ECE 4213/5213

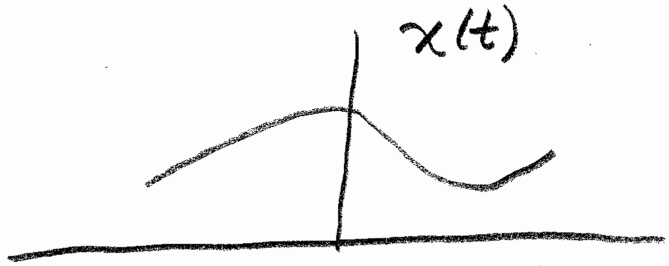
MODULE 5

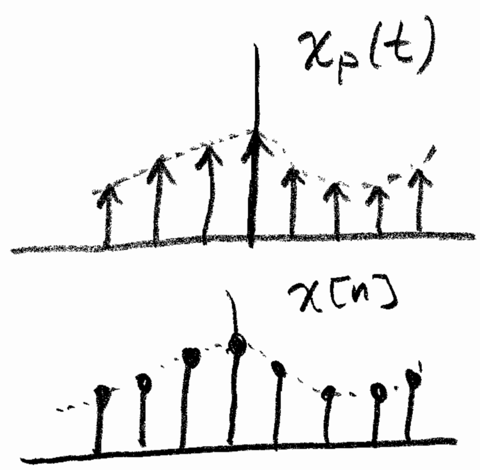
Index

Chapter 5: The DFT

- For a discrete-time signal $x[n]$ with domain $\mathbb{Z}$ and range $\mathbb{R}$ or $\mathbb{C}$, we have seen that the DTFT $X(e^{j\omega})$ is <u>always</u> 2π-periodic.

- For continuous-time signals like $e^{j\Omega_0 t}$, $\cos\Omega_0 t$, and $\sin\Omega_0 t$ that are periodic in the time domain, we saw that the F.T. $X(\Omega)$ consisted exclusively of impulses (Dirac deltas).
  - → Fundamentally, a continuous domain signal that contains only impulses <u>is</u> discrete, because it takes nonzero values only at a <u>countable</u> number of places.
  - → We used this concept in sampling when we "picked off" the weights of the impulses in $x_p(t)$ to make $x[n]$:





- So we have seen that

Discrete in time $\longrightarrow$ Periodic in frequency

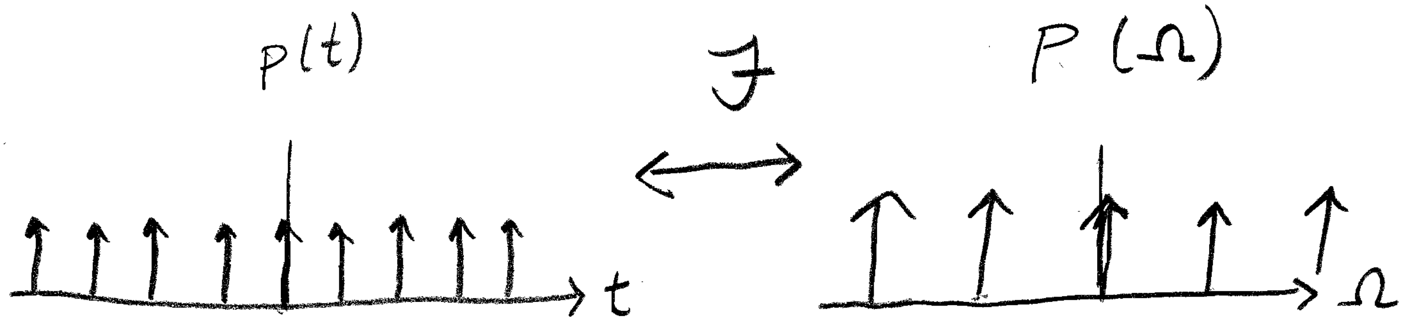
Periodic in time $\longrightarrow$ Discrete in frequency

- In fact, one of the most fundamental duality properties of <u>all</u> Fourier representations is:

☆ A signal that is periodic in one domain is discrete in the other domain.

$\Rightarrow$ This implies that a signal that is <u>both</u> <u>periodic</u> <u>and</u> <u>discrete</u> in one domain must also be <u>both</u> <u>periodic</u> <u>and</u> <u>discrete</u> in the other domain.

<u>EX</u>: the periodic impulse train $p(t)$ that we used for sampling:



- This concept that

$$\text{discrete \& periodic} \overset{\mathcal{F}}{\longleftrightarrow} \text{discrete \& periodic}$$

is the main idea behind the <u>Discrete Fourier Transform</u>, or "DFT."

- The DFT is a Fourier representation for a <u>finite length</u> signal $x[n]$ that is defined only for $0 \le n < N$.
  - → In other words, the domain of the function $x[n]$ is the integer interval $[0, N-1]$.
  - → outside of 0 to $N-1$, $x[n]$ is undefined; it does not take a value.
  - → We refer to $x[n]$ as a "length-N signal (or sequence)" or as an "N-point signal (or sequence)."

- This is very useful for applications, since a finite-length signal $x[n]$ can be represented in a computer or DSP chip.

- The DFT is a way to write $x[n]$ as a sum of N sinusoidal basis signals $e^{j\omega_k n}$ for $k = 0, 1, \ldots, N-1$.

- The frequencies of the basis signals are given by $\omega_k = \frac{2\pi}{N} k$ for $0 \le k \le N-1$.

