Fundamentals of the DFT (fft) Algorithms

D. Sundararajan

November 6, 2009

Contents

1 The PM DIF DFT Algorithm **2**

1.1 Half-wave symmetry of periodic waveforms 3

1.2 The DFT definition and the half-wave symmetry 5

1.3 The PM DIF DFT Algorithm 5

2 The PM DIT DFT Algorithm **10**

3 Summary **12**

Abstract

In this article, a physical explanation of the fundamentals of the DFT (fft) algorithms is presented in terms of waveform decomposition. After reading the article and trying the examples, the reader is expected to gain a clear understanding of the basics of the mysterious DFT (fft) algorithms.

In practice, the functions, whose Fourier representation is to be computed, are defined by a set of samples. In addition, the transform values over a finite range of frequencies are adequate. The discrete Fourier transform (DFT), with finite number of values in both time- and frequency-domains, ideally suits the purpose of approximating the other versions of Fourier representations [1]. However, the computational complexity of the order of N^2 of computing a N-point DFT, from its defined equation, does not give any advantages over alternate methods of analysis. For example, the computational complexity of the direct evaluation of the convolution operation is also of the order of N^2. Fortunately, the existence of DFT algorithms of computational complexity of the order of $N \log_2 N$ makes the DFT the heart and soul of many applications in science and engineering. Further, the DFT algorithms are regular and reduce the quantization noise. As the IDFT can be computed by a DFT algorithm with trivial modifications, deriving the DFT algorithm is sufficient.

The N-point DFT computation involves, for each of the N DFT coefficients, the summation of N products. The products are formed using the N given data values and a set of N twiddle factors. As the N data values remain the same in the computation of each of the N DFT coefficients and the N twiddle factors (made up of $N/4$ distinct values of cosine or sine function) are repeatedly used, there is a large amount of redundancy in computing the DFT from its defined equation. The redundant operations are reduced by computing partial sums and using them for the computation of more than one DFT coefficient.

In this article, two of the most useful of the family of PM DFT algorithms are presented. The presentation of the algorithms is based, for easier understanding, on waveform decomposition rather than on a mathematical derivation. Formal mathematical derivations of the algorithms and the details of implementation can be found in the book [2].

1 The PM DIF DFT Algorithm

Any periodic sequence $x(n)$ of period N is composed of N complex exponentials. That is,

$$x(n) = X(0)e^{j0\frac{2\pi}{N}n} + X(1)e^{j1\frac{2\pi}{N}n} + X(2)e^{j2\frac{2\pi}{8}n} + \cdots + X(N-1)e^{j(N-1)\frac{2\pi}{N}n}$$

The task is to separate the frequency components, that is to find the values $X(k),\ k = 0, 1, \ldots, N-1$. A finite sequence of values is assumed to be periodic in DFT computation. Assume that period N is an integral power of 2. In practice, this constraint is not so severe as most of the signals to be analyzed are aperiodic and zero-padding can be used to extend their length so that the number of samples is equal to 2^M for some integer M. A real sinusoid, at a given frequency, is characterized by its amplitude and phase. The mathematically equivalent complex exponential is characterized, at a given frequency, by its single complex amplitude. Although, most practical signals are real, for obtaining the highest efficiency as well as ease of use, it is a necessity to formulate the DFT algorithms using complex exponentials as basis functions, rather than real sinusoids.

1.1 Half-wave symmetry of periodic waveforms

The easiest way to understand the basics of DFT algorithms is through the half-wave symmetry of periodic waveforms. Any periodic sequence $x(n)$ of period N can be decomposed into its even and odd half-wave symmetric components $x_{eh}(n)$ and $x_{oh}(n)$, respectively. That is $x(n) = x_{eh}(n) + x_{oh}(n)$, where

$$x_{eh}(n) = \frac{1}{2}\left(x(n) + x\left(n \pm \frac{N}{2}\right)\right) \quad \text{and} \quad x_{oh}(n) = \frac{1}{2}\left(x(n) - x\left(n \pm \frac{N}{2}\right)\right)$$

The sequence values of the even half-wave symmetric waveform $x_{eh}(n)$ over any half period are the same over the preceding or succeeding half period

$$x_{eh}\left(n \pm \frac{N}{2}\right) = x_{eh}(n)$$

That is, the fundamental period of $x_{eh}(n)$ is $N/2$. Therefore, the waveform remains the same by a shift of $N/2$ sample intervals. The sequence values of the odd half-wave symmetric waveform $x_{oh}(n)$ over any half period are the negatives of those over the preceding or succeeding half period

$$x_{oh}\left(n \pm \frac{N}{2}\right) = -x_{oh}(n)$$

The waveform remains the same by a shift of $N/2$ sample intervals, except that it is inverted about the horizontal axis. Therefore, $N/2$ values of each of $x_{eh}(n)$ and $x_{oh}(n)$ are adequate to uniquely represent them and, thereby, representing the N values of one period of $x(n)$. Figures 1(a), (b), and (c) show, respectively, the periodic waveform, with period 8,

$$x(n) = \frac{1}{2} + \cos(\frac{\pi}{4}n + \frac{\pi}{8}) + \cos(2\frac{\pi}{4}n + \frac{\pi}{6}) + \cos(3\frac{\pi}{4}n + \frac{\pi}{3}),$$

