

# **The Discrete Fourier Transform**

The spectrum of a sampled function is given by

$$\begin{aligned} X_s(\Omega) &= X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-jn\Omega T} \\ &= \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n} \end{aligned}$$

where  $-\pi < \omega < \pi$  or  $0 < \omega < 2\pi$ .

Since it *impossible* to carry-out the summation from  $n=-\infty$  to  $\infty$ , let us consider a *truncated* version of  $x[n]$ :

$$\tilde{x}[n] = \begin{cases} x[n] & (0 \leq n < N - 1), \\ 0 & (\text{elsewhere}). \end{cases}$$

The corresponding Fourier transform is

$$\begin{aligned}\tilde{X}(e^{j\omega}) &= \sum_{n=-\infty}^{\infty} \tilde{x}[n]e^{-j\omega n} \\ &= \sum_{n=0}^{N-1} x[n]e^{-j\omega n}.\end{aligned}$$

Because the series is truncated, the resultant spectrum will exhibit Gibb's phenomenon: there will be ripples near the edges of the spectrum.

Now let us choose ***N** frequency points* from  $\omega = 0$  to  $2\pi$ :

$$\omega = 2\pi \frac{k}{N} \quad k = 0, 1, \dots, N-1.$$

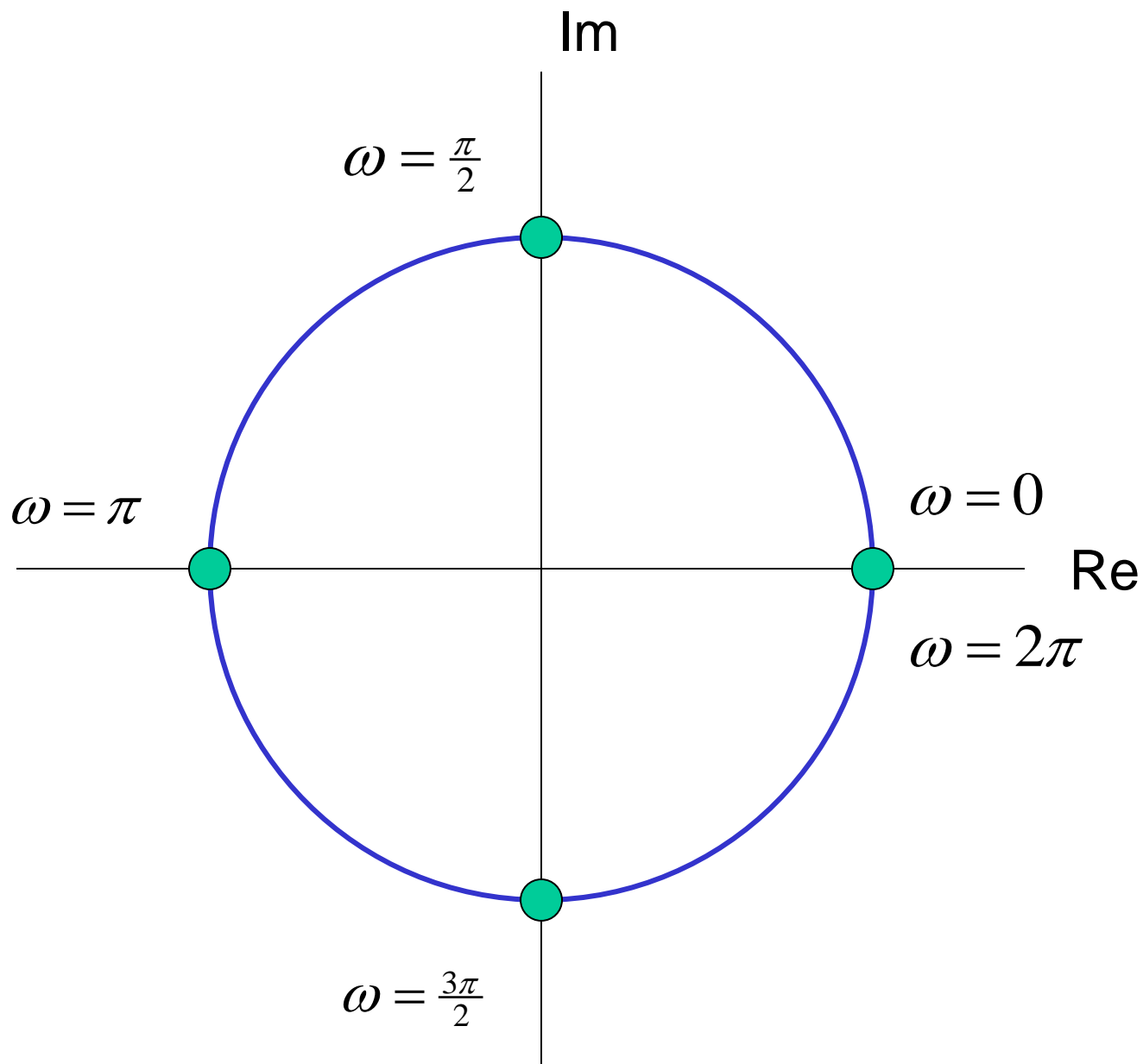
When plugged into  $e^{j\omega}$ , these values correspond to points along the unit circle.