- In other words, the basis signals are given by $e^{j2\pi nk/N}$, $0 \le k \le N-1$.

- It is customary to use the notation

$$W_N = e^{-j2\pi/N}. \quad \text{(a complex number)}$$

→ With this notation, the basis signals are written W_N^{-nk}. $\left(= e^{+j2\pi nk/N}\right)$

<u>Note:</u> The conjugates of the basis signals (which will appear in inner products) are given by $W_N^{nk} = e^{-j2\pi nk/N}$.

- These basis signals are all periodic functions of n with period N.

- We will add them up in a linear combination so that the sum is equal to $x[n]$ on the interval $0 \le n \le N-1$.

  $\Rightarrow$ Outside this interval, the sum will be periodic with period N.

- So it is useful to think of <u>three</u> related signals:

  1. $x[n]$: a length-N signal defined for $0 \le n \le N-1$. He has a DFT $X[k]$ that is also length-N; it is defined for $0 \le k \le N-1$.

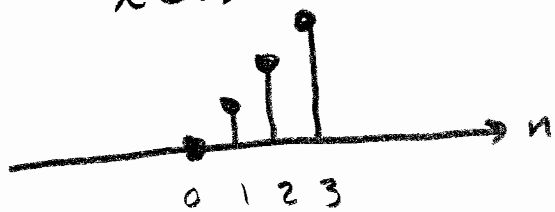
  2. $\tilde{x}[n]$: the periodic extension of $x[n]$. He is defined on all of $\mathbb{Z}$. For $0 \le n \le N-1$, $\tilde{x}[n] = x[n]$. Outside of this interval, $\tilde{x}[n]$ <u>is</u> defined and repeats periodically with period N.
     He has a Fourier series representation and a distributional DTFT.

③ $\hat{x}[n]$: the zero-padded extension of $x[n]$.

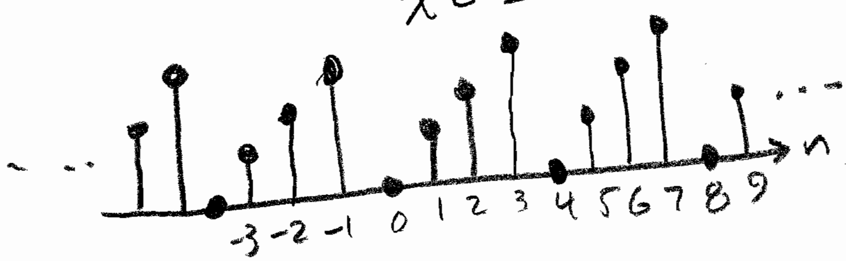
- He is defined on all of $\mathbb{Z}$.
- For $0 \le n \le N-1$, $\hat{x}[n] = x[n]$.
- outside of this interval, $\hat{x}[n] = 0$.
- He has a DTFT $\hat{X}(e^{j\omega})$. (2π-periodic)

<u>EX</u> : $x[n] = n, \quad 0 \le n \le 3$ (length $N=4$).

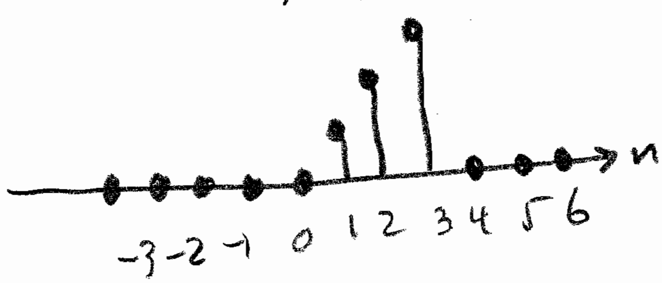
$x[n]$



$\tilde{x}[n]$



$\hat{x}[n]$



- The notions of time shifting, time reversal, and symmetry for the N-point sequence $x[n]$ are defined in terms of the periodic extension $\tilde{x}[n]$.

  → This kind of shifting is called "circular shifting."

  → This kind of symmetry is called "periodic symmetry." (or "circular symmetry)

  → There is also a special type of convolution that naturally goes along with the DFT called "circular convolution" or "periodic convolution."

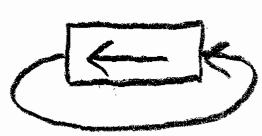
  ⇒ More on this later.

Circular Shifting

- circular shifting a length-N sequence $x[n]$ is like shifting bits in an N-bit circular shift register.

- Instead of falling off the end, the samples that are shifted out of one side come back in on the other side.

Circular Shift Right

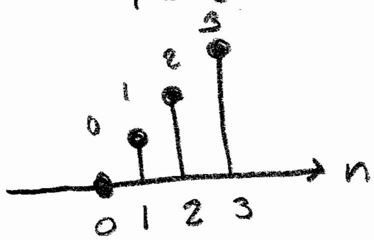


Circular Shift Left

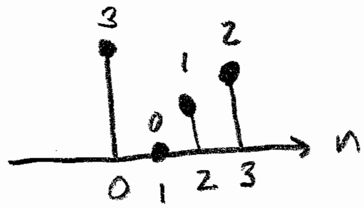
EX: $N = 4$. $x_1[n] = n, \quad 0 \le n \le 3$. $n_0 = 2$

