[7] J. M. Tribolet, "A new phase unwrapping algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-25, pp. 170-175, Apr. 1977.

[8] A. V. Oppenheim, "Speech analysis-synthesis system based on homomorphic filtering," *J. Acoust. Soc. Amer.*, vol. 45, pp. 458-464, Feb. 1969.

[9] A. V. Oppenheim and R. W. Shafer, "Homomorphic analysis of speech," *IEEE Trans. Audio Electroacoust.*, vol. AU-16, pp. 221-226, June 1968.

[10] D. G. Childers, D. P. Skinner, and R. C. Kemerait, "The cepstrum: A guide to processing," *Proc. IEEE*, vol. 65, pp. 1428-1443, Oct. 1977.

On the Use of Symmetry in FFT Computation

LAWRENCE R. RABINER, FELLOW, IEEE

Abstract—It is well known that if a finite duration, N-point sequence $x(n)$ possesses certain symmetries, the computation of its discrete Fourier transform (DFT) can be obtained from an FFT of size $N/2$ or smaller. This is accomplished by first preprocessing the sequence, taking the FFT of the processed sequence, and then postprocessing the results to give the desired transform. In this paper we show how a similar approach can be used for sequences which are known to have only odd harmonics. The approach is shown to be essentially the dual of the known method for time symmetry. Computer programs are included for implementing the special procedures discussed in this paper.

I. Time Domain Symmetries and the FFT

Consider the N-point, finite duration sequence $x(n)$, defined for $0 \leqslant n \leqslant N-1$, with discrete Fourier transform $X(k)$ defined as [1]-[3]

$$X(k) = \sum_{n=0}^{N-1} x(n)\, W_N^{nk}, \qquad k = 0, 1, \cdots, N-1 \tag{1}$$

where

$$W_N = e^{-j(2\pi/N)}. \tag{2}$$

In general, an N-point complex FFT is required to give the DFT $X(k)$ for an arbitrary sequence. However, when $x(n)$ is real, an $N/2$-point complex FFT can be used to give $X(k)$ [4]. (Equivalently, one can use an FFT routine which accepts real inputs and gives the complex DFT as output [5], [6]. We denote such a routine as a (real) FFT.)

In many cases either $x(n)$ or $X(k)$ possesses certain desirable properties which can be exploited to reduce the amount of computation to obtain the desired DFT. The most notable of these properties are the time symmetries. A *symmetric* sequence is defined as one for which

$$x(n) = x(N-n), \qquad n = 1, 2, \cdots, N/2 - 1 \tag{3}$$

Manuscript received August 2, 1978; revised December 22, 1978.

The author is with the Acoustics Research Department, Bell Laboratories, Murray Hill, NJ 07974.

and an *antisymmetric* sequence is defined as one for which

$$x(n) = -x(N-n), \qquad n = 1, 2, \cdots, N/2 - 1. \tag{4}$$

Cooley *et al.* [4] have shown that if the sequence is either symmetric, or antisymmetric, a simple procedure can be used to reduce the computation of the DFT to that for an $N/4$ point (complex input) FFT with preprocessing and postprocessing. The algorithm works as follows (assuming $x(n)$ is a symmetric, real sequence). We express the N-point DFT as

$$X(k) = \sum_{\substack{n=0 \\ n \text{ even}}}^{N-1} x(n)\, W_N^{nk} + \sum_{\substack{n=0 \\ n \text{ odd}}}^{N-1} x(n)\, W_N^{nk}. \tag{5}$$

We then define the DFT's $A(k)$ and $B(k)$ as

$$A(k) = \sum_{\substack{n=0 \\ n \text{ even}}}^{N-1} x(n)\, W_N^{nk}, \qquad k = 0, 1, \cdots, N-1 \tag{6}$$

$$B(k) = \sum_{\substack{n=0 \\ n \text{ odd}}}^{N-1} x(n)\, W_N^{nk}, \qquad k = 0, 1, \cdots, N-1 \tag{7}$$

so

$$X(k) = A(k) + B(k), \qquad k = 0, 1, \cdots, N-1. \tag{8}$$

We next preprocess $x(n)$ to give the $N/2$-point real sequence $y(n)$, defined as

$$y(n) = x(2n) + [x(2n+1) - x(2n-1)], \qquad n = 0, 1, \cdots, N/2 - 1 \tag{9a}$$

$$= a(n) + c(n) \tag{9b}$$

where

$$a(n) = x(2n) \tag{10a}$$

$$c(n) = x(2n+1) - x(2n-1) \tag{10b}$$

and all indices are interpreted modulo N. It is readily seen that the sequence $a(n)$ is itself a symmetric, real sequence; hence its DFT is purely real (and, of course, even). The sequence $c(n)$ is an antisymmetric, real sequence; hence its DFT is purely imaginary (and odd). Thus, if we take the $N/2$-point (real input) FFT of $y(n)$, we get the result

$$Y(k) = \sum_{n=0}^{N/2-1} y(n) W_{N/2}^{nk}, \qquad k = 0, 1, \cdots, N/2 - 1$$

$$= \sum_{n=0}^{N/2-1} x(2n) W_{N/2}^{nk} + \sum_{n=0}^{N/2-1} (x(2n+1) - x(2n-1)) W_{N/2}^{nk}. \tag{11}$$