As an example, if  $N = 4$ , we have

$$\omega = 2\pi \frac{k}{4} \quad k = 0, 1, 2, 3.$$

$$\omega = 0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}.$$

The values of  $e^{j\omega}$  are shown on the following slide.



When we insert

$$\omega = 2\pi \frac{k}{N}.$$

into

$$\tilde{X}(e^{j\omega}) = \sum_{n=0}^{N-1} x[n]e^{-j\omega n}$$

we get



$$X[k] = \tilde{X}(e^{j\omega}) \Big|_{\omega=2\pi\frac{k}{N}} = \sum_{n=0}^{N-1} x[n] e^{-j2\pi\frac{nk}{N}}.$$

This function of  $k$  is the definition of the discrete Fourier transform.

$$X[k] = \text{DFT}\{x[n]\} \equiv \sum_{n=0}^{N-1} x[n] e^{-j2\pi\frac{nk}{N}}.$$

**Example:** Find the discrete Fourier transform of  $x[n]=\delta[n]$ , for  $N=4$ .

**Solution:** we have only four time samples and four frequency samples. The values of the time samples are 1 0 0 0 (for  $n=0, 1, 2, 3$ ). Inserting these values into the DFT definition, we have

$$X[k] = \text{DFT}\{\delta[n]\} = \sum_{n=0}^3 \delta[n] e^{-j2\pi \frac{nk}{4}}$$

$$\begin{aligned} &= (1)e^{-j2\pi\frac{nk}{4}} \Big|_{n=0} \\ &= e^{-j2\pi\frac{0k}{4}} = 1. \end{aligned}$$

Note that this result is *independent* of *k*.

The result is consistent with the Fourier transform of a delta function.

**Example:** Find the discrete Fourier transform of  $x[n]=1$ , for  $N=4$ .

**Solution:** as before, we have only four time samples and four frequency samples. The values of the time samples are 1 1 1 1 (for  $n=0, 1, 2, 3$ ). Inserting these values into the DFT definition, we have

