Chapter 4

Discrete Fourier Transform (DFT)
And
Signal Spectrum

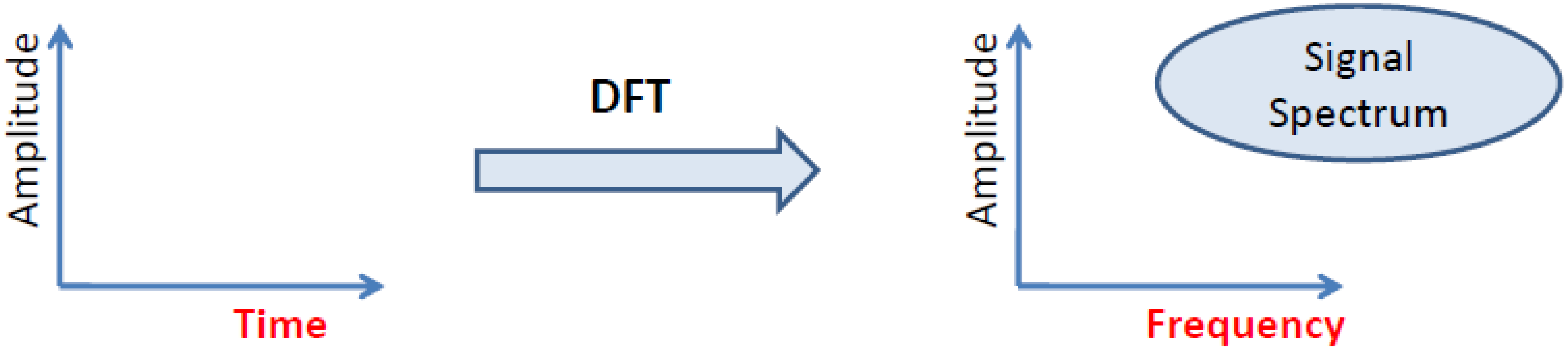
Fourier Transform History

- Born 21 March 1768 (Auxerre).
- **Died** 16 May 1830 (Paris)
- French mathematician and physicist.
- Best known for initiating the investigation of Fourier series.
- Fourier series applications to problems of heat transfer and vibrations.
- The Fourier series is used to represent a periodic function by a discrete sum of complex exponentials.
- Fourier transform is then used to represent a general, non-periodic function by a continuous superposition or integral of complex exponentials (the period approaches to infinity).

Jean-Baptiste Joseph Fourier

Discrete Fourier Transform

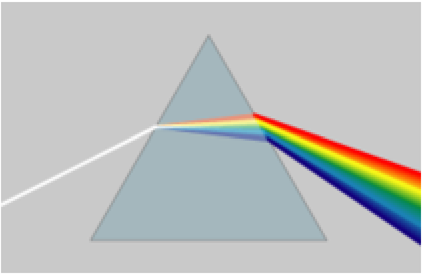
- In this chapter we introduce the concept of Fourier or ***frequency-domain representation of signals***.
- ***Discrete Fourier Transform*** (***DFT***) transforms (*break up the signal into summations of sinusoidal components*) the time domain signal samples to the frequency domain components (*frequency analysis*).



In the ***time domain***, representation of digital signals describes the signal amplitude versus the sample number (time).

The representation of the digital signal in terms of its frequency component in a ***frequency domain***, displays the frequency information of a digital signal (signal spectrum).

- Fourier analysis is like a *glass prism*, which splits a beam of light into frequency components corresponding to different colors.



Continuous-time sinusoids

- A *continuous-time sinusoidal* signal may be represented as a function of time *t* by the equation

Amplitude — frequency — phase in radians

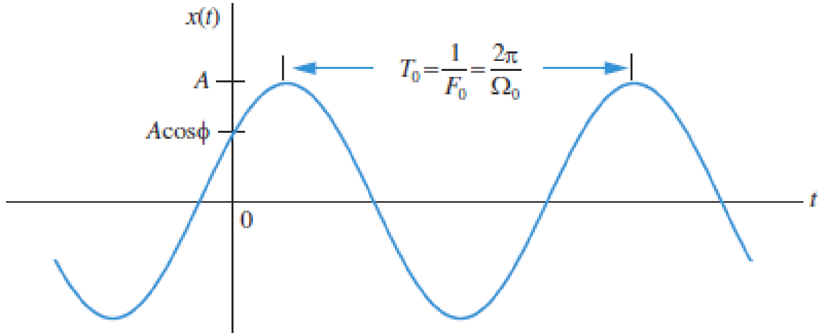
$$x(t) = A\cos(2\pi F_0 t + \theta), \quad -\infty < t < \infty$$

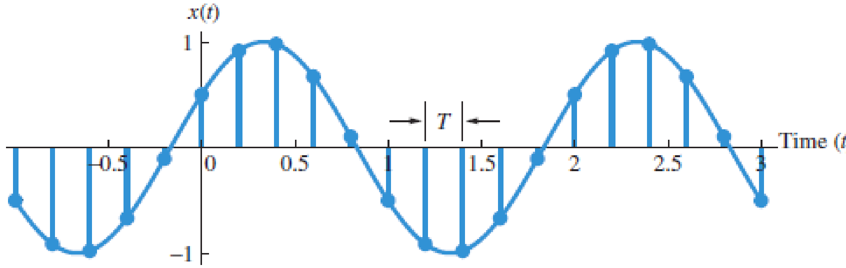
- *The angular* or *radian* frequency (radians per second.) $\Omega_0 = 2\pi F_0$
- A *discrete-time sinusoidal signal* is conveniently obtained by sampling the continuous-time sinusoid at equally spaced points $t = nT$

$$x[n] = x(nT) = A\cos(2\pi F_0 nT + \theta) = A\cos\left(2\pi \frac{F_0}{F_s} n + \theta\right)$$
$$= A\cos(\omega_0 n + \theta), \quad -\infty < n < \infty$$

$f \triangleq \frac{F}{F_s} = FT$, *normalized frequency*

$\omega \triangleq 2\pi f = 2\pi \frac{F}{F_s} = \Omega T$, *normalized angular frequency*

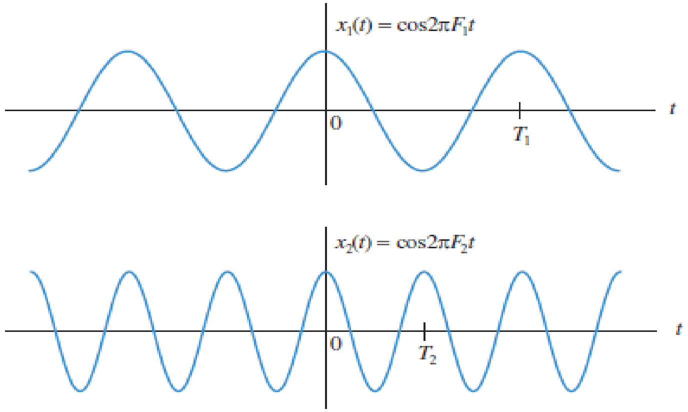




- Using Euler's identity $e^{\pm j\phi} = \cos\phi \pm j\sin\phi$, we can express every sinusoidal signal in terms of two complex exponentials with the same frequency

$$A\cos(\Omega_0 t + \theta) = \frac{A}{2} e^{j\theta} e^{j\Omega_0 t} + \frac{A}{2} e^{-j\theta} e^{-j\Omega_0 t}.$$

  - *Frequency* (positive quantity.), viewed as the *number of cycles completed per unit of time*.
  - *Negative frequencies* is a convenient way to describe signals in terms of complex exponentials.



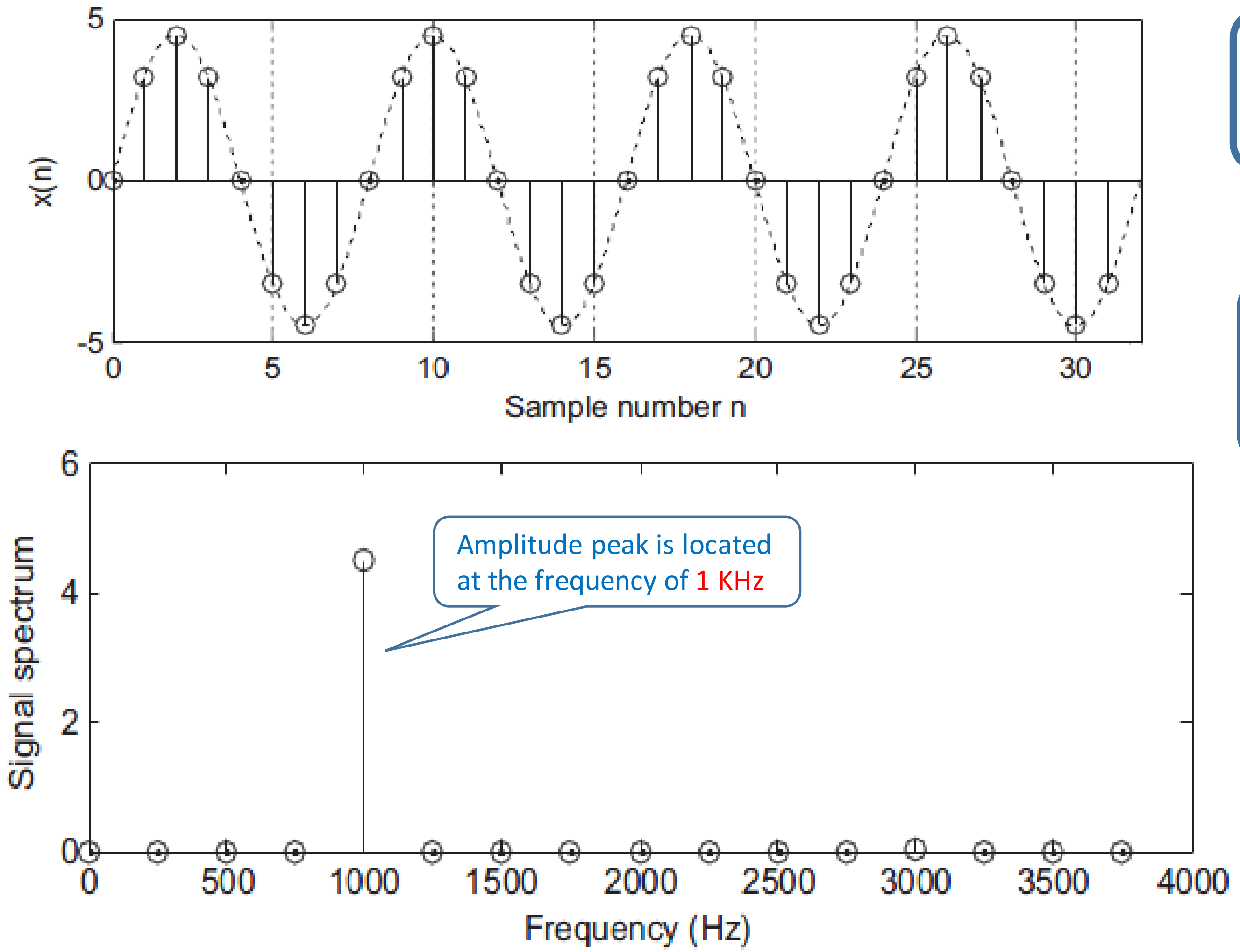
For continuous-time sinusoids, $F_1 < F_2$ always implies that $T_1 > T_2$.

DFT: Graphical Example

Time domain representation

DFT

Frequency domain representation



Time domain

1000-Hz sinusoid with 32 samples at a sampling rate of 8000 Hz in

Sampling Rate

8000 samples = 1 second

-> sampling period $T_s = \frac{1}{8000} = 125\mu s$

Duration of 32 samples = 32*0.125 ms = 4 ms

Signal Frequency

1000-Hz sinusoid -> $T = 1ms$

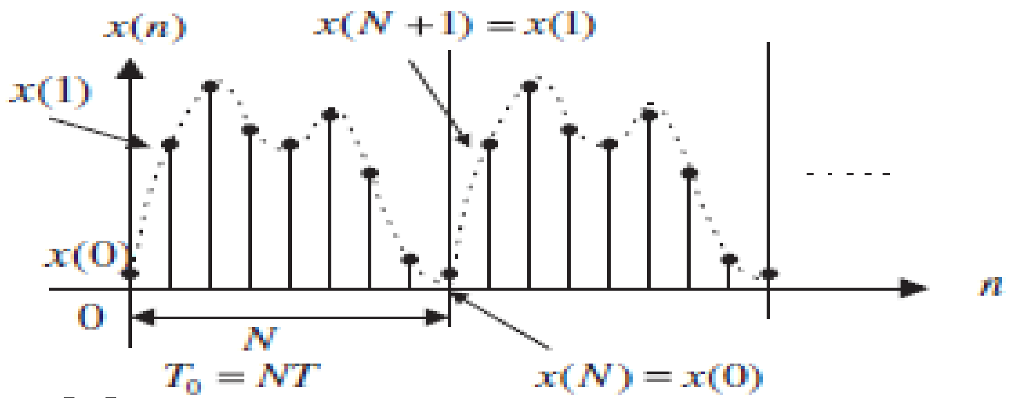
32 samples = 4 ms -> 4 cycles.

DFT Coefficients of Periodic Signals

- Given a set of N harmonically related complex exponentials $e^{j\frac{2\pi}{N}kn}$, We can synthesize a signal $x[n]$

$$x[n] = \sum_{k=0}^{N-1} c_k e^{j\frac{2\pi}{N}kn}.$$

$x[n]$ is sampled at a rate of $f_s\ Hz$ (period $T_0 = NT = N\ \frac{1}{f_s}$)



Periodic Digital Signal $x[n]$

Equation of DFT coefficients:

We determine the coefficients c_k from the values of the periodic signal $x[n]$

Sum over one period

$$c_k = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn}, \quad k = 0, \pm 1, \pm 2, \cdots,$$

DTFS

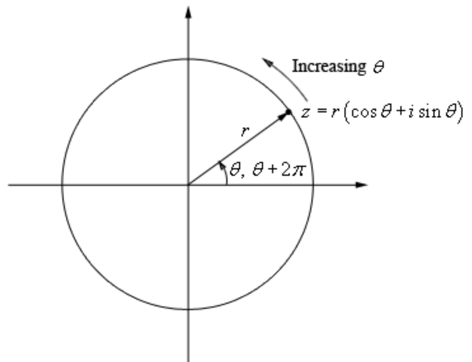
Discrete-Time Fourier Series

Fourier Synthesis Equation — **Fourier Analysis Equation**

$$x[n] = \sum_{k=0}^{N-1} c_k e^{j\frac{2\pi}{N}kn} \quad \xleftrightarrow{\text{DTFS}} \quad c_k = \frac{1}{N} \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn}.$$

We have: $e^{j\theta} = \cos(\theta) + j\sin(\theta)$ *and* $e^{j(\theta+2\pi)} = e^{j\theta}$ *period of* 2π

For $\theta(t) = \omega t \rightarrow e^{j\omega t} = \cos(\omega t) + j\sin(\omega t)$ Rotation of a point on a circle



DFT Coefficients of Periodic Signals

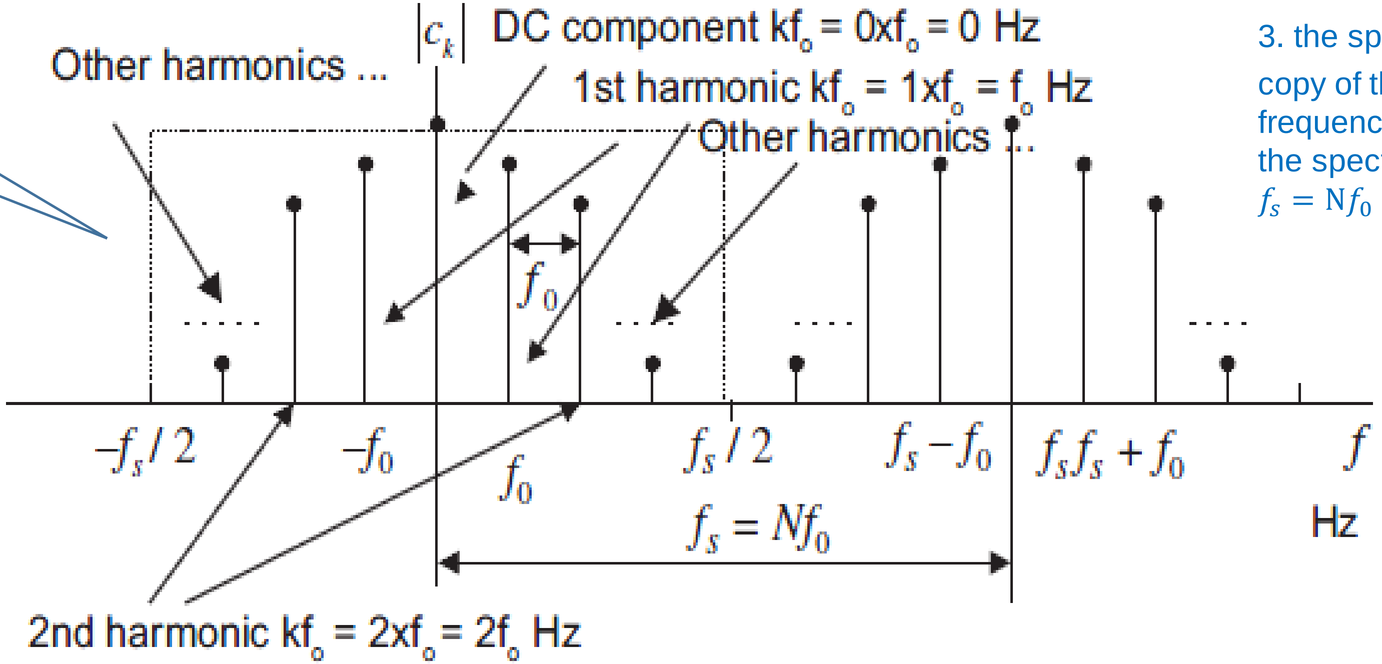
- Fourier series coefficient C_k is periodic of N

$$c_{k+N} = \frac{1}{N}\sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi(k+N)n}{N}} = \frac{1}{N}\sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi kn}{N}}e^{-j2\pi n}$$

Since $e^{-j2\pi n} = \cos(2\pi n) - j\sin(2\pi n) = 1$ → $C_{K+N} = C_K$

Amplitude spectrum of the periodic digital signal →

The spectrum C_k has two sides.



Remarks

1. spectral portion between the frequency $-f_s$ and f_s (folding frequency) represents frequency information of the periodic signal.

2. For the k[th] harmonic, the frequency is $f = kf_0$ Hz (f_0 is the frequency resolution =The frequency spacing between the consecutive spectral lines)

3. the spectral portion from $\frac{f_s}{2}$ to f_s is a copy of the spectrum in the negative frequency range from $-f_s/2$ to 0 Hz due to the spectrum being periodic for every $f_s = Nf_0$ Hz.

Example 1

The periodic signal $x(t)$ is sampled at $f_s = 4Hz$ $x(t) = sin(2\pi t)$

a. Compute the spectrum C_k using the samples in one period.

b. Plot the two-sided amplitude spectrum $|C_k|$ over the range from -2 to 2 Hz.