The first term in (11) is $A(k)$, the $N/2$-point DFT of the even components of $x(n)$ (defined for $0 \leq k \leq N/2 - 1$). The second term in (11) can readily be shown to be of the form

$$\sum_{n=0}^{N/2-1} (x(2n+1) - x(2n-1)) W_{N/2}^{nk}$$

$$= \sum_{\substack{n'=0 \\ n' \text{ odd}}}^{N-1} x(n') W_N^{n'k} W^{-k} - \sum_{\substack{n'=0 \\ n' \text{ odd}}}^{N-1} x(n') W_N^{n'k} W^{+k} \tag{12}$$

$$= (W^{-k} - W^{k}) \sum_{\substack{n'=0 \\ n' \text{ odd}}}^{N-1} x(n') W_N^{n'k} \tag{13}$$

$$= 2j \sin\left(\frac{2\pi}{N} k\right) B(k), \qquad k = 0, 1, \cdots, N/2 - 1. \tag{14}$$

Thus, $Y(k)$ of (11) can be written as

$$Y(k) = A(k) - 2j \sin\left(\frac{2\pi}{N} k\right) B(k). \tag{15}$$

Using the properties of $A(k)$ and $B(k)$, we get

$$A(k) = \text{Re}\,[Y(k)], \qquad k = 0, 1, \cdots, N/2 - 1 \tag{16a}$$

$$B(k) = \frac{\text{Im}\,[Y(k)]}{2 \sin\left(\frac{2\pi}{N} k\right)}, \qquad k = 1, 2, \cdots, N/2 - 1. \tag{16b}$$

For $k = 0$ and $k = N/2$, $B(k)$ is not defined from (16b); instead these values are obtained directly as

$$B(0) = \sum_{\substack{n=0 \\ n \text{ odd}}}^{N-1} x(2n+1) \tag{17a}$$

$$B(N/2) = -B(0). \tag{17b}$$

Thus, from the $N/2$-point real FFT of $y(n)$, the N-point real FFT of $x(n)$ can be recovered using (8), (16), and (17). In addition, $x(n)$ need only be specified for $0 \leq n \leq N/2$. A summary of the procedure for obtaining $X(k)$ is as follows.

1) Compute $B(0)$ as

$$B(0) = \sum_{n=0}^{N-1} x(2n+1) = 2 \sum_{n=0}^{N/4-1} x(2n+1).$$

2) Form the sequence $y(n)$ as

$$y(n) = x(2n) + (x(2n+1) - x(2n-1)), \qquad n = 1, 2, \cdots, N/4 - 1$$

$$y(N/2 - n) = x(2n) - (x(2n+1) - x(2n-1)), \qquad n = 1, 2, \cdots, N/4 - 1$$

$$y(0) = x(0)$$

$$y(N/4) = x(N/2).$$

3) Take the $N/2$-point real FFT of $y(n)$; call this result $Y(k)$, $0 \leq k \leq N/2 - 1$.

4) Form $A(k)$, $B(k)$ as

$$A(k) = \text{Re}\,[Y(k)], \qquad k = 0, 1, 2, \cdots, N/4$$

$$B(k) = \frac{\text{Im}\,[Y(k)]}{2 \sin\left(\frac{2\pi}{N} k\right)}, \qquad k = 1, 2, \cdots, N/4.$$

5) Form $X(k)$ as

$$X(k) = A(k) + B(k), \qquad k = 1, 2, \cdots, N/4$$

$$X(N/2 - k) = A(k) - B(k), \qquad k = 1, 2, \cdots, N/4$$

$$X(0) = B(0) + A(0)$$

$$X(N/2) = A(0) - B(0).$$

An implementation of this procedure is given in Appendix I.

A similar procedure is used when the sequence $x(n)$ is odd. The sequence $y(n)$ of (9) is again formed; however, $a(n)$ is now an odd, real sequence, and $c(n)$ is an even, real sequence. Appropriate modifications are made in the procedure to reflect these differences. An implementation of this procedure is given in Appendix II.

II. Reduction in Computation for Odd Harmonic Sequences

In the special case where $x(n)$ is a real sequence that is known to have a DFT for which only the odd harmonics are present, i.e.,

$$X(k) = 0, \qquad k \text{ even}, \tag{18}$$

we can also take advantage of this special symmetry by using a frequency domain approach. In this case, we first form the inverse DFT of $X(k)$, giving

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-nk}, \qquad n = 0, 1, \cdots, N - 1 \tag{19}$$

$$= \frac{1}{N} \sum_{\substack{k=0 \\ k \text{ odd}}}^{N-1} X(k) W_N^{-nk} + \frac{1}{N} \sum_{\substack{k=0 \\ k \text{ even}}}^{N-1} X(k) W_N^{-nk} \tag{20}$$

$$= a(n) + b(n) \tag{21}$$

where

$$a(n) = \frac{1}{N} \sum_{\substack{k=0 \\ k \text{ odd}}}^{N-1} X(k) W_N^{-nk} \tag{22a}$$

$$b(n)=\frac{1}{N}\sum_{\substack{k=0\\ k\text{ even}}}^{N-1} X(k)\,W_N^{-nk}. \tag{22b}$$

For sequences with only odd harmonics, (18) shows that $b(n)=0$. We now form the (complex) sequence $Y(k)$ as

$$Y(k)=X(2k)+j[X(2k+1)-X(2k-1)],\quad k=0,1,\cdots,N/2-1. \tag{23}$$

The $N/2$-point inverse DFT of $Y(k)$ is obtained as

$$y(n)=\frac{1}{(N/2)}\sum_{k=0}^{N/2-1} Y(k)\,W_{N/2}^{-nk},\quad n=0,1,\cdots,\frac{N}{2}-1 \tag{24}$$

$$=\frac{1}{(N/2)}\sum_{k=0}^{N/2-1} X(2k)\,W_{N/2}^{-nk}+\frac{j}{(N/2)}\sum_{k=0}^{N/2-1}[X(2k+1)-X(2k-1)]\,W_{N/2}^{-nk}. \tag{25}$$

The first term in (25) is readily seen to be $2b(n)$ (which is identically 0 for this case since $X(2k)=0$ for all k). The second term in (25) can be written as

$$\frac{j}{(N/2)}\sum_{k=0}^{N/2-1}[X(2k+1)-X(2k-1)]\,W_{N/2}^{-nk}=2j\,[W_N^{n}-W_N^{-n}]\,a(n). \tag{26}$$

Therefore,

$$y(n)=4\sin\left(\frac{2\pi n}{N}\right)a(n)=4\sin\left(\frac{2\pi n}{N}\right)x(n),\quad n=0,1,2,\cdots,N/2-1. \tag{27}$$

Thus, the procedure for obtaining $X(k)$ is as follows.