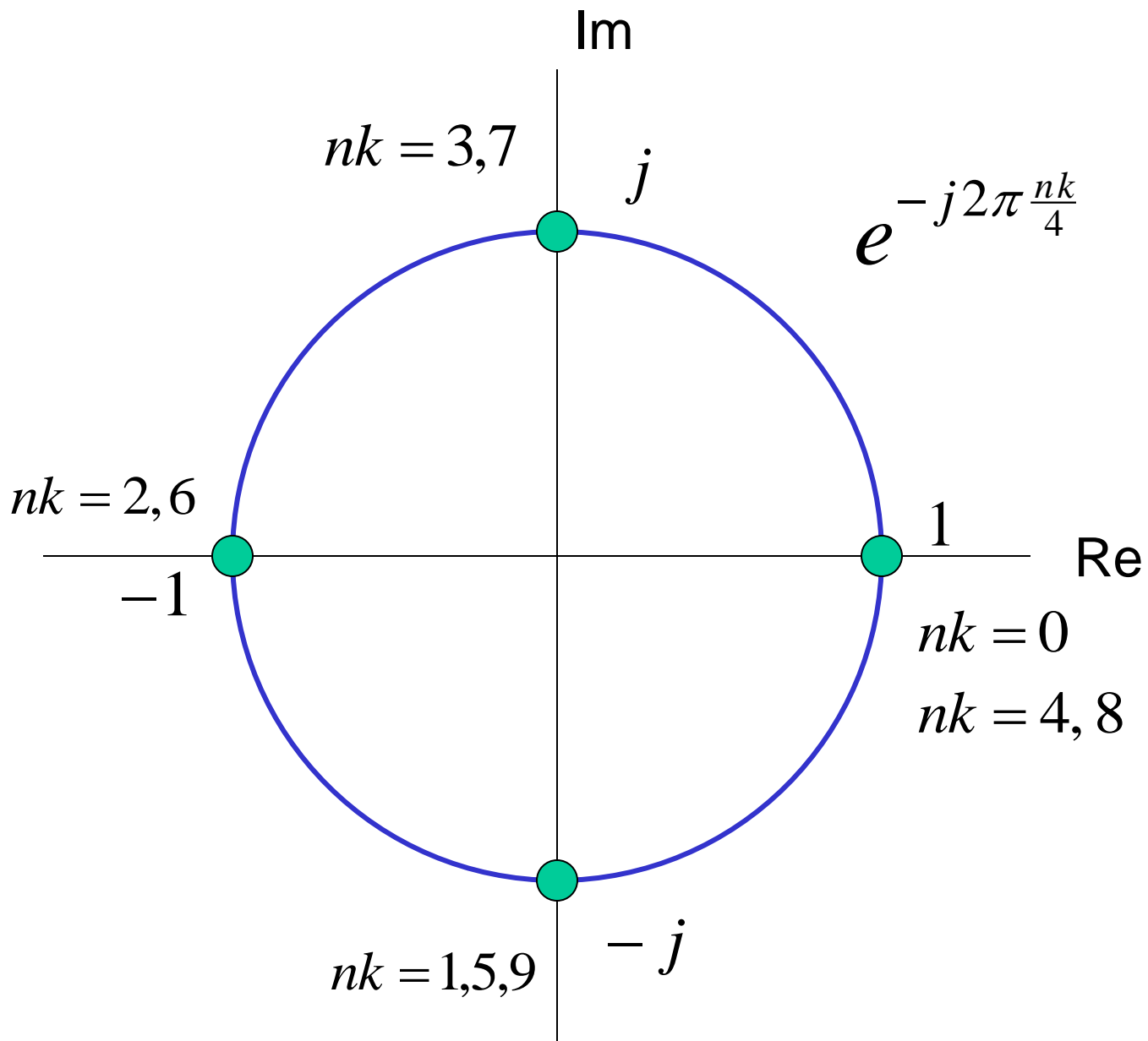
$$X[k] = \text{DFT}\{1\} = \sum_{n=0}^3 (1) e^{-j2\pi \frac{nk}{4}}$$

$$\begin{aligned}
 &= (1)e^{-j2\pi\frac{0k}{4}} + (1)e^{-j2\pi\frac{1k}{4}} + (1)e^{-j2\pi\frac{2k}{4}} + (1)e^{-j2\pi\frac{3k}{4}} \\
 &= 1 + e^{-j2\pi\frac{k}{4}} + e^{-j2\pi\frac{2k}{4}} + e^{-j2\pi\frac{3k}{4}}.
 \end{aligned}$$

This result *is* dependent upon *k*.

$$\begin{aligned}
 X(0) &= 1 + e^{-j2\pi\frac{0}{4}} + e^{-j2\pi\frac{2(0)}{4}} + e^{-j2\pi\frac{3(0)}{4}} \\
 &= 1 + 1 + 1 + 1 = 4.
 \end{aligned}$$

To perform the rest of the calculations, it is good to have our circle of value of  $e^{-j2\pi nk/4}$



$$\begin{aligned}
 X(1) &= 1 + e^{-j2\pi\frac{1}{4}} + e^{-j2\pi\frac{2(1)}{4}} + e^{-j2\pi\frac{3(1)}{4}} \\
 &= 1 - j + (-1) + j = 0.
 \end{aligned}$$

$$\begin{aligned}
 X(2) &= 1 + e^{-j2\pi\frac{2}{4}} + e^{-j2\pi\frac{2(2)}{4}} + e^{-j2\pi\frac{3(2)}{4}} \\
 &= 1 + (-1) + 1 + (-1) = 0.
 \end{aligned}$$

$$\begin{aligned}
 X(3) &= 1 + e^{-j2\pi\frac{3}{4}} + e^{-j2\pi\frac{2(3)}{4}} + e^{-j2\pi\frac{3(3)}{4}} \\
 &= 1 + j + (-1) - j = 0.
 \end{aligned}$$

The previous example suggests that the discrete Fourier transform can be calculated using a matrix equation:

$$X(0) = 1 + e^{-j2\pi\frac{0}{4}} + e^{-j2\pi\frac{2(0)}{4}} + e^{-j2\pi\frac{3(0)}{4}}.$$

$$X(1) = 1 + e^{-j2\pi\frac{1}{4}} + e^{-j2\pi\frac{2(1)}{4}} + e^{-j2\pi\frac{3(1)}{4}}.$$

$$X(2) = 1 + e^{-j2\pi\frac{2}{4}} + e^{-j2\pi\frac{2(2)}{4}} + e^{-j2\pi\frac{3(2)}{4}}.$$

$$X(3) = 1 + e^{-j2\pi\frac{3}{4}} + e^{-j2\pi\frac{2(3)}{4}} + e^{-j2\pi\frac{3(3)}{4}}.$$



$$\begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ X[3] \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{bmatrix}.$$

For the previous examples we have

$$x[n] = \delta[n]$$

$$\begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ X[3] \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

$$x[n]=1$$

$$\begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ X[3] \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

In both cases the results are consistent with Fourier transforms: the Fourier transform of an impulse is a constant (white) spectrum, and the Fourier transform of a constant is an impulse in frequency domain (just a D.C. component).

