ECE503: Digital Signal Processing
Lecture 3

D. Richard Brown III

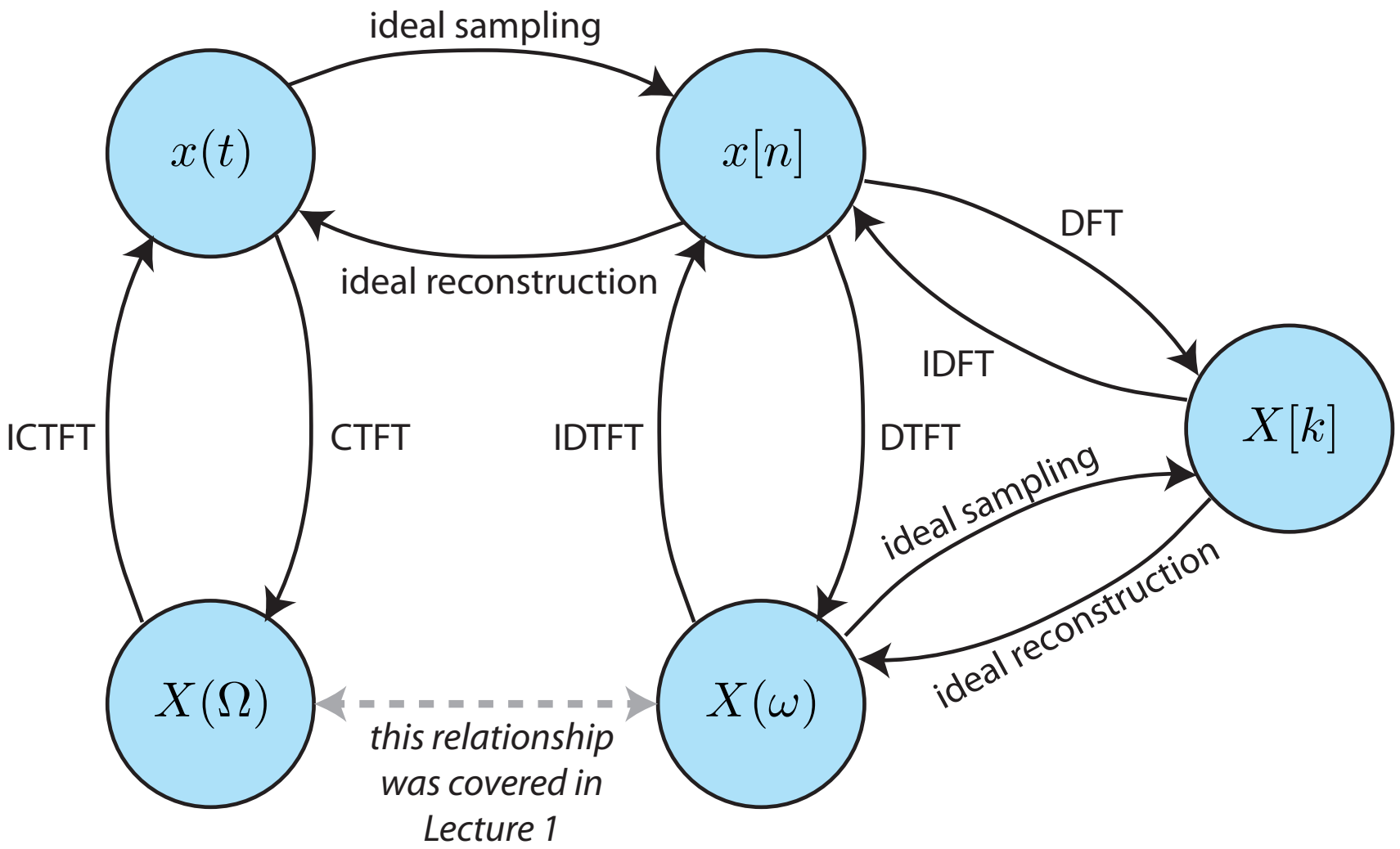
WPI

30-January-2012

Lecture 3 Topics

1. Discrete Fourier Transform (DFT)
   - Relationship to DTFT
   - Intuition
   - Properties
   - Convolution
2. Short-Time Fourier Transform (STFT)
3. Discrete Cosine Transform (DCT)

Big Picture


ideal sampling
$x(t)$
$x[n]$
ideal reconstruction
DFT
IDFT
ICTFT
CTFT
IDTFT
DTFT
$X[k]$
ideal sampling
ideal reconstruction
$X(\Omega)$
$X(\omega)$
this relationship was covered in Lecture 1

The Discrete Fourier Transform (DFT)

Definition: Given a length-$N < \infty$ sequence $\{x[n]\}$, the DFT $\{X[k]\}$ is

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N} = \sum_{n=0}^{N-1} x[n] W_N^{kn} \qquad \text{for } k = 0, \ldots, N-1$$

where the "twiddle factors" $W_N := e^{-j2\pi/N}$. Recall the DTFT of a finite-length sequence $\{x[n]\}$ with length N:

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

Remarks:

- The DTFT takes a discrete-time sequence and produces a continuous-frequency output.
- The DFT is a length-N sampled version of the DTFT, i.e. $X[k] = X(\omega)|_{\omega=2\pi k/N}$ for $k = 0, \ldots, N-1$.
- The DFT takes a finite-length discrete-time sequence and produces a finite-length discrete-frequency output.

Sampling and Periodicity

Recall all discrete-time signals are periodic in the frequency domain. Given a discrete-time sequence $\{x[n]\}$, the DTFT $X(\omega)$ is periodic with period 2π, i.e. $X(\omega + m2\pi) = X(\omega)$ for any integer m. The DFT is also periodic.

Now, when we compute the DFT, we are sampling the continuous-frequency signal. The DFT $X[k] = X(\omega)_{\omega=2\pi k/N}$ is a **discrete-frequency signal**. What does this imply about the time-domain signal $x[n] = \text{IDFT}(X[k])$?