1) First form the coefficients Re $[X(1)]$ and Im $[X(N/2)-1)]$ directly from the relations

$$\text{Re}\,[X(1)]=\sum_{n=0}^{N-1}x(n)\cos\left(\frac{2\pi}{N}n\right) \tag{28a}$$

$$\text{Im}\,[X(N/2-1)]=\sum_{n=0}^{N-1}x(n)\cos\left(\frac{2\pi}{N}n\right)(-1)^n. \tag{28b}$$

It is readily shown that for a sequence containing only odd harmonics, the signal $x(n)$ satisfies the relation

$$x(n+N/2)=-x(n),\quad n=0,1,\cdots,N/2-1, \tag{29}$$

i.e., the upper half of the sequence is the negative of the lower half of the sequence. Thus, exploiting (29), only $N/2$ values of $x(n)$ need be stored, and Re $[X(1)]$ and Re $[X(N/2-1)]$ can be efficiently computed as

$$T1=\sum_{\substack{n=0\\ n\text{ even}}}^{N/2-1}x(n)\cos\left(\frac{2\pi}{N}n\right) \tag{30a}$$

$$T2=\sum_{\substack{n=0\\ n\text{ odd}}}^{N/2-1}x(n)\cos\left(\frac{2\pi}{N}n\right) \tag{30b}$$

$$\text{Re}\,[X(1)]=2[T1+T2] \tag{30c}$$

$$\text{Re}\left[X\left(\frac{N}{2}-1\right)\right]=2[T1-T2]. \tag{30d}$$

2) Weight the sequence $x(n)$ by $4\sin((2\pi/N)n)$ for $n=0,1,\cdots,N/2-1$, i.e.,

$$y(n)=x(n)4\sin\left(\frac{2\pi}{N}n\right),\quad 0\leqslant n\leqslant N/2-1.$$

3) Take the $N/2$-point real FFT of $y(n)$. Call this result $Y(k)$, $0\leqslant k\leqslant N/2-1$.

4) Recursively solve (23) for $X(2k+1)$ (since $X(2k)=0$) for $k=1,2,\cdots,N/4-2$, with appropriate initial conditions, i.e.,

$$\begin{aligned}
\text{Im}\,[X(1)]&=-\text{Re}\,[Y(0)]/2\\
\text{Re}\,[X(1)]&=2\cdot(T1+T2)\\
\text{Re}\,[X(2k+1)]&=\text{Im}\,[Y(k)]+\text{Re}\,[X(2k-1)],\\
&\qquad k=1,2,\cdots,N/4-2\\
\text{Im}\,[X(2k+1)]&=-\text{Re}\,[Y(k)]+\text{Im}\,[X(2k-1)],\\
&\qquad k=1,2,\cdots,N/4-2\\
\text{Im}\,[X(N/2-1)]&=\text{Re}\,[Y(N/4)]/2\\
\text{Re}\,[X(N/2-1)]&=2(T1-T2).
\end{aligned}$$

A computer program that implements this procedure is given in Appendix III.

III. Sequences with Both Time Symmetry and Having Only Odd Harmonics

The sequence $x(n)$ can simultaneously possess special properties in both the time domain (i.e., time symmetry) and the frequency domain (i.e., odd harmonics only). For such sequences, one can essentially apply the algorithms discussed in Sections I and II in sequence. The simplest procedure is to use one algorithm first, and at the place where the FFT is called, simply insert the call for the second algorithm with appropriate code to account for the format in which the transform is returned. However, as in most cases, such a simple approach is not generally as efficient as it can be made, since the additional properties of the sequence can be exploited to reduce computation and/or memory.

By way of example, consider a sequence which is symmetric in time and which is known to contain only odd harmonics. We again denote this sequence as $x(n)$, defined for $0\leqslant n\leqslant N-1$. It is readily shown that $x(n)$ satisfies the relations

$$x(N/2-n)=-x(n),\quad n=0,1,\cdots,N/4-1 \tag{31a}$$

$$x(N/4)=x(3N/4)=0 \tag{31b}$$

in addition to the usual symmetric sequence (3) and odd harmonic sequence (18), (29) relations. Thus, one need specify $x(n)$ for $0\leqslant n\leqslant N/4-1$ to uniquely define this sequence. As such, simply imbedding, for example, the odd harmonic subroutine within the symmetric sequence subroutine yields an inefficient algorithm. For maximum efficiency, we first preprocess $x(n)$ to form the $N/2$-point sequence $y(n)$ as

$$y(n) = x(2n) + [x(2n+1) - x(2n-1)], \quad n = 1, 2, \cdots, N/8 - 1 \tag{32a}$$

$$y(N/4 - n) = -x(2n) + [x(2n+1) - x(2n-1)], \quad n = 1, 2, \cdots, N/8 - 1 \tag{32b}$$

with initial and final values

$$y(0) = x(0) \tag{32c}$$

$$y(N/8) = -2x(N/4 - 1). \tag{32d}$$

The sequence $y(n)$ represents the first $N/4$ points of an odd harmonic $N/2$-point sequence. Thus, we can use the procedure of Appendix III directly on $y(n)$ to give the $N/2$-point complex odd harmonic sequence $Y(k)$. The desired DFT $X(k)$ is obtained from $Y(k)$ via the relation

$$X(2k+1) = \left[\operatorname{Im}[Y(2k+1)] + \operatorname{Re}[Y(2k+1)] \bigg/ \left[2 \sin\left(\frac{2\pi}{N}(2k+1)\right)\right]\right], \quad k = 0, 1, \cdots, N/8 - 1 \tag{33a}$$

$$X(N/2 - 2k - 1) = \left[\operatorname{Im}[Y(2k+1)] + \operatorname{Re}[Y(2k+1)] \bigg/ \left[2 \sin\left(\frac{2\pi}{N}(2k+1)\right)\right]\right], \quad k = 0, 1, \cdots, N/8 - 1. \tag{33b}$$