Suppose that we found the DFT of  $x[n] = \{1 \ -1 \ 1 \ -1\}$ .

$$\begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ X[3] \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 4 \\ 0 \end{bmatrix}.$$

We get the Fourier transform of a sinewave at half the sampling frequency:  $k=2$  (for  $N = 4$ ) corresponds to  $\omega=\pi$ , or  $\Omega=\Omega_s/2$ .

In general we have

$$\frac{k}{N} = \frac{\Omega}{\Omega_s}.$$

**Example:** Suppose we sample a 2 kHz sinewave at 8000 samples/second. If we perform a 1024-point DFT, where are the spikes in the transform?

**Solution:**

$$\frac{k}{1024} = \frac{2}{8} \Rightarrow k = 256.$$

There will *also* be a spike at  $k=1024-256=768$ .

We can perform discrete Fourier transformations in **MATLAB** using the function **fft()**. In MATLAB, as in other software packages, the discrete Fourier transform is implemented using an algorithm called a **fast Fourier transform**. The previous examples were done using MATLAB as will be shown on the following slides.



$$\mathbf{x}[n] = \delta[n]$$

```
>> x = [1 0 0 0];
```

```
>> X = fft(x)
```

$X =$

$1 \quad 1 \quad 1 \quad 1$

**x[n] = 1**

```
>> x = [1 1 1 1];
```

```
>> X = fft(x)
```

$X =$

4            0            0            0

$$\mathbf{x}[n] = \{1 \ -1 \ 1 \ -1\}$$

```
>> x = [1 -1 1 -1];
```

```
>> X = fft(x)
```