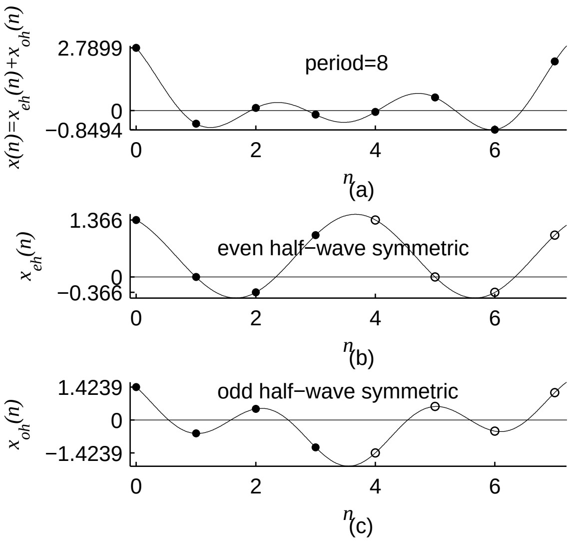

Figure 1: (a) $x(n) = x_{eh}(n) + x_{oh}(n)$; (b) The even half-wave symmetric component of $x(n)$, $x_{eh}(n)$; (c) The odd half-wave symmetric component of $x(n)$, $x_{oh}(n)$

its even half-wave symmetric component (consisting of even frequency components of $x(n)$ with frequency indices 0 and 2)

$$x_{eh}(n) = \frac{1}{2} + \cos(2\frac{\pi}{4}n + \frac{\pi}{6}),$$

and its odd half-wave symmetric component (consisting of odd frequency components of $x(n)$ with frequency indices 1 and 3)

$$x_{oh}(n) = \cos(\frac{\pi}{4}n + \frac{\pi}{8}) + \cos(3\frac{\pi}{4}n + \frac{\pi}{3})$$

In Figure 1(b),

$$\{x_{eh}(4) = x_{eh}(0), x_{eh}(5) = x_{eh}(1), x_{eh}(6) = x_{eh}(2), x_{eh}(7) = x_{eh}(3)\},$$

which implies that samples $\{x_{eh}(4), x_{eh}(5), x_{eh}(6), x_{eh}(7)\}$, shown by unfilled circles, are redundant. In Figure 1(c),

$$\{x_{oh}(4) = -x_{oh}(0), x_{oh}(5) = -x_{oh}(1), x_{oh}(6) = -x_{oh}(2), x_{oh}(7) = -x_{oh}(3)\},$$

which implies that samples $\{x_{oh}(4), x_{oh}(5), x_{oh}(6), x_{oh}(7)\}$, shown by unfilled circles, are redundant.

1.2 The DFT definition and the half-wave symmetry

Sequence $x_{eh}(n)$ contributes the even frequency components of $x(n)$ and $x_{oh}(n)$ contributes the odd frequency components, as, from the DFT definition,

$$\begin{aligned} X(k) &= \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi}{N}kn}, \quad k = 0, 1, \ldots, N-1 \\ &= \begin{cases} \displaystyle\sum_{n=0}^{(N/2)-1} \left(x(n) + x\left(n \pm \frac{N}{2}\right)\right) e^{-j\frac{2\pi}{N}kn} = \sum_{n=0}^{(N/2)-1} 2x_{eh}(n)e^{-j\frac{2\pi}{N}kn}, & k \text{ even} \\ \displaystyle\sum_{n=0}^{(N/2)-1} \left(x(n) - x\left(n \pm \frac{N}{2}\right)\right) e^{-j\frac{2\pi}{N}kn} = \sum_{n=0}^{(N/2)-1} 2x_{oh}(n)e^{-j\frac{2\pi}{N}kn}, & k \text{ odd} \end{cases} \end{aligned}$$

Both the even-indexed exponentials $e^{-jk\frac{2\pi}{N}n}$, $k = 0, 2, \ldots, N-2$, appearing in the defining equation of a N-point DFT, and the even-indexed frequency components that constitute $(x(n) + x(n + N/2))$ are even half-wave symmetric. Both the odd-indexed exponentials $e^{-jk\frac{2\pi}{N}n}$, $k = 1, 3, \ldots, N-1$ and the odd-indexed frequency components that constitute $(x(n) - x(n + N/2))$ are odd half-wave symmetric.

It is by the repeated decomposition of a waveform into its even and odd half-wave symmetric components, along with frequency shifting, and using the temporal redundancy (in time) of these components, we extract the frequency coefficients of its constituent sinusoids. The decomposition operation involves taking two values, finding their sum and difference, and storing the resulting two values. Therefore, as the sum and difference of a and b is $a \pm b$ and the plus-minus operation is the basic to the algorithms, the DFT algorithms are called PM DFT algorithms. Further, as two values are input to the basic operation resulting in two values, it is found that the most efficient data structure for the algorithms is an array of two element vectors.