Sampling in the frequency domain implies a periodic signal in the time domain, i.e. $x[n + mN] = x[n]$ for any integer m. This is easy to see from the definition of the IDFT:

$$x[n+mN] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]e^{j2\pi k(n+mN)/N} = \frac{1}{N}\sum_{k=0}^{N-1} X[k]e^{j2\pi kn/N}\underbrace{e^{j2\pi kmN/N}}_{=1} = x[n]$$

The Discrete Fourier Transform (DFT)

The DFT is a linear operation that can be conveniently represented as a matrix-vector product. For example, suppose $N = 3$. The DFT is

$$X[0] = W_3^{00}x[0] + W_3^{01}x[1] + W_3^{02}x[2]$$
$$X[1] = W_3^{10}x[0] + W_3^{11}x[1] + W_3^{12}x[2]$$
$$X[2] = W_3^{20}x[0] + W_3^{21}x[1] + W_3^{22}x[2]$$

which is the same thing as

$$\begin{bmatrix} X[0] \\ X[1] \\ X[2] \end{bmatrix} = \begin{bmatrix} W_3^{00} & W_3^{01} & W_3^{02} \\ W_3^{10} & W_3^{11} & W_3^{12} \\ W_3^{20} & W_3^{21} & W_3^{22} \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \end{bmatrix}$$
$$= \begin{bmatrix} 1 & 1 & 1 \\ 1 & e^{-j2\pi/3} & e^{-j4\pi/3} \\ 1 & e^{-j4\pi/3} & e^{-j8\pi/3} \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \end{bmatrix}$$
$$\boldsymbol{X} = \boldsymbol{G}\boldsymbol{x}$$

Matrix Hermitian

The notation $\boldsymbol{G}^H$ means "matrix Hermitian" and is defined as the complex conjugate transpose of a matrix. For example:

$$\boldsymbol{G} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \qquad \boldsymbol{G}^H = \begin{bmatrix} a^* & c^* \\ b^* & d^* \end{bmatrix}.$$

Let's look at the $\boldsymbol{G}$ from our length-3 DFT:

$$\boldsymbol{G} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & e^{-j2\pi/3} & e^{-j4\pi/3} \\ 1 & e^{-j4\pi/3} & e^{-j8\pi/3} \end{bmatrix} \qquad \boldsymbol{G}^H = \begin{bmatrix} 1 & 1 & 1 \\ 1 & e^{j2\pi/3} & e^{j4\pi/3} \\ 1 & e^{j4\pi/3} & e^{j8\pi/3} \end{bmatrix}.$$

It is not difficult to confirm that

$$\frac{1}{3}\boldsymbol{G}^H\boldsymbol{G} = \frac{1}{3}\boldsymbol{G}\boldsymbol{G}^H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The DFT is an orthogonal transform. Since $\frac{1}{N}\boldsymbol{G}^H\boldsymbol{G}\boldsymbol{x} = \boldsymbol{x}$, this also tells us how to compute the IDFT.

The Inverse Discrete Fourier Transform (IDFT)

Given our definition of the DFT matrix $\boldsymbol{G}$, the IDFT can be computed as

$$\boldsymbol{x} = \frac{1}{N}\boldsymbol{G}^H\boldsymbol{X}$$

or, if you prefer the more explicit summation notation

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]e^{j2\pi kn/N} = \frac{1}{N}\sum_{k=0}^{N-1} X[k]W_N^{-kn} \qquad \text{for } n = 0,\ldots,N-1$$

See Matlab functions `fft` and `ifft`. You can get an N-point DFT matrix in Matlab with the following code:

```
N = 3;
x = eye(N);
G = fft(x);
```

Note the FFT is the same as the DFT but has less computational complexity (we will discuss this when we cover chapter 11).

DFT Intuition

Look at our $N = 3$ example again:

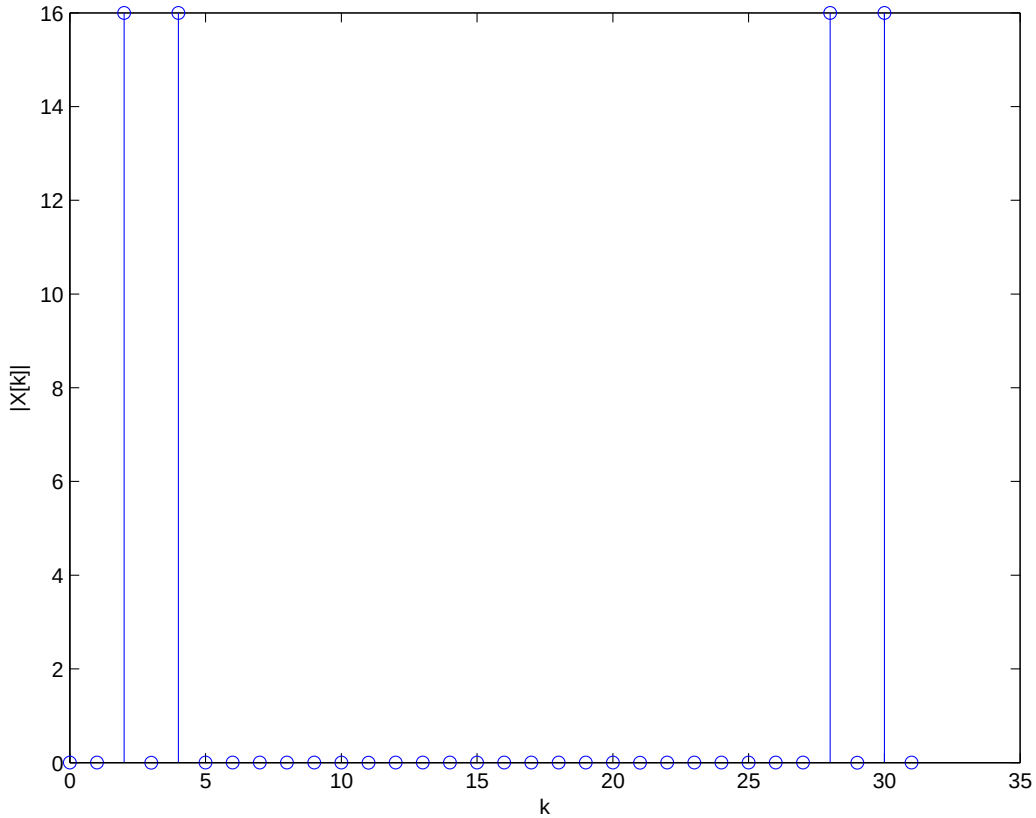
$$\begin{bmatrix} X[0] \\ X[1] \\ X[2] \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & e^{-j2\pi/3} & e^{-j4\pi/3} \\ 1 & e^{-j4\pi/3} & e^{-j8\pi/3} \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \end{bmatrix}$$