Solution:

Fundamental frequency

a. We match $x(t) = sin(2\pi t)$ with $x(t) = sin(2\pi f t)$ and get $f = 1Hz$

Therefore the signal has 1 cycle or 1 period in 1 second

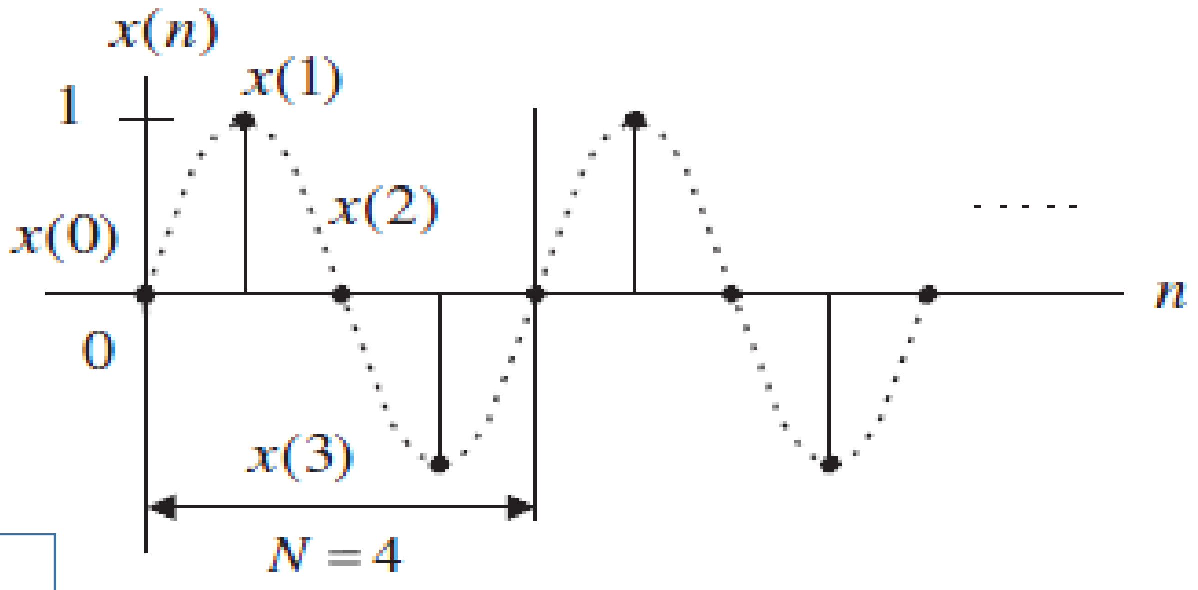
Sampling rate $f_s = 4\ Hz$ → 1 second has 4 samples.

Hence, there are 4 samples in 1 period for this particular signal.

$T = \frac{1}{f_s} = 0.25\ sec$ —Sampled signal→ $x(n) = x(nT) = \sin(2\pi nT) = \sin(0.5\pi n).$

Example 1 -contd. (1)

$$x(n) = x(nT) = \sin(2\pi nT) = \sin(0.5\pi n).$$



$$x(0) = 0; \quad x(1) = 1;$$
$$x(2) = 0; \quad x(3) = -1;$$

b. spectrum

$$c_k = \frac{1}{N}\sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi kn}{N}}, \quad k = 0, 1, \cdots, N-1$$

$$c_0 = \frac{1}{4}\sum_{n=0}^{3} x(n) = \frac{1}{4}\Big(x(0) + x(1) + x(2) + x(3)\Big) = \frac{1}{4}(0 + 1 + 0 - 1) = 0$$

$$c_1 = \frac{1}{4}\sum_{n=0}^{3} x(n)e^{-j2\pi\times 1n/4} = \frac{1}{4}\Big(x(0) + x(1)e^{-j\pi/2} + x(2)e^{-j\pi} + x(3)e^{-j3\pi/2}\Big)$$

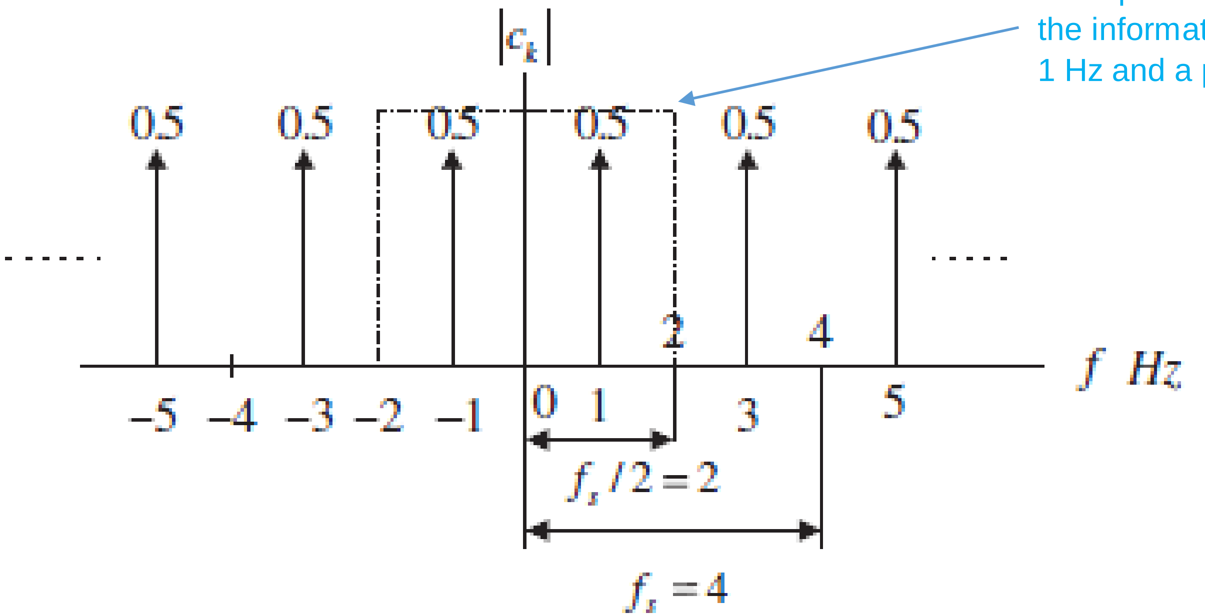
$$= \frac{1}{4}\Big(x(0) - jx(1) - x(2) + jx(3) = 0 - j(1) - 0 + j(-1)\Big) = -j0.5$$

Example 1 –contd. (2)

$$c_2 = \frac{1}{4}\sum_{n=0}^{3} x(n)e^{-j2\pi\times 2n/4} = 0, \text{ and } c_3 = \frac{1}{4}\sum_{n=0}^{3} x(n)e^{-j2\pi\times 3n/4} = j0.5$$

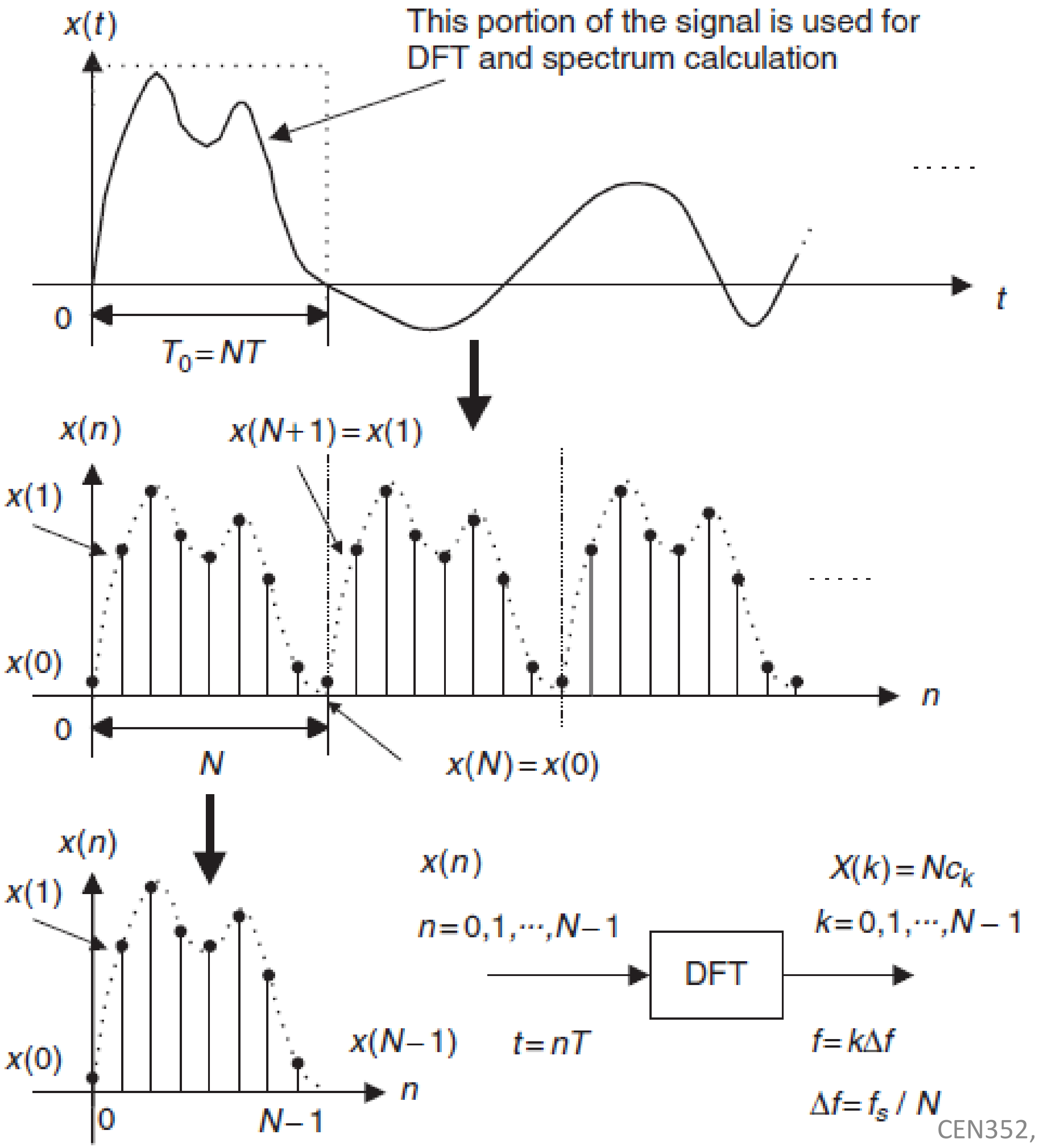
Using periodicity, it follows that $c_{-1} = c_3 = j0.5$, and $c_{-2} = c_2 = 0$

The amplitude spectrum for the digital signal



The spectrum in the range of -2 to 2 Hz presents the information of the sinusoid with a frequency of 1 Hz and a peak value of $2|C_1| = 1$.

Discrete Fourier Transform DFT Formulas



acquired N data samples with duration of T_0

Imagine periodicity of N samples.

Take first N samples (index 0 to N-1) as the input periodic signal $x(n)$ to DFT.

DFT Formulas

Given N data samples of $x[n]$, the N-point *discrete Fourier transform (DFT)* $X(k)$ is defined by:

$$X(k) = Nc_k = \sum_{n=0}^{N-1} x(n)e^{-j\frac{2\pi kn}{N}}, \quad k = 0, 1, \cdots, N-1$$

DFT Coefficients Formula.

Fourier series coefficients

$$= \sum_{n=0}^{N-1} x(n)W_N^{kn}, \quad \text{for } k = 0, 1, \cdots, N-1$$

- k is the *discrete frequency index* (frequency bin number) indicating each calculated DFT coefficient.

$$X(k) = x(0)W_N^{k0} + x(1)W_N^{k1} + x(2)W_N^{k2} + \cdots + x(N-1)W_N^{k(N-1)}, \quad \text{for } k = 0, 1, \cdots, N-1$$

Where the factor W_N is called the *twiddle factor* $W_N = e^{-j2\pi/N} = \cos\left(\frac{2\pi}{N}\right) - j\sin\left(\frac{2\pi}{N}\right)$

Inverse DFT

Given *N* DFT coefficients *X*[*k*], The *inverse of the DFT x[n]* is given by

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)e^{j2\pi kn/N}$$

Inverse DFT (IDFT)

$$= \frac{1}{N}\sum_{k=0}^{N-1} X(k)W_N^{-kn}, \quad \text{for } n = 0, 1, \cdots, N-1$$

$$x(n) = \frac{1}{N}\left(X(0)W_N^{-0n} + X(1)W_N^{-1n} + X(2)W_N^{-2n} + \cdots + X(N-1)W_N^{-(N-1)n}\right),$$

for $n = 0, 1, \cdots, N-1$

Analysis equation		Synthesis equation
$X[k] = \sum_{n=0}^{N-1} x[n]W_N^{kn}$	$\xleftrightarrow[N]{\text{DFT}}$	$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]W_N^{-kn},$

$$x_N = \frac{1}{N}W_N^H X_N = \frac{1}{N}W_N^* X_N. \quad \text{(IDFT)}$$

W_N^H is the conjugate transpose of W_N

MATLAB Functions

We can use MATLAB functions ***fft()*** and ***ifft()*** to compute the DFT coefficients and the inverse DFT with the syntax listed in Table:

FFT: Fast Fourier Transform

MATLAB FFT functions.

X = fft(x)	% Calculate DFT coefficients
x = ifft(X)	% Inverse DFT
x = input vector	
X = DFT coefficient vector	

Example 2

Given a sequence $x(n)$ for $0 \le n \le 3$ where $x(0) = 1, x(1) = 2, x(2) = 3, and\ x(3) = 4.$ evaluate DFT $X(k)$.

Solution:

Since N = 4 and $W_4 = e^{-j\frac{\pi}{2}}$ ⇨ $X(k) = \sum_{n=0}^{3} x(n) W_4^{kn} = \sum_{n=0}^{3} x(n) e^{-j\frac{\pi k n}{2}}$

Thus, for $k = 0$

$$X(0) = \sum_{n=0}^{3} x(n)e^{-j0} = x(0)e^{-j0} + x(1)e^{-j0} + x(2)e^{-j0} + x(3)e^{-j0}$$

$$= x(0) + x(1) + x(2) + x(3)$$

for $k = 1$

$$X(1) = \sum_{n=0}^{3} x(n)e^{-j\frac{\pi n}{2}} = x(0)e^{-j0} + x(1)e^{-j\frac{\pi}{2}} + x(2)e^{-j\pi} + x(3)e^{-j\frac{3\pi}{2}}$$

$$= x(0) - jx(1) - x(2) + jx(3)$$

$$= 1 - j2 - 3 + j4 = -2 + j2$$

Example 2 -contd.

for $k = 2$

$$X(2) = \sum_{n=0}^{3} x(n)e^{-j\pi n} = x(0)e^{-j0} + x(1)e^{-j\pi} + x(2)e^{-j2\pi} + x(3)e^{-j3\pi}$$

$$= x(0) - x(1) + x(2) - x(3)$$

$$= 1 - 2 + 3 - 4 = -2$$

and for $k = 3$

$$X(3) = \sum_{n=0}^{3} x(n)e^{-j\frac{3\pi n}{2}} = x(0)e^{-j0} + x(1)e^{-j\frac{3\pi}{2}} + x(2)e^{-j3\pi} + x(3)e^{-j\frac{9\pi}{2}}$$

$$= x(0) + jx(1) - x(2) - jx(3)$$

$$= 1 + j2 - 3 - j4 = -2 - j2$$

Using MATLAB,

```
>> X = fft([1  2 3 4])
   X = 10.0000   - 2.0000 + 2.0000i   - 2.0000   - 2.0000 - 2.0000i
```

Example 2 -contd.

Using the DFT complex matrix

We first compute the entries of the matrix W_4 using the property: $W_N^{k+N} = W_N^k = e^{-j\frac{2\pi}{N}k} = \cos\left(\frac{2\pi}{N}k\right) - j\sin\left(\frac{2\pi}{N}k\right).$

The result is a complex matrix given by:

$$W_4 = \begin{bmatrix} W_4^0 & W_4^0 & W_4^0 & W_4^0 \\ W_4^0 & W_4^1 & W_4^2 & W_4^3 \\ W_4^0 & W_4^2 & W_4^4 & W_4^6 \\ W_4^0 & W_4^3 & W_4^6 & W_4^9 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & W_4^1 & W_4^2 & W_4^3 \\ 1 & W_4^2 & W_4^0 & W_4^2 \\ 1 & W_4^3 & W_4^2 & W_4^1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}.$$

The DFT coefficients are evaluated by the matrix-by-vector multiplication

$$\begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ X[3] \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 10 \\ -2+2j \\ -2 \\ -2-2j \end{bmatrix}$$

In **MATLAB** these computations are done using the commands:

The DFT

```
x=[0 1 2 3]'; W=dftmtx(4); X=W*x;
```

The inverse DFT

```
x=inv(W)*X
```

Example 3

Using DFT coefficients *X(k)* for $0 \le n \le 3$ of previous example, evaluate the inverse DFT (IDFT) to determine the time domain sequence $x(n)$.

Solution:

$N = 4$ and $W_4^{-1} = e^{j\frac{\pi}{2}}$ ⇒ $$x(n) = \frac{1}{4}\sum^{3} X(k)W_4^{-nk} = \frac{1}{4}\sum^{3} X(k)e^{j\frac{\pi kn}{2}}.$$

for $n = 0$
$$x(0) = \frac{1}{4}\sum_{k=0}^{3} X(k)e^{j0} = \frac{1}{4}\left(X(0)e^{j0} + X(1)e^{j0} + X(2)e^{j0} + X(3)e^{j0}\right)$$
$$= \frac{1}{4}(10 + (-2 + j2) - 2 + (-2 - j2)) = 1$$

for $n = 1$
$$x(1) = \frac{1}{4}\sum_{k=0}^{3} X(k)e^{j\frac{k\pi}{2}} = \frac{1}{4}\left(X(0)e^{j0} + X(1)e^{j\frac{\pi}{2}} + X(2)e^{j\pi} + X(3)e^{j\frac{3\pi}{2}}\right)$$
$$= \frac{1}{4}(X(0) + jX(1) - X(2) - jX(3))$$
$$= \frac{1}{4}(10 + j(-2 + j2) - (-2) - j(-2 - j2)) = 2$$

Example 3 -contd.

for $n = 2$

$$x(2) = \frac{1}{4}\sum_{k=0}^{3} X(k)e^{jk\pi} = \frac{1}{4}\left(X(0)e^{j0} + X(1)e^{j\pi} + X(2)e^{j2\pi} + X(3)e^{j3\pi}\right)$$

$$= \frac{1}{4}(X(0) - X(1) + X(2) - X(3))$$

$$= \frac{1}{4}(10 - (-2 + j2) + (-2) - (-2 - j2)) = 3$$

and for $n = 3$

$$x(3) = \frac{1}{4}\sum_{k=0}^{3} X(k)e^{j\frac{k\pi 3}{2}} = \frac{1}{4}\left(X(0)e^{j0} + X(1)e^{j\frac{3\pi}{2}} + X(2)e^{j3\pi} + X(3)e^{j\frac{9\pi}{2}}\right)$$

$$= \frac{1}{4}(X(0) - jX(1) - X(2) + jX(3))$$

$$= \frac{1}{4}(10 - j(-2 + j2) - (-2) + j(-2 - j2)) = 4$$

Using MATLAB,

$$\gg \mathrm{x} = \mathrm{ifft}([10 \quad -2+2j \quad -2 \quad -2-2j])$$

$$\mathrm{x} = 1 \qquad 2 \qquad 3 \qquad 4.$$

Frequency of bin k

- The calculated ***N*** **DFT** coefficients ***X (k)*** represent the frequency components ranging from $0\ Hz\ to\ f_s\ Hz.$
- The relationship between the frequency ***bin k*** and its associated frequency is computed using:

$$f = k\frac{f_s}{N} = k\ \Delta f\ (Hz)$$

- The ***frequency resolution*** (frequency step between two consecutive DFT coefficients)

$$\Delta f = \frac{f_s}{N}\ (\mathrm{Hz})$$

Example 4

In the previous example, if the sampling rate is 10 Hz,

a. Determine the sampling period, time index, and sampling time instant for a digital sample $x(3)$ in the time domain;

b. Determine the frequency resolution, frequency bin, and mapped frequencies for the DFT coefficients $X(1)$and $X(3)$ in the frequency domain.