$$x_2[n] = x_1[n - n_0] = x_1[n-2]$$

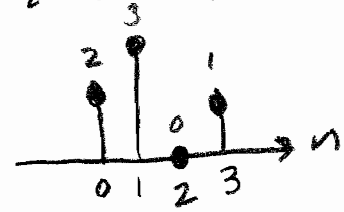
$x_1[n]$



$x_1[n-1]$

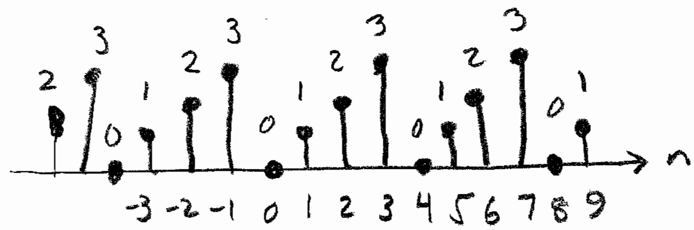


$x_2[n] = x_1[n-2]$

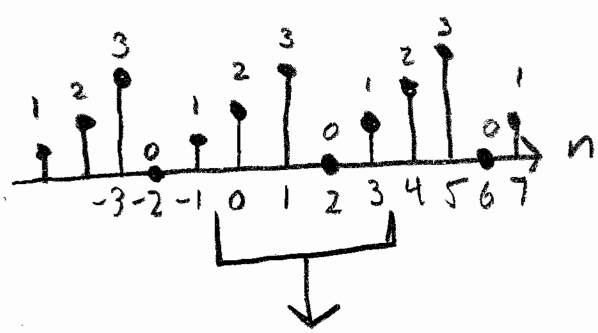


$\Rightarrow$ Notice that this same result can be obtained by performing a plain old time shift on $\tilde{x}[n]$ and then keeping the result only for $0 \le n \le N-1$:

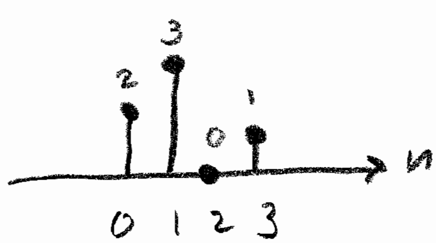
$\tilde{x}[n]$



$\tilde{x}[n-2]$



$x_2[n] = x_1[n-2]$



- Mathematically, the circular shift operation on $x[n]$ can be written using modular arithmetic.
- The Oppenheim & Schafer text uses "double parentheses" with a subscript "N" for this... see, e.g., p. 642 between (8.59) and (8.60):

$$((n))_N = (n \bmod N).$$

- In the notes, we will use angle brackets for a simpler notation to mean the same thing. We will write

$$\langle n \rangle_N = n \bmod N.$$

→ The number $\langle n \rangle_N$ is an integer equal to $n + \ell N$ where $\ell \in \mathbb{Z}$ is chosen such that

$$0 \leq n + \ell N \leq N-1.$$

EX: $N = 7$:

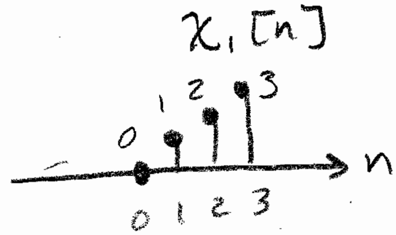
$\langle 25 \rangle_7 = 4 \quad (\ell = -3 \rightarrow 25 + (-3)7 = 4)$

$\langle -16 \rangle_7 = 5 \quad (\ell = 3 \rightarrow -16 + (3)7 = 5)$

- Although time shifting on finite-length sequences is usually understood to be circular shifting, this can be made explicit by using the notation

$$x_2[n] = x_1[\langle n - n_0 \rangle_N].$$

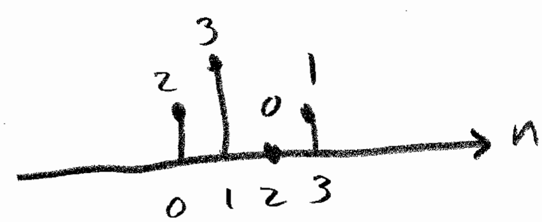
EX: Let $N=4$ and $x_1[n] = n$, $0 \le n \le 3$.



Let $x_2[n] = x_1[\langle n-2 \rangle_4]$

$$x_2[0] = x_1[\langle 0-2 \rangle_4] = x_1[\langle -2 \rangle_4] = x_1[2]$$
$$x_2[1] = x_1[\langle 1-2 \rangle_4] = x_1[\langle -1 \rangle_4] = x_1[3]$$
$$x_2[2] = x_1[\langle 2-2 \rangle_4] = x_1[\langle 0 \rangle_4] = x_1[0]$$
$$x_2[3] = x_1[\langle 3-2 \rangle_4] = x_1[\langle 1 \rangle_4] = x_1[1]$$

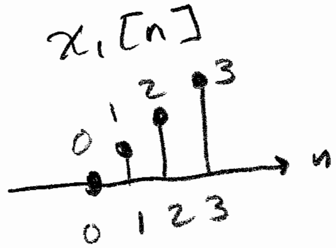
$$x_2[n] = x_1[\langle n-2 \rangle_4]$$



(Same result as before on page 5.8).

- When $n_0 < 0$, you get a circular shift left.