Intuition:

- Each row of the DFT matrix is a sampled complex exponential at a specific frequency.
- The matrix-vector product is a correlation between the time domain sequence $\{x[n]\}$ and each row of the DFT matrix.
- $X[k]$ is a measure of how much $2\pi k/N$-frequency component is present in $\{x[n]\}$.
- Each row of the DFT matrix is orthogonal from the other rows.
- The DFT operation can be thought of as a change of basis.

Interpretation of the DFT Frequency Axis

Suppose you plot the DFT magnitude of an $N = 32$ point signal via `plot(0:31,abs(fft(x)))` and see the following result:



What frequencies are present in the signal? If the signal was sampled at $F_T = 44100$ Hz, what frequencies are present in the original analog signal?

Convolution Theorem: DTFT vs. DFT

Textbook pp. 108-109 proves

$$\text{DTFT}(h[n] \circledast x[n]) = H(\omega)X(\omega)$$

if the DTFTs both exist.

What about

$$\text{DFT}(h[n] \circledast x[n]) \stackrel{?}{=} H[k]X[k]$$

If we think about the lengths of each of these sequences (and their corresponding DFTs), it should be obvious this doesn't make sense. Example:

$$h[n] = \{1, 1\} \Leftrightarrow H[k] = \{2, 0\}$$
$$x[n] = \{1, -1, 1, -1\} \Leftrightarrow X[k] = \{0, 0, 4, 0\}$$
$$h[n] \circledast x[n] = \{1, 0, 0, 0, -1\} \Leftrightarrow \text{DTFT}(h[n] \circledast x[n]) = \{0, 0.69 - 0.95i, 1.81 - 0.59i, 1.81 + 0.59i, 0.69 + 0.95i\}$$

Is the DFT useful for convolution?

Circular Convolution

Recall the definition of (linear) convolution for $y[n] = h[n] \circledast x[n]$:

$$y[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k] = \sum_{k=-\infty}^{\infty} x[n-k]h[k].$$

If $h[n]$ and $x[n]$ are both length-N sequences defined on $n = 0, \ldots, N-1$, we can define circular convolution $y[n] = h[n] \textcircled{N} x[n]$ as

$$y[n] = \sum_{k=0}^{N-1} x[k]h[\langle n-k \rangle_N] = \sum_{k=0}^{N-1} x[\langle n-k \rangle_N]h[k]$$

where $\langle n-k \rangle_N$ means "modulo N". For example: $\langle -1 \rangle_N = N-1$, $\langle N \rangle_N = 0$, $\langle N+1 \rangle_N = 1$, etc.

Note the circular convolution result $y[n]$ will also be length-N.

See Matlab function `cconv`.

Circular Convolution Matrix

To illustrate the idea, suppose we want to circularly convolve $\{a[0], a[1], a[2]\}$ and $\{b[0], b[1], b[2]\}$. Note $N = 3$ here. Applying the definition, we can write

$$\begin{aligned} c[0] &= a[0]b[0] + a[1]b[2] + a[2]b[1] \\ c[1] &= a[0]b[1] + a[1]b[0] + a[2]b[2] \\ c[2] &= a[0]b[2] + a[1]b[1] + a[2]b[0] \end{aligned}$$

This is the same as

$$\begin{bmatrix} c[0] \\ c[1] \\ c[2] \end{bmatrix} = \underbrace{\begin{bmatrix} a[0] & a[2] & a[1] \\ a[1] & a[0] & a[2] \\ a[2] & a[1] & a[0] \end{bmatrix}}_{\text{convolution matrix}} \begin{bmatrix} b[0] \\ b[1] \\ b[2] \end{bmatrix} = \underbrace{\begin{bmatrix} b[0] & b[2] & b[1] \\ b[1] & b[0] & b[2] \\ b[2] & b[1] & b[0] \end{bmatrix}}_{\text{convolution matrix}} \begin{bmatrix} a[0] \\ a[1] \\ a[2] \end{bmatrix}$$

Like linear convolution, the circular convolution matrix has a Toeplitz (actually "circulant") structure. Unlike linear convolution, the circular convolution matrix is square ($N \times N$).

Circular Convolution Theorem

Textbook pp. 226-228 proves

$$h[n] \textcircled{N} x[n] \overset{\text{DFT}}{\longleftrightarrow} H[k]X[k].$$

This is **not** the same thing as

$$h[n] \circledast x[n] \overset{\text{DTFT}}{\longleftrightarrow} H(\omega)X(\omega).$$

Intuition:

- Recall N-point sampling in the frequency domain leads to a periodic signal with period N in the time domain.
- Circular convolution accounts for this periodicity.
- Circular convolution is like the linear convolution of two infinite-length periodic sequences with period N (summing only over one period).

