \cos\theta + \sin\theta$. The inverse relation is

$$f(\tau) = \sum_{\nu=0}^{N-1} H(\nu)\operatorname{cas}(2\pi\nu\tau/N), \qquad \tau = 0,1,\cdots,N-1.$$

The symbol τ can be thought of as a mnemonic for time, while ν/N is like frequency measured in cycles per unit of time.

III. The Fast Algorithm

If the DHT is evaluated numerically from its defining expression, the time taken for long data sequences of length N is proportional to N^2 as with the DFT, because for each value of ν there are N evaluations of the product $f(\tau)\operatorname{cas}(2\pi\nu\tau/N)$ and there are N values of ν. Let N be expressible as 2 raised to some power P, i.e., $N = 2^P$. Just as with the FFT [4], [8]–[11], the number of arithmetic operations can be reduced to the order of $N\log_2 N$ or NP.

One way of proceeding is shown in the flow diagram of

Manuscript received August 9, 1983; revised February 7, 1984 and April 27, 1984. This work was performed while the author was on sabbatical leave granted by Stanford University.

The author is with the Electrical Engineering Department, Stanford University, Stanford, CA 94305, USA.

0018-9219/84/0800-1010$01.00 ©1984 IEEE

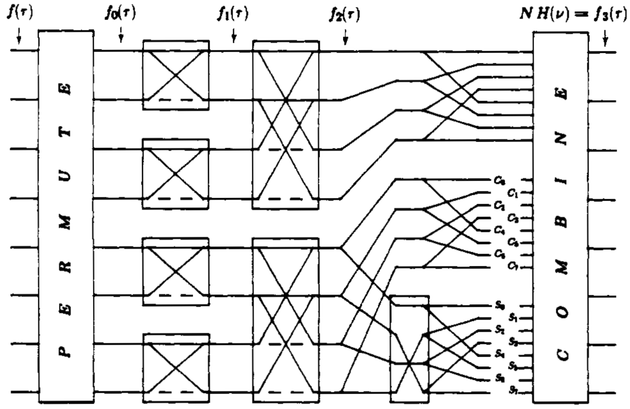

Fig. 1. Flow diagram for the Discrete Hartley Transform with $N = 8$ and $P = 3$. Broken lines represent transfer factors —1 while full lines represent unity transfer factor. The crossover boxes perform the sign reversal called for by the shift theorem which also requires the sine and cosine factors S_n, C_n.

Fig. 1 for the case of $N = 8$, $P = 3$. The operation labeled PERMUTE rearranges the sequence of data. The ith member is placed into the jth position where j is calculated from i as follows:

```
10R = I; J = 0
20 FOR K = 1  TO  P
30      S = R  DIV  2
40      J = J + J + R - S - S
50      R = S
60 NEXT K
```

The purpose of permutation is the same as with the FFT, namely to bisect the data sequence progressively until data pairs are reached. By definition when $N = 2$

$$\{a \quad b\} \text{ has DHT } \tfrac{1}{2}\{a + b \quad a - b\}$$

which is very simple. To superimpose all the two-element transforms requires a decomposition formula that expresses the DFT of a given sequence in terms of the DHTs of subsequences of half length. This formula permits, for example, the DHT of a four-element sequence $\{a_1 \quad a_2 \quad b_1 \quad b_2\}$ to be expressed in terms of the DHTs of the two interleaved two-element sequences $\{a_1 \quad b_1\}$ and $\{a_2 \quad b_2\}$. To derive the decomposition formula we require two theorems, the shift theorem and the similarity theorem. The shift theorem [1] states that if $f(\tau)$ has DHT $H(\nu)$ then

$$f(\tau + a) \text{ has DHT } H(\nu)\cos(2\pi a\nu/N) - H(N - \nu)\sin(2\pi a\nu/N).$$

The similarity theorem, which is the same as for the DFT [4], states that if a sequence $f(\tau)$ is stretched to double its length by inserting a zero element after each given element then the elements of the original DHT are repeated. As examples

$$\{1 \quad 2 \quad 3 \quad 4\} \text{ has DHT } \{2.5 \quad -1 \quad -0.5 \quad 0\}$$
$$\{1 \quad 0 \quad 2 \quad 0 \quad 3 \quad 0 \quad 4 \quad 0\}$$
$$\text{has DHT } \tfrac{1}{2}\{2.5 \quad -1 \quad -0.5 \quad 0 \quad 2.5 \quad -1 \quad -0.5 \quad 0\}.$$

Both these theorems are derivable in one line from the DHT definition. Suppose that

$$\{a_1 \quad a_2 \quad b_1 \quad b_2 \quad c_1 \quad c_2 \cdots\} \text{ has DHT } H(\nu)$$

where there are N elements. Then the a_1-sequence of $N/2$ elements $\{a_1 \quad b_1 \quad c_1 \cdots\}$ has DHT $\{\alpha_1 \quad \beta_1 \quad \gamma_1 \cdots\}$ and the a_2-sequence $\{a_2 \quad b_2 \quad c_2 \cdots\}$ has DHT $\{\alpha_2 \quad \beta_2 \quad \gamma_2 \cdots\}$. We see that

$$\begin{aligned}
H(\nu) &= \text{DHT of } \{a_1 \quad 0 \quad b_1 \quad 0 \quad c_1 \quad 0 \cdots\} \\
&\quad + \text{DHT of } \{0 \quad a_2 \quad 0 \quad b_2 \quad 0 \quad c_2 \quad 0 \cdots\} \\
&= \{\alpha_1 \quad \beta_1 \quad \gamma_1 \cdots \alpha_1 \quad \beta_1 \quad \gamma_1 \cdots\} \\
&\quad + \text{DHT of } \{0 \quad 1\} * \{a_2 \quad b_2 \quad c_2 \cdots\} \\
&= \{\alpha_1 \quad \beta_1 \quad \gamma_1 \cdots \alpha_1 \quad \beta_1 \quad \gamma_1 \cdots\} \\
&\quad + \Big\{\alpha_2 \quad \beta_2\cos(2\pi/N) \quad \gamma_2\cos(2\pi \cdot 2/N) \cdots \\
&\quad -\alpha_2 \quad \beta_2\cos\left(2\pi\left(\frac{N}{2} + 1\right)/N\right) \\
&\quad \gamma_2\cos\left(2\pi\left(\frac{N}{2} + 2\right)/N\right) \cdots\Big\} \\
&\quad + \Big\{0 \cdots \gamma_2\sin(2\pi \cdot 2/N) \quad \beta_2\sin(2\pi \cdot 3/N) \\
&\quad 0 \cdots \gamma_2\sin\left(2\pi\left(\frac{N}{2} + 2\right)/N\right) \\
&\quad \beta_2\sin\left(2\pi\left(\frac{N}{2} + 3\right)/N\right)\Big\}.
\end{aligned}$$