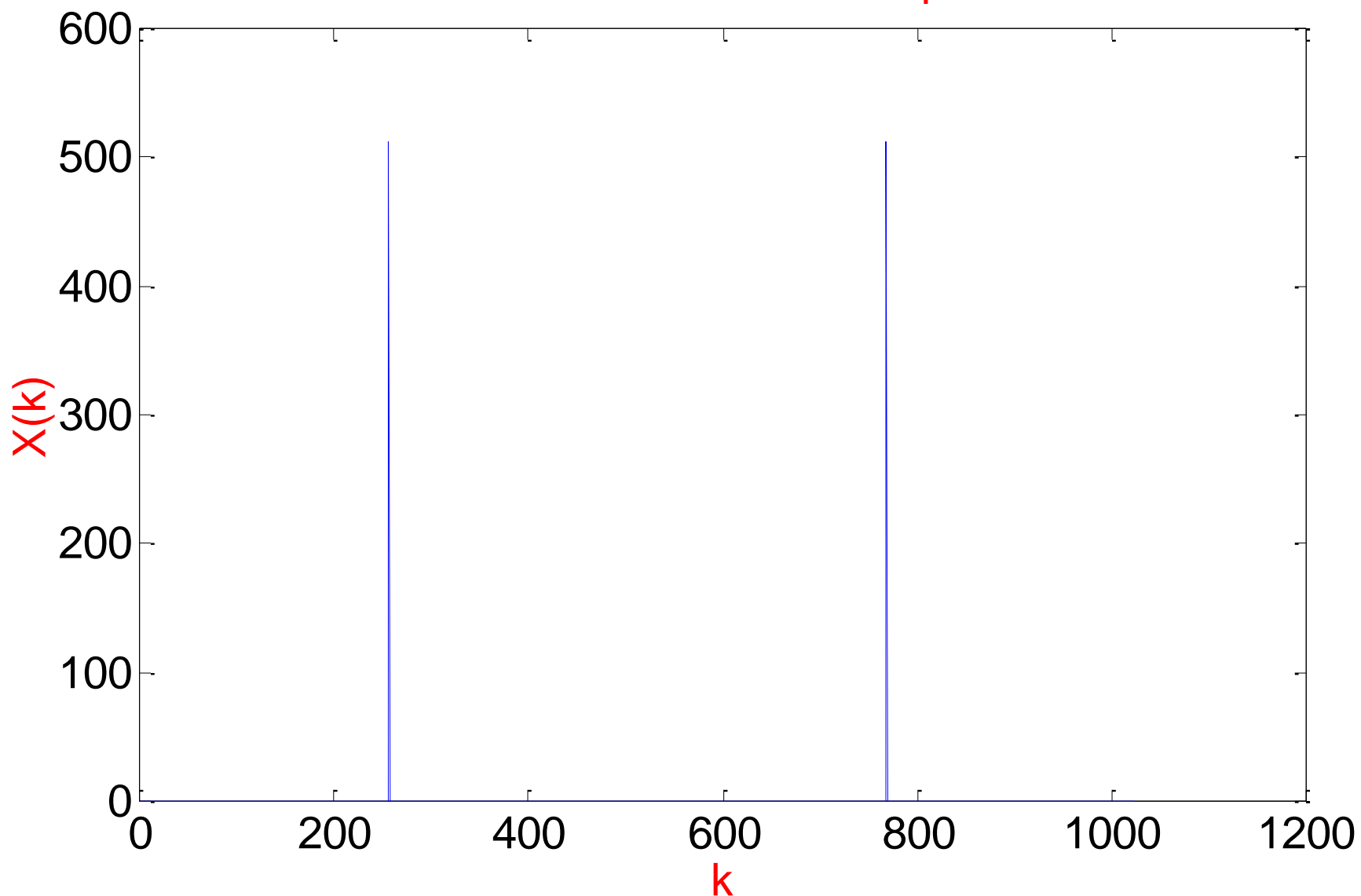
$X =$

$0 \quad 0 \quad 4 \quad 0$

**x is a 2 kHz sinewave**

```
>> n = 0:1023;  
>> t = n/8000;  
>> x = sin(2000*2*pi*t) ;  
>> X = fft(x) ;  
>> plot(n,abs(X)) ;
```

## Discrete Fourier Transform of Sampled Sinewave



**Example:** Suppose we sample a sinewave at 16000 samples/second. After performing a 2048-point DFT, we have frequency spikes at  $k=128$  and  $k = 1920$ . Find the frequency of the sinewave.

**Solution:**

$$\frac{128}{2048} = \frac{f}{16000} \Rightarrow f = 1000 \text{ Hz.}$$

**Example:** Suppose we sample the following signal at 4000 samples/sec:

$$x(t) = \cos 1000\pi t + \cos 500\pi t.$$

If we take a 512-point DFT, where are the frequency spikes?

**Solution:** We find  $k$  for each of the two frequencies:  
 $f_1 = 500$ ,  $f_2 = 250$ .

$$\frac{k}{512} = \frac{500}{4000} \Rightarrow k = 64.$$

$$\frac{k}{512} = \frac{250}{4000} \Rightarrow k = 32.$$

We also get (alias) spikes at  $k = 448$  and  $k = 480$ .



# The Fast Fourier Transform

An  $N$ -point DFT takes  $N^2$  complex multiplications and  $N^2$  complex additions. For large values of  $N$ , the number of computations becomes *very* large. A number of computationally-efficient algorithms have been developed called ***fast Fourier transforms***, whose number of computations is far less than  $N^2$ .

The key to the efficiency is in the exploitation of the properties of  $e^{-j2\pi kn/N}$ .

Let us look at the discrete Fourier transform for  $N=4$ :

$$\begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ X[3] \end{bmatrix} = \begin{bmatrix} e^{-j2\pi \frac{0(0)}{4}} & e^{-j2\pi \frac{0(1)}{4}} & e^{-j2\pi \frac{0(2)}{4}} & e^{-j2\pi \frac{0(3)}{4}} \\ e^{-j2\pi \frac{1(0)}{4}} & e^{-j2\pi \frac{1(1)}{4}} & e^{-j2\pi \frac{1(2)}{4}} & e^{-j2\pi \frac{1(3)}{4}} \\ e^{-j2\pi \frac{2(0)}{4}} & e^{-j2\pi \frac{2(1)}{4}} & e^{-j2\pi \frac{2(2)}{4}} & e^{-j2\pi \frac{2(3)}{4}} \\ e^{-j2\pi \frac{3(0)}{4}} & e^{-j2\pi \frac{3(1)}{4}} & e^{-j2\pi \frac{3(2)}{4}} & e^{-j2\pi \frac{3(3)}{4}} \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{bmatrix}.$$

In order to simplify the notation, let  $\mathbf{W} = e^{-j2\pi/N}$

$$\begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ X[3] \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 \\ W^0 & W^1 & W^2 & W^3 \\ W^0 & W^2 & W^4 & W^6 \\ W^0 & W^3 & W^6 & W^9 \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{bmatrix}.$$