Zero-Padding

Suppose we take a length-N sequence and extend it to length-$L > N$ by appending $L - N$ zeros. The DFT is then

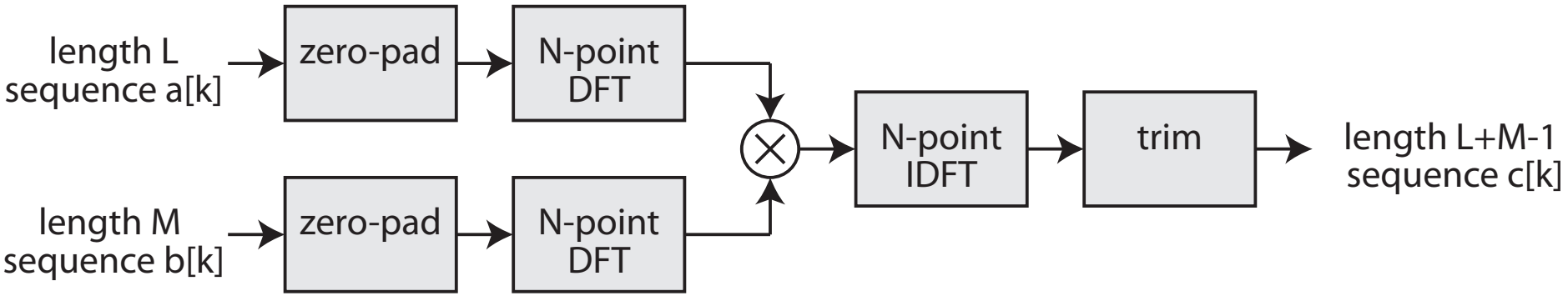
$$X[k] = \sum_{n=0}^{L-1} x[n] e^{-j2\pi kn/L} = \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/L} \qquad \text{for } k = 0, \ldots, L-1$$

This effectively creates a "tall" DFT matrix with L rows and N columns. For example, suppose $L = 4$ and $N = 3$. Then

$$\boldsymbol{G} = \begin{bmatrix} W_4^{00} & W_4^{01} & W_4^{02} \\ W_4^{10} & W_4^{11} & W_4^{12} \\ W_4^{20} & W_4^{21} & W_4^{22} \\ W_4^{30} & W_4^{31} & W_4^{32} \end{bmatrix} \text{ where we had } \boldsymbol{G} = \begin{bmatrix} W_3^{00} & W_3^{01} & W_3^{02} \\ W_3^{10} & W_3^{11} & W_3^{12} \\ W_3^{20} & W_3^{21} & W_3^{22} \end{bmatrix} \text{ before.}$$

Why might this be useful?

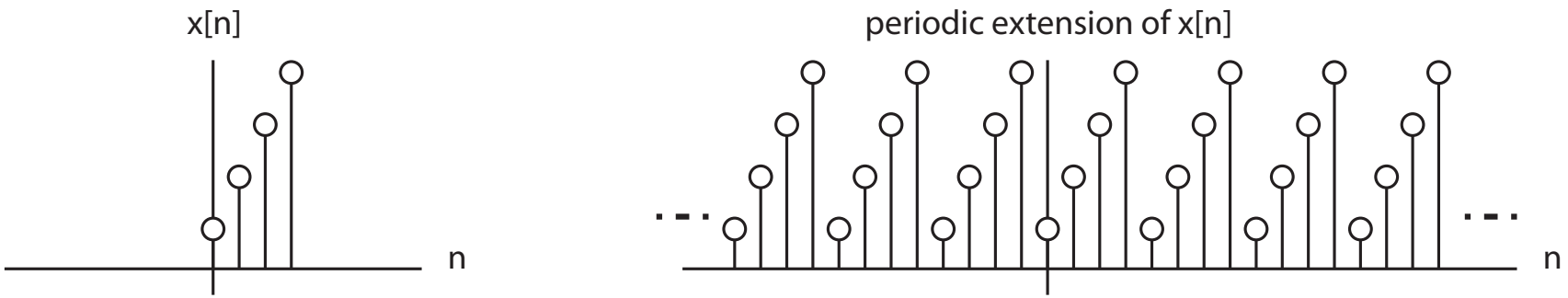
Linear Convolution with the DFT



Remarks:

- The zero-padded sequences are still periodic, but the gaps inserted by the zeros make the circular convolution look like linear convolution.
- N should be selected such that $N \geq L + M - 1$.
- To make the computation efficient, N is usually chosen as the smallest integer power of two satisfying $N \geq L + M - 1$.
- When the DFTs are computed with the FFT, this is called "fast convolution".
- This can be extended to infinite length sequences using "overlap-add" and "overlap-save" methods (see Section 5.10.3 of your textbook).

Periodic Extension: DFT and Fourier Series (1 of 2)



Denote the periodic extension of the length-N sequence $x[n]$ as $\tilde{x}[n]$. Note

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

hence

$$\tilde{X}(\omega) = \text{DTFT}(\tilde{x}[n]) = \frac{1}{N}\sum_{k=0}^{N-1} X[k]\text{DTFT}(e^{j2\pi kn/N})$$

Textbook p. 105 says $\text{DTFT}(e^{j\omega_0 n}) = \sum_{\ell=-\infty}^{\infty} 2\pi\delta(\omega - \omega_0 + 2\pi\ell)$.

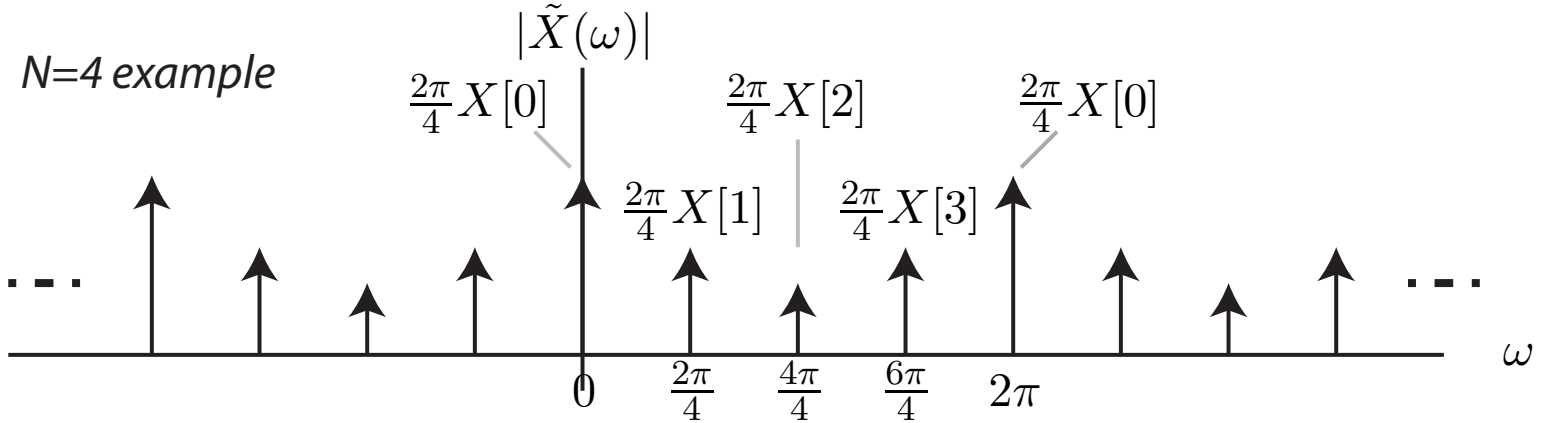
Periodic Extension: DFT and Fourier Series (2 of 2)