The general decomposition formula is thus

$$H(\nu) = H_{a_1}(\nu) + H_{a_2}(\nu)\cos(2\pi\nu/N) + H_{a_2}(N - \nu)\sin(2\pi\nu/N)$$

where $H_{a_1}(\nu)$ and $H_{a_2}(\nu)$ are, respectively, $\tfrac{1}{2}\{\alpha_1 \quad \beta_1 \quad \gamma_1 \cdots \alpha_1 \quad \beta_1 \quad \gamma_1 \cdots\}$ and $\tfrac{1}{2}\{\alpha_2 \quad \beta_2 \quad \gamma_2 \cdots \alpha_2 \quad \beta_2 \quad \gamma_2 \cdots\}$. In this derivation, we have put $a = 1$ in the shift theorem. In the first two operations following permutation on the left of Fig. 1 the sine and cosine factors assume only values of 0, 1, or -1. For this reason, the diagram can make use of two flow line types: unbroken lines that transmit values unchanged and broken lines that transmit with a sign reversal. In later stages sine and cosine factors are associated with two-thirds of the flow lines; of course, many of these factors are still 0, 1, or -1. For the case of $N = 16$, Table 1 summarizes the equations represented by the flow diagram using the level-dependent abbreviations C_n and S_n to stand for $\substack{\cos \\ \sin}(2\pi n/2^L)$, respectively, where L is the level number of the stage. Table 2 shows how the equations in fact simplify when the special values 0, 1, -1, and $r = 2^{-1/2}$ are substituted. The small boxes containing crossover connections, that appear in one-third of the connections in the later stages (Fig. 1) are to implement the sign reversal demanded by the factor $H(N - \nu)$ in the shift theorem. (When $\nu = 0$, $H(N)$ is assigned the value $H(0)$.)

An example (Table 3) will clarify the steps involved. As will be seen, the computations for $N = 8$ can easily be carried out by hand. For illustration, let the given data sequence be $f(\tau) = \{1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8\}$. Permuta-

Table 1 Equations Relating Successive Stages of the FHT as Indexed by Level Number L

Data	Permute	Level 1	Level 2	Level 3	16 × DHT
$F(0,0)$	$f(0,0) = F(0,0)$	$F(1,0) = F(0,0) + F(0,1)$	$F(2,0) = F(1,0) + F(1,2)C_0 + F(1,2)S_0$	$F(3,0) = F(2,0) + F(2,4)C_0 + F(2,4)S_0$	$F(4,0) = F(3,0) + F(3,8)C_0 + F(3,8)S_0$
$F(0,1)$	$F(0,1) = F(0,8)$	$F(1,1) = F(0,0) - F(0,1)$	$F(2,1) = F(1,1) + F(1,3)C_1 + F(1,3)S_1$	$F(3,1) = F(2,1) + F(2,5)C_1 + F(2,7)S_1$	$F(4,1) = F(3,1) + F(3,9)C_1 + F(3,15)S_1$
$F(0,2)$	$F(0,2) = F(0,4)$	$F(1,2) = F(0,2) + F(0,3)$	$F(2,2) = F(1,0) + F(1,2)C_2 + F(1,2)S_2$	$F(3,2) = F(2,2) + F(2,6)C_2 + F(2,6)S_2$	$F(4,2) = F(3,2) + F(3,10)C_2 + F(3,14)S_2$
$F(0,3)$	$F(0,3) = F(0,12)$	$F(1,3) = F(0,2) - F(0,3)$	$F(2,3) = F(1,1) + F(1,3)C_3 + F(1,3)S_3$	$F(3,3) = F(2,3) + F(2,7)C_3 + F(2,5)S_3$	$F(4,3) = F(3,3) + F(3,11)C_3 + F(3,13)S_3$
$F(0,4)$	$F(0,4) = F(0,2)$	$F(1,4) = F(0,4) + F(0,5)$	$F(2,4) = F(1,4) + F(1,6)C_0 + F(1,6)S_0$	$F(3,4) = F(2,0) + F(2,4)C_4 + F(2,4)S_4$	$F(4,4) = F(3,4) + F(3,12)C_4 + F(3,12)S_4$
$f(0,5)$	$F(0,5) = F(0,10)$	$F(1,5) = F(0,4) - F(0,5)$	$F(2,5) = F(1,5) + F(1,7)C_1 + F(1,7)S_1$	$F(3,5) = F(2,1) + F(2,5)C_5 + F(2,7)S_5$	$F(4,5) = F(3,5) + F(3,13)C_5 + F(3,11)S_5$
$F(0,6)$	$F(0,6) = F(0,6)$	$F(1,6) = F(0,6) + F(0,7)$	$F(2,6) = F(1,4) + F(1,6)C_2 + F(1,6)S_2$	$F(3,6) = F(2,2) + F(2,6)C_6 + F(2,6)S_6$	$F(4,6) = F(3,6) + F(3,14)C_6 + F(3,10)S_6$
$F(0,7)$	$F(0,7) = F(0,14)$	$F(1,7) = F(0,6) - F(0,7)$	$F(2,7) = F(1,5) + F(1,7)C_3 + F(1,7)S_3$	$F(3,7) = F(2,3) + F(2,7)C_7 + F(2,5)S_7$	$F(4,7) = F(3,7) + F(3,15)C_7 + F(3,9)S_7$
$F(0,8)$	$F(0,8) = F(0,1)$	$F(1,8) = F(0,8) + F(0,9)$	$F(2,8) = F(1,8) + F(1,10)C_0 + F(1,10)S_0$	$F(3,8) = F(2,8) + F(2,12)C_0 + F(2,12)S_0$	$F(4,8) = F(3,0) + F(3,8)C_8 + F(3,8)S_8$
$F(0,9)$	$F(0,9) = F(0,9)$	$F(1,9) = F(0,8) - F(0,9)$	$F(2,9) = F(1,9) + F(1,11)C_1 + F(1,11)S_1$	$F(3,9) = F(2,9) + F(2,13)C_1 + F(2,15)S_1$	$F(4,9) = F(3,1) + F(3,9)C_9 + F(3,15)S_9$
$F(0,10)$	$F(0,10) = F(0,5)$	$F(1,10) = F(0,10) + F(0,11)$	$F(2,10) = F(1,8) + F(1,10)C_2 + F(1,10)S_2$	$F(3,10) = F(2,10) + F(2,14)C_2 + F(2,14)S_2$	$F(4,10) = F(3,2) + F(3,10)C_{10} + F(3,14)S_{10}$
$F(0,11)$	$F(0,11) = F(0,13)$	$F(1,11) = F(0,10) - F(0,11)$	$F(2,11) = F(1,9) + F(1,11)C_3 + F(1,11)S_3$	$F(3,11) = F(2,11) + F(2,15)C_3 + F(2,13)S_3$	$F(4,11) = F(3,3) + F(3,11)C_{11} + F(3,13)S_{11}$
$F(0,12)$	$F(0,12) = F(0,3)$	$F(1,12) = f(0,12) + F(0,13)$	$F(2,12) = F(1,12) + F(1,14)C_0 + F(1,14)S_0$	$F(3,12) = F(2,8) + F(2,12)C_4 + F(2,12)S_4$	$F(4,12) = F(3,4) + F(3,12)C_{12} + F(3,12)S_{12}$
$F(0,13)$	$F(0,13) = F(0,11)$	$F(1,13) = F(0,12) - F(0,13)$	$F(2,13) = F(1,13) + F(1\ 15)C_1 + F(1,15)S_1$	$F(3,13) = F(2,9) + F(2,13)C_5 + F(2,15)S_5$	$F(4,13) = F(3,5) + F(3,13)C_{13} + F(3,11)S_{13}$
$F(0,14)$	$F(0,14) = F(0,7)$	$F(1,14) = F(0,14) + F(0,15)$	$F(2,14) = F(1,12) + F(1,14)C_2 - F(1,14)S_2$	$F(3,14) = F(2,10) + F(2,14)C_6 + F(2,14)S_6$	$F(4,14) = F(3,6) + F(3,14)C_{14} + F(3,10)S_{14}$
$F(0,15)$	$F(0,15) = F(0,15)$	$F(1,15) = F(0,14) - F(0,15)$	$F(2,15) = F(1,13) + F(1,15)C_3 + F(1,15)S_3$	$F(3,15) = F(2,11) + F(2,15)C_7 + F(2,13)S_7$	$F(4,15) = F(3,7) + F(3,15)C_{15} + F(3,9)S_{15}$