1.3 The PM DIF DFT Algorithm

Given a waveform $x(n)$ composed of N frequency components (let us assume $N = 8$),

$$\begin{aligned} x(n) &= X(0)e^{j0\frac{2\pi}{8}n} + X(1)e^{j1\frac{2\pi}{8}n} + X(2)e^{j2\frac{2\pi}{8}n} + X(3)e^{j3\frac{2\pi}{8}n} \\ &\quad + X(4)e^{j4\frac{2\pi}{8}n} + X(5)e^{j5\frac{2\pi}{8}n} + X(6)e^{j6\frac{2\pi}{8}n} + X(7)e^{j7\frac{2\pi}{8}n}, \; n = 0, 1, \ldots, 7 \end{aligned}$$

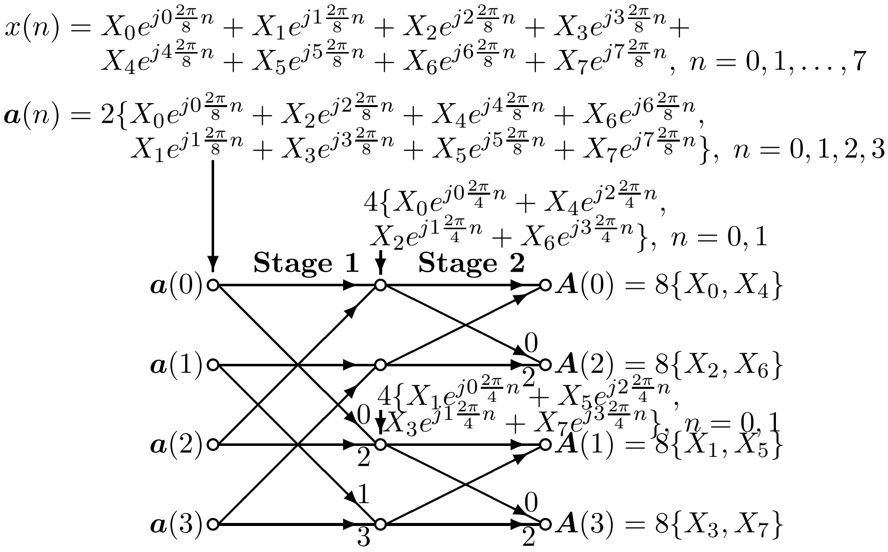

Figure 2: The signal-flow graph of the PM DIF DFT algorithm, with $N = 8$, showing the decomposition of a waveform.

the first step is to decompose $x(n)$ into its even and odd half-wave symmetric components $x_{eh}(n)$ and $x_{oh}(n)$, respectively. The decomposition results in

$$\begin{aligned}
\boldsymbol{a}(n) &= \{a_0(n), a_1(n)\} = 2\{x_{eh}(n), x_{oh}(n)\} \\
&= 2\{X(0)e^{j0\frac{2\pi}{8}n} + X(2)e^{j2\frac{2\pi}{8}n} + X(4)e^{j4\frac{2\pi}{8}n} + X(6)e^{j6\frac{2\pi}{8}n}, \\
&\qquad X(1)e^{j1\frac{2\pi}{8}n} + X(3)e^{j3\frac{2\pi}{8}n} + X(5)e^{j5\frac{2\pi}{8}n} + X(7)e^{j7\frac{2\pi}{8}n}\}, \ n = 0, 1, 2, 3
\end{aligned}$$

The division operation by two, required in finding the symmetric components, is not carried out and hence the factor two appears in the result. Since a half-wave symmetric component is symmetric about the midpoint of any period, the values over half the period is sufficient for further processing. Therefore, the components are found only for $n = 0, 1, 2,$ and 3. We reformulate the problem of separating the frequency components of an arbitrary $x(n)$ into that of its even and odd half-wave symmetric components $x_{eh}(n)$ and $x_{oh}(n)$. We form the data structure $\boldsymbol{a}(n) = \{a_0(n), a_1(n)\} = 2\{x_{eh}(n), x_{oh}(n)\}, \ n = 0, 1, \ldots (N/2) - 1$, an array of two element vectors. For $N = 8$, we get an array of length four, with each element of the array being a pair of ordered complex numbers. This array is stored in the nodes at the beginning of the signal-flow graph of the algorithm shown in Figure 2.

A signal-flow graph is an interconnection of nodes and branches. The direction of signal flow along a branch is indicated by an arrow. A node, shown by unfilled circles, stores two values. In addition, except the first set of nodes at the beginning of the graph, each node finds the sum and difference of the two values supplied by the two incoming branches. An upper node is a node where a branch with a positive slope terminates. This type of nodes receive the first element of the vectors of the nodes from which their incoming branches originate. A lower node is a node where a branch with a negative slope terminates. This type of nodes receive the second element of the vectors of the nodes from which their incoming branches originate. A value passing along a branch is multiplied by $e^{-j\frac{2\pi}{8}n}$, where n is an integer that appears near the arrow of the branch. No integer near an arrow implies that the branch simply passes the value to its connecting node.