Noting $\omega_0 = 2\pi k/N$ here, we can use the prior result to write

$$\text{DTFT}(e^{j2\pi kn/N}) = \sum_{\ell=-\infty}^{\infty} 2\pi\delta(\omega - 2\pi k/N + 2\pi\ell)$$

which implies

$$\tilde{X}(\omega) = \sum_{\ell=-\infty}^{\infty} \sum_{k=0}^{N-1} \underbrace{\frac{2\pi}{N} X[k]}_{\text{Fourier series coefficient}} \delta(\omega - 2\pi k/N + 2\pi\ell)$$



Two-Dimensional DFT

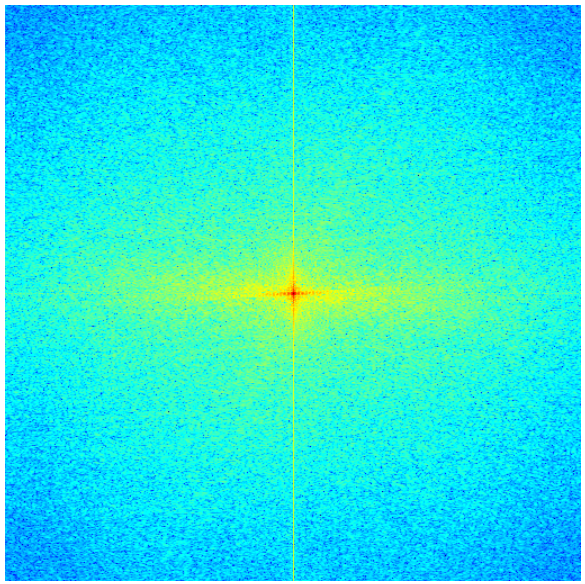
Given a two-dimensional time-domain signal $x[m,n]$, the two-dimensional DFT is defined as

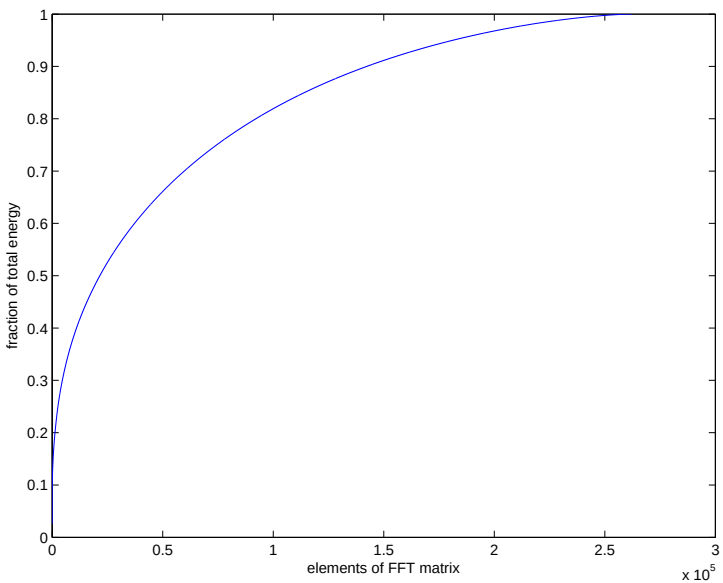
$$\begin{aligned} X[k,\ell] &= \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} x[m,n]e^{-j2\pi(mk/M+n\ell/N)} \\ &= \sum_{m=0}^{M-1} \underbrace{\left(\sum_{n=0}^{N-1} x[m,n]e^{-j2\pi n\ell/N}\right)}_{\text{1-D DFT of row } m} e^{-j2\pi mk/M} \\ &= \sum_{n=0}^{N-1} \underbrace{\left(\sum_{m=0}^{M-1} x[m,n]e^{-j2\pi mk/M}\right)}_{\text{1-D DFT of column } n} e^{-j2\pi n\ell/N} \end{aligned}$$

for $k = 0, \ldots, M-1$ and $\ell = 0, \ldots, N-1$. 2D-FFT has the same properties as the 1D-FFT. See Matlab functions `fft2` and `ifft2`.

Two-Dimensional DFT Example

```
X = imread('post512.tif');
figure(1), imshow(X)
J = fft2(X);
figure(2), imshow(fftshift(log(abs(J))),[])
colormap(jet(64))
figure(3), plot(cumsum(sort(abs(J(:)),'descend'))/sum(sum(abs(J))))
```