Table 2 Simplification Permitted by Explicit Substitution for the Sine and Cosine Factors of Table 1, many of which are 0, 1, or −1

Data	Permute	Level 1	Level 2	Level 3
$F(0,0)$	$F(0,0) = F(0,0)$	$F(1,0) = F(0,0) + F(0,1)$	$F(2,0) = F(1,0) + F(1,2)$	$F(3,0) = F(2,0) + F(2,4)$
$F(0,1)$	$F(0,1) = F(0,8)$	$F(1,1) = F(0,0) - F(0,1)$	$F(2,1) = F(1,1) + F(1,3)$	$F(3,1) = F(2,1) + rF(2,5) + rF(2,7)$
$F(0,2)$	$F(0,2) = F(0,4)$	$F(1,2) = F(0,2) + F(0,3)$	$F(2,2) = F(1,0) - F(1,2)$	$F(3,2) = F(2,2) \qquad + F(2,6)$
$F(0,3)$	$F(0,3) = F(0,12)$	$F(1,3) = F(0,2) - F(0,3)$	$F(2,3) = F(1,1) - F(1,3)$	$F(3,3) = F(2,3) - rF(2,7) + rF(2,5)$
$F(0,4)$	$F(0,4) = F(0,2)$	$F(1,4) = F(0,4) + F(0,5)$	$F(2,4) = F(1,4) + F(1,6)$	$F(3,4) = F(2,0) - F(2,4)$
$F(0,5)$	$F(0,5) = F(0,10)$	$F(1,5) = F(0,4) - F(0,5)$	$F(2,5) = F(1,5) + F(1,7)$	$F(3,5) = F(2,1) - rF(2,5) - rF(2,7)$
$F(0,6)$	$F(0,6) = F(0,6)$	$F(1,6) = F(0,6) + F(0,7)$	$F(2,6) = F(1,4) - F(1,6)$	$F(3,6) = F(2,2) \qquad - F(2,6)$
$F(0,7)$	$F(0,7) = F(0,14)$	$F(1,7) = F(0,6) - F(0,7)$	$F(2,7) = F(1,5) - F(1,7)$	$F(3,7) = F(2,3) + rF(2,7) - rF(2,5)$
$F(0,8)$	$F(0,8) = F(0,1)$	$F(1,8) = F(0,8) + F(0,9)$	$F(2,8) = F(1,8) + F(1,10)$	$F(3,8) = F(2,8) + F(2,12)$
$F(0,9)$	$F(0,9) = F(0,9)$	$F(1,9) = F(0,8) - F(0,9)$	$F(2,9) = F(1,9) + F(1,11)$	$F(3,9) = F(2,9) + rF(2,13) + rF(2,15)$
$F(0,10)$	$F(0,10) = F(0,5)$	$F(1,10) = F(0,10) + F(0,11)$	$F(2,10) = F(1,8) - F(1,10)$	$F(3,10) = F(2,10) \qquad + F(2,14)$
$F(0,11)$	$F(0,11) = F(0,13)$	$F(1,11) = F(0,10) - F(0,11)$	$F(2,11) = F(1,9) - F(1,11)$	$F(3,11) = F(2,11) - rF(2,15) + rF(2,13)$
$F(0,12)$	$F(0,12) = F(0,3)$	$F(1,12) = F(0,12) + F(0,13)$	$F(2,12) = F(1,12) + F(1,14)$	$F(3,12) = F(2,8) - F(2,12)$
$F(0,13)$	$F(0,13) = F(0,11)$	$F(1,13) = F(0,12) - F(0,13)$	$F(2,13) = F(1,13) + F(1,15)$	$F(3,13) = F(2,9) - rF(2,13) + rF(2,15)$
$F(0,14)$	$F(0,14) = F(0,7)$	$F(1,14) = F(0,14) + F(0,15)$	$F(2,14) = F(1,12) - F(1,14)$	$F(3,14) = F(2,10) \qquad - F(2,14)$
$F(0,15)$	$F(0,15) = F(0,15)$	$F(1,15) = F(0,14) - F(0,15)$	$F(2,15) = F(1,13) - F(1,15)$	$F(3,15) = F(2,11) + rF(2,15) - rF(2,13)$