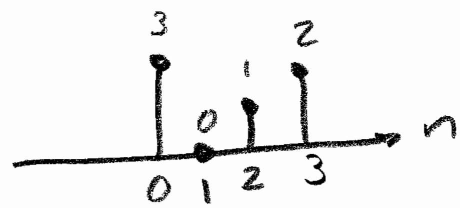
<u>EX</u>: $N = 4$. $x_1[n] = n$, $0 \le n \le 3$. $n_0 = -3$.



$$x_2[n] = x_1[\langle n - n_0 \rangle_4]$$
$$= x_1[\langle n+3 \rangle_4].$$

$$x_2[0] = x_1[\langle 3 \rangle_4] = x_1[3]$$
$$x_2[1] = x_1[\langle 4 \rangle_4] = x_1[0]$$
$$x_2[2] = x_1[\langle 5 \rangle_4] = x_1[1]$$
$$x_2[3] = x_1[\langle 6 \rangle_4] = x_1[2]$$

$$x_2[n] = x_1[\langle n+3 \rangle_4]$$



<u>Note</u>: this same result could have been obtained by doing a plain old shift left by 3 on $\tilde{x}[n]$ and only keeping the result for $0 \le n \le 3$. You may find this easier to think about.

- Modular arithmetic "quick tricks":

* $\langle n-n_0\rangle_N = n-n_0$ for $n_0 \le n \le N-1$

* $\langle n-n_0\rangle_N = N-n_0+n$ for $0 \le n < n_0$

* $\underbrace{\langle n-n_0\rangle_N}_{\text{circular shift right by } n_0} = \underbrace{\langle n+(N-n_0)\rangle_N}_{\text{circular shift left by } N-n_0}$

circular shift right by n_0 = circular shift left by $N-n_0$

* For $n_0 > N$, a circular shift left (or right) by n_0 is the same as a circular shift left (or right) by $\langle n_0\rangle_N$.

Circular Time Reversal

- For an N-point sequence $x[n]$, the circular time reversal is defined by

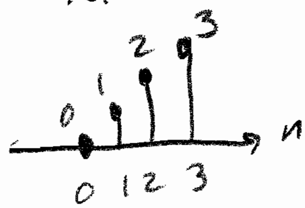
$$x[\langle -n\rangle_N] = x[\langle N-n\rangle_N].$$

- This gives the same result as performing a plain old time reversal on the periodically extended sequence $\tilde{x}[n]$ and only keeping the resulting samples for $0 \le n \le N-1$.

EX: $N = 4$. $x_1[n] = n, \quad 0 \le n \le 3$.

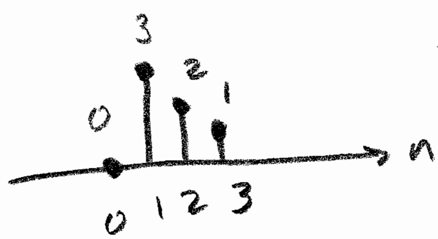
$$x_2[n] = x_1[\langle -n \rangle_4]$$

$x_1[n]$



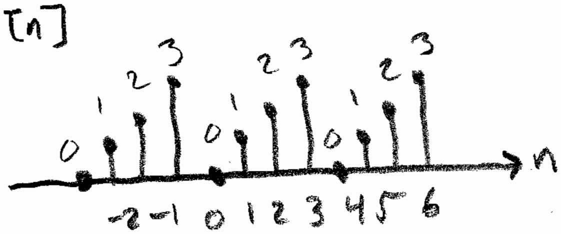
$$x_2[0] = x_1[\langle 0 \rangle_4] = x_1[\langle 4-0 \rangle_4] = x_1[0]$$
$$x_2[1] = x_1[\langle -1 \rangle_4] = x_1[\langle 4-1 \rangle_4] = x_1[3]$$
$$x_2[2] = x_1[\langle -2 \rangle_4] = x_1[\langle 4-2 \rangle_4] = x_1[2]$$
$$x_2[3] = x_1[\langle -3 \rangle_4] = x_1[\langle 4-3 \rangle_4] = x_1[1]$$

$x_2[\langle -n \rangle_4]$

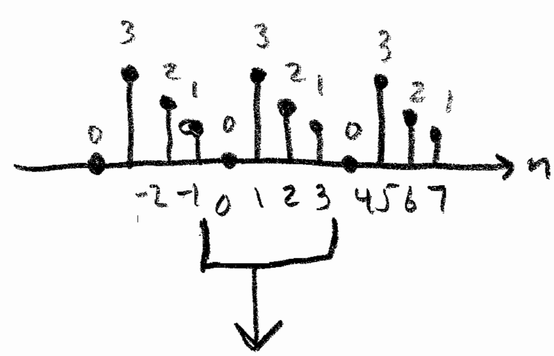


$\Rightarrow$ This same result can be obtained by performing a plain old time reversal on $\tilde{x}[n]$ and only keeping the result for $0 \le n \le 3$:

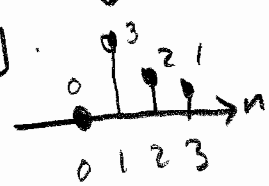
$\tilde{x}[n]$



$\tilde{x}[-n]$



$x_2[n]$



Periodic Symmetry (also known as Circular Symmetry)

- if $x[n] = x[\langle -n \rangle_N]$ for $0 \le n \le N-1$,

  then $x[n]$ is called <u>periodically even</u>.

→ This implies that $\tilde{x}[n]$ is an even signal in the plain old sense.