The even half-wave symmetric component can be expressed as

$$\begin{aligned}
& 2(X(0)e^{j0\frac{2\pi}{8}n} + X(2)e^{j2\frac{2\pi}{8}n} + X(4)e^{j4\frac{2\pi}{8}n} + X(6)e^{j6\frac{2\pi}{8}n}) \\
= \; & 2(X(0)e^{j0\frac{2\pi}{4}n} + X(2)e^{j1\frac{2\pi}{4}n} + X(4)e^{j2\frac{2\pi}{4}n} + X(6)e^{j3\frac{2\pi}{4}n}), \; n = 0,1,2,3
\end{aligned}$$

This is a waveform composed of $\frac{N}{2} = 4$ frequency components with frequency coefficients $2\{X(0), X(2), X(4), X(6)\}$.

The odd half-wave symmetric component is multiplied by the exponential $e^{-j1\frac{2\pi}{8}n}$ to get

$$\begin{aligned}
& 2(X(1)e^{j1\frac{2\pi}{8}n} + X(3)e^{j3\frac{2\pi}{8}n} + X(5)e^{j5\frac{2\pi}{8}n} + X(7)e^{j7\frac{2\pi}{8}n})e^{-j1\frac{2\pi}{8}n} \\
= \; & 2(X(1)e^{j0\frac{2\pi}{8}n} + X(3)e^{j2\frac{2\pi}{8}n} + X(5)e^{j4\frac{2\pi}{8}n} + X(7)e^{j6\frac{2\pi}{8}n}) \\
= \; & 2(X(1)e^{j0\frac{2\pi}{4}n} + X(3)e^{j1\frac{2\pi}{4}n} + X(5)e^{j2\frac{2\pi}{4}n} + X(7)e^{j3\frac{2\pi}{4}n}), \; n = 0,1,2,3
\end{aligned}$$

This is a waveform composed of $\frac{N}{2} = 4$ frequency components with frequency coefficients $2\{X(1), X(3), X(5), X(7)\}$. Note that multiplication by $e^{-j1\frac{2\pi}{8}n}$ is the frequency shifting operation, which shifts the spectrum but the coefficients of the spectral components $X(k)$ are not changed. The multiplication operation is indicated by the numbers (the value of n in $e^{-j1\frac{2\pi}{8}n}$) close to the first set of arrows near the two bottom nodes in the signal-flow graph.

We have reduced the problem of decomposing a waveform composed of N frequency components into two problems, each of decomposing a waveform composed of $\frac{N}{2}$ frequency components. Now, we repeat the same process to these two waveforms. Decomposing the two waveforms into their even and odd half-wave symmetric components, we get

$$4\{X_0e^{j0\frac{2\pi}{4}n} + X(4)e^{j2\frac{2\pi}{4}n}, X(2)e^{j1\frac{2\pi}{4}n} + X(6)e^{j3\frac{2\pi}{4}n}\}, \; n = 0,1 \tag{1}$$

$$4\{X(1)e^{j0\frac{2\pi}{4}n} + X(5)e^{j2\frac{2\pi}{4}n}, X(3)e^{j1\frac{2\pi}{4}n} + X(7)e^{j3\frac{2\pi}{4}n}\}, \; n = 0,1 \tag{2}$$

These vector arrays are stored in the nodes at the middle of the signal-flow graph of the algorithm shown in Figure 2. The nodes, except the first set, carries out the add-subtract operation, in addition to providing storage.

The even half-wave symmetric component of the waveform given by Equation 1 can be expressed as

$$\begin{aligned} & 4(X_0 e^{j0\frac{2\pi}{4}n} + X(4)e^{j2\frac{2\pi}{4}n}) \\ = \; & 4(X_0 e^{j0\frac{2\pi}{2}n} + X(4)e^{j1\frac{2\pi}{2}n}), \; n = 0,1 \end{aligned}$$

This is a waveform composed of $\frac{N}{4} = 2$ frequency components with frequency coefficients $4\{X_0, X(4)\}$. The coefficients $\boldsymbol{A}(0) = \{A_0(0), A_1(0)\} = 8\{X_0, X(4)\}$ are obtained by simply adding and subtracting the two sample values. These coefficients are stored in the top node at the end of the signal-flow graph shown in Figure 2.

The odd half-wave symmetric component of the waveform given by Equation 1 is multiplied by the exponential $e^{-j1\frac{2\pi}{8}(2n)} = e^{-j1\frac{2\pi}{4}n}$ to get

$$\begin{aligned} & 4(X(2)e^{j1\frac{2\pi}{4}n} + X(6)e^{j3\frac{2\pi}{4}n})e^{-j1\frac{2\pi}{4}n} \\ = \; & 4(X(2)e^{j0\frac{2\pi}{4}n} + X(6)e^{j2\frac{2\pi}{4}n}) \\ = \; & 4(X(2)e^{j0\frac{2\pi}{2}n} + X(6)e^{j1\frac{2\pi}{2}n}), \; n = 0,1 \end{aligned}$$

This is a waveform composed of $\frac{N}{4} = 2$ frequency components with frequency coefficients $4\{X(2), X(6)\}$. The coefficients $\boldsymbol{A}(2) = \{A_0(2), A_1(6)\} = 8\{X(2), X(6)\}$ are obtained by simply adding and subtracting the two sample values. These coefficients are stored in the second node from top at the end of the signal-flow graph shown in Figure 2.