Table 3 Numerical Example of a Short FHT with $N = 8$, $P = 3$

τ	$f(\tau)$	π	$f_0(\tau)$	$f_1(\tau)$	$f_2(\tau)$	$f_3(\tau)$	$H(\nu)$	ν
0	1	1	1	6	16	36	4.5	0
1	2	3	5	−4	−8	−13.6	−1.7	1
2	3	5	3	10	−4	−8	−1	2
3	4	7	7	−4	0	−5.6	−0.7	3
4	5	2	2	8	20	−4	−0.5	4
5	6	4	6	−4	−8	−2.4	−0.3	5
6	7	6	4	12	−4	0	0	6
7	8	8	8	−4	0	5.6	0.7	7

tion proceeds in $P - 1$ steps. The first step is to separate out the data into two four-element sequences $\{1 \quad 3 \quad 5 \quad 7\}$ and $\{2 \quad 4 \quad 6 \quad 8\}$, as in the column headed π. The second step separates each four-element sequence into two two-element sequences $\{1 \quad 5\}$, $\{3 \quad 7\}$, $\{2 \quad 6\}$, and $\{4 \quad 8\}$. As $P = 3$, the second step is the last step of permutation.

The sequence $\{1 \quad 5 \quad 3 \quad 7 \quad 2 \quad 6 \quad 4 \quad 8\}$ is the permuted sequence $f_0(\tau)$. Each two-element sequence $\{a \quad b\}$ is now transformed; thus $\{1 \quad 5\}$ gives $\frac{1}{2}\{6 \quad -4\}$, but we suppress the factors $\frac{1}{2}$ until the end. When concatenated, these elementary transforms constitute the first stage $f_1(\tau) = \{6 \quad -4 \quad 10 \quad -4 \quad 8 \quad -4 \quad 12 \quad -4\}$. Owing to the degeneracy exhibited in Table 1 the steps for obtaining $f_2(\tau)$ are also very simple. Thus $16 = 6 + 10$, $-8 = -4 - 4$, $-4 = 6 - 10$, $0 = -4 + 4$, etc. Some of the flow lines from Fig. 1 are included in Table 3 to facilitate cross reference. In the final stage, computation of $f_3(\tau)$ involves sines and cosines of eighths of a turn. In the operation labeled COMBINE there are three sets of eight inputs each and one set of eight outputs. The first of the eight outputs is the sum of the first elements of the three input sets and similarly for the others.

We remember that a factor $\frac{1}{2}$ was suppressed P times so the result $f_3(\tau)$ must be divided by 8 to conclude that

$$\{1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8\}$$

has DHT $\{4.5 \quad -1.7 \quad -1 \quad -0.7 \quad -0.5 \quad -0.3 \quad 0 \quad 0.7\}$.

As a check we note that the sum of the data equals 8 times the first element of the DHT.

If we wished to go on and get the DFT $R(\nu) + jX(\nu)$ we would proceed as follows. The real part $R(\nu)$ is equal to the even part of the DHT $H(\nu)$ and the imaginary part $X(\nu)$ equals the negative odd part. If we want the "power spectrum" $Z^2 = R^2 + X^2$ we do not have to go via the complex quantity $R + jX$; the power spectrum can be calculated directly from

$$Z^2 = \{[H(\nu)]^2 + [H(-\nu)]^2\}/2.$$

IV. Running Time

It is of interest to know how the FHT compares with the FFt in speed. The ratio depends on several factors including the language and the machine; this paper reports on experience using BASIC on the HP-85 personal computer. In order to normalize the numerical results one may introduce a figure of merit $\mathfrak{T}$ as follows. In the FFT the innermost arithmetic operation is a complex multiplication which can be carried out by four real multiplies, although a very clever method with three multiplies was invented by Buneman [12]. One may determine the time T_4 for four real multiplies by the following arbitrary but definite prescription:

```
10 A = RND;  B = RND;  C = RND;  D = RND
20 T0 = TIME
30 FOR I = 1 TO 1000;  R = A * C - B * D;
       X = B * C + A * D; NEXT I
40 T4 = TIME - T0;  PRINT T4; END
```

Then the running time required to take the FHT of a sequence of length $N = 2^P$ may be expressed as

$$T_{\text{FHT}} = \mathfrak{T} N P T_4.$$

For a standardized comparison I translated a well-known FFT program (Appendix I) published by Brigham [13] so as to run on my computer ($T_4 = 0.038$ s). In the range $1 < P < 11$, $\mathfrak{T}$ ranged from 3.1 to 5.8. Using the FHT to get the Fourier Transform yielded $\mathfrak{T} = 1.3$ approximately. Comparison with other programs [2], [14], [3] will require tests in a common language or languages, including assembly language, on a common machine. Programming style, which is important and accounts for some of the discrepancy in the test reported, also needs to be balanced. The fact that $\mathfrak{T}$ turns out to depend on N is an interesting reality that is not brought out by counting operations in the inner loop and will now be examined.

V. The Stripe Diagram

Running time dependence on N has been studied using a program given in Appendix II. For this purpose, and to keep the program as clear as possible, some simplifications have been adopted. For example, each successive stage is allocated to a separate array $F(L, I)$, where L is the "level" or stage number and I ranges from 0 to $N - 1$. In the notation used so far, $F(L, I) \equiv f_L(I)$. However $F(0, I)$ is used for the data $f(\tau)$ and reused for the permuted data $f_0(\tau)$. Stages 1 and 2 are then computed directly from the short equations of Table 2, without recourse to sines or cosines. After that, an iterative loop is entered; as each stage is completed, execution returns to the beginning of the loop.

By varying N we see how the components of $\mathfrak{T}$ behave in the "stripe diagram" of Fig. 2. There are three kinds of components to be prepared for. First, there are some "overhead" operations that do not depend much, if at all, on N and can be expected to die out rapidly when the ordinate is normalized with respect to NP.