The reader is reminded that $X(k)$ is 0 for k even, and $X(k)$ is purely real because $x(n)$ is symmetric. An implementation of this procedure is given in Appendix IV.

Next we consider an N-point sequence which is antisymmetric in time and which is known to contain only odd harmonics. Denoting this sequence as $x(n)$, it can be shown that $x(n)$ satisfies the relations

$$x(N/2 - n) = x(n), \quad n = 0, 1, \cdots, N/4 - 1 \tag{34a}$$

$$x(0) = x(N/2) = 0 \tag{34b}$$

in addition to the antisymmetric sequence (4) and odd harmonic sequence (18), (29) relations. Thus, $x(n)$ need only be specified for $0 \leqslant n \leqslant N/4$ to uniquely define this sequence.

For maximum efficiency in obtaining the DFT of antisymmetric, odd harmonic sequences, we combine the procedure for antisymmetric sequences with the one for odd harmonic sequences. We first form the sequence $y(n)$ as

$$y(n) = x(2n) + [x(2n+1) - x(2n-1)], \quad n = 1, 2, \cdots, N/8 - 1 \tag{35a}$$

$$y(N/4 - n) = x(2n) - [x(2n+1) - x(2n-1)], \quad n = 1, 2, \cdots, N/8 - 1 \tag{35b}$$

with initial and final values

$$y(0) = 2x(1) \tag{35c}$$

$$y(N/8) = x(N/4). \tag{35d}$$

The sequence $y(n)$ contains the first $N/4$ points of an $N/2$-point odd harmonic sequence. Hence, we again use the procedure of Appendix III to give the $N/2$-point complex odd harmonic sequence $Y(k)$. The desired DFT $X(k)$ is obtained from $Y(k)$ via the relation

$$\operatorname{Im}[X(2k+1] = \operatorname{Im}[Y(2K+1)] - \operatorname{Re}[Y(2k+1)] \bigg/ \left[\left[2 \sin\left(\frac{2\pi}{N}(2k+1)\right)\right]\right], \quad k = 0, 1, \cdots, N/8 - 1 \tag{36a}$$

$$\operatorname{Im}[X(N/2 - 2k - 1)] = \left[-\operatorname{Im}[Y(2k+1)] - \operatorname{Re}[Y(2k+1)] \bigg/ \left[2 \sin\left(\frac{2\pi}{N}(2k+1)\right)\right]\right], \quad k = 0, 1, \cdots, N/8 - 1. \tag{36b}$$

The reader is reminded that $X(k)$ is 0 for k even, and $X(k)$ is purely imaginary because $x(n)$ is antisymmetric. An implementation of this procedure is given in Appendix V.

IV. Computation Time

The computation time for computing the transform of an arbitrary real N-point sequence is $\tau_1 = \alpha(N/2) \log_2 (N/2)$ where α is a proportionally constant. When one employs any of the symmetries and special properties discussed in this paper, the computation time becomes

$$\tau_2 = \alpha(N/4) \log_2 (N/4) + \beta N \tag{37}$$

where βN represents the time for preprocessing and postprocessing the sequences. For most practical cases, $\beta N \ll \alpha(N/4) \log_2 (N/4)$, and thus the savings in computation using the efficient algorithms is on the order of 2 to 1.

A test was run to measure the actual time required to run the subroutines of Appendixes I–V for a fixed size sequence. Using the real FFT subroutine FAST [6], [7], a 1024 FFT required 0.31 s. The subroutines FFTSYM and FFTASM required 0.19 s to run this same size transform, and the subroutine FFTOHM took 0.18 s for this case. The subroutines FFTSOH and FFTAOH took 0.12 s to run this same size transform. Thus, for a 1024-point transform, the overhead due to preprocessing and postprocessing (the βN term) is on the order of 20–30 percent of the time it takes to do the $N/2$-point FFT.

V. Summary

We have shown in this paper how one can exploit special properties of some sequences to reduce the computation time for obtaining its DFT. We have proposed a novel approach in frequency (the dual of the time algorithm) which can be used for sequences that are known to consist of only odd harmonics. The details of the computation have been described along with a set of subroutines which perform this basic computation.

Appendix I

The program given in this Appendix takes the real N-point symmetric sequence $x(n)$ and returns the $(N/2 + 1)$ real points of its DFT. On input only the first $(N/2 + 1)$ points are supplied to the routine; on output the first $(N/2 + 1)$ real values of $X(k)$, the DFT of $x(n)$, are returned.

The FFT subroutine used in this and subsequent routines in Appendixes II–V is the routine FAST described originally by Bergland [6], and modified by Bergland and Dolan [7]. For the direct transform [i.e., (1)], the input is an N-point real sequence, and the first $(N/2 + 1)$ complex values of its transform are stored in the original array with the real part of each DFT point preceding the imaginary part, i.e., Re $[X(0)]$ is followed by Im $[X(0)]$, which is followed by Re $[X(1)]$, which is followed by Im $[X(1)]$, etc. The last $(N/2 - 1)$ complex values of the transform are not returned, as they can be obtained via the DFT symmetry relations for real sequences, i.e.,

$$X(N - k) = X^*(k), \qquad k = 1, 2, \cdots.$$

For the inverse transform (the subroutine FSST), the input $X(k)$ is assumed to be in the format returned by FAST. The reader should note that for an N-point FFT, a total of $(N + 2)$ storage locations are required [7].