The even half-wave symmetric component of the waveform defined by Equation 2 can be expressed as

$$\begin{aligned} & 4(X(1)e^{j0\frac{2\pi}{4}n} + X(5)e^{j2\frac{2\pi}{4}n}) \\ = \; & 4(X(1)e^{j0\frac{2\pi}{2}n} + X(5)e^{j1\frac{2\pi}{2}n}), \; n = 0,1 \end{aligned}$$

This is a waveform composed of $\frac{N}{4} = 2$ frequency components with frequency coefficients $4\{X(1), X(5)\}$. The coefficients $\boldsymbol{A}(1) = \{A_0(1), A_1(5)\} = 8\{X(1), X(5)\}$ are obtained by simply adding and subtracting the two sample values. These coefficients are stored in the third node from top at the end of the signal-flow graph shown in Figure 2.

The odd half-wave symmetric component of the waveform defined by Equation 2 is multiplied by the exponential $e^{-j1\frac{2\pi}{4}n}$ to get

$$\begin{aligned} & 4(X(3)e^{j1\frac{2\pi}{4}n} + X(7)e^{j3\frac{2\pi}{4}n})e^{-j1\frac{2\pi}{4}(n)} \\ = \; & 4(X(3)e^{j0\frac{2\pi}{4}n} + X(7)e^{j2\frac{2\pi}{4}n}) \\ = \; & 4(X(3)e^{j0\frac{2\pi}{2}n} + X(7)e^{j1\frac{2\pi}{2}n}), \; n = 0,1 \end{aligned}$$

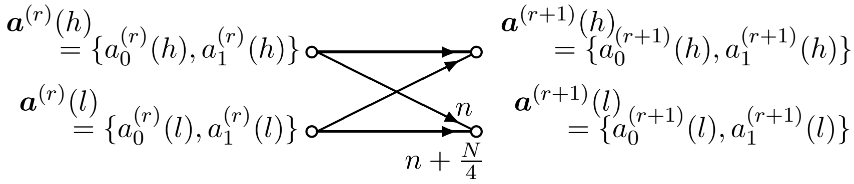

Figure 3: The signal-flow graph of the butterfly of the PM DIF DFT algorithm, where $0 \leq n < \frac{N}{4}$. A twiddle factor $W_N^n = e^{-j\frac{2\pi}{N}n}$ is indicated only by its variable part of the exponent, n

This is a waveform composed of $\frac{N}{4} = 2$ frequency components with frequency coefficients $4\{X(3), X(7)\}$. The coefficients $\boldsymbol{A}(3) = \{A_0(3), A_1(7)\} = 8\{X(3), X(7)\}$ are obtained by simply adding and subtracting the two sample values. These coefficients are stored in the fourth node from top at the end of the signal-flow graph shown in Figure 2.

The output vectors $\{\boldsymbol{A}(0), \boldsymbol{A}(1), \boldsymbol{A}(2), \boldsymbol{A}(3)\}$ appear in bit-reversed order. The binary number representation of the frequency indices $\{0, 1, 2, 3\}$ is $\{00, 01, 10, 11\}$. By reversing the order of bits, we get the bit-reversed order $\{00, 10, 01, 11\}$ in binary form and $\{0, 2, 1, 3\}$ in decimal form. The bit-reversed order occurs at the output because of the repeated splitting of the frequency components into groups consisting of odd- and even-indexed frequency indices over the stages of the algorithm.

Each stage of the algorithm requires N complex additions and $N/2$ complex multiplications, where N is the sequence length. There are $(\log_2 N) - 1$ stages. In addition, the initial vector formation requires N complex additions. Therefore, the algorithm reduces the computational complexity to $N \log_2 N$ compared from that of N^2 required for the direct computation of the N-point DFT.

The algorithm is so regular that one can easily get the signal-flow graph for any value of N that is an integral power of two. The algorithm is basically an interconnection of butterflies (a computational structure), shown in Figure 3. The defining equations of a butterfly at the rth stage are given by

$$\begin{aligned}
a_0^{(r+1)}(h) &= a_0^{(r)}(h) + a_0^{(r)}(l) \\
a_1^{(r+1)}(h) &= a_0^{(r)}(h) - a_0^{(r)}(l) \\
a_0^{(r+1)}(l) &= W_N^n a_1^{(r)}(h) + W_N^{n+\frac{N}{4}} a_1^{(r)}(l) \\
a_1^{(r+1)}(l) &= W_N^n a_1^{(r)}(h) - W_N^{n+\frac{N}{4}} a_1^{(r)}(l),
\end{aligned}$$

There are $(\log_2 N) - 1$ stages, each with $N/4$ butterflies. With $N = 8$, therefore, we see four butterflies in Figure 2.

The extraction of the coefficient $e^{j\frac{\pi}{3}} = \frac{1}{2} + j\frac{\sqrt{3}}{2}$, multiplied by 8 $(= 4 + j4\sqrt{3})$, of the waveform $x(n) = e^{j(2\frac{2\pi}{8}n+\frac{\pi}{3})}$, is shown in Figure 4. Figure 5 shows the values at various stages of the algorithm.

The extraction of the coefficient $3e^{j\frac{\pi}{6}} = \frac{3\sqrt{3}}{2}+j\frac{3}{2}$, multiplied by 8 $(= 12\sqrt{3}+j12)$, of the waveform $x(n) = 3e^{j(7\frac{2\pi}{8}n+\frac{\pi}{6})}$, is shown in Figure 6. Figure 7 shows the values at various stages of the algorithm.