Secondly, there are operations such as the precalculation of the needed sines and cosines of multiples of one Nth of a turn that are carried out a number of times proportional to N. (It is desirable to precalculate these factors, otherwise many of them must be redone many times.) Therefore, the contribution of the precalculation to the running time after

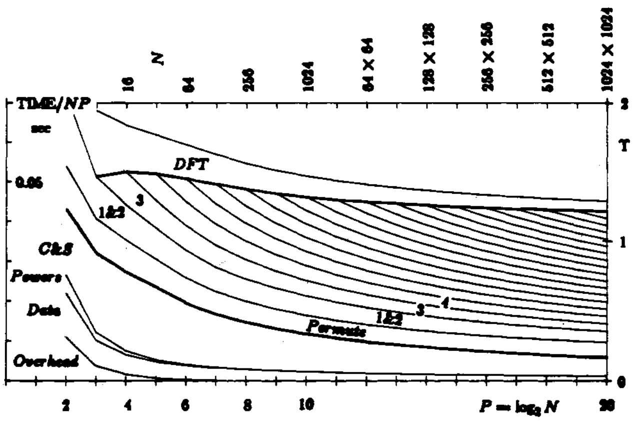

Fig. 2. Stripe diagram showing the figure of merit T versus P (upper heavy curve) for the FHT program of Appendix II and contributions made by overhead, data insertion, precalculation of powers of 2 and sines and cosines, permutation, and the various numbered stages. The additional time involved in going on to the real and imaginary parts of the Discrete Fourier Transform is shown by the top stripe. The ordinate scale on the left, which is specific to the author's personal computer, shows running time divided by NP.

division by NP dies out as P^{-1} as exhibited by the declining vertical thickness of the stripe labeled "C & S." A second example is the contribution of, let us say, stage 3. As N becomes larger, stage 3 requires more time in proportion to N. Therefore, the thickness of the stripe labeled "3" also decays as P^{-1}. But the number of such stripes increases as P; the zone bounded by heavy lines remains of approximately constant vertical thickness.

The third kind of behavior is exhibited by the contribution from permutation. Permutation involves N reassignments of variables, repeated P times; so the time required is proportional to NP and the permutation stripe therefore does not die out. Any effort made to streamline permutation is consequently most valuable. Merely replacing $J/2$ by $0.5*J$ or $2*J$ by $J+J$ makes a significant improvement in the end.

Stages 1 and 2 have been lumped together for the purposes of the stripe diagram. It is apparent that the stripe labeled "1 & 2" is of much the same width as the stripe for stage 3. This is the justification for using the short equations of Table 2. Further, but diminishing, gains would also be realized from further postponement of iteration.

If the final step to the DFT is taken, a thin decaying band of the second kind is added on top, ultimately becoming negligible.

The nondecaying zone, which is indicated by heavy lines in the stripe diagram, has a thickness of 0.038 s. This means that the running time, in the limit as P approaches infinity, will be 0.038 NP. In the neighborhood of $P = 10$ the running time is about one-third more than this. By special tricks one might hope to erode this one-third. The widest component comes from pretabulation using built-in sine and cosine functions and can be substantially reduced by another Buneman algorithm [15] or, as he explains, eliminated entirely when runs are repeated or dedicated hardware is introduced. Overhead and precalculation of powers of two are both negligible. Fetching and loading data are shown by this analysis to be also negligible in the current implementation. Permutation is possibly near the ideal limit.

VI. Matrix Formulation

A different and condensed view of the fast Hartley operator may be gained by formulating the eighty equations of Table 1 in matrix form. We may write

$$\mathbf{H} = N^{-1}\mathbf{L}_4\mathbf{L}_3\mathbf{L}_2\mathbf{L}_1\mathbf{P}\mathbf{f}$$

where $\mathbf{f}$ and $\mathbf{H}$ are respectively the N-element column matrices representing the data $f(\tau)$ and the Discrete Hartley Transform $H(\nu)$, $\mathbf{P}$ is the permutation matrix and the $\mathbf{L}_i$ are matrix operators which convert the column matrix operand to the column matrix of level i. In this example with $N = 16$ and $P = 4$, i runs from 1 to 4. A subsequent step of conversion to the Discrete Fourier Transform $F(\nu)$ may also be represented by the (complex) matrix multiplication

$$\mathbf{F} = \mathbf{\Phi}\mathbf{H}.$$

Combining the above equations, we obtain a new expression for the discrete Fourier transform

$$\mathbf{F} = N^{-1}\mathbf{\Phi}\mathbf{L}_P\mathbf{L}_{P-1}\cdots\mathbf{L}_1\mathbf{P}\mathbf{f}.$$

The matrix operator $N^{-1}\mathbf{\Phi}\mathbf{L}_P \cdots \mathbf{L}_1\mathbf{P}$ thus represents a new factorization of the DFT matrix operator $\mathbf{W}$, where $\mathbf{F} = \mathbf{W}\mathbf{f}$, $W = \exp(-i2\pi/N)$, and

$$\mathbf{W} = \begin{bmatrix} 1 & 1 & 1 & & & & \\ 1 & W & W^2 & & & & \\ 1 & W^2 & W^4 & & & & \\ & & & & & & \\ & & & & W^{(N-3)^2} & W^{(N-3)(N-2)} & W^{(N-1)(N-3)} \\ & & & & W^{(N-2)(N-3)} & W^{(N-2)^2} & W^{(N-2)(N-1)} \\ & & & & W^{(N-1)(N-3)} & W^{(N-1)(N-2)} & W^{(N-1)^2} \end{bmatrix}$$

The factors, which are directly verifiable from Table 1 are as follows:

$$\mathbf{P} = \left[\begin{array}{cccc|cccc|cccc|cccc}
1&&&&&&&&&&&&&&& \\
&&&&&&&&1&&&&&&& \\
&&&&1&&&&&&&&&&& \\
&&&&&&&&&&&&1&&& \\ \hline
&&1&&&&&&&&&&&&& \\
&&&&&&&&&&1&&&&& \\
&&&&&&1&&&&&&&&& \\
&&&&&&&&&&&&&&1& \\ \hline
&1&&&&&&&&&&&&&& \\
&&&&&&&&&1&&&&&& \\
&&&&&1&&&&&&&&&& \\
&&&&&&&&&&&&&1&& \\ \hline
&&&1&&&&&&&&&&&& \\
&&&&&&&&&&&1&&&& \\
&&&&&&&1&&&&&&&& \\
&&&&&&&&&&&&&&&1
\end{array}\right]$$

$$\mathbf{L}_1 = \left[\begin{array}{cccc|cccc|cccc|cccc}
1&1&&&&&&&&&&&&&& \\
1&-1&&&&&&&&&&&&&& \\
&&1&1&&&&&&&&&&&& \\
&&1&-1&&&&&&&&&&&& \\ \hline
&&&&1&1&&&&&&&&&& \\
&&&&1&-1&&&&&&&&&& \\
&&&&&&1&1&&&&&&&& \\
&&&&&&1&-1&&&&&&&& \\ \hline
&&&&&&&&1&1&&&&&& \\
&&&&&&&&1&-1&&&&&& \\
&&&&&&&&&&1&1&&&& \\
&&&&&&&&&&1&-1&&&& \\ \hline
&&&&&&&&&&&&1&1&& \\
&&&&&&&&&&&&1&-1&& \\
&&&&&&&&&&&&&&1&1 \\
&&&&&&&&&&&&&&1&-1
\end{array}\right]$$