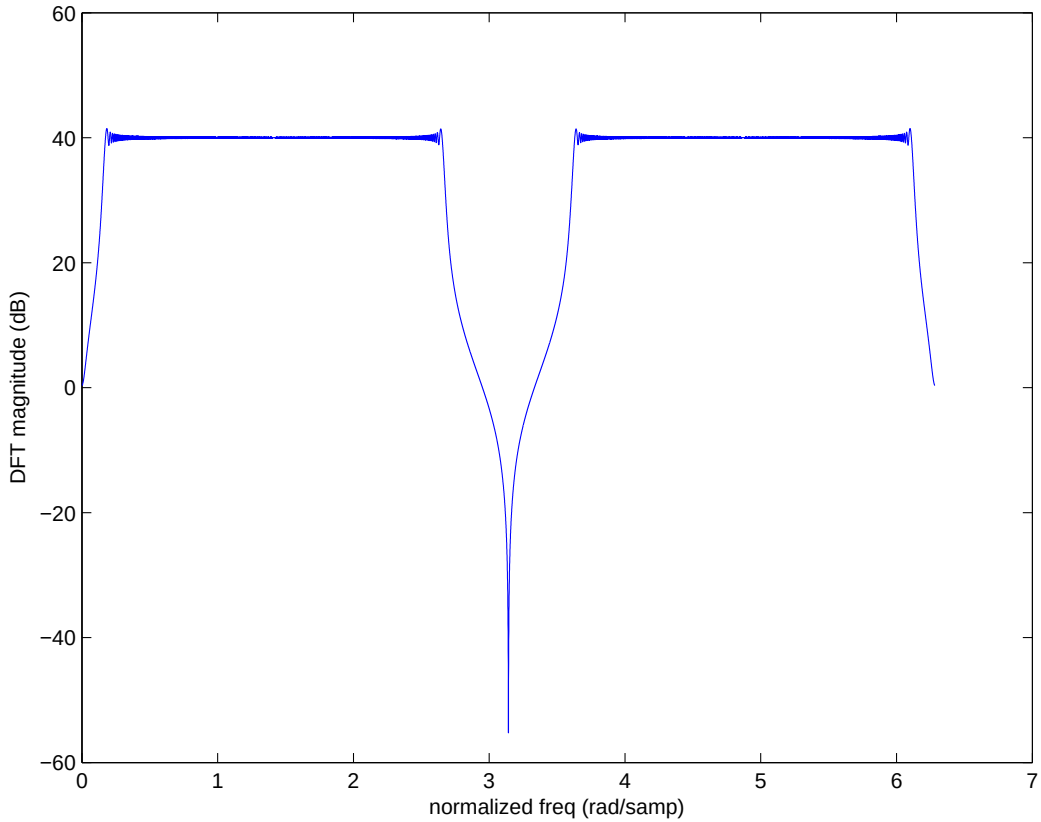




Short-Time Fourier Transform (1 of 4)

For signals that have frequency content that is changing over time, e.g. music or speech, taking the DFT of the whole signal usually isn't very interesting.

```
x = cos(2*pi/80000*(n+1000).^2); % linear chirp
soundsc(x,8000)
plot([0:length(x)-1]/length(x)*2*pi,20*log10(abs(fft(x))))
```



Short-Time Fourier Transform (2 of 4)

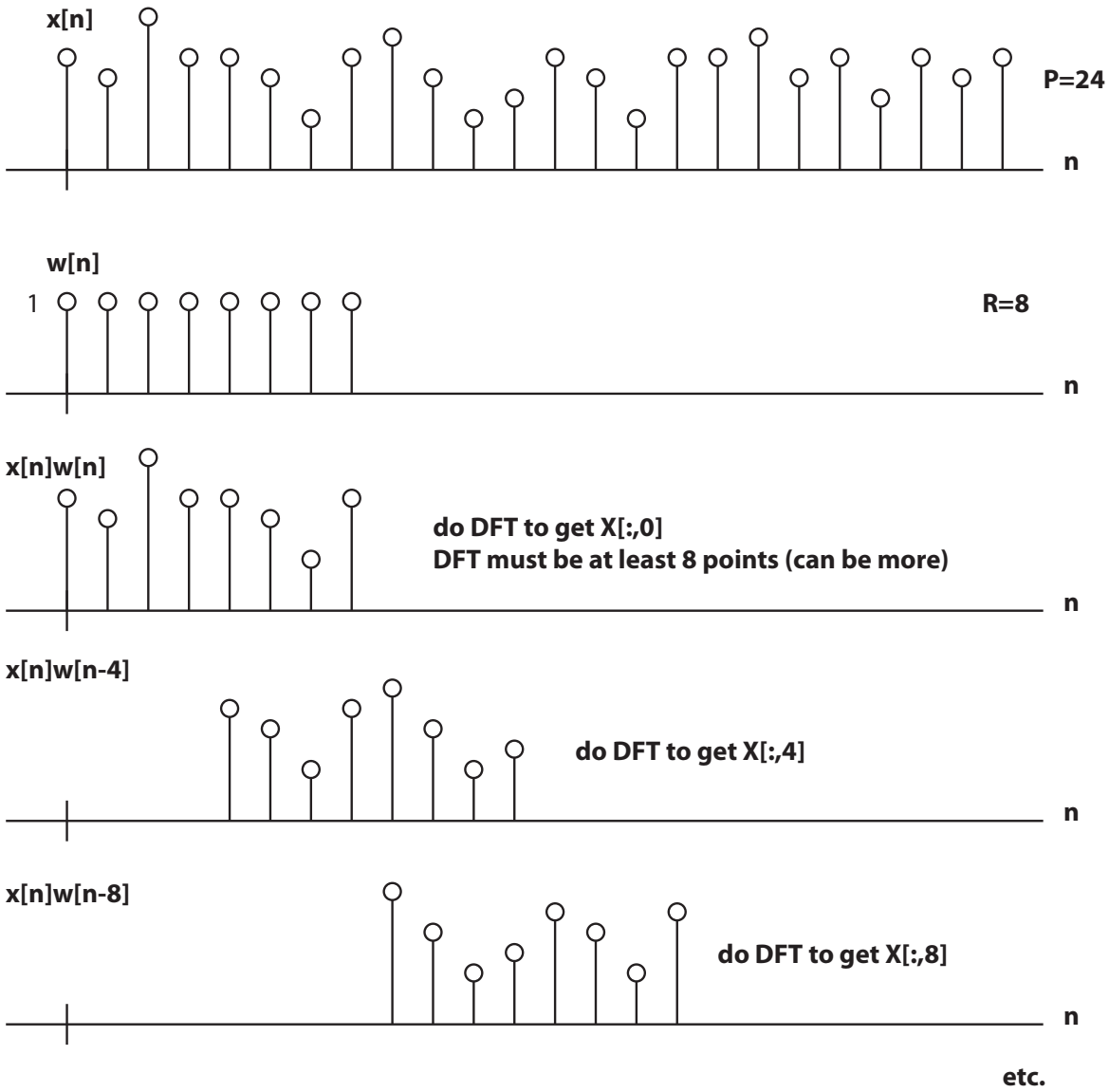
We can get more insight on these signals by performing a "short-time Fourier transform" (STFT). Suppose $x[n]$ is a signal with P samples. The STFT defined as