2 The PM DIT DFT Algorithm

We have given the physical explanation of the decomposition of waveforms in the DIF DFT algorithm. In a decimation-in-frequency (DIF) algorithm, the transform sequence, $X(k)$, is successively divided into smaller subsequences. For example, in the beginning of the first stage, the computation of an N-point DFT is decomposed into two problems: (i) computing the $(N/2)$ even-indexed $X(k)$ and (ii) computing the $(N/2)$ odd-indexed $X(k)$. In a decimation-in-time (DIT) algorithm, the data sequence, $x(n)$, is successively divided into smaller subsequences. For example, in the beginning of the last stage, the computation of an N-point DFT is decomposed into two problems: (i) computing the $(N/2)$-point DFT of even-indexed $x(n)$ and (ii) computing the $(N/2)$-point DFT of odd-indexed $x(n)$. The DIT DFT algorithm is based on zero-padding, time-shifting, and spectral redundancy. For understanding, the DIF DFT algorithms are easier. However, the DIT algorithms are used more often, as taking care of the data scrambling problem occurring at the beginning of the algorithm is relatively easier. The DIT DFT algorithms can be considered as the algorithms obtained by transposing the signal-flow graph of the corresponding DIF algorithms, that is by reversing the direction of (signal flow) all the arrows and interchanging the input and the output.

The defining equations of a butterfly, shown in Figure 8, of the PM DIT DFT algorithm at the rth stage are given by

$$\begin{aligned}
A_0^{(r+1)}(h) &= A_0^{(r)}(h) + W_N^n A_0^{(r)}(l) \\
A_1^{(r+1)}(h) &= A_0^{(r)}(h) - W_N^n A_0^{(r)}(l) \\
A_0^{(r+1)}(l) &= A_1^{(r)}(h) + W_N^{n+\frac{N}{4}} A_1^{(r)}(l) \\
A_1^{(r+1)}(l) &= A_1^{(r)}(h) - W_N^{n+\frac{N}{4}} A_1^{(r)}(l),
\end{aligned}$$

The signal-flow graph of the PM DIT DFT algorithm is shown in Figure 9 with $N = 8$.

Figure 10 shows the values at various stages of the algorithm for the same input values given in Figure 5.

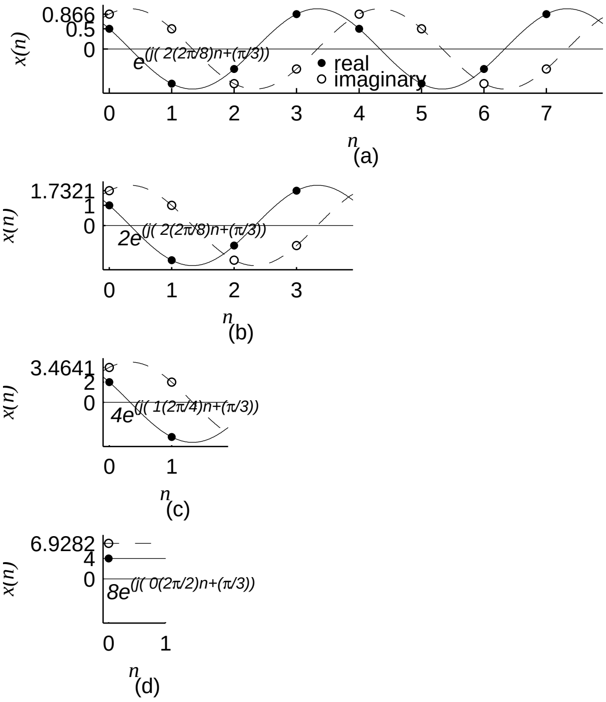

Figure 4: Extracting the coefficient $e^{j\frac{\pi}{3}}$, multiplied by 8, of $x(n) = e^{j(2\frac{2\pi}{8}n+\frac{\pi}{3})}$: (a) Eight samples of the waveform, $x(n) = e^{j(2\frac{2\pi}{8}n+\frac{\pi}{3})}$; (b) Four samples of the waveform, $x(n) = 2e^{j(2\frac{2\pi}{8}n+\frac{\pi}{3})}$, stored in the first element of the vectors at the nodes in the beginning of the signal-flow graph in Figure 2; (c) Two samples of the waveform, $x(n) = 4e^{j(1\frac{2\pi}{4}n+\frac{\pi}{3})}$, stored in the second element of the vectors at the upper two nodes in the middle of the signal-flow graph in Figure 2; (d) One sample of the waveform, $x(n) = 8e^{j(0\frac{2\pi}{2}n+\frac{\pi}{3})}$, stored in the first element of the vector at the second node from the top at the end of the signal-flow graph in Figure 2