$$\mathbf{L}_2 = \left[\begin{array}{cccccccccccccccc}
1 & & 1 & & & & & & & & & & & & & \\
& 1 & & 1 & & & & & & & & & & & & \\
1 & & -1 & & & & & & & & & & & & & \\
& 1 & & -1 & & & & & & & & & & & & \\
& & & & 1 & & 1 & & & & & & & & & \\
& & & & & 1 & & 1 & & & & & & & & \\
& & & & 1 & & -1 & & & & & & & & & \\
& & & & & 1 & & -1 & & & & & & & & \\
& & & & & & & & 1 & & 1 & & & & & \\
& & & & & & & & & 1 & & 1 & & & & \\
& & & & & & & & 1 & & -1 & & & & & \\
& & & & & & & & & 1 & & -1 & & & & \\
& & & & & & & & & & & & 1 & & 1 & \\
& & & & & & & & & & & & & 1 & & 1 \\
& & & & & & & & & & & & 1 & & -1 & \\
& & & & & & & & & & & & & 1 & & -1
\end{array}\right]$$

$$\mathbf{L}_3 = \left[\begin{array}{cccccccccccccccc}
1 & & & & C_0 & & & & & & & & & & & \\
& 1 & & & & C_1 & & S_1 & & & & & & & & \\
& & 1 & & & & K_2 & & & & & & & & & \\
& & & 1 & & S_3 & & C_3 & & & & & & & & \\
1 & & & & C_4 & & & & & & & & & & & \\
& 1 & & & & C_5 & & S_5 & & & & & & & & \\
& & 1 & & & & K_4 & & & & & & & & & \\
& & & 1 & & S_6 & & C_6 & & & & & & & & \\
& & & & & & & & 1 & & & & C_0 & & & \\
& & & & & & & & & 1 & & & & C_1 & & S_1 \\
& & & & & & & & & & 1 & & & & K_2 & \\
& & & & & & & & & & & 1 & & S_3 & & C_3 \\
& & & & & & & & 1 & & & & C_4 & & & \\
& & & & & & & & & 1 & & & & C_5 & & S_5 \\
& & & & & & & & & & 1 & & & & K_6 & \\
& & & & & & & & & & & 1 & & S_7 & & C_7
\end{array}\right]$$

$$\mathbf{L}_4 = \left[\begin{array}{cccccccccccccccc}
1 & & & & & & & & C_0 & & & & & & & \\
& 1 & & & & & & & & C_1 & & & & & & S_1 \\
& & 1 & & & & & & & & C_2 & & & & S_2 & \\
& & & 1 & & & & & & & & C_3 & & S_3 & & \\
& & & & 1 & & & & & & & & K_4 & & & \\
& & & & & 1 & & & & & & S_5 & & C_5 & & \\
& & & & & & 1 & & & & S_6 & & & & C_6 & \\
& & & & & & & 1 & & S_7 & & & & & & C_7 \\
1 & & & & & & & & K_8 & & & & & & & \\
& 1 & & & & & & & & C_9 & & & & & & S_9 \\
& & 1 & & & & & & & & C_{10} & & & & S_{10} & \\
& & & 1 & & & & & & & & C_{11} & & S_{11} & & \\
& & & & 1 & & & & & & & & K_{12} & & & \\
& & & & & 1 & & & & & & S_{13} & & C_{13} & & \\
& & & & & & 1 & & & & S_{14} & & & & C_{14} & \\
& & & & & & & 1 & & S_{15} & & & & & & C_{15}
\end{array}\right]$$

$$2\boldsymbol{\Phi} = \left[\begin{array}{cccccccccccccccc}
2 & & & & & & & & & & & & & & & \\
& 1-i & & & & & & & & & & & & & & 1+i \\
& & 1-i & & & & & & & & & & & & 1+i & \\
& & & 1-i & & & & & & & & & & 1+i & & \\
& & & & 1-i & & & & & & & & 1+i & & & \\
& & & & & 1-i & & & & & & 1+i & & & & \\
& & & & & & 1-i & & & & 1+i & & & & & \\
& & & & & & & 1-i & & 1+i & & & & & & \\
& & & & & & & & 2 & & & & & & & \\
& & & & & & & 1+i & & 1-i & & & & & & \\
& & & & & & 1+i & & & & 1-i & & & & & \\
& & & & & 1+i & & & & & & 1-i & & & & \\
& & & & 1+i & & & & & & & & 1-i & & & \\
& & & 1+i & & & & & & & & & & 1-i & & \\
& & 1+i & & & & & & & & & & & & 1-i & \\
& 1+i & & & & & & & & & & & & & & 1-i
\end{array}\right]$$

In the above equations C_n and S_n are the level-dependent abbreviations for $\begin{smallmatrix}\cos\\ \sin\end{smallmatrix}(2\pi n/2^L)$, respectively, and $K_n = C_n + S_n$.

The matrix representation offers a different way of viewing the Fast Hartley procedure. For example, $\mathbf{L}_4$ shows the retrograde indexing of the sine factors as elements on lines of slope 45°. $\mathbf{L}_1$ and $\mathbf{L}_2$ do not have such elements at all. One might also notice that $\mathbf{L}_1$ and $\mathbf{L}_2$ have only $2N$ nonzero elements compared with $3N$ in the limit for factors of higher level. Allowing for the time taken to access the trigonometric values one may gain an intimate understanding of the empirical timing results of Fig. 2.

VII. Convolution

In the vast majority of image processing applications, convolution is carried out between two functions of which one is symmetrical. Since two-dimensional convolution performed on an image can be reduced to one dimension by spreading the image out serially as in a television waveform, it will suffice here to speak in terms of one dimension. Under conditions where $f_1(\tau)$ has no particular symmetry and is to be convolved with $f_2(\tau)$ which is an even function, the convolution theorem for the DHT is [1]

$$f_1(\tau) * f_2(\tau) \text{ has DHT } H_1(\nu)H_2(\nu).$$

In other words, the DHT of a convolution of this type is the product of the two separate Hartley transforms. Therefore, to perform convolution, we take the two DHTs, multiply them together term by term, and take the DHT again. This procedure represents an improvement over taking the two DFTs, multiplying the complex values together, and inverting, since one complex multiplication $(a + jb)(c + jd) = ac - bd + j(ad + bc)$ stands for four real multiplications.