The calling sequence for the subroutine is

CALL FFTSYM (X, N, Y)

where

X = array of size $(N/2 + 1)$ which on input contains the first half of the real symmetric sequence $x(n)$, $n = 0, 1, \cdots, N/2$, and on output contains the real part of the DFT, $X(k)$, $k = 0, 1, \cdots, N/2$;

N = size of sequence $x(n)$;

Y = scratch array of size $(N/2 + 2)$.

```
C
C -----------------------------------------------------------------
C SUBROUTINE: FFTSYM
C COMPUTE DFT FOR REAL, SYMMETRIC, N-POINT SEQUENCE X(M) USING
C N/2-POINT FFT
C SYMMETRIC SEQUENCE MEANS X(M)=X(N-M), M=1,...,N/2-1
C NOTE: INDEX M IS SEQUENCE INDEX--NOT FORTRAN INDEX
C -----------------------------------------------------------------
C
      SUBROUTINE FFTSYM(X, N, Y)
      DIMENSION X(1), Y(1)
C
C X = REAL ARRAY WHICH ON INPUT CONTAINS THE N/2+1 POINTS OF THE
C     INPUT SEQUENCE (SYMMETRICAL)
C     ON OUTPUT X CONTAINS THE N/2+1 REAL POINTS OF THE TRANSFORM OF
C     THE INPUT--I.E. THE ZERO VALUED IMAGINARY PARTS ARE NOT RETURNED
C N = TRUE SIZE OF INPUT
C Y = SCRATCH ARRAY OF SIZE N/2+2
C
C
C FOR N = 2, COMPUTE DFT DIRECTLY
C
      IF (N.GT.2) GO TO 10
      T = X(1) + X(2)
      X(2) = X(1) - X(2)
      X(1) = T
      RETURN
   10 TWOPI = 8.*ATAN(1.0)
C
C FIRST COMPUTE B0 TERM, WHERE B0=SUM OF ODD VALUES OF X(M)
C
      NO2 = N/2
      NO4 = N/4
      NIND = NO2 + 1
      B0 = 0.
      DO 20 I=2,NIND,2
        B0 = B0 + X(I)
   20 CONTINUE
      B0 = B0*2.
C
C FOR N = 4 SKIP RECURSION LOOP
C
      IF (N.EQ.4) GO TO 40
C
C FORM NEW SEQUENCE, Y(M)=X(2*M)+(X(2*M+1)-X(2*M-1))
C
      DO 30 I=2,NO4
        IND = 2*I
        T1 = X(IND) - X(IND-2)
        Y(I) = X(IND-1) + T1
        IND1 = NO2 + 2 - I
        Y(IND1) = X(IND-1) - T1
   30 CONTINUE
   40 Y(1) = X(1)
      Y(NO4+1) = X(NO2+1)
C
C TAKE N/2 POINT (REAL) FFT OF Y
C
      CALL FAST(Y, NO2)
C
C FORM ORIGINAL DFT BY UNSCRAMBLING Y(K)
C  USE RECURSION TO GIVE SIN(TPN*I) MULTIPLIER
C
      TPN = TWOPI/FLOAT(N)
      COSI=2.*COS(TPN)
      SINI=2.*SIN(TPN)
      COSD=COSI/2.
      SIND=SINI/2.
      NIND = NO4 + 1
      DO 50 I=2,NIND
        IND = 2*I
        BK = Y(IND)/SINI
        AK = Y(IND-1)
        X(I) = AK + BK
        NIND1 = N/2 + 2 - I
        X(NIND1) = AK - BK
        TEMP=COSI*COSD-SINI*SIND
        SINI=COSI*SIND+SINI*COSD
        COSI=TEMP
   50 CONTINUE
      X(1) = B0 + Y(1)
      X(NO2+1) = Y(1) - B0
      RETURN
      END
```

Appendix II

The program given in this Appendix requires as input the first $(N/2)$ points of the antisymmetric, real N-point sequence $x(n)$, and returns the $(N/2 + 1)$ imaginary values of $X(k)$, the antisymmetric, imaginary N-point DFT of $x(n)$.

The calling sequence is

CALL FFTASM (X, N, Y)

where

X = array of size $(N/2 + 1)$ which on input contains the first half of the real, antisymmetric sequence $x(n)$, $N = 0, 1, \cdots, N/2 - 1$, and on output contains the imaginary parts of the DFT, $X(k)$, $k = 0, 1, \cdots, N/2$;

N = size of sequence, $x(n)$;