$$X_{\text{STFT}}[k,n] = \sum_{m=-\infty}^{\infty} x[m]w[m-n]e^{-j2\pi km/N}$$

where $k = 0, \ldots, N-1$ is the frequency index (same as the DFT), n is a time index, and $w[n]$ is a "window function". For now, assume a rectangular window of length $R \leq N \ll P$ with $w[n] = 1$ for $n = 0, \ldots, R-1$ and $w[n] = 0$ otherwise.

In practice, we don't usually compute the STFT for all $n = 0, 1, \ldots$ since the frequency content of the signal does not change much from sample to sample. Instead, we typically pick an integer $L \leq R$ and compute the STFT for values of $n = 0, L, 2L, \ldots$.

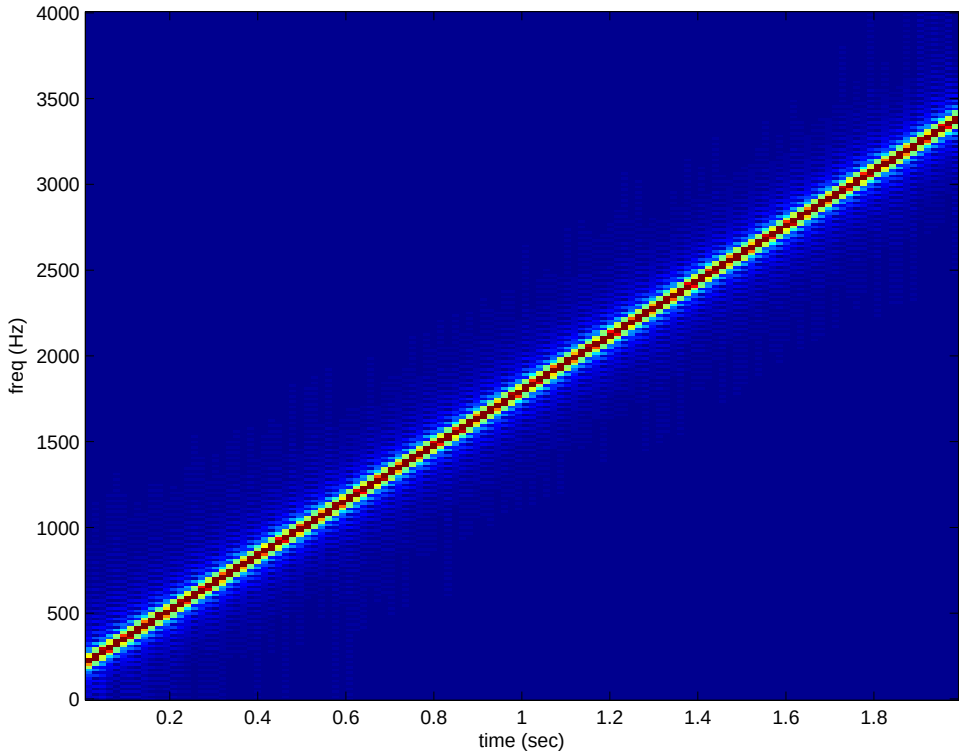
Short-Time Fourier Transform (3 of 4)



Short-Time Fourier Transform (4 of 4)

Matlab function `spectrogram` is useful for computing STFTs.

```
x = cos(2*pi/80000*(n+1000).^2); % linear chirp
[s,f,t] = spectrogram(x,ones(1,256),128,512,8000);
image(t,f,abs(s)); set(gca,'ydir','normal')
```



Google "Aphex face" for an example of a more interesting spectrogram.

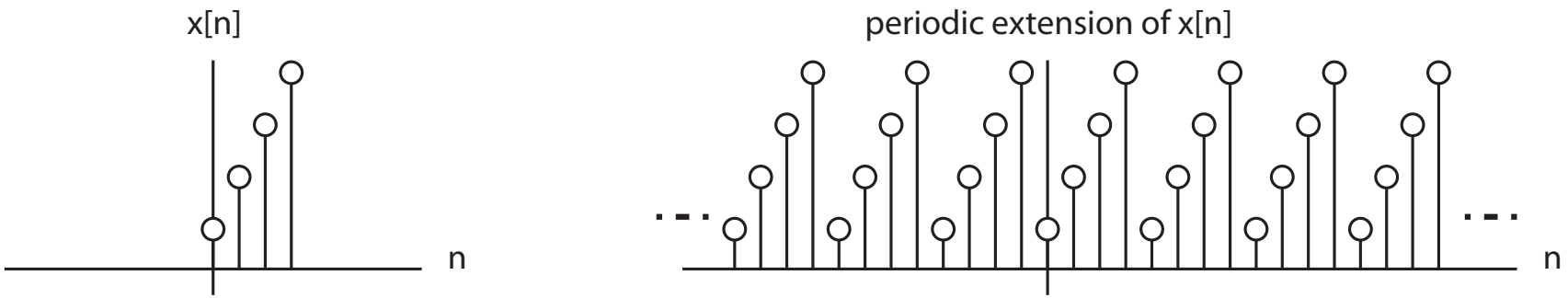
Discrete Cosine Transform (1 of 3)

When working with real-valued signals, the DFT is typically complex and has some redundancy:

- Odd symmetric phase
- Even symmetric magnitude

If you know $X[1]$, you know $X[N-1]$ since they are just conjugates of each other (easy to prove or try it for yourself in Matlab).