Input values	Vector formation	Stage 1 output	Stage 2 output
$x(0)=\frac{1}{2}+j\frac{\sqrt{3}}{2}$ $x(4)=\frac{1}{2}+j\frac{\sqrt{3}}{2}$	$a_0(0)=1+j\sqrt{3}$ $a_1(0)=0$	0 $2+j2\sqrt{3}$	$X(0)=A_0(0)=0$ $X(4)=A_1(0)=0$
$x(1)=-\frac{\sqrt{3}}{2}+j\frac{1}{2}$ $x(5)=-\frac{\sqrt{3}}{2}+j\frac{1}{2}$	$a_0(1)=-\sqrt{3}+j1$ $a_1(1)=0$	0 $-2\sqrt{3}+j2$	$X(2)=A_0(2)=4+j4\sqrt{3}$ $X(6)=A_1(2)=0$
$x(2)=-\frac{1}{2}-j\frac{\sqrt{3}}{2}$ $x(6)=-\frac{1}{2}-j\frac{\sqrt{3}}{2}$	$a_0(2)=-1-j\sqrt{3}$ $a_1(2)=0$	0 0	$X(1)=A_0(1)=0$ $X(5)=A_1(1)=0$
$x(3)=\frac{\sqrt{3}}{2}-j\frac{1}{2}$ $x(7)=\frac{\sqrt{3}}{2}-j\frac{1}{2}$	$a_0(3)=\sqrt{3}-j1$ $a_1(3)=0$	0 0	$X(3)=A_0(3)=0$ $X(7)=A_1(3)=0$

Figure 5: The trace of the PM DIF DFT algorithm, with $N = 8$, in extracting the coefficient $e^{j\frac{\pi}{3}}$, multiplied by 8, of $x(n) = e^{j(2\frac{2\pi}{8}n+\frac{\pi}{3})}$.

3 Summary

- DFT algorithms are based on decomposing a given waveform, repeatedly, into its even- and odd half-wave symmetric components.
- The constraint that the sequence length must be a power-of-two is acceptable in almost all applications of the DFT.
- The DFT algorithms using complex exponentials are the most efficient, since the most efficient way, for computational purposes, of representing a real sinusoid is by a complex exponential.
- As the fundamental operation in DFT algorithms is computing the sum and difference of two values, a two-element complex vector is the most suitable data structure for DFT computation.

References

1. Sundararajan, D. (2008) *A Practical Approach to Signals and Systems*, Wiley, Singapore.
2. Sundararajan, D. (2001) *Discrete Fourier Transform, Theory, Algorithms, and Applications*, World Scientific, Singapore.

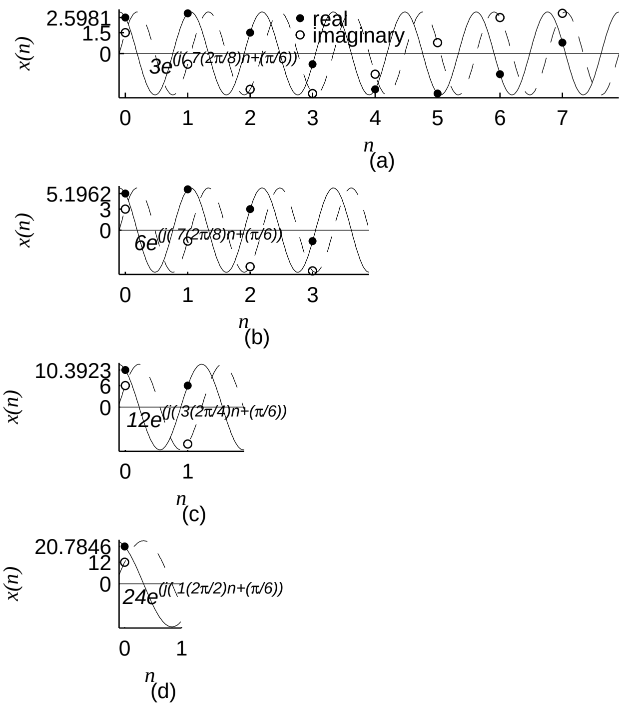

Figure 6: Extracting the coefficient $3e^{j\frac{\pi}{6}}$, multiplied by 8, of $x(n) = 3e^{j(7\frac{2\pi}{8}n+\frac{\pi}{6})}$: (a) Eight samples of the waveform, $x(n) = 3e^{j(7\frac{2\pi}{8}n+\frac{\pi}{6})}$; (b) Four samples of the waveform, $x(n) = 6e^{j(7\frac{2\pi}{8}n+\frac{\pi}{6})}$, stored in the second element of the vectors at the nodes in the beginning of the signal-flow graph in Figure 2; (c) Two samples of the waveform, $x(n) = 12e^{j(3\frac{2\pi}{4}n+\frac{\pi}{6})}$, stored in the second element of the vectors at the lower two nodes in the middle of the signal-flow graph in Figure 2; (d) One sample of the waveform, $x(n) = 24e^{j(1\frac{2\pi}{2}n+\frac{\pi}{6})}$, stored in the second element of the vector at the fourth node in the end of the signal-flow graph in Figure 2