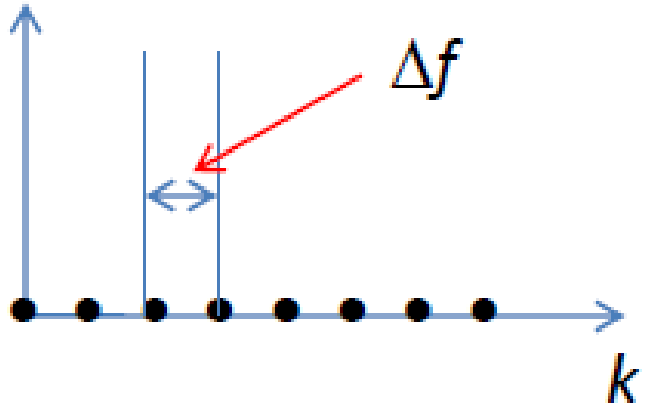
Solution:

a. Sampling period: $T = 1/f_s = 1/10 = 0.1$ second

For x(3), the time index is n = 3 and the sampling time instant is determined by

$$t = nT = 3 \cdot 0.1 = 0.3 \text{ second}$$

b. Frequency resolution: $\Delta f = \frac{f_s}{N} = \frac{10}{4} = 2.5\,\text{Hz}.$



Frequency bin number for X(1) is k = 1, and its corresponding frequency is $f = \frac{kf_s}{N} = \frac{1 \times 10}{4} = 2.5\,\text{Hz}.$

Similarly, for X(3) is k = 3, and its corresponding frequency is $f = \frac{kf_s}{N} = \frac{3 \times 10}{4} = 7.5\,\text{Hz}.$

Amplitude and Power Spectrum

- Since each calculated DFT coefficient is a complex number, it is not convenient to plot it versus its frequency index.
- Hence, after evaluating the N DFT coefficients, the magnitude and phase of each DFT coefficient can be determined and plotted versus its frequency index.

Amplitude Spectrum:

$$A_k = \frac{1}{N}|X(k)| = \frac{1}{N}\sqrt{(\text{Real}[X(k)])^2 + (\text{Imag}[X(k)])^2}, \quad k = 0, 1, 2, \cdots, N-1$$

To find one-sided amplitude spectrum, we double the amplitude keeping the original ***DC*** term at k =0.

$$\bar{A}_k = \begin{cases} \frac{1}{N}|X(0)|, & k = 0 \\ \frac{2}{N}|X(k)|, & k = 1, \cdots, N/2 \end{cases}$$

Amplitude and Power Spectrum -contd.

Power Spectrum:

$$P_k = \frac{1}{N^2}|X(k)|^2 = \frac{1}{N^2}\left\{(\text{Real}[X(k)])^2 + (\text{Imag}[X(k)])^2\right\},$$
$$k = 0,\ 1,\ 2, \ldots,\ N-1.$$

For, one-sided power spectrum: $\bar{P}_k = \begin{cases} \frac{1}{N^2}|X(0)|^2 & k = 0 \\ \frac{2}{N^2}|X(k)|^2 & k = 0,\ 1, \ldots,\ N/2 \end{cases}$

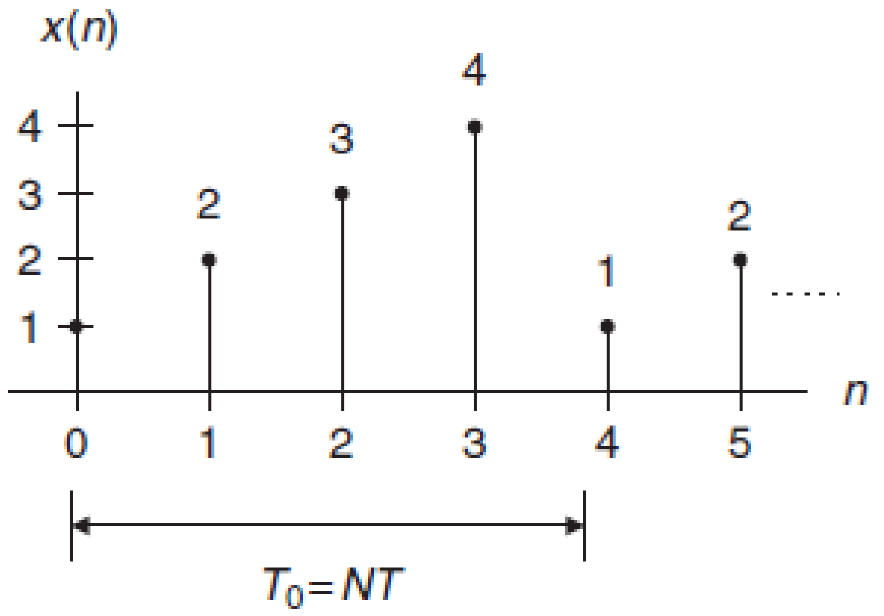
Phase Spectrum:

$$\varphi_k = \tan^{-1}\left(\frac{\text{Imag}[X(k)]}{\text{Real}[X(k)]}\right),\ k = 0,\ 1,\ 2, \ldots,\ N-1.$$

Example 5

Consider the sequence in the Figure, assuming that $f_s = 100\ Hz$, compute the amplitude spectrum, phase spectrum, and power spectrum.

Solution:



$$X(0) = 10$$
$$X(1) = -2 + j2$$
$$X(2) = -2$$
$$X(3) = -2 - j2.$$

See Example 2.

For $k = 0$, $f = k \cdot f_s/N = 0 \times 100/4 = 0\,\text{Hz}$,

$$A_0 = \frac{1}{4}|X(0)| = 2.5,\ \varphi_0 = \tan^{-1}\left(\frac{\text{Imag}[X(0)]}{\text{Real}([X(0)]}\right) = 0^0,$$

$$P_0 = \frac{1}{4^2}|X(0)|^2 = 6.25.$$

Example 5 –contd. (1)

For $k = 1$, $f = 1 \times 100/4 = 25\,\text{Hz}$,

$$A_1 = \frac{1}{4}|X(1)| = 0.7071,\ \varphi_1 = \tan^{-1}\left(\frac{\text{Imag}[X(1)]}{\text{Real}[X(1)]}\right) = 135^0,$$

$$P_1 = \frac{1}{4^2}|X(1)|^2 = 0.5000.$$

For $k = 2$, $f = 2 \times 100/4 = 50\,\text{Hz}$,

$$A_2 = \frac{1}{4}|X(2)| = 0.5,\ \varphi_2 = \tan^{-1}\left(\frac{\text{Imag}[X(2)]}{\text{Real}[X(2)]}\right) = 180^0,$$

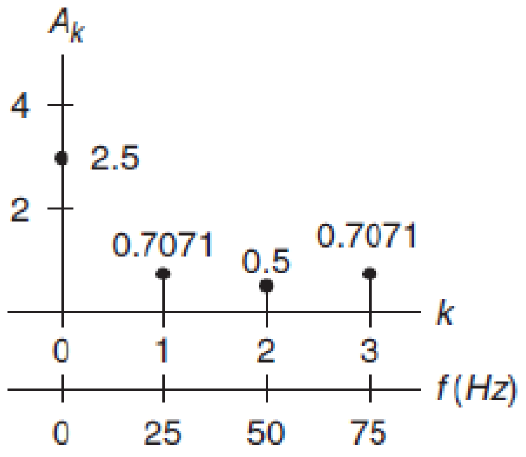
$$P_2 = \frac{1}{4^2}|X(2)|^2 = 0.2500.$$

Similarly, for $k = 3$, $f = 3 \times 100/4 = 75\,\text{Hz}$,

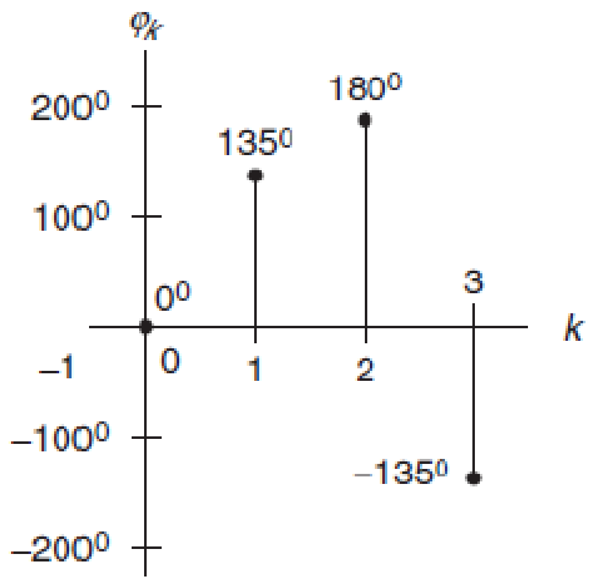
$$A_3 = \frac{1}{4}|X(3)| = 0.7071,\ \varphi_3 = \tan^{-1}\left(\frac{\text{Imag}[X(3)]}{\text{Real}[X(3)]}\right) = -135^0,$$

$$P_3 = \frac{1}{4^2}|X(3)|^2 = 0.5000.$$

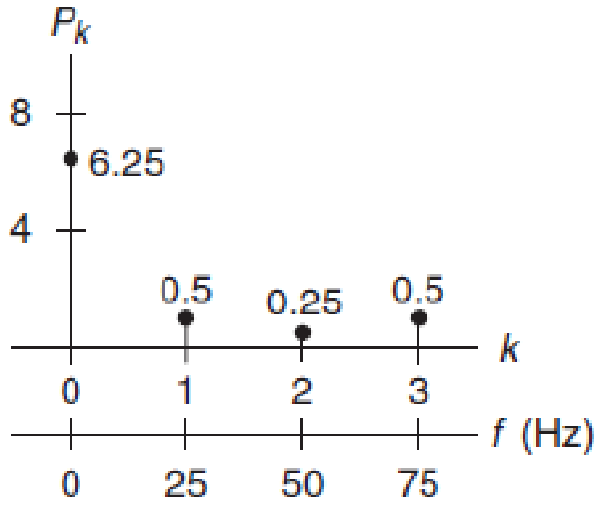
Example 5 -contd. (2)



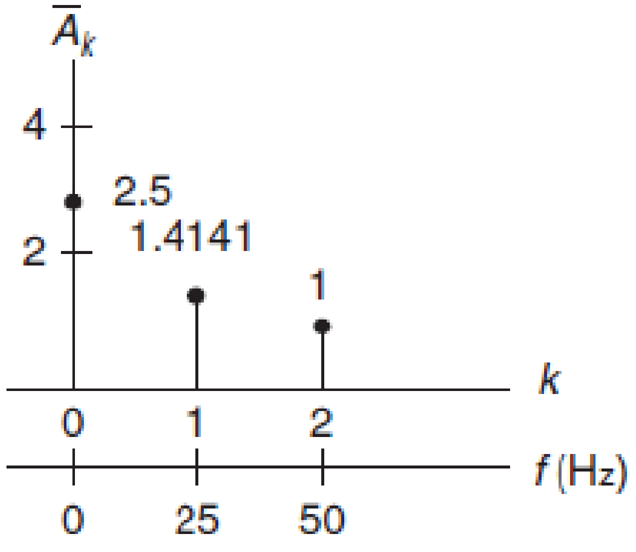
Amplitude Spectrum



Phase Spectrum



Power Spectrum



One sided Amplitude Spectrum

Example 6

Consider a digital sequence sampled at the rate of 10 kHz. If we use 1,024 data points and apply the 1,024-point DFT to compute the spectrum,

a. Determine the frequency resolution;

b. Determine the highest frequency in the spectrum.

Solution:

a. $\Delta f = \frac{f_s}{N} = \frac{10000}{1024} = 9.776 \text{ Hz}$

b. The highest frequency is the folding frequency, given by

$$f_{max} = \frac{N}{2}\Delta f = \frac{f_s}{2}$$

$$= 512 \cdot 9.776 = 5000 \text{ Hz.}$$

Zero Padding for FFT

FFT: Fast Fourier Transform.

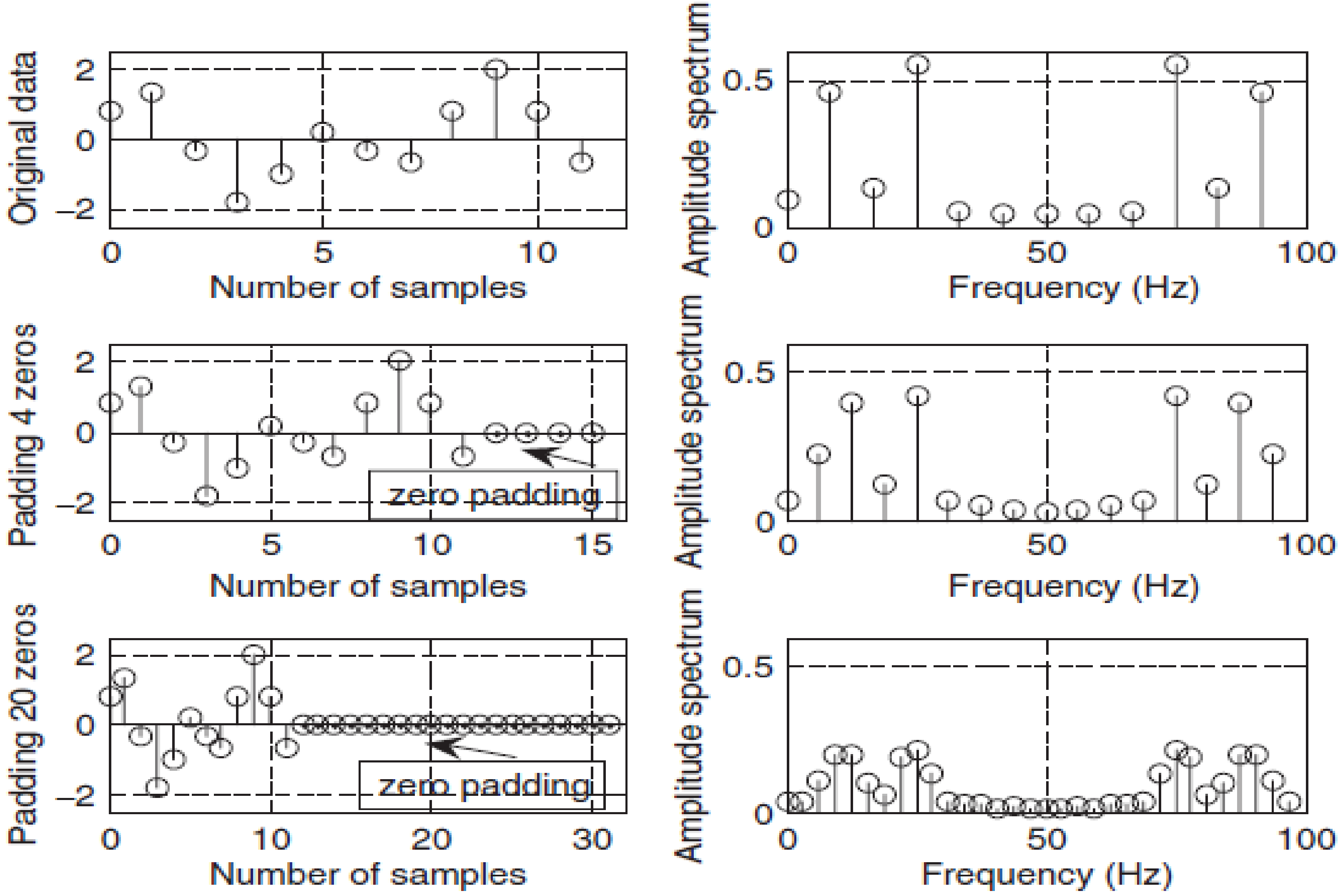
→ A fast version of DFT; It requires signal length to be power of 2 (N = 2, 4, 8, 16, …).

Therefore, we need to pad zero at the end of the signal.

$$\bar{x}(n) = \begin{cases} x(n) & 0 \le n \le N-1 \\ 0 & N \le n \le \bar{N}-1 \end{cases}$$

However, it does not add any new information.

The frequency spacing is reduced due to more DFT points



Example 7

Consider a digital signal has sampling rate = 10 kHz. For amplitude spectrum we need frequency resolution of less than 0.5 Hz. For FFT how many data points are needed?

Solution:

$$\Delta f = 0.5 \text{ Hz} \Rightarrow N = \frac{f_s}{\Delta f} = \frac{10{,}000}{0.5} = 20{,}000$$

For FFT, we need N to be power of 2.

$2^{14} = 16384 < 20000$ And $2^{15} = 32768 > 20000$

Recalculated frequency resolution,

$$\Delta f = \frac{f_s}{N} = \frac{10000}{32768} = 0.31 \text{ Hz.}$$

MATLAB Example -1

$2\pi\, 1000\, nT_s \rightarrow f = 1Khz$

Consider the sinusoid with a sampling rate of $f_s = 8{,}000\ Hz$. $x(n) = 2 \cdot \sin\left(2{,}000\pi \frac{n}{8{,}000}\right)$

Use the MATLAB DFT to compute the signal spectrum with the frequency resolution to be equal to or less than 8 Hz.

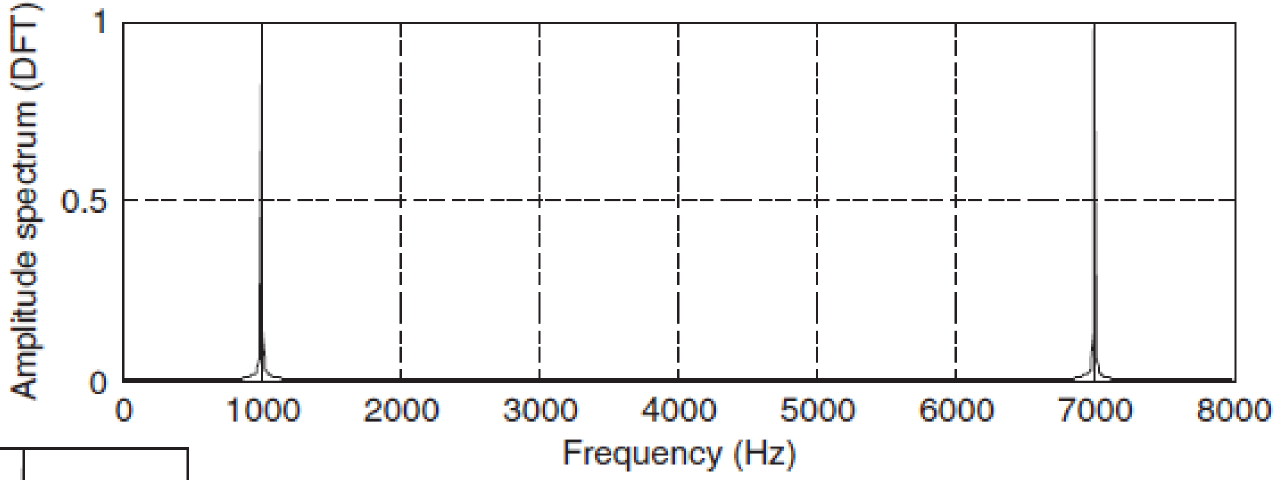
Solution:

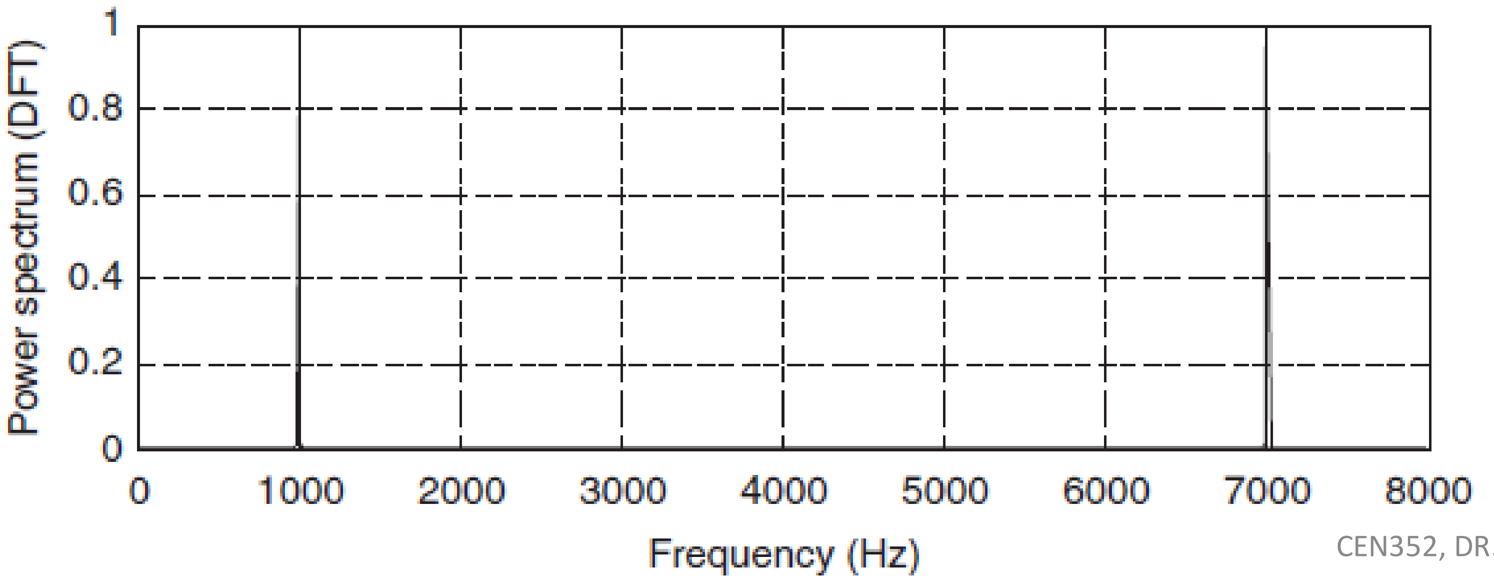
The number of data points is $N = \frac{f_s}{\Delta f} = \frac{8{,}000}{8} = 1{,}000$

```
% Generate the sine wave sequence
fs=8000;                                % Sampling rate
N=1000;                                 % Number of data points
x=2*sin(2000*pi*[0:1:N-1]/fs);

xf=abs(fft(x))/N;              %Compute the amplitude spectrum

P = xf.*xf;          %Compute the power spectrum
f = [0:1:N - 1]*fs/N;          %Map the frequency bin to the frequency (Hz)
```

MATLAB Example –contd. (1)

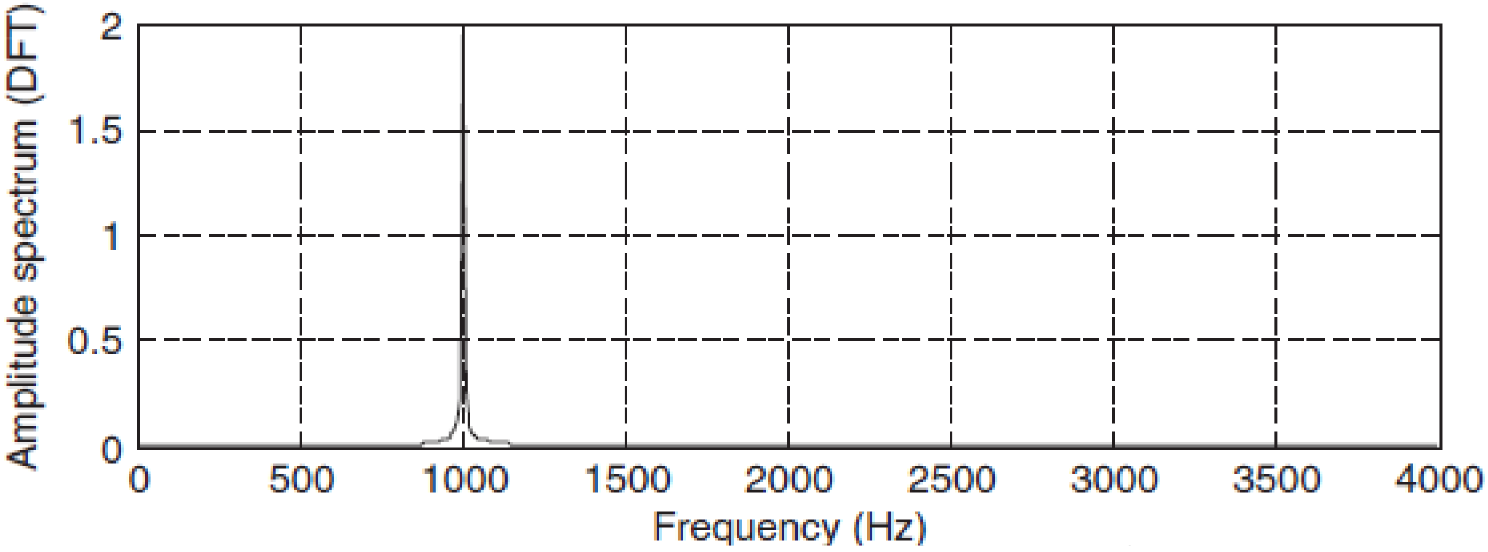
```
subplot(2,1,1); plot(f,xf);grid
xlabel('Frequency (Hz)'); ylabel('Amplitude spectrum (DFT)');
subplot(2,1,2);plot(f,P);grid
xlabel('Frequency (Hz)'); ylabel('Power spectrum (DFT)');
```



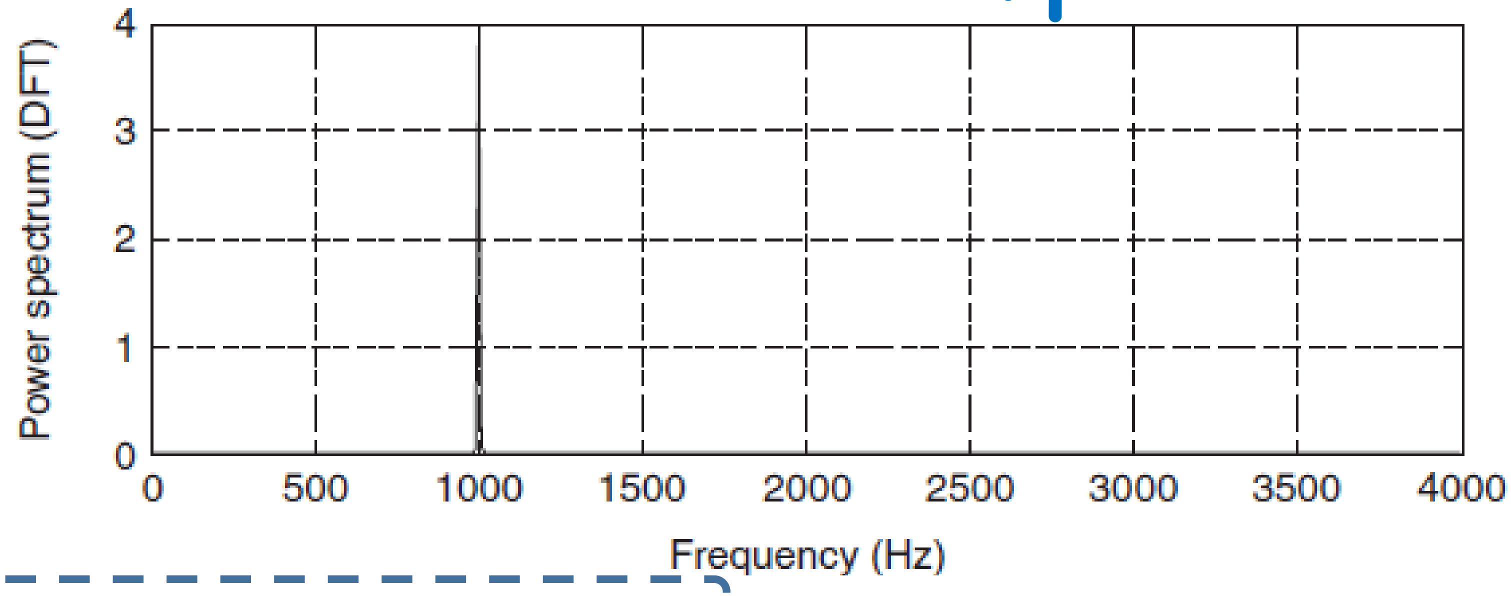


MATLAB Example -contd. (2)

```
% Convert it to one-sided spectrum
xf(2:N) = 2*xf(2:N);            % Get the single-sided spectrum
P = xf.*xf;  % Calculate the power spectrum
f = [0:1:N/2]*fs/N  % Frequencies up to the folding frequency
subplot(2,1,1); plot(f,xf(1:N/2+1));grid
xlabel('Frequency (Hz)'); ylabel('Amplitude spectrum (DFT)');
subplot(2,1,2);plot(f,P(1:N/2+1));grid
xlabel('Frequency (Hz)'); ylabel('Power spectrum (DFT)');
```



MATLAB Example –contd. (3)



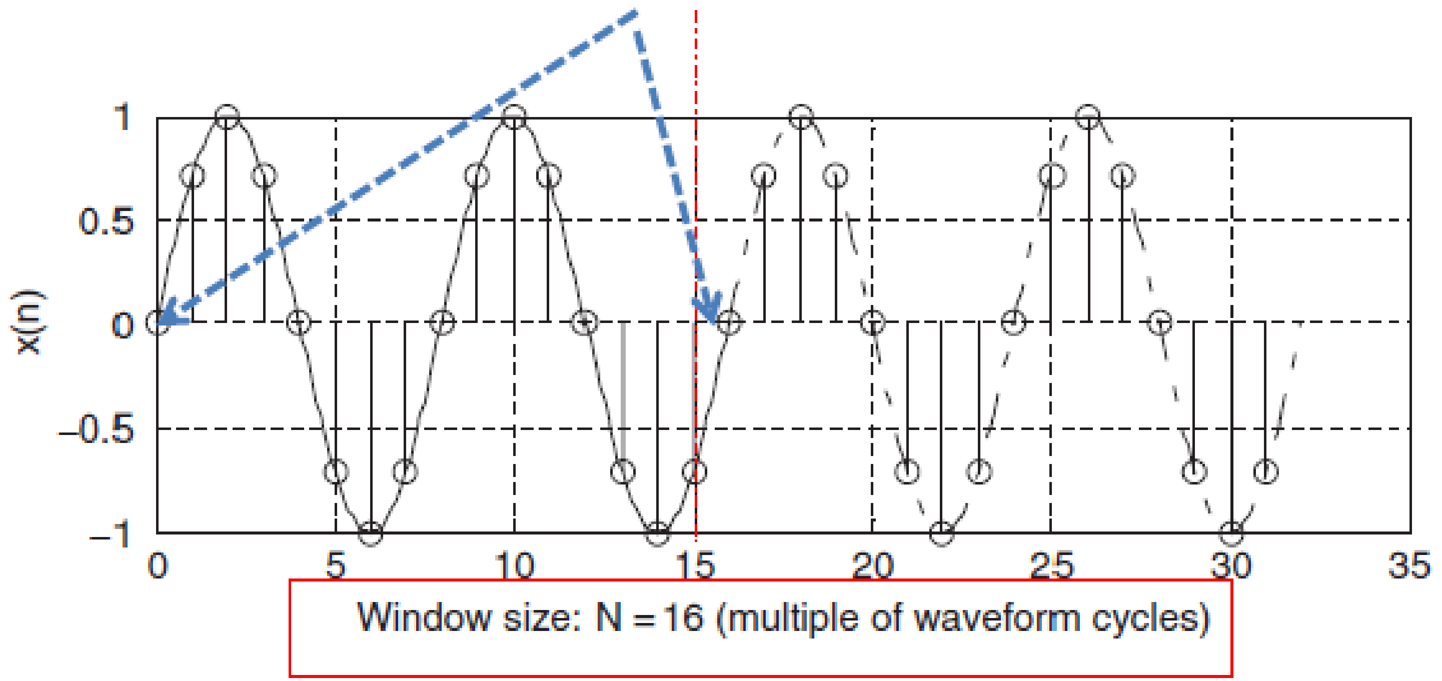

```
% Zero padding to the length of 1024
x = [x, zeros(1,24)];
N = length(x);
xf = abs(fft(x))/N;            %Compute the amplitude spectrum with zero padding
P = xf.*xf;           %Compute the power spectrum
f = [0:1:N - 1]*fs/N;           %Map frequency bin to frequency (Hz)
subplot(2,1,1); plot(f,xf);grid
xlabel('Frequency (Hz)'); ylabel('Amplitude spectrum (FFT)');
subplot(2,1,2);plot(f,P);grid
xlabel('Frequency (Hz)'); ylabel('Power spectrum (FFT)');
```

Effect of Window Size

When applying DFT, we assume the following:

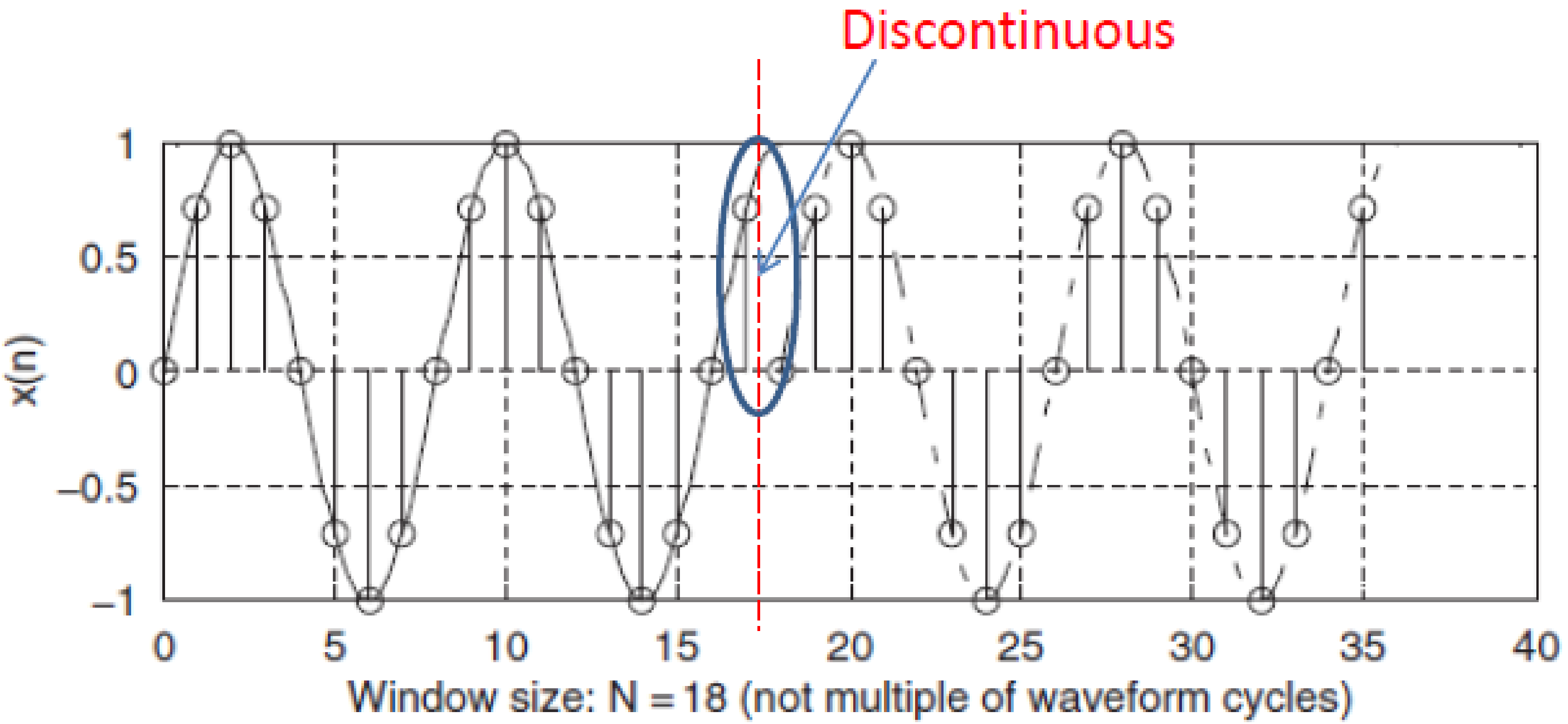
1. Sampled data are periodic to themselves (repeat).
2. Sampled data are continuous to themselves and band limited to the folding frequency.

1 Hz sinusoid, with 32 samples



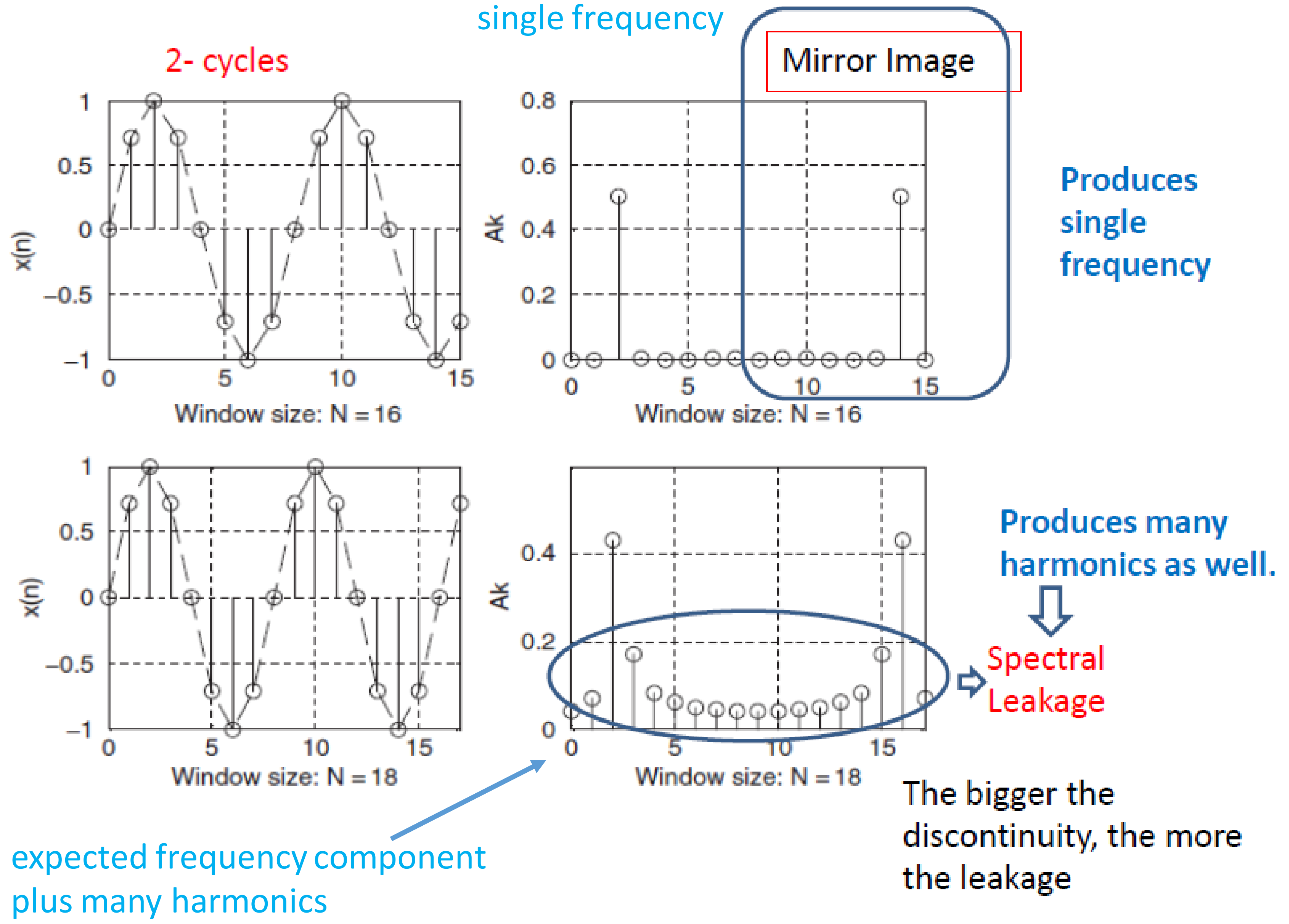
Effect of Window Size -contd. (1)

If the window size is not multiple of waveform cycles, the discontinuity produces undesired harmonic frequencies:



Effect of Window Size -contd. (2)

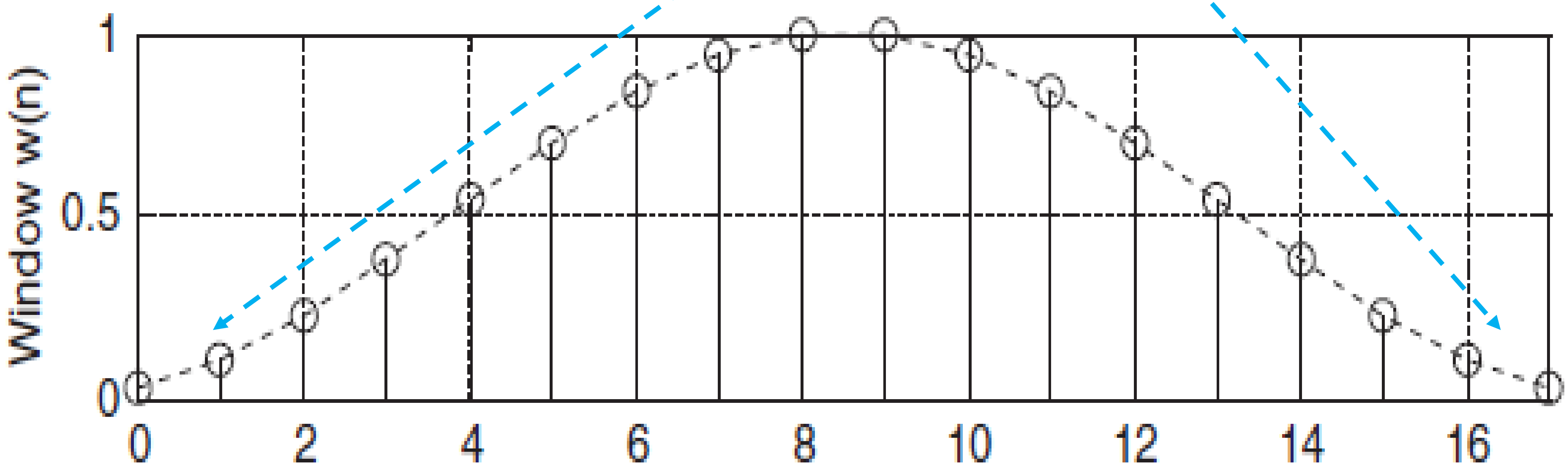
Signal samples and spectra without spectral leakage and with spectral leakage.