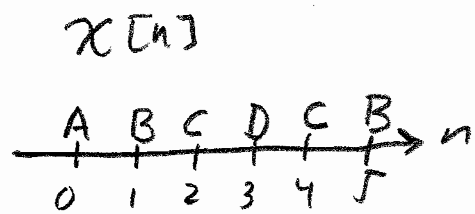
<u>EX</u>: if $N=6$ and $x[n]$ is periodically even, then:

$$x[0] = x[\langle -0 \rangle_6] = x[0] \quad \text{(no restriction)}$$
$$x[1] = x[\langle -1 \rangle_6] = x[5]$$
$$x[2] = x[\langle -2 \rangle_6] = x[4]$$
$$x[3] = x[\langle -3 \rangle_6] = x[3] \quad \text{(no restriction)}$$

$$x[4] = x[\langle -4 \rangle_6] = x[2]$$
$$x[5] = x[\langle -5 \rangle_6] = x[1]$$

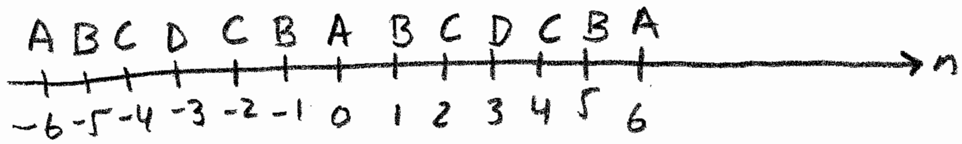
if, say, $x[0] = A$, $x[1] = B$, $x[2] = C$, $x[3] = D$, then $x[4] = C$, $x[5] = B$

And the graph is



- Then the graph of $\tilde{x}[n]$ looks like

$\tilde{x}[n]$



which is even in the plain old sense.

- <u>How to spot it</u>: if N is even and $x[n]$ is periodically even symmetric, then $x[0]$ and $x[\frac{N}{2}]$ are free parameters. But $x[1] \ldots x[\frac{N}{2}-1]$ must be the mirror image of $x[\frac{N}{2}+1] \ldots x[N-1]$.

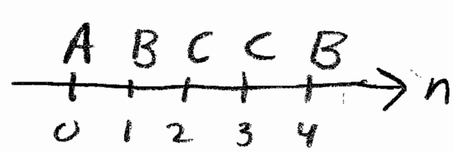
<u>EX</u>: if $N=5$ and $x[n]$ is periodically even, then:

$$x[0] = x[\langle -0 \rangle_5] = x[0] \quad \text{(no restriction)}$$
$$x[1] = x[\langle -1 \rangle_5] = x[4]$$
$$x[2] = x[\langle -2 \rangle_5] = x[3]$$
$$x[3] = x[\langle -3 \rangle_5] = x[2]$$
$$x[4] = x[\langle -4 \rangle_5] = x[1]$$

if, say $x[0] = A$, $x[1] = B$, $x[2] = C$ then $x[3] = C$, $x[4] = B$

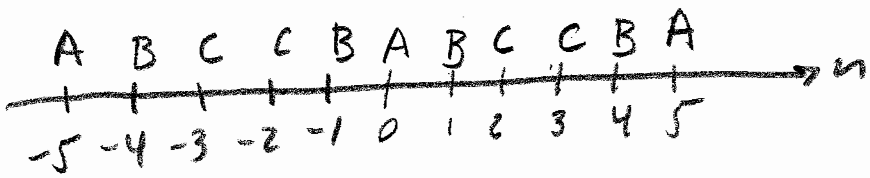
and the graph is

$x[n]$



- Then the graph of $\tilde{x}[n]$ looks like

$\tilde{x}[n]$



which is even in the plain old sense.

- How to spot it: if N is odd and $x[n]$ is periodically even, then $x[0]$ is a free parameter. But $x[1] \ldots x[\frac{N-1}{2}]$ must be the mirror image of $x[\frac{N+1}{2}] \ldots x[N-1]$.

- if $x[n] = -x[\langle -n \rangle_N]$ for $0 \leq n \leq N-1$, then $x[n]$ is called periodically odd.

$\rightarrow$ This implies that $\tilde{x}[n]$ is an odd signal in the plain old sense.

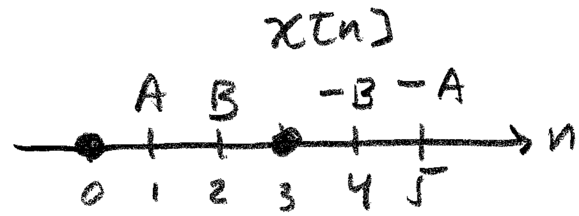
EX: if $N=6$ and $x[n]$ is periodically odd, then:

$$x[0] = -x[\langle -0\rangle_6] = -x[0] \Longrightarrow x[0]=0$$
$$x[1] = -x[\langle -1\rangle_6] = -x[5]$$
$$x[2] = -x[\langle -2\rangle_6] = -x[4]$$
$$x[3] = -x[\langle -3\rangle_6] = -x[3] \Longrightarrow x[3]=0$$