In the general case where $f_2(\tau)$ is not symmetrical, the convolution theorem has a second term. Let $H_2(\nu) = H_{2e}(\nu) + H_{2o}(\nu)$ where H_{2e} and H_{2o} are the even and odd parts of $H_2(\nu)$. (If $f_2(\tau)$ were symmetrical then the odd part of $H_{2o}(\nu)$ would be zero.) The general convolution theorem then reads

$$f_1(\tau) * f_2(\tau) = H_1(\nu)H_{2e}(\nu) + H_1(-\nu)H_{2o}(\nu).$$

This theorem is immediately deducible from the convolution theorem for the DFT.

Acknowledgment

Emeritus Professor Arthur Samuel caused me to tackle the FHT by casually remarking that my first paper did not go far enough, thus thrusting me into the arcane world of fast algorithms. Oscár Buneman was quick to recognize the potential of the FHT and generously shared his knowledge of FFT esoterica; as a result I was able to improve my first FHT program rather rapidly. I am grateful to Professor Buneman fo his interest in developing advanced assembly language versions of the FHT.

Appendix I
A Fast Fourier Program

A program for the FFT translated into HP BASIC from a widely accessible version published by Brigham [13]. Data are inserted in arrays $R(\ )$ and $I(\ )$ and the FFT is returned in the same arrays.

```
10 !          "FFT"
20 DIM R(1024), I(1024)
30 P=3 @ N=2^P
40 !  Insert data
50 FOR I=0 TO N-1
60    R(I)=I+1
```

```
 70    I(I)=0
 80 NEXT I
 90 GOSUB 220
100 END

110 !  Bit reversal function
120 DEF FN B(J)
130    J1=J
140    K0=0
150    FOR Q=1 TO P
160       J2=J1 DIV 2
170       K0=K0*2+(J1-2*J2)
180       J1=J2
190    NEXT Q
200    FN B=K0
210 FN END

220 !  Subr Get FFT
230 N2=N DIV 2
240 F1=P-1
250 K=0
260 FOR L=1 TO P
270    FOR I=1 TO N2
280       J=FN B(K) DIV 2^F1)
290       A=2*PI*J/N
300       C=COS(A)
310       S=SIN(A)
320       U=R(K+N2)*C+I(K+N2)*S
330       V=I(K+N2)*C-R(K+N2)*S
340       R(K+N2)=R(K)-U
350       I(K)=I(K)+V
380       K=K+1
390    NEXT I
400    K=K+N2
410    IF K<N THEN GOTO 270
420    K=0
430    F1=F1-1
440    N2=N2 DIV 2
450 NEXT L
460 FOR K=0 TO N-1
470    I=FN B(K)
480    IF I<K THEN GOTO 550
490    U=R(K)/N
500    V=I(K)/N
510    R(K)=R(I)/N
520    I(K)=I(I)/N
530    R(I)=U
540    I(I)=V
550 NEXT K
560 RETURN
```

Appendix II
A Fast Hartley Program that Also Gives the Fourier Transform

The following explanatory program computes the DHT of an N-element data set, $2 < N < 512$, and places it in $F(P,)$, where $P = \log_2 N$. The real and imaginary parts of the DFT appear in $R( )$ and $X( )$.

User furnishes the 2^P data values and changes line 50 to the appropriate value of $P, 1 < P < 9$. Data in algebraic form may be expressed on line 70 by substituting one's own function for the sample function $I + 1$. For numerical data, append the necessary DATA statements, delete lines 70–90, and change line 1010 to read FOR I = 0 TO N7 @ READ F(0, I) @ F(I, 1) = F(0, I) @ NEXT I.

The program is written for clarity rather than for speed. However, careful practice is exemplified. For example, in subroutine 7000 where the DFT is obtained from the DHT, two divisions by 2 are avoided (by relegation to the display subroutine 8000) and a multiplication by 2 is handled by addition.

Note that @ stands for a semicolon between separate statements under the one line number, that ! stand for REM (remark), and that the subroutine on line 8000 that tabulates the results needs adapting to local circumstances. Running time to completion of the DFT is assigned to the variable *T0* and is printed out at the end (see sample). Time taken by the HP-85 for $N = 256$ will be 2 min or in general 0.06 *NP*.

If $P > 8$ the 32K memory of the HP-85 overflows. On a machine with more memory change line 20 to read DIM$(P, 2^P)$; alternatively, edit to reuse the arrays.