The bigger the discontinuity, the more the leakage

Reducing Leakage Using Window

To reduce the effect of spectral leakage, a window function $w(n)$ can be used whose amplitude tapers smoothly and gradually toward zero at both ends



Window function, $w(n)$

Data sequence, $x(n)$

Obtained windowed sequence, $x_w(n)$

$$x_w(n) = x(n)w(n), \quad \text{for } n = 0, 1, \cdots, N-1$$

Example 8

Given,

$$x(2) = 1 \text{ and } w(2) = 0.2265;$$

$$x(5) = -0.7071 \text{ and } w(5) = 0.7008,$$

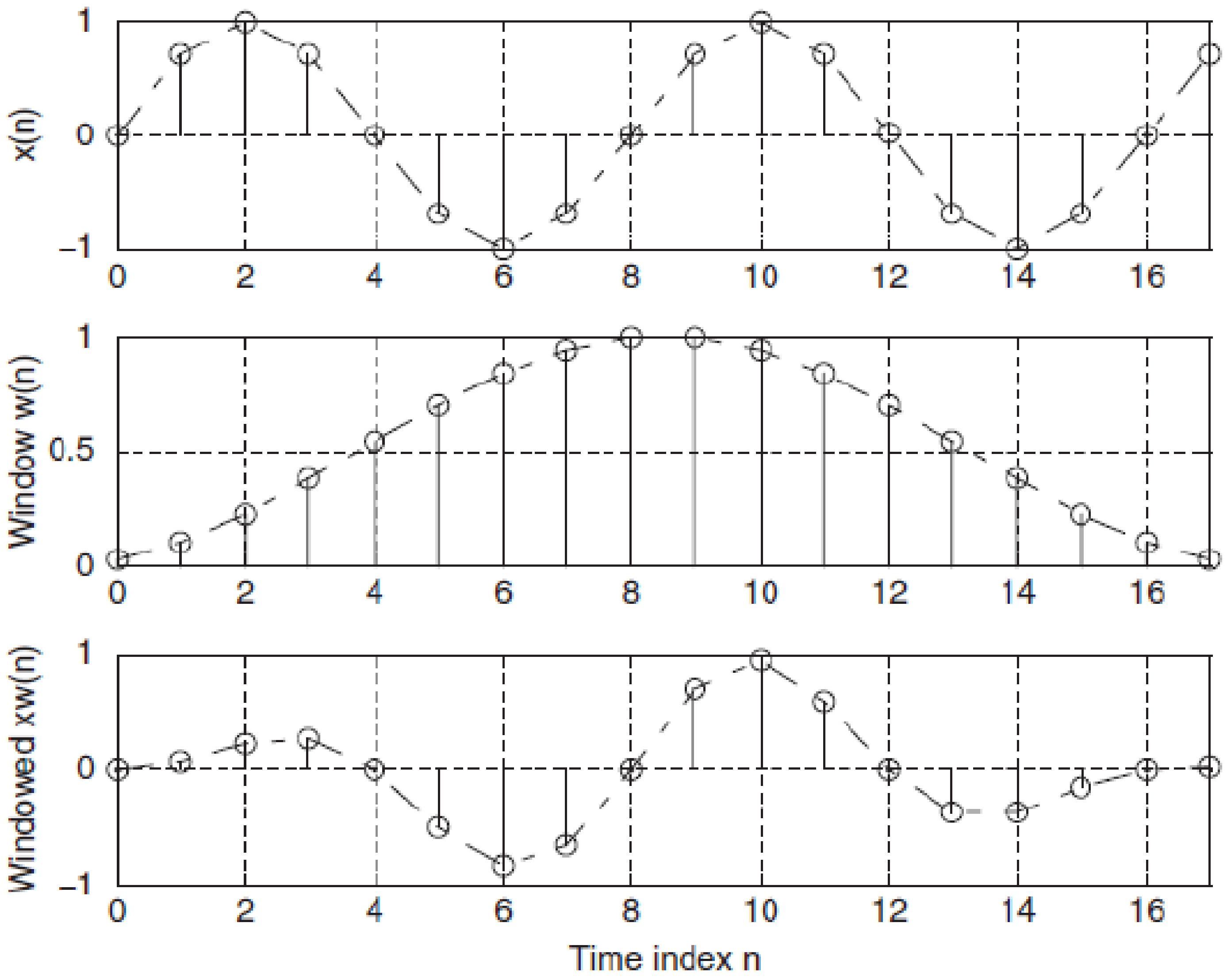
Calculate, windowed sequence data

$$x_w(2) and\ x_w(5)$$

Applying the window function operation leads to

$$x_w(2) = x(2) \times w(2) = 1 \times 0.2265 = 0.2265 \text{ and}$$
$$x_w(5) = x(5) \times w(5) = -0.7071 \times 0.7008 = -0.4956$$

Using the window function the spectral leakage is greatly reduced.



Different Types of Windows

Rectangular Window (no window): $w_R(n) = 1 \qquad 0 \le n \le N-1$

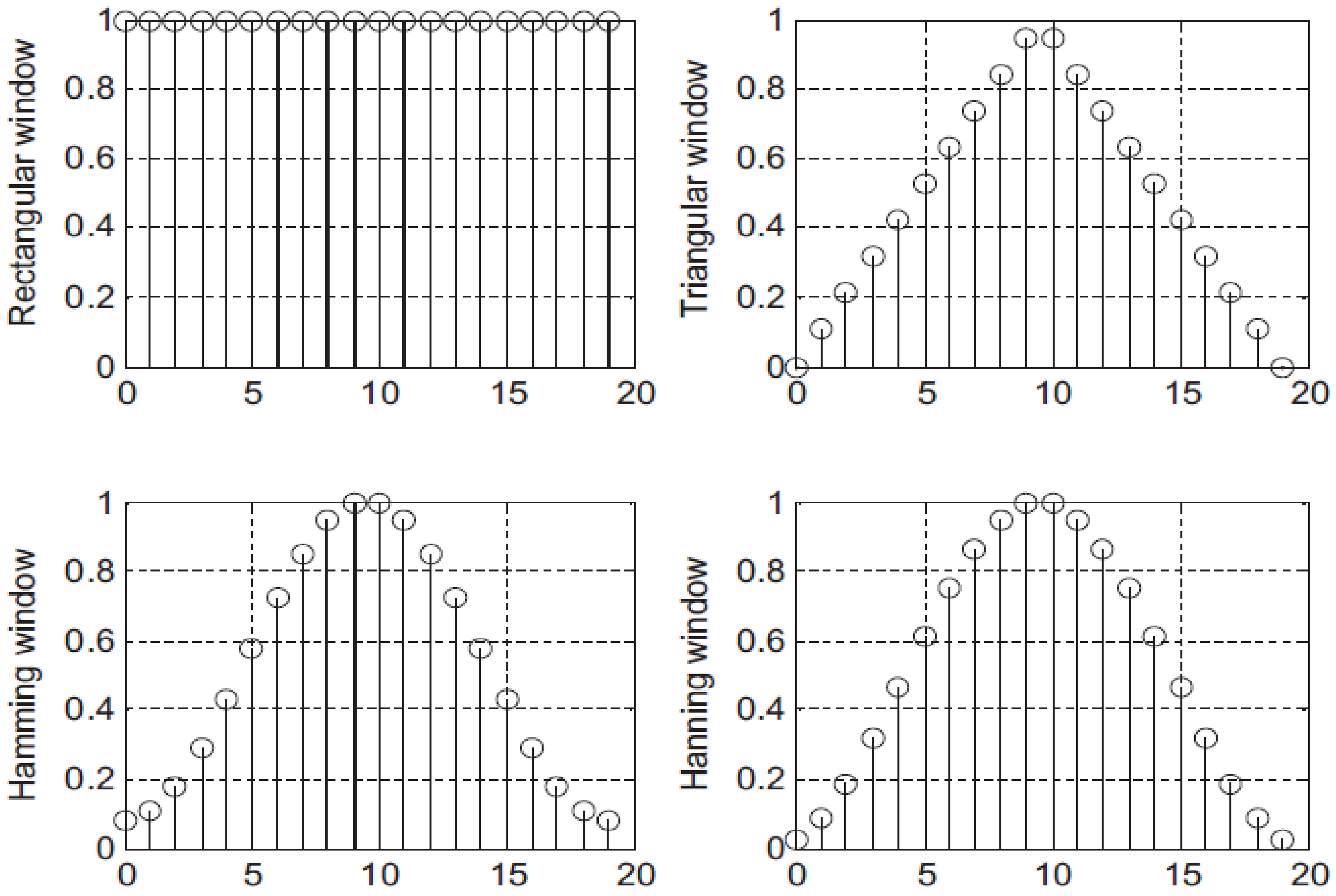
Triangular Window: $w_{tri}(n) = 1 - \frac{|2n - N + 1|}{N-1}, \; 0 \le n \le N-1$

Hamming Window: $w_{hm}(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right), \; 0 \le n \le N-1$

Hanning Window: $w_{hn}(n) = 0.5 - 0.5\cos\left(\frac{2\pi n}{N-1}\right), \; 0 \le n \le N-1$

Different Types of Windows -contd.

Window size of 20 samples



Example 9

Considering the sequence $x(0) = 1,\ x(1) = 2,\ x(2) = 3,\ \ x(3) = 4$ and given $f_s = 100\ \text{Hz}$, $T = 0.01$ seconds, compute the amplitude spectrum, phase spectrum, and power spectrum using the ***Hamming window function***.

Solution:

Since N = 4, Hamming window function can be found as:

$$w_{hm}(0) = 0.54 - 0.46\cos\left(\frac{2\pi \times 0}{4-1}\right) = 0.08$$

$$w_{hm}(1) = 0.54 - 0.46\cos\left(\frac{2\pi \times 1}{4-1}\right) = 0.77.$$

Similarly, $w_{hm}(2) = 0.77$, $w_{hm}(3) = 0.08$.

Example 9 -contd. (1)

- The windowed sequence is computed as:

$$
\begin{aligned}
x_w(0) &= x(0) \times w_{hm}(0) = 1 \times 0.08 = 0.08 \\
x_w(1) &= x(1) \times w_{hm}(1) = 2 \times 0.77 = 1.54 \\
x_w(2) &= x(2) \times w_{hm}(2) = 3 \times 0.77 = 2.31 \\
x_w(0) &= x(3) \times w_{hm}(3) = 4 \times 0.08 = 0.32
\end{aligned}
$$

- DFT Sequence:

$$X(k) = x(0)\, W_N^{k0} + x(1)\, W_N^{k1} + x(2) W_N^{k2} + \ldots + x(N-1) W_N^{k(N-1)}$$

$$\Rightarrow \quad X(k) = x_w(0) W_4^{k\times 0} + x_w(1)\, W_4^{k\times 1} + x_w(2) W_4^{k\times 2} + x_w(3) W_4^{k\times 3}$$

We obtain:

$$
\begin{cases}
X(0) = 4.25 \\
X(1) = -2.23 - j1.22 \\
X(2) = 0.53 \\
X(3) = -2.23 + j1.22
\end{cases}
$$

$$\Delta f = \frac{1}{NT} = \frac{1}{4 \cdot 0.01} = 25 \text{ Hz}$$

Example 9 -contd. (2)

Amplitude spectrum	Power spectrum	Phase spectrum
$A_0 = \frac{1}{4}\|X(0)\| = 1.0625,$	$P_0 = \frac{1}{4^2}\|X(0)\|^2 = 1.1289$	$\phi_0 = \tan^{-1}\left(\frac{0}{4.25}\right) = 0^0,$
$A_1 = \frac{1}{4}\|X(1)\| = 0.6355,$	$P_1 = \frac{1}{4^2}\|X(1)\|^2 = 0.4308$	$\phi_1 = \tan^{-1}\left(\frac{-1.22}{-2.23}\right) = -151.32^0,$
$A_2 = \frac{1}{4}\|X(2)\| = 0.1325,$	$P_2 = \frac{1}{4^2}\|X(2)\|^2 = 0.0176$	$\phi_2 = \tan^{-1}\left(\frac{0}{0.53}\right) = 0^0,$
$A_3 = \frac{1}{4}\|X(3)\| = 0.6355,$	$P_3 = \frac{1}{4^2}\|X(3)\|^2 = 0.4308$	$\phi_3 = \tan^{-1}\left(\frac{1.22}{-2.23}\right) = 151.32^0,$

MATLAB Example -2

Given the sinusoid obtained using a sampling rate of $f_s = 8{,}000\ Hz$

$$x(n) = 2\cdot\sin\left(2{,}000\pi\frac{n}{8{,}000}\right)$$

Use the DFT to compute the spectrum of a Hamming window function with window size = 100.

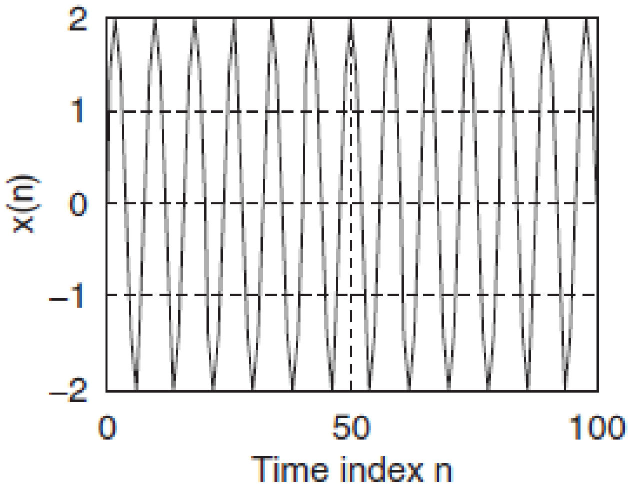
Solution:

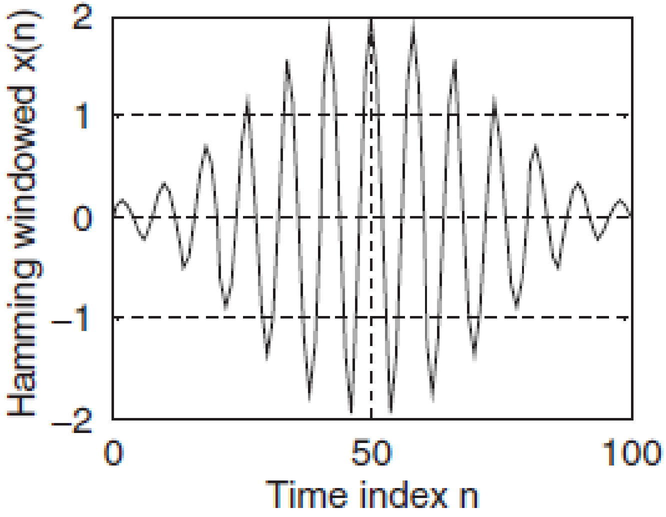
```
% Generate the sine wave sequence
fs = 8000; T = 1/fs;            % Sampling rate and sampling period

% Generate the sine wave sequence
x = 2* sin (2000*pi*[0:1:100]*T);
% Apply the FFT algorithm
N=length(x);
index_t = [0:1:N-1];
f = [0:1:N-1]*fs/N;
xf = abs(fft(x))/N;
```

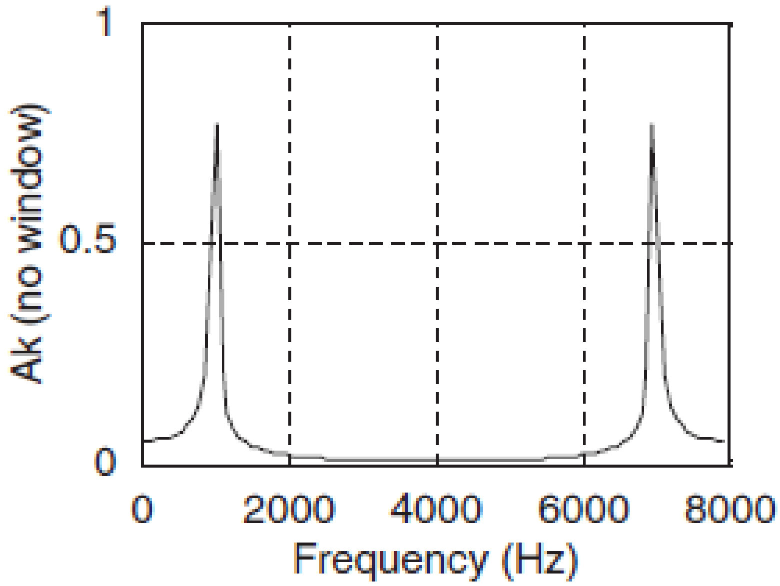
```
%Using the Hamming window
x_hm = x.*hamming(N)';         %Apply the Hamming window function
xf_hm=abs(fft(x_hm))/N;        %Calculate the amplitude spectrum
```

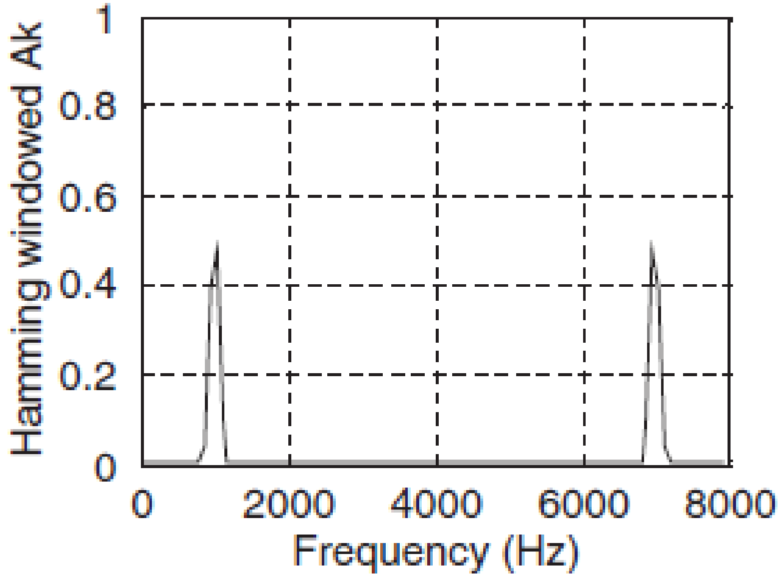
MATLAB Example -2 contd.

```
subplot(2,2,1);plot(index_t,x);grid
xlabel('Time index n'); ylabel('x(n)');
subplot(2,2,3); plot(index_t,x_hm);grid
xlabel('Time index n'); ylabel('Hamming windowed x(n)');
subplot(2,2,2);plot(f,xf);grid;axis([0 fs 0 1]);
xlabel('Frequency (Hz)'); ylabel('Ak (no window)');
subplot(2,2,4); plot(f,xf_hm);grid;axis([0 fs 0 1]);
xlabel('Frequency (Hz)'); ylabel('Hamming windowed Ak');
```