Input values	Vector formation	Stage 1 output	Stage 2 output
$x(0)=\frac{3\sqrt{3}}{2}+j\frac{3}{2}$ $x(4)=-\frac{3\sqrt{3}}{2}-j\frac{3}{2}$	$a_0(0)=0$ $a_1(0)=3\sqrt{3}+j3$	0 0	$X(0)=A_0(0)=0$ $X(4)=A_1(0)=0$
$x(1)=\frac{3(\sqrt{6}+\sqrt{2})}{4}-j\frac{3(\sqrt{6}-\sqrt{2})}{4}$ $x(5)=-\frac{3(\sqrt{6}+\sqrt{2})}{4}+j\frac{3(\sqrt{6}-\sqrt{2})}{4}$	$a_0(1)=0$ $a_1(1)=\frac{3(\sqrt{6}+\sqrt{2})}{2}-j\frac{3(\sqrt{6}-\sqrt{2})}{2}$	0 0	$X(2)=A_0(2)=0$ $X(6)=A_1(2)=0$
$x(2)=\frac{3}{2}-j\frac{3\sqrt{3}}{2}$ $x(6)=-\frac{3}{2}+j\frac{3\sqrt{3}}{2}$	$a_0(2)=0$ $a_1(2)=3-j3\sqrt{3}$	0 $6\sqrt{3}+j6$	$X(1)=A_0(1)=0$ $X(5)=A_1(1)=0$
$x(3)=-\frac{3(\sqrt{6}-\sqrt{2})}{4}-j\frac{3(\sqrt{6}+\sqrt{2})}{4}$ $x(7)=\frac{3(\sqrt{6}-\sqrt{2})}{4}+j\frac{3(\sqrt{6}+\sqrt{2})}{4}$	$a_0(3)=0$ $a_1(3)=-\frac{3(\sqrt{6}-\sqrt{2})}{2}-j\frac{3(\sqrt{6}+\sqrt{2})}{2}$	0 $6-j6\sqrt{3}$	$X(3)=A_0(3)=0$ $X(7)=A_1(3)=12\sqrt{3}+j12$

Figure 7: The trace of the PM DIF DFT algorithm, with $N=8$, in extracting the coefficient $3e^{j\frac{\pi}{6}}$, multiplied by 8, of $x(n)=3e^{j(7\frac{2\pi}{8}n+\frac{\pi}{6})}$.

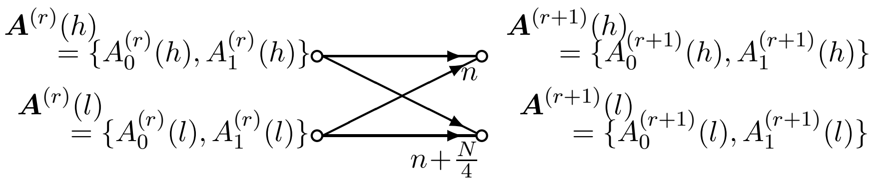

Figure 8: The signal-flow graph of the butterfly of the PM DIT DFT algorithm, where $0 \le n < \frac{N}{4}$. A twiddle factor $W_N^n = e^{-j\frac{2\pi}{N}n}$ is indicated only by its variable part of the exponent, n

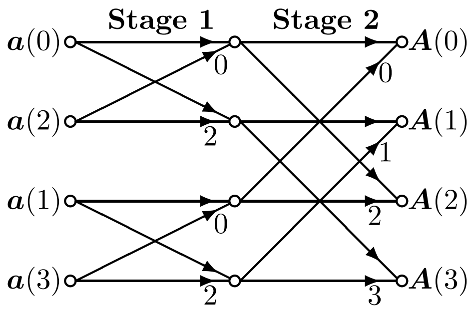

Figure 9: The signal-flow graph of the PM DIT DFT algorithm with $N = 8$.

Input values	Vector formation and swapping	Stage 1 output	Stage 2 output
$x(0)=\frac{1}{2}+j\frac{\sqrt{3}}{2}$ $x(4)=\frac{1}{2}+j\frac{\sqrt{3}}{2}$	$a_0(0)=1+j\sqrt{3}$ $a_1(0)=0$	0 $2+j2\sqrt{3}$	$X(0)=A_0(0)=0$ $X(4)=A_1(0)=0$
$x(1)=-\frac{\sqrt{3}}{2}+j\frac{1}{2}$ $x(5)=-\frac{\sqrt{3}}{2}+j\frac{1}{2}$	$a_0(2)=-1-j\sqrt{3}$ $a_1(2)=0$	0 0	$X(1)=A_0(1)=0$ $X(5)=A_1(1)=0$
$x(2)=-\frac{1}{2}-j\frac{\sqrt{3}}{2}$ $x(6)=-\frac{1}{2}-j\frac{\sqrt{3}}{2}$	$a_0(1)=-\sqrt{3}+j1$ $a_1(1)=0$	0 $-2\sqrt{3}+j2$	$X(2)=A_0(2)=4+j4\sqrt{3}$ $X(6)=A_1(2)=0$
$x(3)=\frac{\sqrt{3}}{2}-j\frac{1}{2}$ $x(7)=\frac{\sqrt{3}}{2}-j\frac{1}{2}$	$a_0(3)=\sqrt{3}-j1$ $a_1(3)=0$	0 0	$X(3)=A_0(3)=0$ $X(7)=A_1(3)=0$

Figure 10: The trace of the PM DIT DFT algorithm, with $N = 8$, in extracting the coefficient $e^{j\frac{\pi}{3}}$, multiplied by 8, of $x(n) = e^{j(2\frac{2\pi}{8}n+\frac{\pi}{3})}$.