Y = scratch array of size $(N/2 + 2)$.

```
C
C -----------------------------------------------------------------
C SUBROUTINE: FFTASM
C COMPUTE DFT FOR REAL, ANTISYMMETRIC, N-POINT SEQUENCE X(M) USING
C N/2-POINT FFT
C ANTISYMMETRIC SEQUENCE MEANS X(M)=-X(N-M), M=1,...,N/2-1
C NOTE: INDEX M IS SEQUENCE INDEX--NOT FORTRAN INDEX
C -----------------------------------------------------------------
C
      SUBROUTINE FFTASM(X, N, Y)
      DIMENSION X(1), Y(1)
C
C X = REAL ARRAY WHICH ON INPUT CONTAINS THE N/2 POINTS OF THE
C     INPUT SEQUENCE (ASSYMMETRICAL)
C     ON OUTPUT X CONTAINS THE N/2+1 IMAGINARY POINTS OF THE TRANSFORM
C     OF THE INPUT--I.E. THE ZERO VALUED REAL PARTS ARE NOT RETURNED
C N = TRUE SIZE OF INPUT
C Y = SCRATCH ARRAY OF SIZE N/2+2
C
C
C FOR N = 2, ASSUME X(1)=0, X(2)=0, COMPUTE DFT DIRECTLY
C
      IF (N.EQ.2) GO TO 30
      TWOPI = 8.*ATAN(1.0)
C
C FORM NEW SEQUENCE, Y(M)=X(2*M)+(X(2*M+1)-X(2*M-1))
C
      NO2 = N/2
      NO4 = N/4
      DO 10 I=2,NO4
        IND = 2*I
        T1 = X(IND) - X(IND-2)
        Y(I) = X(IND-1) + T1
        IND1 = NO2 + 2 - I
        Y(IND1) = -X(IND-1) + T1
   10 CONTINUE
      Y(1) = 2.*X(2)
      Y(NO4+1) = -2.*X(NO2)
C
C TAKE N/2 POINT (REAL) FFT OF Y
C
      CALL FAST(Y, NO2)
```

```
C
C FORM ORIGINAL DFT BY UNSCRAMBLING Y(K)
C  USE RECURSION RELATION TO GENERATE SIN(TPN*I) MULTIPLIER
C
      TPN = TWOPI/FLOAT(N)
      COSI=2.*COS(TPN)
      SINI=2.*SIN(TPN)
      COSD=COSI/2.
      SIND=SINI/2.
      NIND = NO4 + 1
      DO 20 I=2,NIND
        IND = 2*I
        BK = Y(IND-1)/SINI
        AK = Y(IND)
        X(I) = AK - BK
        IND1 = NO2 + 2 - I
        X(IND1) = -AK - BK
        TEMP=COSI*COSD-SINI*SIND
        SINI=COSI*SIND+SINI*COSD
        COSI=TEMP
  20  CONTINUE
  30  X(1) = 0.
      X(NO2+1) = 0.
      RETURN
      END
```

Appendix III

The program given in this Appendix requires as input the first $(N/2)$ points of the real, odd harmonic N-point sequence $x(n)$, and returns the $(N/4)$ complex values of the odd harmonics of $X(k)$, the N-point DFT of $x(n)$. The output format is the real part of each DFT point, followed by its imaginary part.

The calling sequence is

CALL FFTOHM (X, N)

where

X = array of size $N/2$ which on input contains the first half of the odd harmonic sequence $x(n)$, $0 \leqslant n \leqslant N/2 - 1$, and on output contains the odd harmonics of the DFT; $X(k)$, $k = 1, 3, \cdots$ stored as $\mathrm{Re}[X(1)]$, $\mathrm{Im}[X(1)]$, $\mathrm{Re}[X(3)]$, $\mathrm{Im}[X(3)]$, $\cdots$;

N = size of sequence, $x(n)$.

```
C
C -----------------------------------------------------------------------
C SUBROUTINE: FFTOHM
C COMPUTE DFT FOR REAL, N-POINT, ODD HARMONIC SEQUENCES USING AN
C N/2 POINT FFT
C ODD HARMONIC MEANS X(2*K)=0, ALL K WHERE X(K) IS THE DFT OF X(M)
C NOTE: INDEX M IS SEQUENCE INDEX--NOT FORTRAN INDEX
C -----------------------------------------------------------------------
C
      SUBROUTINE FFTOHM(X, N)
      DIMENSION X(1)
C
C X = REAL ARRAY WHICH ON INPUT CONTAINS THE FIRST N/2 POINTS OF THE
C     INPUT
C     ON OUTPUT X CONTAINS THE N/4 COMPLEX VALUES OF THE ODD
C     HARMONICS OF THE INPUT--STORED IN THE SEQUENCE RE(X(1)),IM(X(1)),
C     RE(X(2)),IM(X(2)),...
C ****NOTE: X MUST BE DIMENSIONED TO SIZE N/2+2 FOR FFT ROUTINE
C N = TRUE SIZE OF X SEQUENCE
C
C FIRST COMPUTE REAL(X(1)) AND REAL(X(N/2-1)) SEPARATELY
C ALSO SIMULTANEOUSLY MULTIPLY ORIGINAL SEQUENCE BY SIN(TWOPI*(M-1)/N)
C SIN AND COS ARE COMPUTED RECURSIVELY
C
C
C FOR N = 2, ASSUME X(1)=X0, X(2)=-X0, COMPUTE DFT DIRECTLY
C
      IF (N.GT.2) GO TO 10
      X(1) = 2.*X(1)
      X(2) = 0.
      RETURN
  10  TWOPI = 8.*ATAN(1.0)
      TPN = TWOPI/FLOAT(N)
C
C COMPUTE X1=REAL(X(1)) AND X2=IMAGINARY(X(N/2-1))
C X(N) = X(N)*4.*SIN(TWOPI*(I-1)/N)
C
      T1 = 0.
C
C COSD AND SIND ARE MULTIPLIERS FOR RECURSION FOR SIN AND COS
C COSI AND SINI ARE INITIAL CONDITIONS FOR RECURSION FOR SIN AND COS
C
      COSD = COS(TPN*2.)
      SIND = SIN(TPN*2.)
      COSI = 1.
      SINI = 0.
      NO2 = N/2
      DO 20 I=1,NO2,2
        T = X(I)*COSI
        X(I) = X(I)*4.*SINI
        TEMP = COSI*COSD - SINI*SIND
        SINI = COSI*SIND + SINI*COSD
        COSI = TEMP
        T1 = T1 + T
  20  CONTINUE
C
C RESET INITIAL CONDITIONS (COSI,SINI) FOR NEW RECURSION
C
      COSI = COS(TPN)
      SINI = SIN(TPN)
      T2 = 0.
      DO 30 I=2,NO2,2
        T = X(I)*COSI
        X(I) = X(I)*4.*SINI
        TEMP = COSI*COSD - SINI*SIND
        SINI = COSI*SIND + SINI*COSD
        COSI = TEMP
        T2 = T2 + T
  30  CONTINUE
      X1 = 2.*(T1+T2)
      X2 = 2.*(T1-T2)
C
C TAKE N/2 POINT (REAL) FFT OF PREPROCESSED SEQUENCE X
C
      CALL FAST(X, NO2)
C
C FOR N = 4--SKIP RECURSION AND INITIAL CONDITIONS
C
      IF (N.EQ.4) GO TO 50
C
C INITIAL CONDITIONS FOR RECURSION
C
      X(2) = -X(1)/2.
      X(1) = X1
C
C FOR N = 8, SKIP RECURSION
C
      IF (N.EQ.8) GO TO 50
C
C UNSCRAMBLE Y(K) USING RECURSION FORMULA
C
      NIND = NO2 - 2
      DO 40 I=3,NIND,2
        T = X(I)
        X(I) = X(I-2) + X(I+1)
        X(I+1) = X(I-1) .- T
  40  CONTINUE
  50  X(NO2) = X(NO2+1)/2.
      X(NO2-1) = X2
      RETURN
      END
```