DFT Matrix

- The N equations for the DFT coefficients can be expressed in matrix form as: Let, $w_N = e^{-2j\pi/N}$ then,

Frequency Spectrum DFT Matrix Time-Domain Samples

$$\begin{bmatrix} X[0] \\ X[1] \\ \vdots \\ X[N-1] \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & W_N & \cdots & W_N^{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & W_N^{N-1} & \cdots & W_N^{(N-1)(N-1)} \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix}$$

DFT coefficient vector X_N *DFT matrix* W_N signal vector x_N

Compact form : $X_N = W_N \cdot x_N$

DFT Equation:

$$X(k) = \sum_{m=0}^{N-1} x(m) w_N^{mk} \qquad k = 0, \ldots, N-1$$

DFT requires N^2 complex multiplications

DFT Matrix Example

Determine the DFT coefficients of the four point segment $x[0] = 0, x[1] = 1, x[2] = 2, x[3] = 3$ of a sequence $x[n]$

Solution

We first compute the entries of the matrix W_4 using the property $W_N^{k+N} = W_N^k = e^{-j\frac{2\pi}{N}k} = cos\left(\frac{2\pi}{N}k\right) - jsin\left(\frac{2\pi}{N}k\right)$

The result is a complex matrix given by

$$\mathbf{W}_4 = \begin{bmatrix} W_4^0 & W_4^0 & W_4^0 & W_4^0 \\ W_4^0 & W_4^1 & W_4^2 & W_4^3 \\ W_4^0 & W_4^2 & W_4^4 & W_4^6 \\ W_4^0 & W_4^3 & W_4^6 & W_4^9 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & W_4^1 & W_4^2 & W_4^3 \\ 1 & W_4^2 & W_4^0 & W_4^2 \\ 1 & W_4^3 & W_4^2 & W_4^1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

The DFT coefficients are evaluated by the matrix-by-vector multiplication

$$\begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ X[3] \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \end{bmatrix} = \begin{bmatrix} 6 \\ -2+j2 \\ -2 \\ -2-j2 \end{bmatrix}$$

In MATLAB these computations are done using the commands:

```
x=[0 1 2 3]'; W=dftmtx(4); X=W*x;
```

FFT

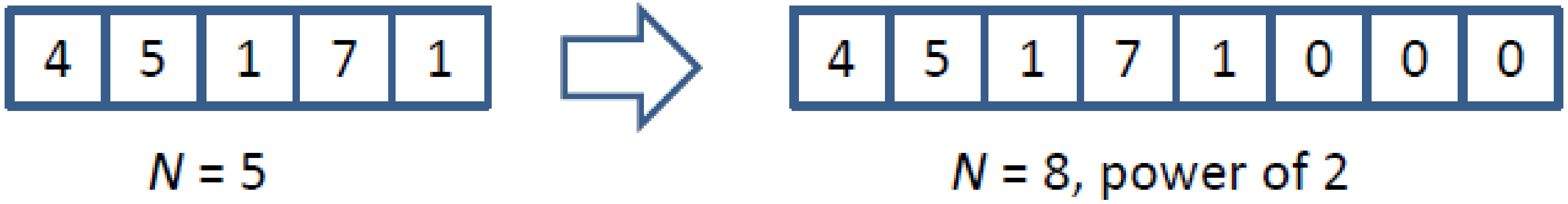
FFT: Fast Fourier Transform

A very efficient algorithm to compute DFT; it requires less multiplication.

- The length of input signal, x(n) must be 2^m samples, where m is an integer.

Samples N= 2, 4, 8, 16 or so.

- If the input length is not 2^m, append (pad) zeros to make it 2^m.



DFT to FFT: Decimation in Frequency

DFT: $X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}$ for $k = 0, 1, \ldots, N-1,$

$w_N = e^{-j2\pi/N}$

twiddle factor

$$X(k) = x(0) + x(1)W_N^k + \ldots + x(N-1)W_N^{k(N-1)}$$

split Equation

$$X(k) = x(0) + x(1)W_N^k + \ldots + x\left(\frac{N}{2}-1\right)W_N^{k(N/2-1)} + x\left(\frac{N}{2}\right)W^{kN/2} + \ldots + x(N-1)W_N^{k(N-1)}$$

$$X(k) = \sum_{n=0}^{(N/2)-1} x(n)W_N^{kn} + \sum_{n=N/2}^{N-1} x(n)W_N^{kn}$$

$n \leftarrow n + \frac{N}{2}$ to sum from zero

$$X(k) = \sum_{n=0}^{(N/2)-1} x(n)W_N^{kn} + W_N^{(N/2)k} \sum_{n=0}^{(N/2)-1} x\left(n+\frac{N}{2}\right)W_N^{kn}$$

$$W_N^{\left(n+\frac{N}{2}\right)k} = W_N^{nk} W_N^{\left(\frac{N}{2}\right)k}$$

$$= W_N^{nk} e^{-\frac{2\pi N}{N\ 2}jk} = W_N^{nk} e^{-j\pi k}$$

$$= W_N^{nk}(-1)^k$$

$$W_N^{N/2} = e^{-j\frac{2\pi(N/2)}{N}} = e^{-j\pi} = -1$$

$$X(k) = \sum_{n=0}^{(N/2)-1} \left(x(n) + (-1)^k x\left(n+\frac{N}{2}\right)\right) W_N^{kn}$$

$for\ k = 2m\ (even) \rightarrow (-1)^k = 1\ compute\ X(2m)$

$for\ k = 2m+1\ (odd) \rightarrow (-1)^k = -1\ compute\ X(2m+1)$

DFT to FFT: Decimation in Frequency

Now decompose into even (k = 2m) and odd (k = 2m+1) sequences.

$$X(2m) = \sum_{n=0}^{(N/2)-1} \left(x(n) + x\left(n + \frac{N}{2}\right) \right) W_N^{2mn}$$

$$X(2m+1) = \sum_{n=0}^{(N/2)-1} \left(x(n) - x\left(n + \frac{N}{2}\right) \right) W_N^{n} W_N^{2mn}$$

$W_N^{(2m+1)n}$

Using the fact that

$$W_N^2 = e^{-j\frac{2\pi \times 2}{N}} = e^{-j\frac{2\pi}{(N/2)}} = W_{N/2},$$

$$X(2m) = \sum_{n=0}^{(N/2)-1} a(n) W_{N/2}^{mn} = DFT\{a(n) with (N/2) \text{points}\}$$

$$X(2m+1) = \sum_{n=0}^{(N/2)-1} b(n) W_N^{n} W_{N/2}^{mn} = DFT\{b(n) W_N^{n} \text{ with } (N/2) \text{points}\}$$

With:

$$a(n) = x(n) + x\left(n + \frac{N}{2}\right), \quad \text{for } n = 0, 1 \cdots, \frac{N}{2} - 1$$

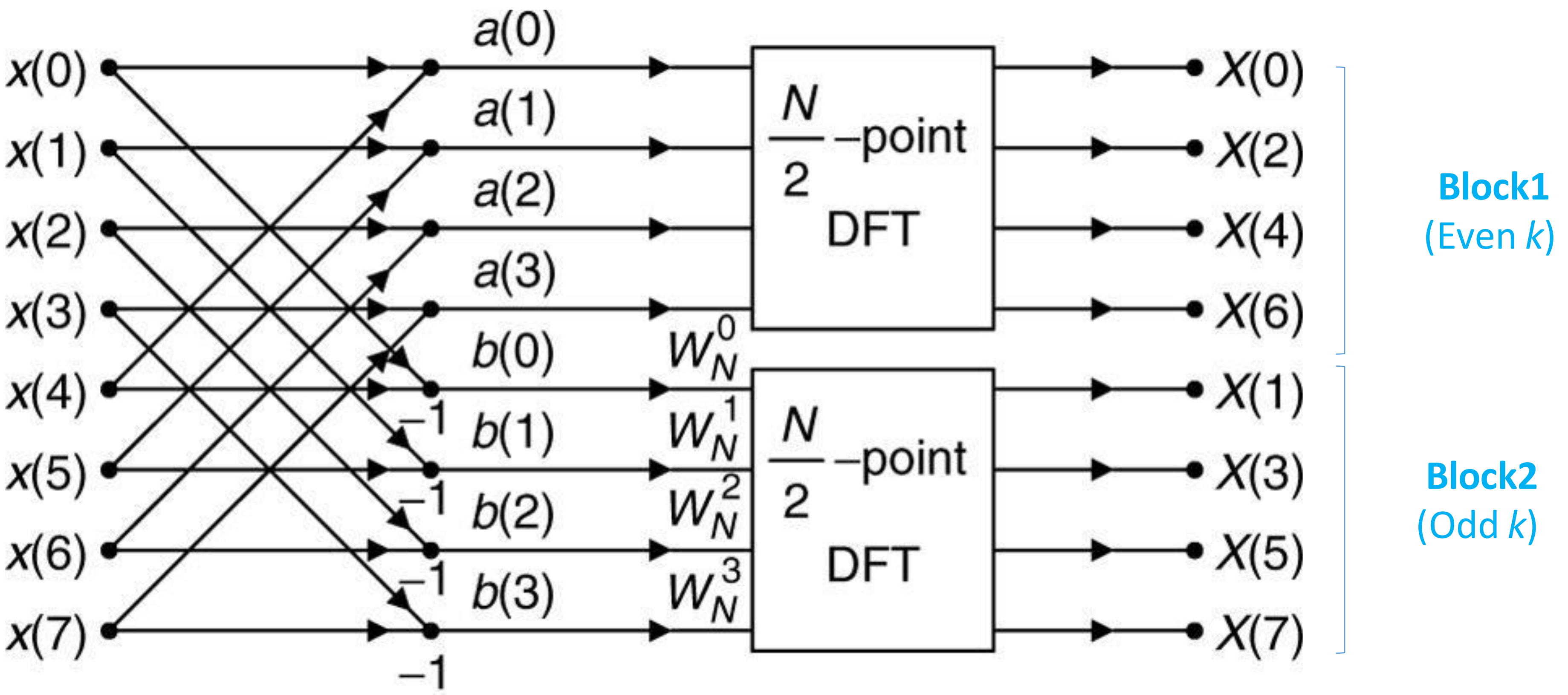
$$b(n) = x(n) - x\left(n + \frac{N}{2}\right), \quad \text{for } n = 0, 1, \cdots, \frac{N}{2} - 1$$

DFT to FFT: Decimation in Frequency

The computation process is

$$DFT\{x(n)\ with\ N\ \text{points}\} = \begin{cases} DFT\{a(n)\ with\ (N/2)\ \text{points}\} \\ DFT\{b(n)\,W_N^n\ with\ (N/2)\ \text{points}\} \end{cases}$$

First iteration



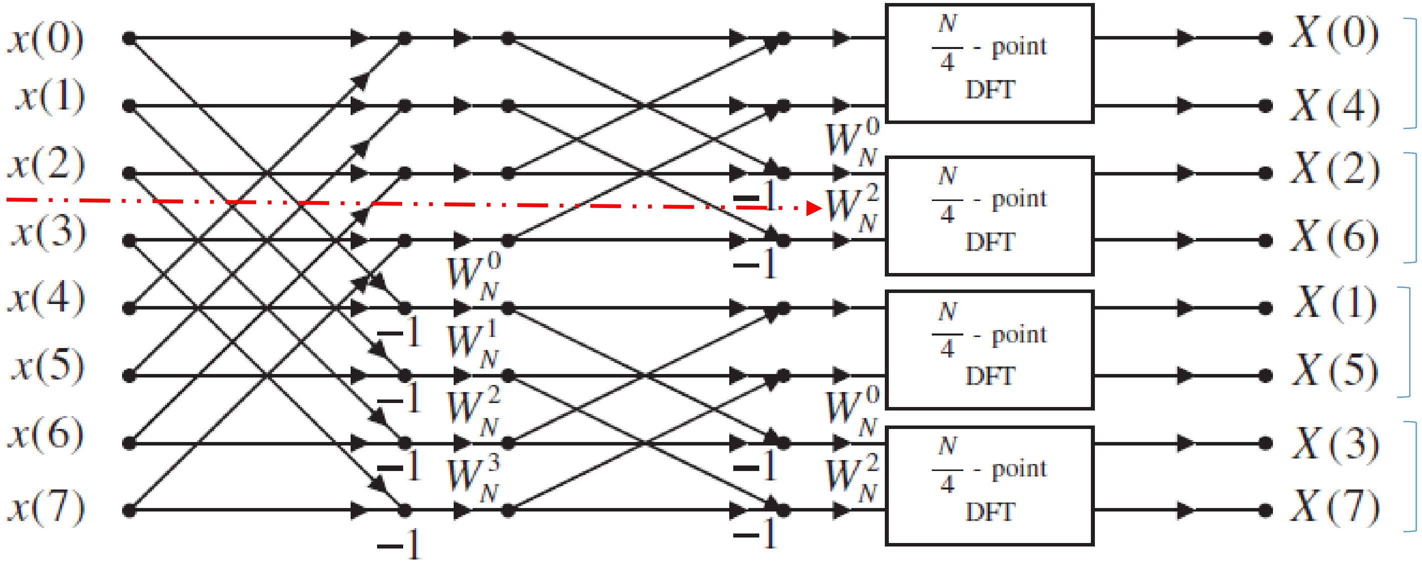
$\frac{N}{2}$ Point DFT ➡ $\sum_{n=0}^{(N/2)-1} x(n)\, W_{N/2}^{mn}$ and $x(n)$ is $a(n)$ for even k and $b(n)W_N^n$ for odd k

DFT to FFT: Decimation in Frequency

Using same process

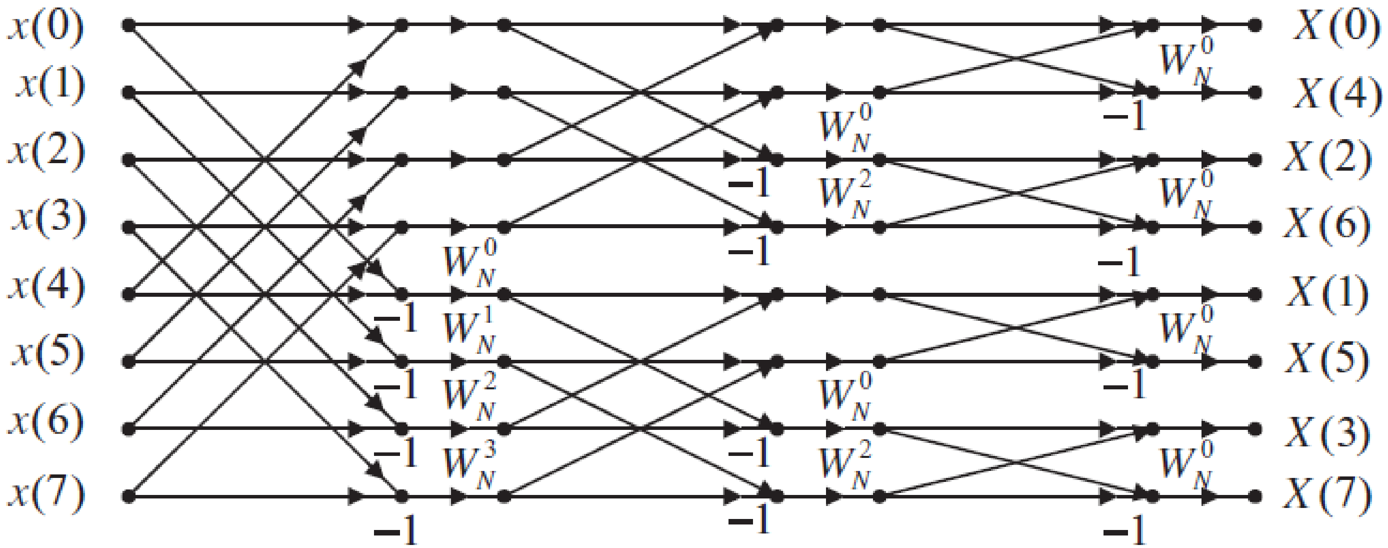
$W_{N/2}^{1} = e^{-j\frac{2\pi}{\left(\frac{N}{2}\right)}(1)} = e^{-j\frac{2\pi}{(N)}(2)} = W_N^2$

Second iteration



The splitting process continues to the end (until having 2 input points to the DFT block, in this case third iteration).

Third iteration



12 complex multiplication

DFT to FFT: Decimation in Frequency

The index (bin number) of the eight-point DFT coefficient becomes inverted, and can be fixed by applying reversal bits.

Binary	index	1st split	2nd split	3rd split	Bit reversal
000	0	0	0	0	000
001	1	2	4	4	100
010	2	4	2	2	010
011	3	6	6	6	110
100	4	1	1	1	001
101	5	3	5	5	101
110	6	5	3	3	011
111	7	7	7	7	111

For data length of N, the number of complex multiplications:

$$\text{Complex multiplications of DFT} = N^2,$$

For each k (N) we need N multiplications

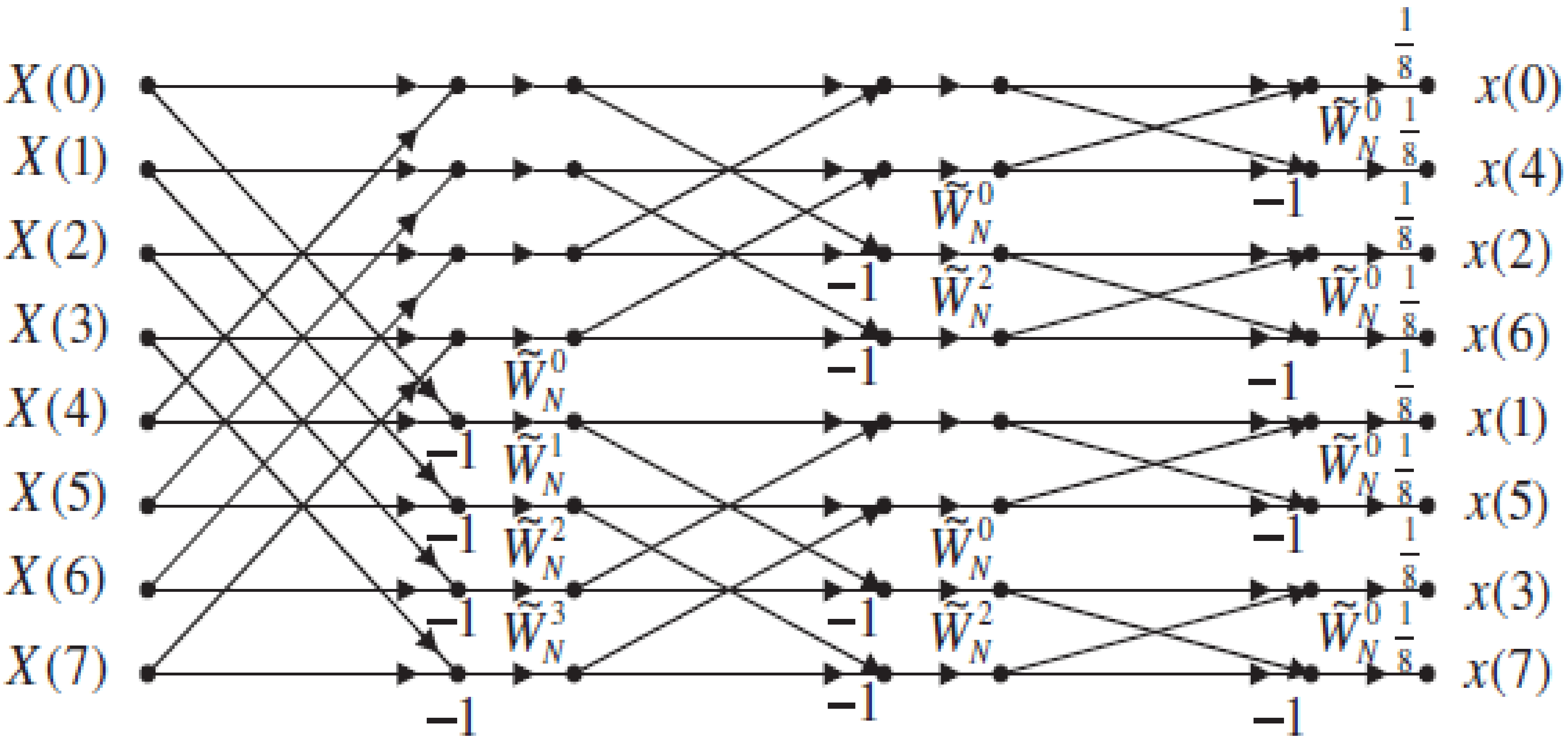
$$\text{Complex multiplications of FFT} = \frac{N}{2}\log_2(N)$$

For 1024 samples data sequence, DFT requires 1024×1024 = 1048576 complex multiplications. FFT requires (1024/2)log(1024) = 5120 complex multiplications.