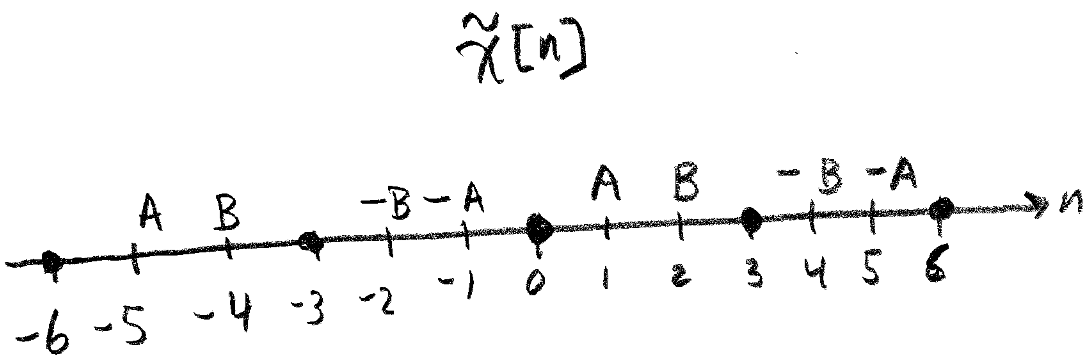
$$x[4] = -x[\langle -4\rangle_6] = -x[2]$$
$$x[5] = -x[\langle -5\rangle_6] = -x[1]$$

So if, say, $x[1]=A$ then $x[4]=-B$
$x[2]=B$ $x[5]=-A$

And the graph is



- Then the graph of $\tilde{x}[n]$ looks like



which is odd in the plain old sense.

- How to spot it: if N is even and $x[n]$ is periodically odd, then $x[0] = x[\frac{N}{2}] = 0$. Moreover, $x[1] \dots x[\frac{N}{2}-1]$ are the negative mirror image of $x[N/2+1] \dots x[N-1]$.

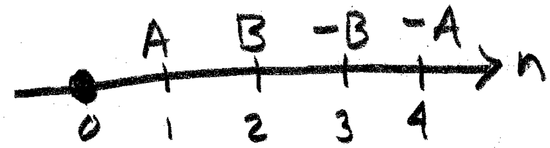
EX: if $N=5$ and $x[n]$ is periodically odd, then:

$$x[0] = -x[\langle -0\rangle_5] = -x[0] \Longrightarrow x[0] = 0$$
$$x[1] = -x[\langle -1\rangle_5] = -x[4]$$
$$x[2] = -x[\langle -2\rangle_5] = -x[3]$$

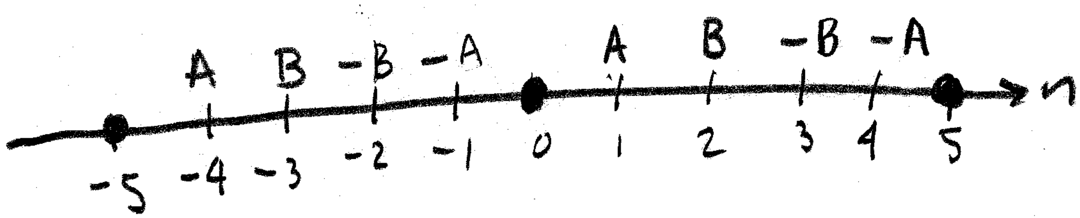
$$x[3] = -x[\langle -3\rangle_5] = -x[2]$$
$$x[4] = -x[\langle -4\rangle_5] = -x[1]$$

So if, say, $x[1] = A$ then $x[3] = -B$
$x[2] = B$ $x[4] = -A$

and the graph looks like $x[n]$



- Then the graph of $\tilde{x}[n]$ looks like



which is odd in the plain old sense.

- __How to spot it:__ if N is odd and $x[n]$ is periodically odd, then $x[0] = 0$ and $x[1] \ldots x[\frac{N-1}{2}]$ must be the negative mirror image of $x[\frac{N+1}{2}] \ldots x[N-1]$.

FACT: any length-N real sequence can be uniquely written as the sum of a periodically even length-N sequence $x_{pe}[n]$ and a periodically odd length-N sequence $x_{po}[n]$, where

$$x_{pe}[n] = \frac{1}{2}\left\{x[n] + x[\langle -n\rangle_N]\right\}, \quad 0 \le n \le N-1$$

and $$x_{po}[n] = \frac{1}{2}\left\{x[n] - x[\langle -n\rangle_N]\right\}, \quad 0 \le n \le N-1.$$

- A complex-valued length-N sequence is called periodically conjugate symmetric if $x[n] = x^*[\langle -n\rangle_N]$, $0 \le n \le N-1$.
- A complex-valued length-N sequence is called periodically conjugate antisymmetric if

$$x[n] = -x^*[\langle -n\rangle_N], \quad 0 \le n \le N-1.$$

$\longrightarrow$ These symmetries imply that the periodic extension $\tilde{x}[n]$ is conjugate symmetric or conjugate antisymmetric in the plain old sense.

FACT: Any length-N complex sequence can be uniquely written as the sum of a periodically conjugate symmetric length-N sequence $x_{cs}[n]$ and a periodically conjugate antisymmetric sequence $x_{ca}[n]$, where

$$x_{cs}[n] = \frac{1}{2}\left\{x[n] + x^*[\langle -n \rangle_N]\right\}, \quad 0 \le n \le N-1$$

and $$x_{ca}[n] = \frac{1}{2}\left\{x[n] - x^*[\langle -n \rangle_N]\right\}, \quad 0 \le n \le N-1.$$

NOTE: if $x[n]$ just happens to be real, then

$$x_{cs}[n] = x_{pe}[n] \text{ and } x_{ca}[n] = x_{po}[n].$$

Discrete Fourier Series (DFS)

- We now return to the discussion from page 5.6 about writing $x[n]$ and $\tilde{x}[n]$ in terms of the basis $W_N^{-kn} = e^{+j2\pi nk/N}$ for $0 \le k \le N-1$.