Because of the periodicity of  $W^{nk}$ , we have

$$\begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ X[3] \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 \\ W^0 & W^1 & W^2 & W^3 \\ W^0 & W^2 & W^0 & W^2 \\ W^0 & W^3 & W^2 & W^1 \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{bmatrix}.$$

Now, suppose we rearrange the **order** of the  $x[n]$  (time-domain) values:

$$\begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ X[3] \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 \\ W^0 & W^2 & W^1 & W^3 \\ W^0 & W^0 & W^2 & W^2 \\ W^0 & W^2 & W^3 & W^1 \end{bmatrix} \begin{bmatrix} x[0] \\ x[2] \\ x[1] \\ x[3] \end{bmatrix}.$$

This reordering lends itself to a *repartitioning* of the matrices:

$$\begin{bmatrix} X[0] \\ X[1] \\ \hline X[2] \\ X[3] \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 \\ W^0 & W^2 & W^1 & W^3 \\ \hline W^0 & W^0 & W^2 & W^2 \\ W^0 & W^2 & W^3 & W^1 \end{bmatrix} \begin{bmatrix} x[0] \\ x[2] \\ \hline x[1] \\ x[3] \end{bmatrix}.$$

Adopting further notation for the partitioned matrix, we have

$$\begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{W}_{11} & \mathbf{W}_{12} \\ \mathbf{W}_{21} & \mathbf{W}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{\text{even}} \\ \mathbf{x}_{\text{odd}} \end{bmatrix},$$

where

$$\mathbf{W}_{11} = \begin{bmatrix} W^0 & W^0 \\ W^0 & W^2 \end{bmatrix} \equiv \mathbf{W}_{N/2},$$

The  $\mathbf{W}_{11}$  matrix is a two-point DFT.

$$\mathbf{W}_{12} = \begin{bmatrix} W^0 & W^0 \\ W^1 & W^3 \end{bmatrix},$$

The  $\mathbf{W}_{12}$  matrix is a two-point DFT with the second row (output) multiplied by  $W^1$ . We denote this operation by

$$\mathbf{W}_{12} = [1, W^1] \mathbf{W}_{N/2}.$$

The  $\mathbf{W}_{21}$  matrix is the same as the  $\mathbf{W}_{11}$  matrix—a two-point DFT.