IFFT: Inverse FFT

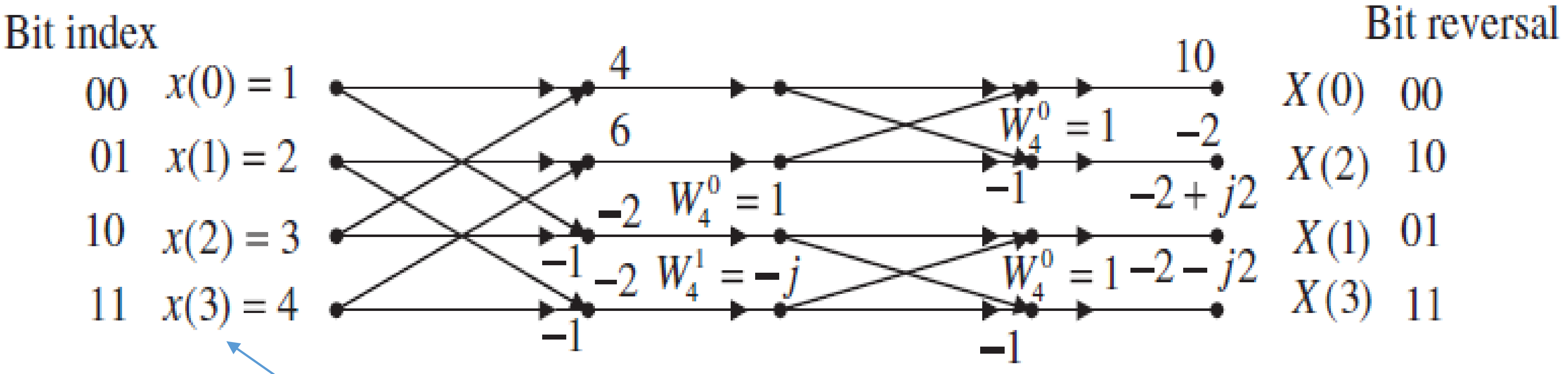
$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)W_N^{-kn} = \frac{1}{N}\sum_{k=0}^{N-1} X(k)\tilde{W}_N^{kn}, \quad \text{for } k = 0, 1, \cdots, N-1$$

The difference is: the twiddle factor w_N is changed to $\widetilde{w}_N = w_N^{-1}$, and the sum is multiplied by a factor of 1/N.



FFT and IFFT Examples

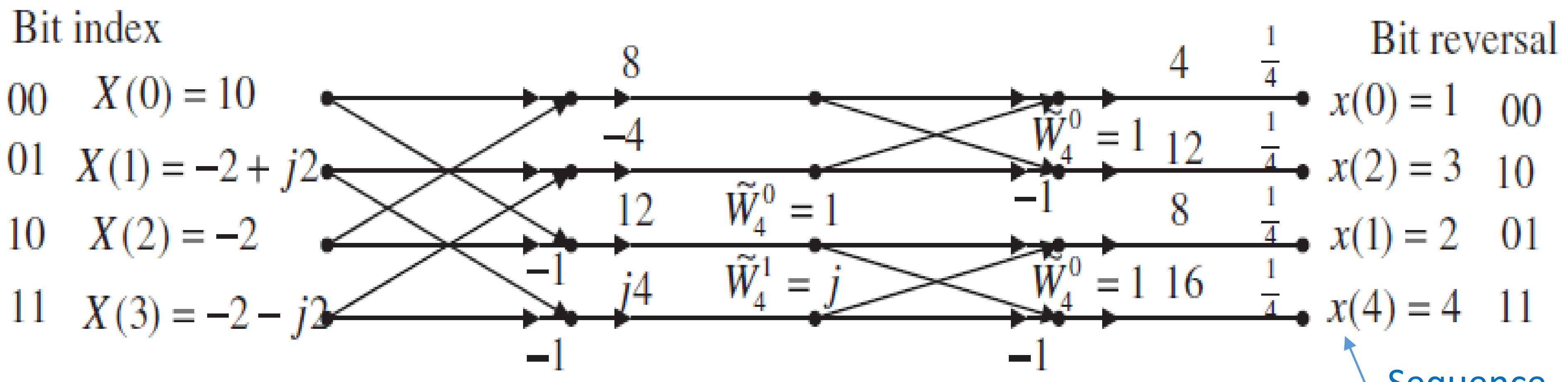
FFT



Sequence $x(n)$

Number of complex multiplication = $\frac{N}{2}\log_2(N) = \frac{4}{2}\log_2(4) = 4.$

IFFT



Sequence $x(n)$

DFT to FFT: Decimation in Time

Split the input sequence $x(n)$ into the even indexed $x(2m)$ and $x(2m+1)$ each with N/2 data points.

$$X(k) = \sum_{m=0}^{(N/2)-1} x(2m) W_N^{2mk} + \sum_{m=0}^{(N/2)-1} x(2m+1) W_N^k W_N^{2mk}, \quad \text{for } k = 0, 1, \cdots, N-1$$

Using $$w_N^2 = \left(e^{-j2\pi/N}\right)^2 = e^{-j2\pi/(N/2)} = w_{N/2}$$

$$X(k) = \sum_{m=0}^{(N/2)-1} x(2m) W_{N/2}^{mk} + W_N^k \sum_{m=0}^{(N/2)-1} x(2m+1) W_{N/2}^{mk}, \quad \text{for } k = 0, 1, \cdots, N-1$$

DFT to FFT: Decimation in Time

Define new functions as

$$G(k) = \sum_{m=0}^{(N/2)-1} x(2m)W_{N/2}^{mk} = DFT\{x(2m)\ with\ (N/2)\ \text{points}\}$$

$$H(k) = \sum_{m=0}^{(N/2)-1} x(2m+1)W_{N/2}^{mk} = DFT\{x(2m+1)\ with\ (N/2)\ \text{points}\}$$

As,

$$G(k) = G\left(k+\frac{N}{2}\right), \quad \text{for } k = 0, 1, \cdots, \frac{N}{2}-1$$

$$H(k) = H\left(k+\frac{N}{2}\right), \quad \text{for } k = 0, 1, \cdots, \frac{N}{2}-1$$

$$W_{N/2}^{k+\frac{N}{2}} = e^{-j\frac{2\pi}{\left(\frac{N}{2}\right)}\left(k+\frac{N}{2}\right)} = e^{-j\frac{2\pi}{\left(\frac{N}{2}\right)}k} e^{-j\frac{2\pi}{\left(\frac{N}{2}\right)}\left(\frac{N}{2}\right)} = e^{-j\frac{2\pi}{\left(\frac{N}{2}\right)}k} e^{-j2\pi} = W_{N/2}^{k}$$

$$X(k) = \sum_{m=0}^{(N/2)-1} x(2m)W_{N/2}^{mk} + W_N^k \sum_{m=0}^{(N/2)-1} x(2m+1)W_{N/2}^{mk}, \quad \text{for } k = 0, 1, \cdots, N-1$$

$$X(k) = G(k) + W_N^k H(k), \quad \text{for } k = 0, 1, \cdots, \frac{N}{2}-1$$

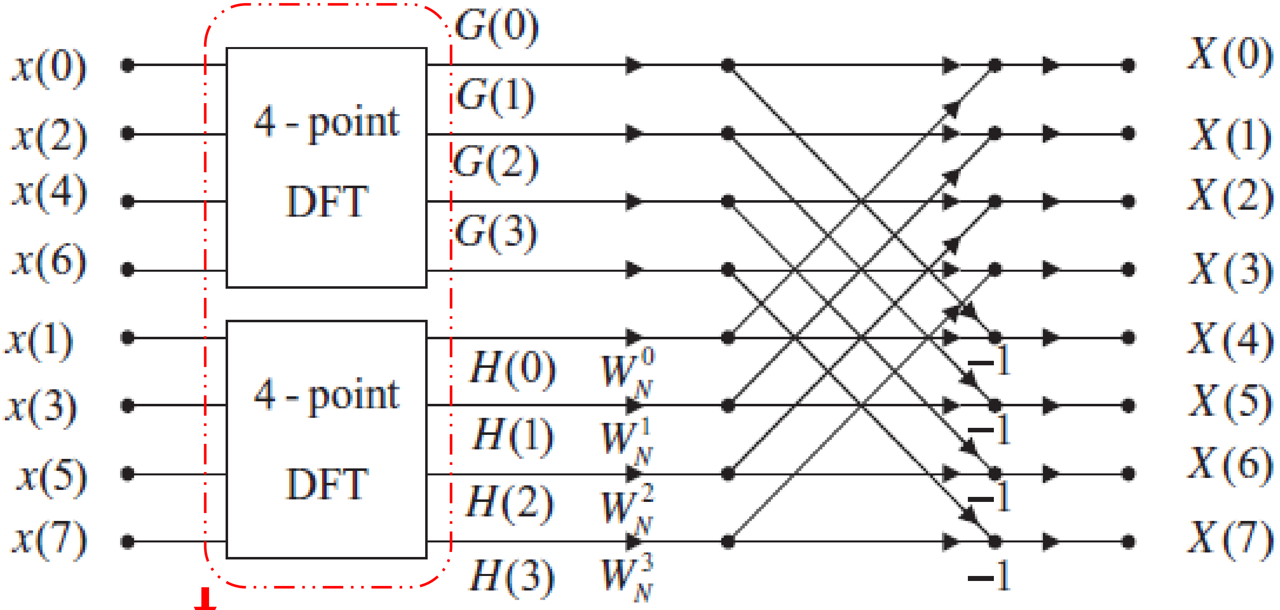
$$X\left(\frac{N}{2}+k\right) = G(k) - W_N^k H(k), \quad \text{for } k = 0, 1, \cdots, \frac{N}{2}-1$$

$$W_N^{k+\frac{N}{2}} = e^{-j\frac{2\pi}{(N)}\left(k+\frac{N}{2}\right)} = e^{-j\frac{2\pi}{(N)}k} e^{-j\frac{2\pi}{(N)}\left(\frac{N}{2}\right)} = e^{-j\frac{2\pi}{(N)}k} e^{-j\pi} = -W_N^k$$

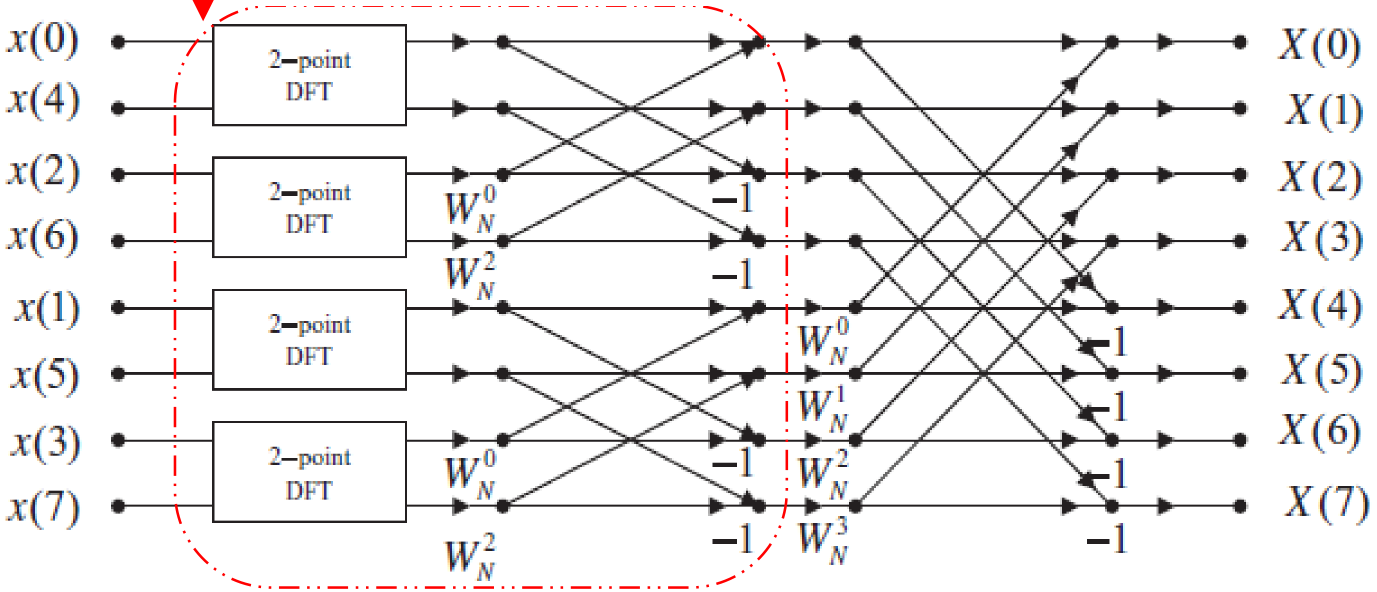
$$W_N^{(N/2+k)} = -W_N^k$$

DFT to FFT: Decimation in Time

First iteration:

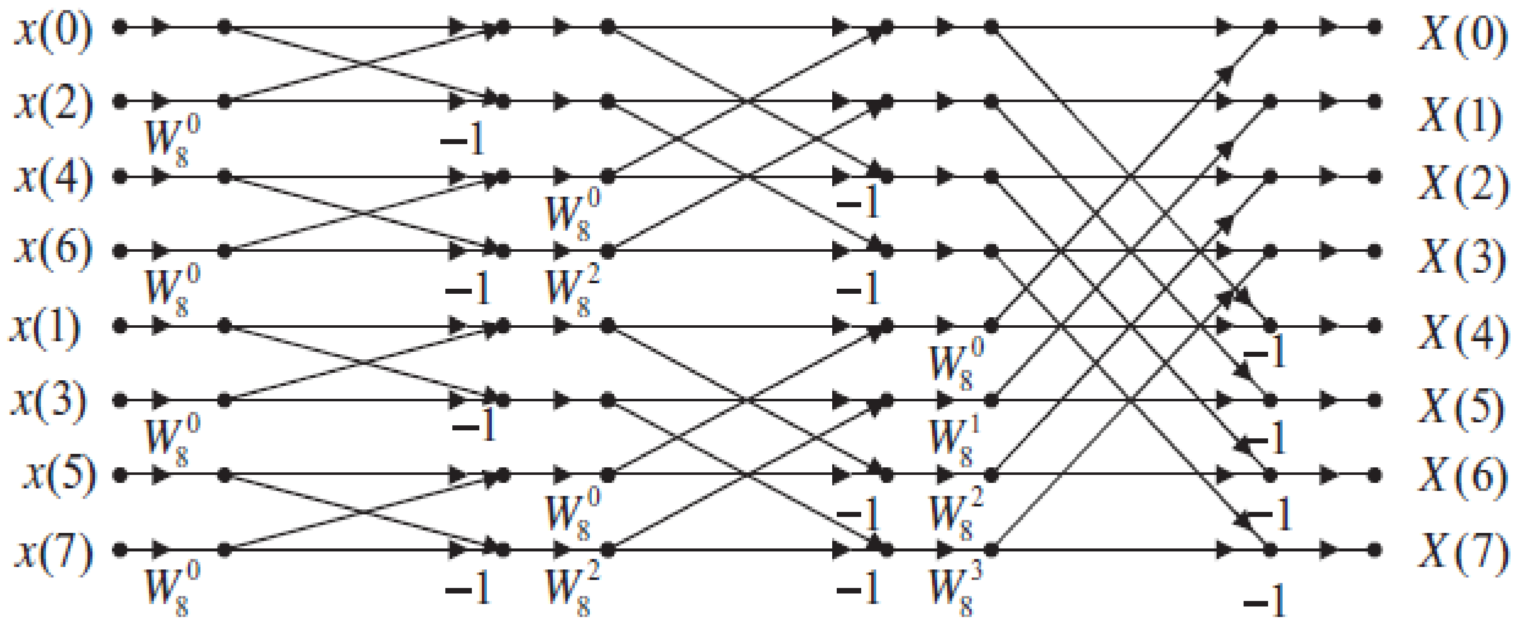


Second iteration:



DFT to FFT: Decimation in Time

Third iteration:



$$W_N = e^{-\frac{2\pi}{N}} = \cos\left(\frac{2\pi}{N}\right) - j\sin\left(\frac{2\pi}{N}\right)$$

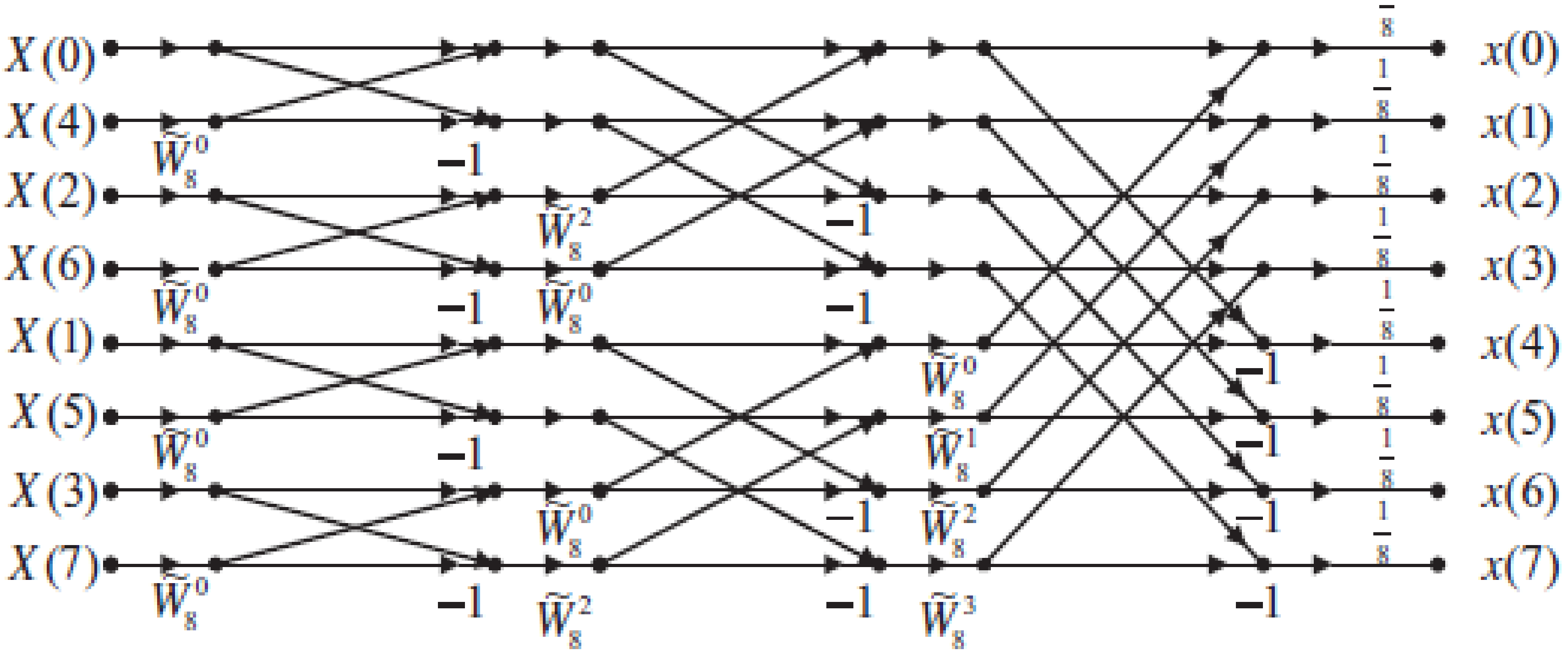
$$W_8^2 = e^{-\frac{2\pi\times 2}{8}} = e^{-\frac{\pi}{2}} = \cos(\pi/2) - j\sin(\pi/2) = -j$$

IFFT: Decimation in Time

Similar to the decimation-in-frequency method, we change W_N to $\widetilde{W}_N$, and the sum is multiplied by a factor of 1/N.