Appendix IV

The program given in this Appendix requires as input the first $(N/4)$ points of the real, symmetric, odd harmonic N-point sequence $x(n)$, and returns the $(N/4)$ real values of the odd harmonics of $X(k)$, the N-point DFT of $x(n)$. The routine calls the subroutine FFTOHM of Appendix III.

The calling sequence is

CALL FFTSOH (X, N, Y)

where

X = array of size $(N/4)$ which on input contains the first quarter of the real, symmetric, odd harmonic sequence $x(n)$, $n = 0, 1, \cdots, N/4 - 1$, and on output contains the real part of the odd harmonics of the DFT, $\mathrm{Re}[X(k)]$, $k = 1, 3, \cdots, N/2 - 1$;

N = size of sequence $x(n)$;

Y = scratch array of size $(N/4 + 2)$.

```
C
C-----------------------------------------------------------------------
C SUBROUTINE: FFTSOH
C COMPUTE DFT FOR REAL, SYMMETRIC, ODD HARMONIC, N-POINT SEQUENCE
C USING N/4-POINT FFT
C SYMMETRIC SEQUENCE MEANS X(M)=X(N-M), M=1,...,N/2-1
C ODD HARMONIC MEANS X(2*K)=0, ALL K, WHERE X(K) IS THE DFT OF X(M)
C X(M) HAS THE PROPERTY X(M)=-X(N/2-M), M=0,1,...,N/4-1,  X(N/4)=0
C NOTE: INDEX M IS SEQUENCE INDEX--NOT FORTRAN INDEX
C-----------------------------------------------------------------------
C
      SUBROUTINE FFTSOH(X,N,Y)
      DIMENSION X(1),Y(1)
C
C  X = REAL ARRAY WHICH ON INPUT CONTAINS THE N/4 POINTS OF THE
C      INPUT SEQUENCE (SYMMETRICAL)
C      ON OUTPUT X CONTAINS  THE N/4 REAL POINTS OF THE ODD HARMONICS
C      OF THE TRANSFORM OF THE INPUT--I.E. THE ZERO VALUED IMAGINARY
C      PARTS ARE NOT GIVEN NOR ARE THE ZERO-VALUED EVEN HARMONICS
C  N = TRUE SIZE OF INPUT
C  Y = SCRATCH ARRAY OF SIZE N/4+2
C
C
C  HANDLE N = 2 AND N = 4 CASES SEPARATELY
C
      IF(N.GT.4) GO TO 5
      IF(N.EQ.4) GO TO 4
C
C  FOR N=2, ASSUME X(1)=X0, X(2)=-X0, COMPUTE DFT DIRECTLY
C
      X(1)=2.*X(1)
      RETURN
C
C  N = 4 CASE, COMPUTE DFT DIRECTLY
C
 4    X(1)=2.*X(1)
      RETURN
 5    TWOPI=8.*ATAN(1.0)
```

```
C
C  FORM NEW SEQUENCE, Y(M)=X(2*M)+(X(2*M+1)-X(2*M-1))
C
      NO2=N/2
      NO4=N/4
      NO8=N/8
      IF(NO8.EQ.1) GO TO 25
      DO 20 I=2,NO8
      IND=2*I
      T1=X(IND)-X(IND-2)
      Y(I)=X(IND-1)+T1
      IND1=N/4+2-I
      Y(IND1)=-X(IND-1)+T1
 20   CONTINUE
 25   Y(1)=X(1)
      Y(NO8+1)=-2.*X(NO4)
C
C  THE SEQUENCE Y (N/4 POINTS) HAS ONLY ODD HARMONICS
C  CALL SUBROUTINE FFTOHM TO EXPLOIT ODD HARMONICS
C
      CALL FFTOHM(Y,NO2)
C
C  FORM ORIGINAL DFT FROM COMPLEX ODD HARMONICS OF Y(K)
C      BY UNSCRAMBLING Y(K)
C
      TPN=TWOPI/FLOAT(N)
      COSI=2.*COS(TPN)
      SINI=2.*SIN(TPN)
      COSD=COS(TPN*2.)
      SIND=SIN(TPN*2.)
      DO 40 I=1,NO8
      IND=2*I
      BK=Y(IND)/SINI
      TEMP=COSI*COSD-SINI*SIND
      SINI=COSI*SIND+SINI*COSD
      COSI=TEMP
      AK=Y(IND-1)
      X(I)=AK+BK
      IND1=N/4+1-I
 40   X(IND1)=AK-BK
      RETURN
      END
```

Appendix V

The program given in this Appendix requires as input the first $(N/4 + 1)$ points of the real, antisymmetric, odd harmonic N-point sequence $x(n)$ and returns the $(N/4)$ imaginary values of the odd harmonics of $X(k)$, the N-point DFT of $x(n)$. The routine calls the subroutine FFTOHM of Appendix III.