Notice

The following software program is under copyright to Stanford University and is published as a part of this paper by permission. It may not be separately copied or used in any way directly or indirectly in a computer. For information of licensing arrangements, please contact Lisa Kuuttila, Consulting Associate, Office of Technology Licensing, 105 Encina Hall, Stanford University, Stanford CA 94305; telephone (415) 497-0651.

```
  10 !            "FHTBAS"
  20 DIM F(8,256) ! Hartley
  30 DIM R(129),X(129) ! Fourier
  40 DIM S(256), C(256) ! sin & cos
  50 P=3 ! 1<P<9
  60 N4=2^(P-2) @ N2=N4+N4 @ N=N2
     +N2 @ N7=N-1 @ P7=P-1
  70 DEF FN F(I)
  80    FN F=I+1 ! Sample function
  90 FN END
 100 T0=TIME ! Start timer
 110 GOSUB 1000 ! Insert data
 120 GOSUB 2000 ! Get powers of 2
 130 GOSUB 3000 ! Get C() & S()
 140 GOSUB 4000 ! Permute
 150 GOSUB 5000 ! Stages 1 & 2
 160 GOSUB 6000 ! Stages 3,4...8
 170 GOSUB 7000 ! Get DFT
 180 T0=TIME-T0 ! Stop timer
 190 GOSUB 8000 ! Show results
 200 END

1000 ! Subr Insert data
1010 FOR I=0 TO N7 @ F(0,I),F(1,
     I)=FNF(I) @ NEXT I
1020 RETURN
1550 NEXT I
2000 ! Subr Get powers of 2
```

```
2010 I=1 @ M(0)=1 @ M(1)=2
2020 M(I+1)=M(I)+M(I)
2030 I=I+1
2040 IF I<P THEN GOTO 2020
2050 RETURN
3000 ! Subr Get sines & cosines
3010 W=2*PI/N @ A=0
3020 FOR I=1 TO N @ A=A+W @ S(I)
     =SIN(A) @ C(I)=COS(A) @ NEX
     T I
3030 RETURN
4000 ! Subr Permute
4010 J, I=-1
4020 I=I+1 @ T=P
4030 T=T-1 @ J=J-M(T)
4040 IF J>=-1 THEN GOTO 4030
4050 J=J+M(T+1)
4060 IF I<=J THEN GOTO 4020
4070 T=F(0,I+1)
4080 F(0,I+1)=F(0,J+1)
4090 F(0,J+1)=T
4100 IF I<N-3 THEN GOTO 4020
4110 RETURN
5000 ! Subr Stages 1 & 2
5010 ! Get F(1,I), 2 element DFTs
5020 FOR I=0 TO N-2 STEP 2
5030   F(1,I)=F(0,I)+F(0,I+1)
5040   F(1,I+1)=F(0,I)-F(0,I+1)
5050 NEXT I
5060 IF P=1 THEN GOTO 170 ! Done
5070 ! Get F(2,I), 4 element DFTs
     using Table B
5080 L,M=2
5090 FOR I=0 TO N-4 STEP 4
5100   F(2,I)=F(1,I)+F(1,I+2)
5110   F(2,I+1)=F(1,I+1)+F(1,I+3)
5120   F(2,I+2)=F(1,I)-F(1,I+2)
5130   F(2,I+3)=F(1,I+1)-F(1,I+3)
5140 NEXT I
5150 IF P=2 THEN GOTO 170 ! Done
5160 RETURN
6000 ! Subr Stages 3,4...8
6010 U=P7
6020 S=4
6030 FOR L=2 to P7
6040    S2=S+S
6050    U=U-1
6060    S0=M(U-1)
6070    FOR Q=0 TO N7 STEP S2
6080        I=Q
6090        D=I+S
6100        F(L+1,I)=F(L,I)+F(L,D)
6110        F(L+1,D)=F(L,I)-F(L,D)
6120        K=D-1
6130        FOR J=S0 TO N4 STEP S0
6140           I=I+1
6150           D=I+S
6160           E=K+S
6170           Y=F(L,D)*C(J)+F(L,E)*S(J)
6180           Z=F(L,D)*S(J)-F(L,E)*C(J)
6190           F(L+1,I)=F(L,I)+Y
6200           F(L+1,D)=F(L,I)-Y
6210           F(L+1,K)=F(L,K)+Z
6220           F(L+1,E)=F(L,K)-Z
6230           K=K-1
6240        NEXT J
6250        E=K+S
6260    NEXT Q
6270    S=S2
6280 NEXT L
6290 RETURN
7000 ! Subr Get DFT
7010 R(0)=F(L,0)+F(L,0) @ X(0)=0
7020 FOR I=1 TO N2
7030    B=F(L,N-I)
7040    R(I)=F(L,I)+B
7050    X(I)=F(L,I)-B
7060 NEXT I
7070 RETURN
8000 ! Subr Show results
8010 CLEAR @ PRINT @ PRINT "τ,ν f(τ)
     H(ν)     R(ν)  + j X(ν)"@PRINT
8020 J$=""
8030 FOR I=0 TO N7
8040    J=MIN(I,N-I) ! Reflect
8050    F=INT(.5+1000*FNF(I))/1000
8060    H=INT(.5+1000/N*F(L,I))/1000
8070    R=INT(.5+1000*R(J)/N)/2000
        @ X=INT(.5+1000*X(J)/N)/2000
        *SGN(N2-I)
8080    J$="+j "&VAL$(ABS(X)) @ IF
        SGN(X)=-1 THEN J$="-j "&VAL
        $(ABS(X))
8090    IF X =0 THEN J$=""
8100    PRINT I;TAB(6);F;TAB(12);H;
        TAB(21);R;J$
8110 NEXT I
8120 PRINT @ PRINT " N="&VAL$(N)
     &", Time was &VAL$(T0)&
     "sec"
8130 RETURN
```

```
τ,ν     f(τ)     H(ν)      R(ν) +j X(ν)

0       1         4.5       4.5
1       2        -1.707    -.5 -j 1.207
2       3        -1        -.5 -j .5
3       4        -.707     -.5 -j .207
4       5        -.5       -.5
5       6        -.293     -.5 +j .207
6       7         0        -.5 +j .5
7       8         .707     -.5 +j 1.207

 N=8,  Time was 1.436 sec
```

References

[1] R. N. Bracewell, "The Discrete Hartley Transform," *J. Opt. Soc. Amer.*, vol. 73, pp. 1832–1835, Dec. 1983.

[2] Digital Signal Processing Committee, IEEE ASSP Soc., Eds., *Programs for Digital Signal Processing.* New York: IEEE PRESS,

1979.

[3] T. W. Parson, "A Winograd-Fourier transform algorithm for real-valued data," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-27, no. 4, pp. 398–402, Aug. 1979.

[4] R. N. Bracewell, *The Fourier Transform and Its Applications*, 2nd ed. New York: McGraw-Hill, 1965, 1978, p. 179.

[5] S. Goldman, *Frequency Analysis, Modulation and Noise*. New York: McGraw-Hill, 1948, pp. 60 and 66.

[6] R. V. L. Hartley, "A more symmetrical Fourier analysis applied to transmission problems," *Proc. IRE*, vol. 30, pp. 144–150, Mar. 1942.

[7] F. E. Terman, "*Radio Engineers' Handbook*. New York: McGraw-HIll, 1943. The oscillator was first described in a patent disclosure dated Feb. 10, 1915.

[8] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Math. Comput.*, vol. 19, pp. 297–301, Apr. 1965.

[9] Papers in *IEEE Trans. Audio Electroacoust.*, vol. AU-2, June 1967.

[10] G. D. Bergland, "A guided tour of the Fast Fourier Transform," *Spectrum*, vol. 6, pp. 41–52, July 1969.

[11] Digital Signal Processing Committee, IEEE ASSP Soc., Eds., *Selected Papers in Digital Signal Processing—II*. New York: IEEE PRESS, 1979.

[12] O. Buneman, "Inversion of the Helmholtz (or Laplace-Poisson) operator for slab geometry," *J. Comput. Phys.*, vol. 12, no. 1, pp. 124–130. May 1973.

[13] E. O. Brigham, *The Fast Fourier Transform*. Englewood Cliffs, NJ: Prentice-Hall, 1974.

[14] G. D. Bergland, "A fast Fourier algorithm for real-valued series," *Commun. Assoc. Comput. Mach.*, vol. 11, pp. 703–710, Oct. 1968.

[15] O. Buneman, "Two Hartley Transforms for the price of one FFT," submitted for publication.