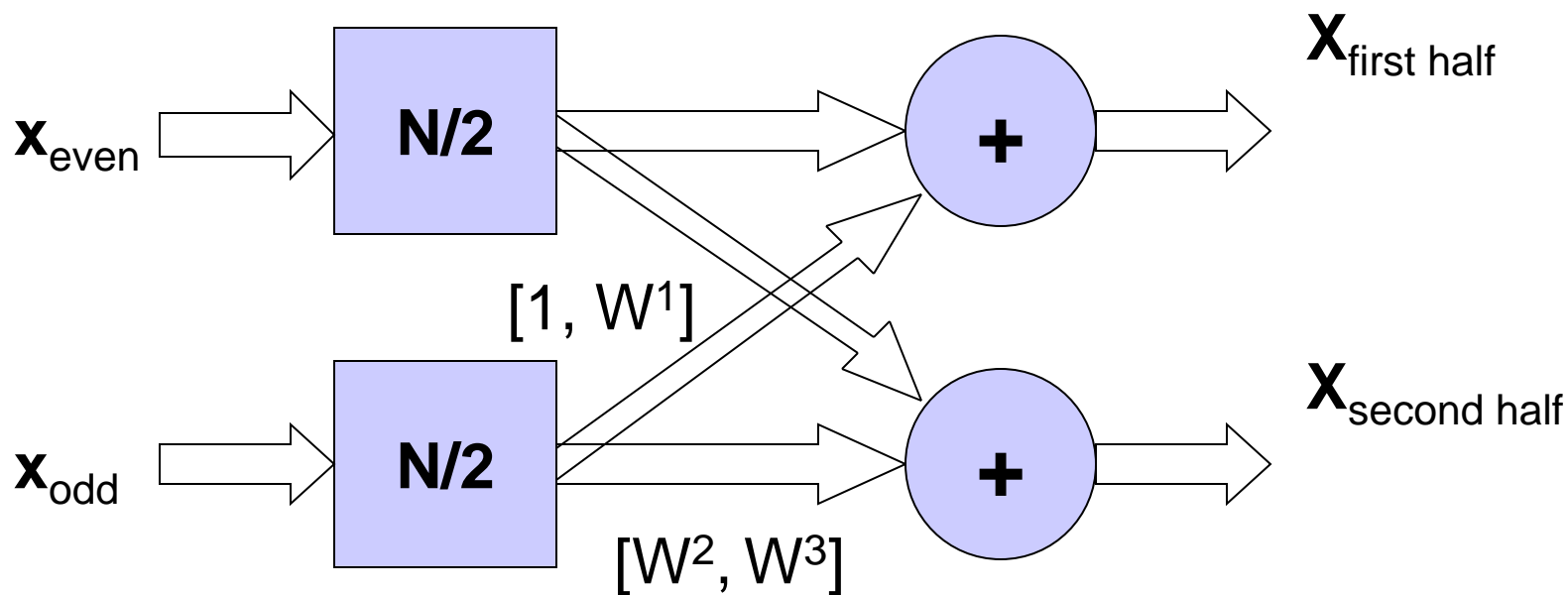
and

$$\mathbf{W}_{22} = \begin{bmatrix} W^2 & W^2 \\ W^3 & W^1 \end{bmatrix}.$$

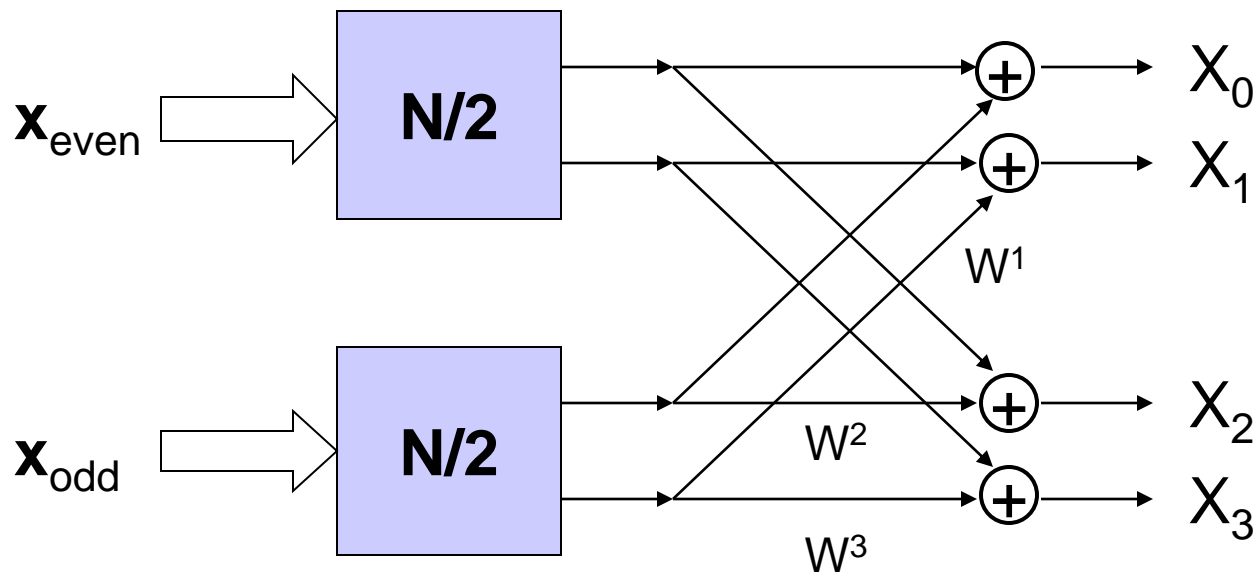
The  $\mathbf{W}_{22}$  matrix is a two-point DFT with the first row multiplied by  $W^2$ , and the second row (output) multiplied by  $W^3$ :