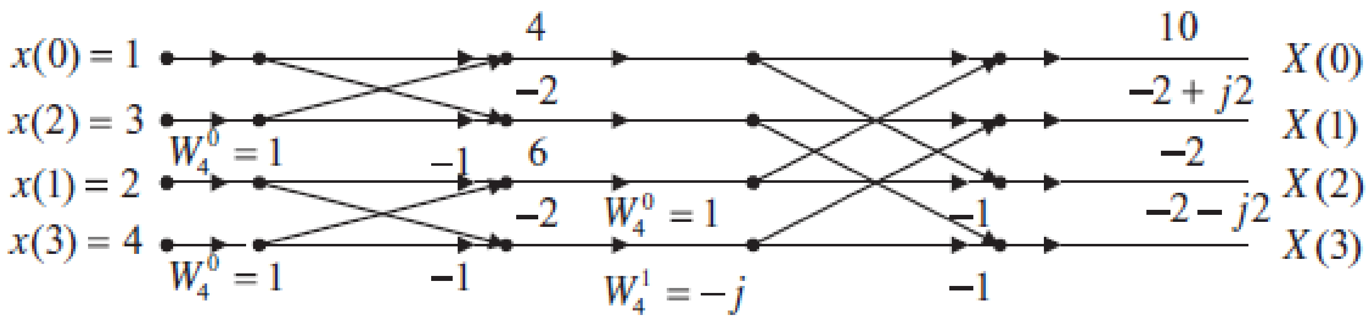
inverse FFT (IFFT) block diagram for the eight-point inverse FFT

IFFT

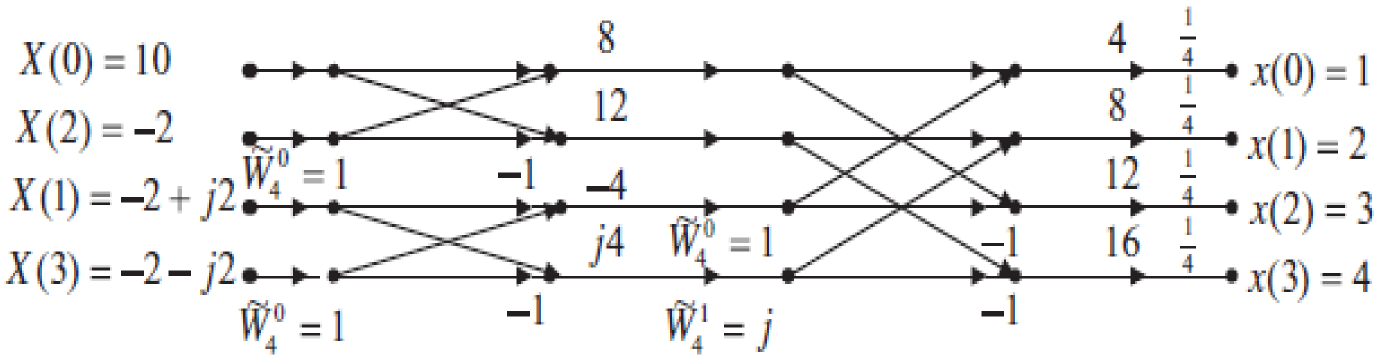


FFT and IFFT Examples

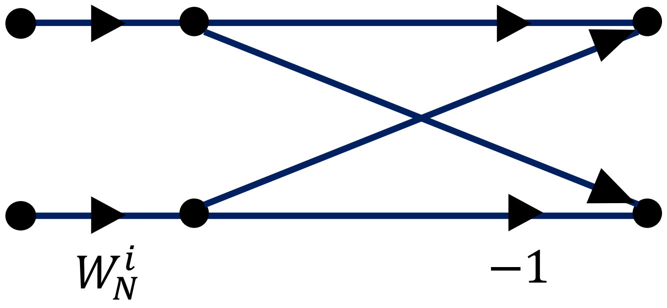
FFT

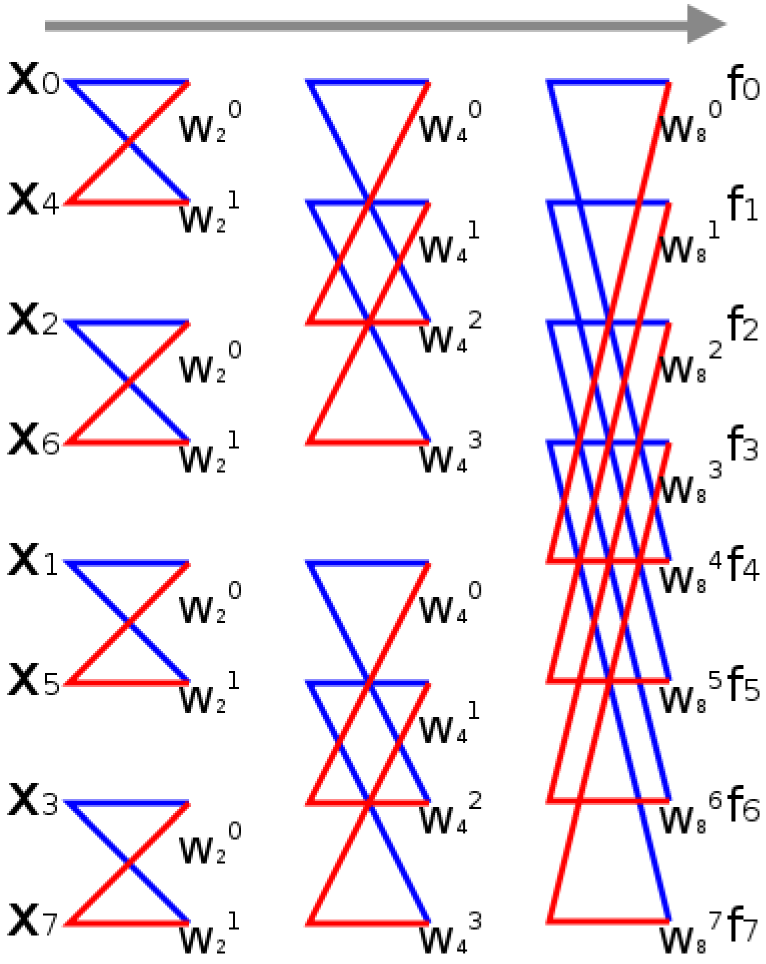


IFFT

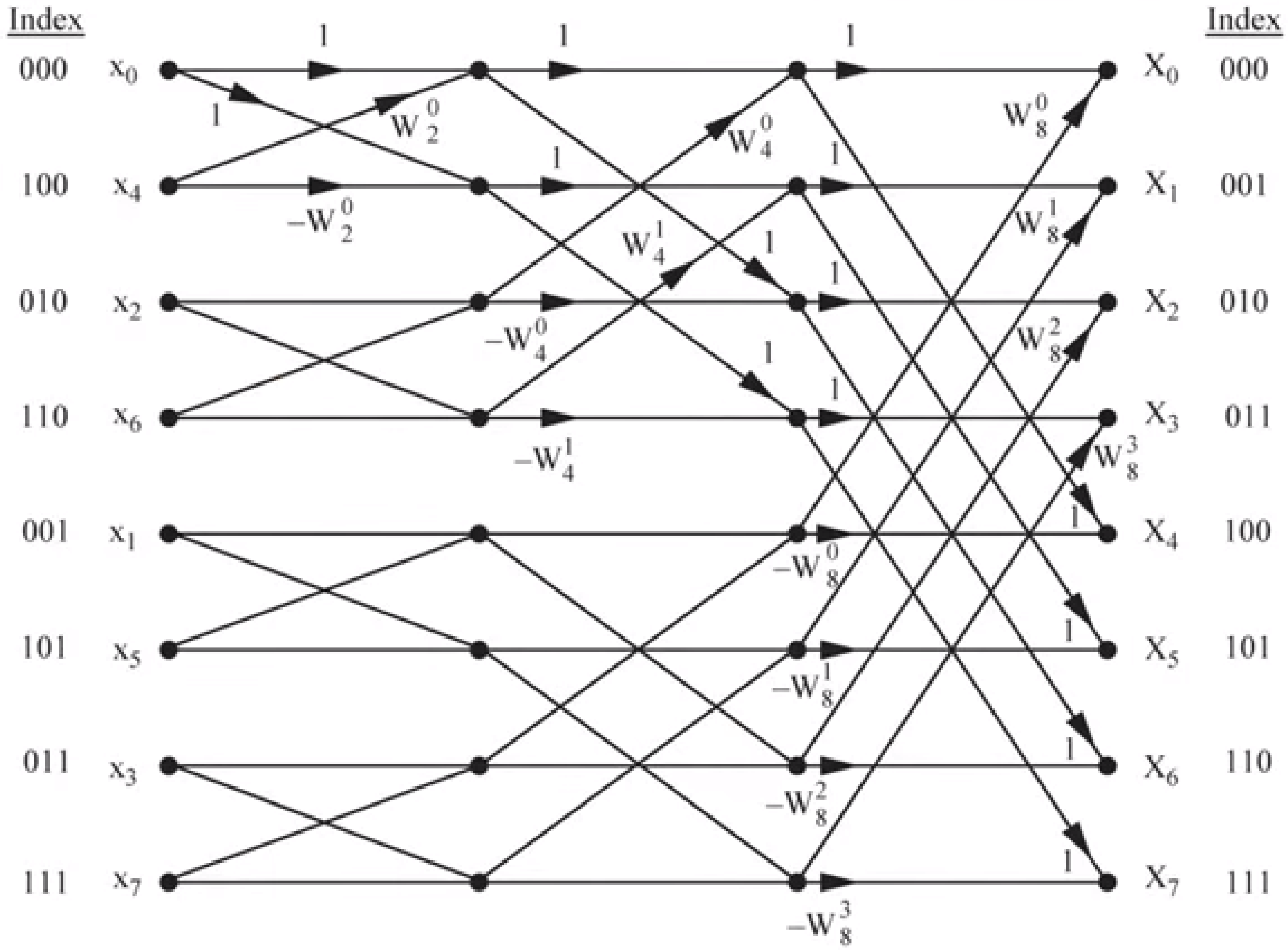


FFT Butterfly

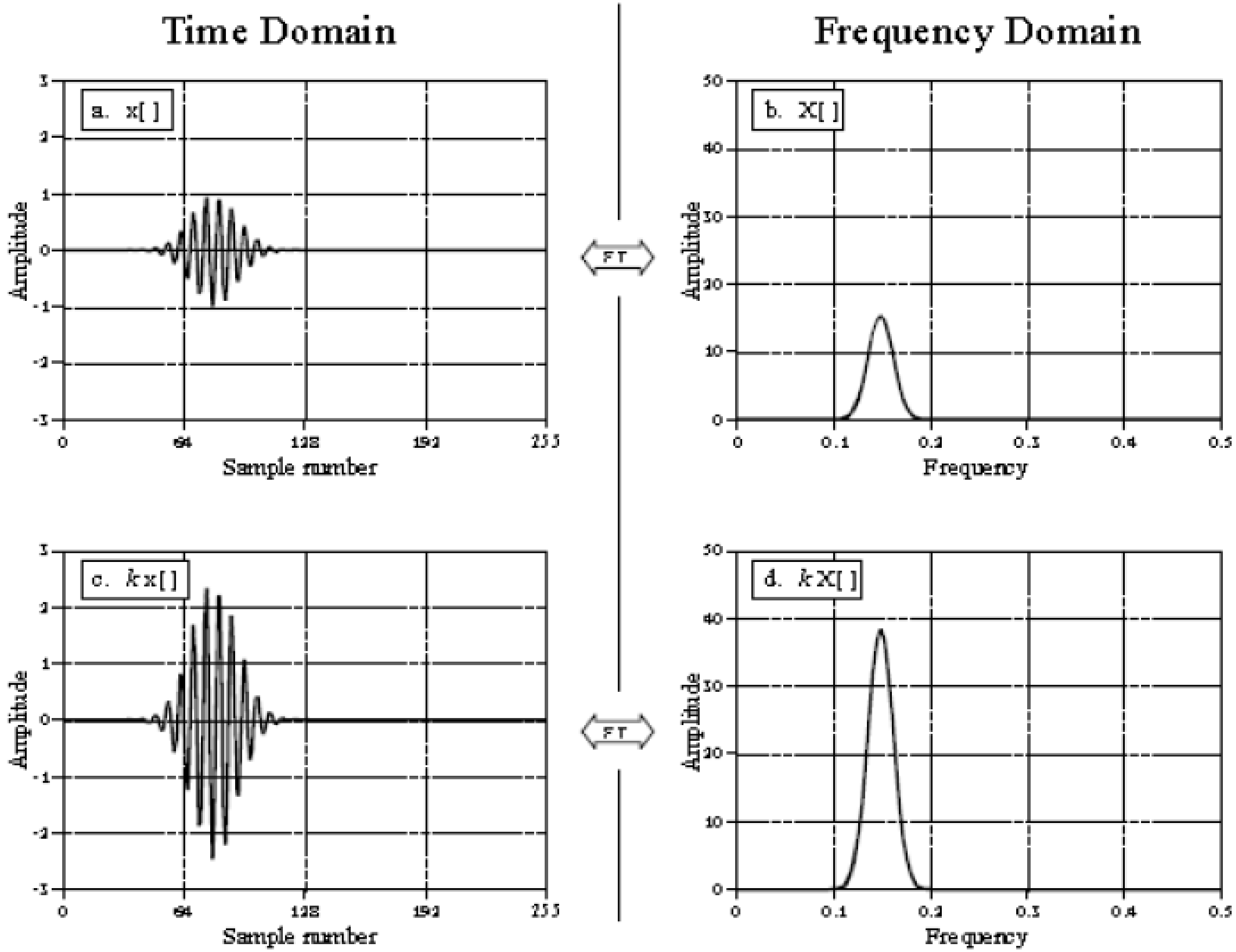




Stage 1 W has base N, stage 2 has base $\frac{N}{2}$, $\frac{N}{4}$, divide by 2 as you add stages.


Index
000
100
010
110
001
101
011
111
x_0
x_4
x_2
x_6
x_1
x_5
x_3
x_7
1
W_2^0
$-W_2^0$
W_4^0
W_4^1
$-W_4^0$
$-W_4^1$
W_8^0
W_8^1
W_8^2
W_8^3
$-W_8^0$
$-W_8^1$
$-W_8^2$
$-W_8^3$
X_0
X_1
X_2
X_3
X_4
X_5
X_6
X_7
Index
000
001
010
011
100
101
110
111

Fourier Transform Properties (1)



FT is linear:

- **Homogeneity**
- **Additivity**

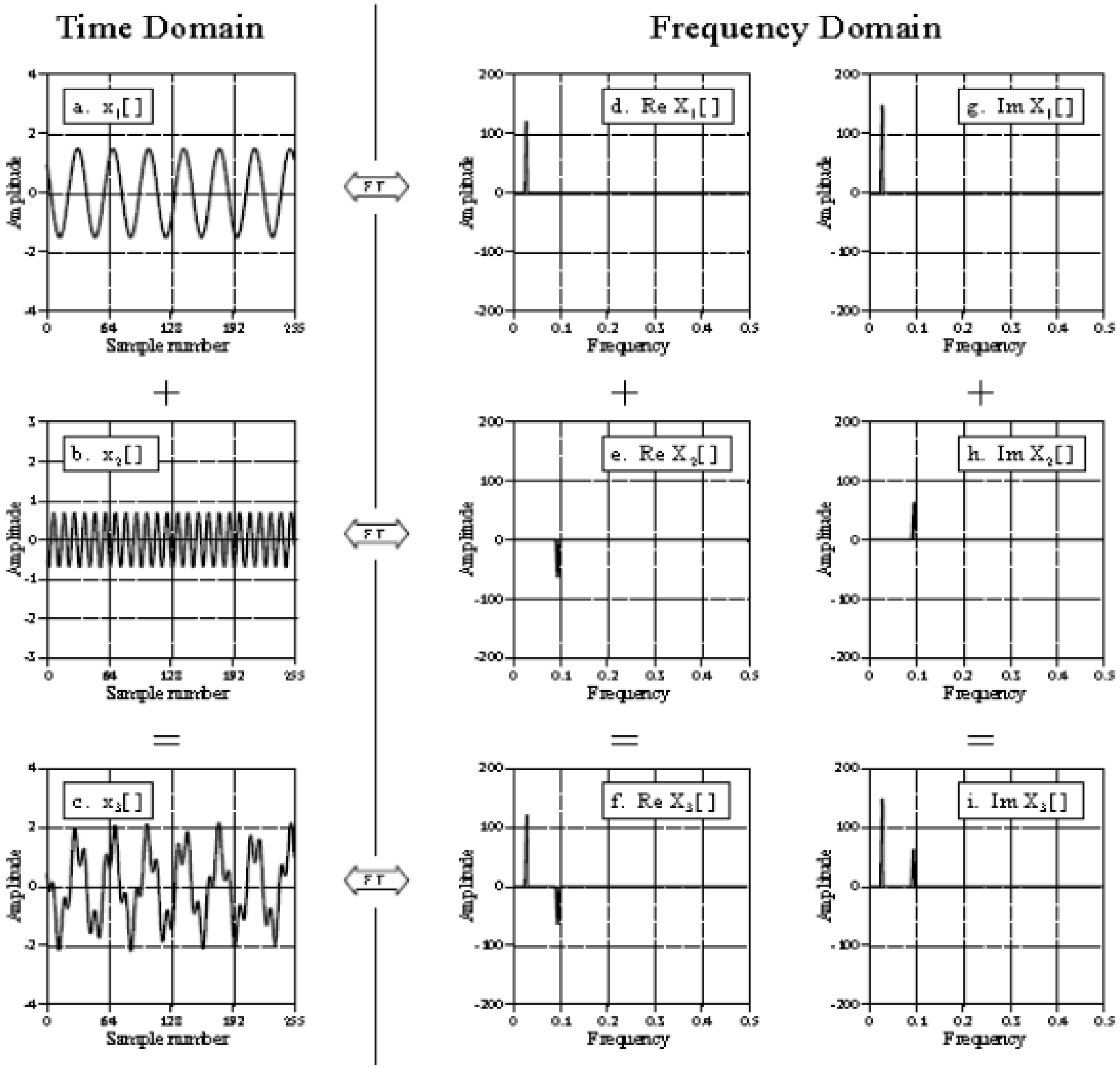
Homogeneity:

x[] $\xrightarrow{\text{DFT}}$ X[]

kx[] $\xrightarrow{\text{DFT}}$ kX[]

Frequency is not changed.

Fourier Transform Properties (2)



Additivity

$$If: x_1[n] + x_2[n] = x_3[n]$$

$$Then: \operatorname{Re} X_1[f] + \operatorname{Re} X_2[f] = \operatorname{Re} X_3[f]$$

$$and \quad \operatorname{Im} X_1[f] + \operatorname{Im} X_2[f] = \operatorname{Im} X_3[f]$$

Fourier Transform Pairs

Delta Function Pairs in Polar Form

Delta Function

Shifted Delta Function

Same Magnitude, Different Phase

Shifted Delta Function