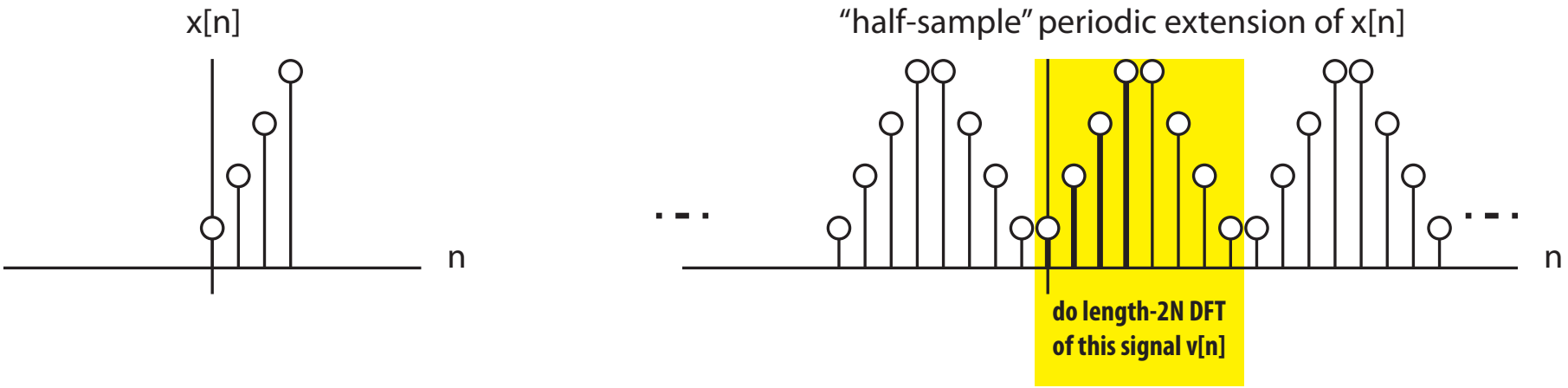
Also, the periodic extension implicit in the DFT can create sharp discontinuities, as we saw before:



These discontinuities can cause the spectral content of the DFT to be spread out.

Discrete Cosine Transform (2 of 3)

The "discrete cosine transform" DCT is an orthogonal transform, like the DFT, but generates a length-N real valued output from a length-N real valued input. It also assumes a different periodic extension, as shown below:



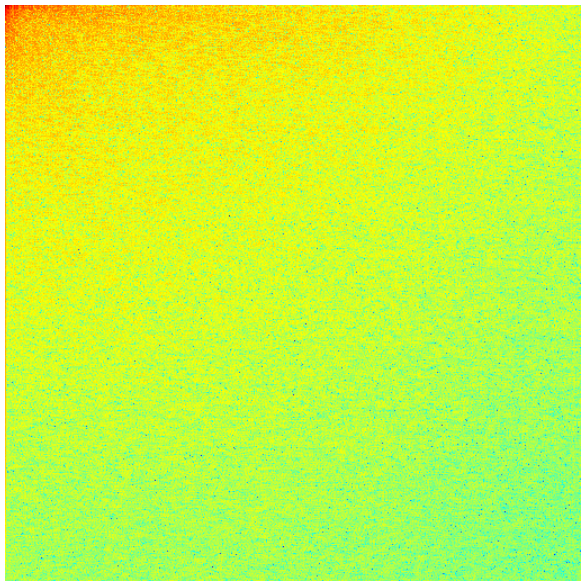
After forming $v[n]$ from $x[n]$, the $2N$-point DFT can be computed as

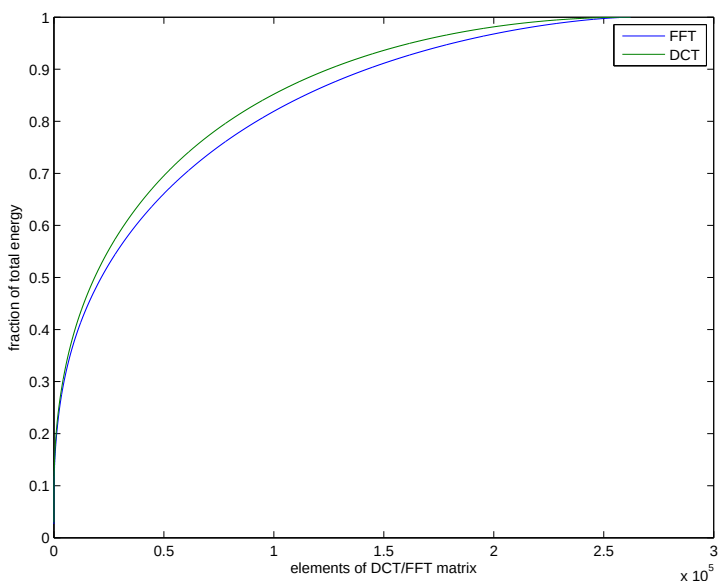
$$V[k] = \sum_{n=0}^{2N-1} v[n] e^{-j2\pi nk/(2N)} = 2W_{2N}^{-k/2} \sum_{n=0}^{N-1} x[n] \cos\left(\frac{\pi k(2n+1)}{2N}\right)$$

and $X_{\text{DCT}}[k] = W_{2N}^{k/2} V[k]$ for $n = 0, \ldots, N-1$. IDCT is similar (see textbook). See Matlab functions `dct`, `idct`, `dct2`, and `idct2`.

Discrete Cosine Transform (3 of 3)

```
X = imread('post512.tif');
figure(1), imshow(X)
J = dct2(X);
figure(2), imshow(log(abs(J)),[]),
colormap(jet(64))
figure(3), plot(cumsum(sort(abs(J(:)),'descend'))/sum(sum(abs(J))))
```





The DCT is used in signal compression due to its better "energy compaction".

Conclusions

1. This concludes Chapter 5. You are responsible for all of the material in this chapter except 5.13 (Haar transform), even if it wasn't covered in lecture.
2. Please read Chapter 6 before the next lecture and have some questions prepared.
3. The next lecture is on Monday 06-Feb-2012 at 6pm.