$$\mathbf{W}_{22} = [W^2, W^3] \mathbf{W}_{N/2}.$$

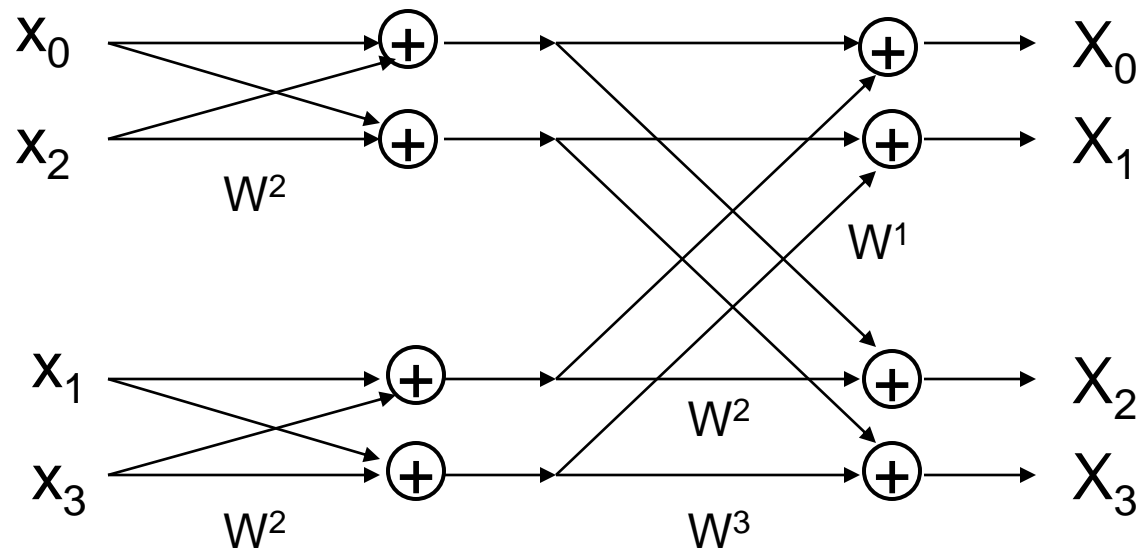
$$\begin{bmatrix} \mathbf{X}_{\text{first half}} \\ \mathbf{X}_{\text{second half}} \end{bmatrix} = \begin{bmatrix} \mathbf{W}_{N/2} \mathbf{x}_{\text{even}} & [1, W^1] \mathbf{W}_{N/2} \mathbf{x}_{\text{odd}} \\ \mathbf{W}_{N/2} \mathbf{x}_{\text{even}} & [W^2, W^3] \mathbf{W}_{N/2} \mathbf{x}_{\text{odd}} \end{bmatrix},$$



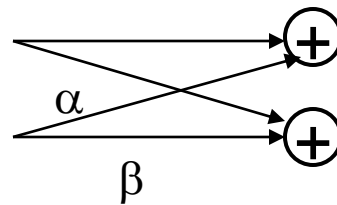
Further detail can be given to the block diagram:



Finally,



The resultant algorithm is called a [decimation in time, radix-2] **fast Fourier transform** (FFT) and uses fewer complex multiplications than a conventional discrete Fourier transform (DFT). The [radix-2] FFT algorithm consists of a number of operations that look like this:



This operation is called a **butterfly** and consists of two (complex) additions and **two** (complex) **multiplications**.

This general computational idea can be extended to higher-order DFT's.

For example, an 8-point DFT can be constructed from two 4-point DFT's and four butterflies to connect their outputs. A 16-point DFT can be constructed from two 8-point DFT's and eight butterflies. The savings in number of computations can be seen in the following table.

<b>N</b>	<b>DFT</b>	<b>FFT</b>
4	16	8
8	64	24
16	256	64
32	1024	160
64	4096	384

The *number of complex multiplications* to perform an **N**-point DFT using the **conventional (DFT)** algorithm and the **FFT** algorithm.



**Exercise:** Construct an 8-point [decimation-in-time, radix-2] FFT from two 4-point FFT's (already constructed). Verify that the number of (complex) multiplications is 24.

There are other types of fast Fourier transforms. The FFT just developed was created by reordering the time samples into even and odd parts. A *second* type of FFT reorders the *frequency* samples into even and odd parts. The first FFT algorithm is called a decimation-in-time FFT. The second algorithm is called a **decimation-in-frequency FFT**.

## 4-Point Decimation-in-Frequency FFT Algorithm

$$\begin{bmatrix} X[0] \\ X[2] \\ X[1] \\ X[3] \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 \\ W^0 & W^2 & W^0 & W^2 \\ W^0 & W^1 & W^2 & W^3 \\ W^0 & W^3 & W^2 & W^1 \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{bmatrix}.$$

$$\begin{bmatrix} X[0] \\ X[2] \\ \hline X[1] \\ X[3] \end{bmatrix} = \begin{bmatrix} W^0 & W^0 & W^0 & W^0 \\ W^0 & W^2 & W^0 & W^2 \\ \hline W^0 & W^1 & W^2 & W^3 \\ W^0 & W^3 & W^2 & W^1 \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ \hline x[2] \\ x[3] \end{bmatrix},$$

$$\begin{bmatrix} \mathbf{X}_{\text{even}} \\ \mathbf{X}_{\text{odd}} \end{bmatrix} = \begin{bmatrix} \mathbf{W}_{N/2} & \mathbf{W}_{N/2} \\ \mathbf{W}_{N/2}[1, W^1] & \mathbf{W}_{N/2}[W^2, W^3] \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix}.$$