- Assume that $\tilde{x}[n]$ is a periodic discrete-time signal with period N.

- Then, similar to the continuous-time case, $\tilde{x}[n]$ can be written in a Fourier series

$$\tilde{x}[n] = \frac{1}{N}\sum_{k=0}^{N-1} A_k e^{j\omega_k n} \qquad (*)$$

where $\omega_k = \frac{2\pi}{N}k$.

<u>Note:</u> $e^{j\omega_k n} = e^{j2\pi nk/N} = W_N^{-nk}$.

- Eq. (*) is called the <u>Discrete</u> <u>Fourier</u> <u>Series</u> (DFS).
  - Since the signal $\tilde{x}[n]$ is discrete and periodic with period N, it has only N distinct values... or N "degrees of freedom."
  - This implies that only N basis functions $e^{j\omega_k n}$ are needed in the sum (*)... just like only three basis vectors $\vec{i}, \vec{j}, \vec{k}$ are needed in $\mathbb{R}^3$.
  - So, whereas the continuous-time Fourier series generally required an infinite number of terms in the sum

    → The DFS sum (*) has only N terms.

- The DFS coefficients A_k are given by

$$A_k = \sum_{n=0}^{N-1} \tilde{x}[n] e^{-j\omega_k n} = \sum_{n=0}^{N-1} \tilde{x}[n] W_N^{nk} \qquad (**)$$

- Note that the DFS coefficients in (**) are a dot product between the signal and the basis functions (as always).

- Even though there are only N terms in the DFS (*):

$$\tilde{x}[n] = \frac{1}{N}\sum_{k=0}^{N-1} A_k W_N^{-nk},$$

we can think of the coeffients A_k as being a periodic sequence with period N.

→ Thus, we could for example write instead

$$\tilde{x}[n] = \frac{1}{N}\sum_{k=N}^{2N-1} A_k W_N^{-nk}.$$

→ This is because

$$A_{k+N} = \sum_{n=0}^{N-1} \tilde{x}[n] W_N^{n(k+N)}$$

$$= \sum_{n=0}^{N-1} \tilde{x}[n] e^{-j2\pi n(k+N)/N}$$

$$= \sum_{n=0}^{N-1} \tilde{x}[n] e^{-j2\pi nk/N} e^{-j2\pi nN/N}$$

$$= \sum_{n=0}^{N-1} \tilde{x}[n] e^{-j2\pi nk/N} \underbrace{e^{-j2\pi n}}_{1}$$

$$= \sum_{n=0}^{N-1} \tilde{x}[n] W_N^{nk} = \underline{\underline{A_k}}$$

$$\text{So } \frac{1}{N}\sum_{k=N}^{2N-1} A_k W_N^{-nk} = \frac{1}{N}\sum_{k=0}^{N-1} A_{k+N} W_N^{-n(k+N)}$$

$$= \frac{1}{N}\sum_{k=0}^{N-1} A_k W_N^{-n(k+N)}$$

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

$$= \frac{1}{N}\sum_{k=0}^{N-1} A_k e^{j2\pi nk/N} \underbrace{e^{j2\pi n}}_{1}$$

$$= \tilde{x}[n].$$

- So we have the following:

① $\tilde{x}[n]$ periodic with period N, specified completely by any N consecutive samples.

② A_k periodic with period N, specified completely by any N consecutive samples.

③ To find the fundamental period $0 \le k \le N-1$ of the A_k's, take dot products between the basis functions $W_N^{-nk} = e^{j2\pi nk/N}$ and any one period of $\tilde{x}[n]$.

④ To find the fundamental period $0 \le n \le N-1$ of $\tilde{x}[n]$, add up any one period of the A_k^s times their respective basis functions, multiply by $\frac{1}{N}$, and keep the resulting samples for $0 \le n \le N-1$ only.

DFT

- Now suppose that we acquire N samples <u>only</u> of a discrete-time signal or that we sample a continuous-time signal for a finite time so that we get N samples <u>only</u>.
- This gives us a finite length discrete-time signal defined for $0 \le n \le N-1$ <u>only</u>.
- Suppose further that
  ① We want a frequency domain representation of $x[n]$ to facilitate processing the signal with an LTI system
  ② We want to be able to process $x[n]$ and the frequency representation with a computer (DSP).

- The first thing we might try is to zero pad $x[n]$ to infinite length to get the discrete-time signal

$$\hat{x}[n] = \begin{cases} x[n], & 0 \le n \le N-1 \\ 0, & \text{other.} \end{cases}$$

- Then we could compute the DTFT

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

⇒ But this is <u>no good</u>, because ω is a continuous variable. So the function $\hat{X}(e^{j\omega})$ cannot be represented in the DSP chip.

⇒ The solution is to <u>assume</u> that $x[n]$, $0 \le n \le N-1$, is <u>one</u> <u>period</u> of the periodic discrete-time signal $\tilde{x}[n]$.

⇒ Then we can write a DFS for $\tilde{x}[n]$.

→ In the computer, the time domain signal is represented by the fundamental period of $\tilde{x}[n]$, which is the N samples $x[n]$, $0 \le n \le N-1$.

→ The frequency domain version is represented by the fundamental period of the DFS coefficients A_k, which will now be written as $X[k]$, $0 \le k \le N-1$.