The calling sequence is

CALL FFTAOH (X, N, Y)

where

X = array of size $(N/4 + 1)$ which on input contains the first quarter of the real, antisymmetric, odd harmonic sequence $x(n)$, $n = 0, 1, \cdots, N/4 - 1$, and on output contains the imaginary parts of the odd harmonic of the DFT, $\mathrm{Im}\,[X(k)]$, $k = 1, 3, \cdots, N/2 - 1$;

N = size of sequence $x(n)$;

Y = scratch array of size $(N/4 + 2)$.

```
C
C-----------------------------------------------------------------------
C SUBROUTINE: FFTAOH
C COMPUTE DFT FOR REAL, ANTISYMMETRIC, ODD HARMONIC, N-POINT SEQUENCE
C USING N/4-POINT FFT
C ANTISYMMETRIC SEQUENCE MEANS X(M)=-X(N-M), M=1,...,N/2-1
C ODD HARMONIC MEANS X(2*K)=0, ALL K, WHERE X(K) IS THE DFT OF X(M)
C X(M) HAS THE PROPERTY X(M)=X(N/2-M), M=0,1,...,N/4-1,  X(0)=0
C NOTE: INDEX M IS SEQUENCE INDEX--NOT FORTRAN INDEX
C-----------------------------------------------------------------------
C
      SUBROUTINE FFTAOH(X,N,Y)
      DIMENSION X(1),Y(1)
C
C  X = REAL ARRAY WHICH ON INPUT CONTAINS THE (N/4+1) POINTS OF THE
C      INPUT SEQUENCE (ANTISYMMETRICAL)
C      ON OUTPUT X CONTAINS THE N/4 IMAGINARY POINTS OF THE ODD
C      HARMONICS OF THE TRANSFORM OF THE INPUT--I.E. THE ZERO
C      VALUED REAL PARTS ARE NOT GIVEN NOR ARE THE ZERO-VALUED
C      EVEN HARMONICS
C  N = TRUE SIZE OF INPUT
C  Y = SCRATCH ARRAY OF SIZE N/4+2
C
C
C  HANDLE N = 2 AND N = 4 CASES SEPARATELY
C
      IF(N.GT.4) GO TO 5
      IF(N.EQ.4) GO TO 4
C
C  FOR N=2, ASSUME X(1)=0, X(2)=0, COMPUTE DFT DIRECTLY
C
      X(1)=0.
      RETURN
C
C  N = 4 CASE, ASSUME X(1)=X(3)=0, X(2)=-X(4)=X0, COMPUTE DFT DIRECTLY
C
 4    X(1)=-2.*X(2)
      RETURN
 5    TWOPI=8.*ATAN(1.0)
C
C  FORM NEW SEQUENCE, Y(M)=X(2*M)+(X(2*M+1)-X(2*M-1))
C
      NO2=N/2
      NO4=N/4
      NO8=N/8
      IF(NO8.EQ.1) GO TO 25
      DO 20 I=2,NO8
      IND=2*I
      T1=X(IND)-X(IND-2)
      Y(I)=X(IND-1)+T1
      IND1=N/4+2-I
      Y(IND1)=X(IND-1)-T1
 20   CONTINUE
 25   Y(1)=2.*X(2)
      Y(NO8+1)=X(NO4+1)
C
C  THE SEQUENCE Y (N/4 POINTS) HAS ONLY ODD HARMONICS
C  CALL SUBROUTINE FFTOHM TO EXPLOIT ODD HARMONICS
C
      CALL FFTOHM(Y,NO2)
C
C  FORM ORIGINAL DFT FROM COMPLEX ODD HARMONICS OF Y(K)
C      BY UNSCRAMBLING Y(K)
C
      TPN=TWOPI/FLOAT(N)
      COSI=2.*COS(TPN)
      SINI=2.*SIN(TPN)
      COSD=COS(TPN*2.)
      SIND=SIN(TPN*2.)
      DO 40 I=1,NO8
      IND=2*I
      BK=Y(IND-1)/SINI
      TEMP=COSI*COSD-SINI*SIND
      SINI=COSI*SIND+SINI*COSD
      COSI=TEMP
      AK=Y(IND)
      X(I)=AK-BK
      IND1=N/4+1-I
 40   X(IND1)=-AK-BK
      RETURN
      END
```


Acknowledgment

The author acknowledges the impetus provided by Dr. J. Allen and Dr. M. Sondhi of Bell Laboratories who proposed the physical problem with the properties discussed in this paper.

References

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

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

[3] A. V. Oppenheim and R. W. Schafer, *Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1975.

[4] J. W. Cooley, P. A. Lewis, and P. D. Welch, "The fast Fourier transform algorithm: Programming considerations in the calculation of sine, cosine and Laplace transforms," *J. Sound Vib.*, vol. 12, pp. 315-337, July 1970.

[5] G. D. Bergland, "A radix-eight fast Fourier transform subroutine for real-valued series," *IEEE Trans. Audio Electroacoust.*, vol. AU-17, pp. 138-144, June 1969.

[6] G. D. Bergland, "Subroutines FAST and FSST, radix 4 routines for computing the DFT of real sequences," unpublished.

[7] G. D. Bergland and M. T. Dolan, "Fast Fourier transform algorithms," to be published in IEEE Press Book on Programs for Digital Signal Processing.