- Writing the DFS in terms of $x[n]$ and $X[k]$, we have

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi nk/N}$$

$$= \sum_{n=0}^{N-1} x[n] W_N^{nk}, \quad 0 \le k \le N-1 \quad (8.67)$$

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

$$= \frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-nk}, \quad 0 \le n \le N-1 \quad (8.68)$$

- Both $x[n]$ and $X[k]$ are finite length and can be stored and processed in a computer.

- Eq. (8.67) is called the <u>Discrete Fourier Transform</u> (DFT) of $x[n]$.

- Eq. (8.68) is called the Inverse Discrete Fourier Transform (IDFT) of $X[k]$.

- We often write

$$X[k] = \mathrm{DFT}\{x[n]\}$$

$$\text{or} \quad X[k] = \mathrm{DFT}_N\{x[n]\}$$

$$x[n] = \mathrm{IDFT}\{X[k]\}$$

$$\text{or} \quad x[n] = \mathrm{IDFT}_N\{X[k]\}$$

$$x[n] \xleftrightarrow{\mathrm{DFT}} X[k].$$

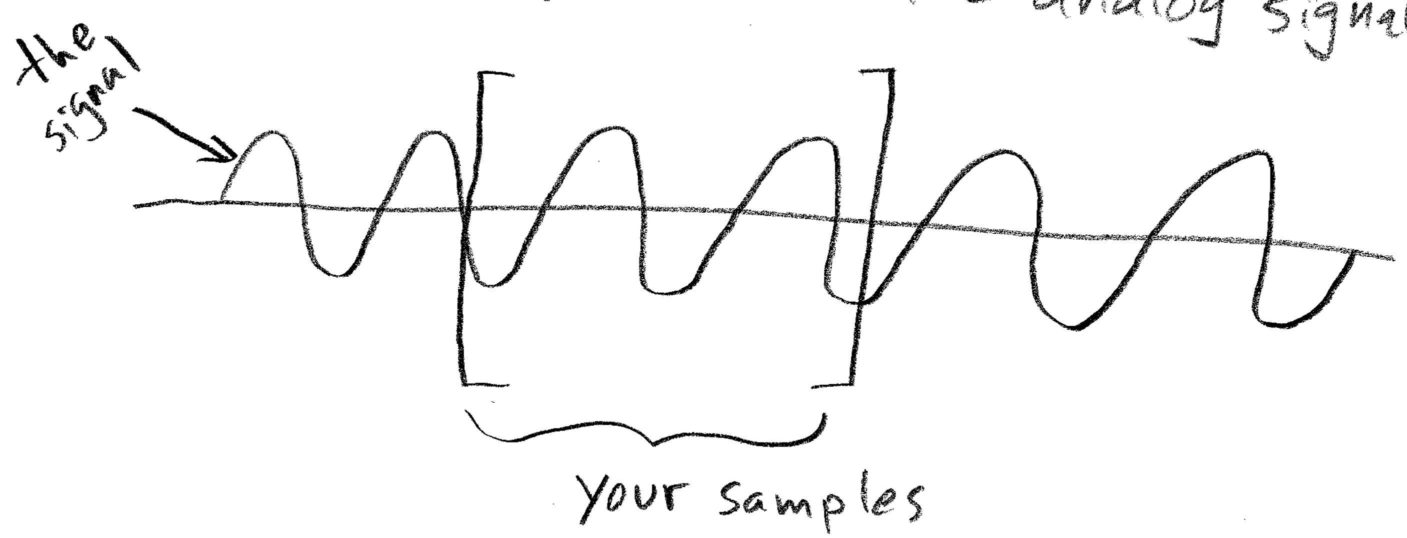
☆☆☆ Always remember:

- There is no inherent frequency representation for a finite length batch of samples $x[n]$.
- The DFT is a frequency representation for a periodic infinite-length signal $\tilde{x}[n]$ where it is assumed that the N samples $x[n]$ are one period of $\tilde{x}[n]$.
- The DFT coefficients $X[k]$ for $0 \le k \le N-1$ are one period of the periodic DFS spectrum of $\tilde{x}[n]$.

- The DFT representation is inherently discrete and periodic in both domains.
- The inherent implied periodicity in the time domain leads to some peculiarities of the DFT that will seem weird and confusing if you forget these things.

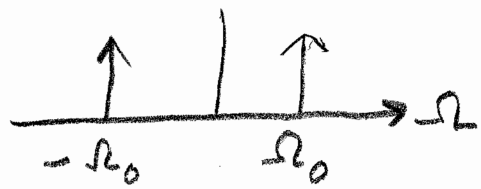
<u>EX</u>:

- you want to study a sinusoidal oscillation that is present in a communication system.
- You take a "test set" out into the field and acquire 1024 samples of the signal.
- Your samples do not capture an integer number of periods of the analog signal:



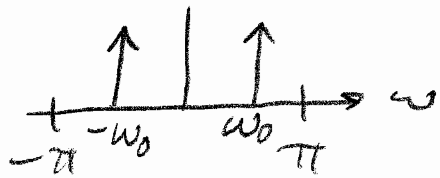
- You compute the DFT.
- You expect to see only two nonzero DFT coefficients, since

$$\cos \Omega_0 t \overset{\mathcal{F}}{\longleftrightarrow}$$



and

$$\cos \omega_0 n \overset{\mathcal{F}}{\longleftrightarrow}$$



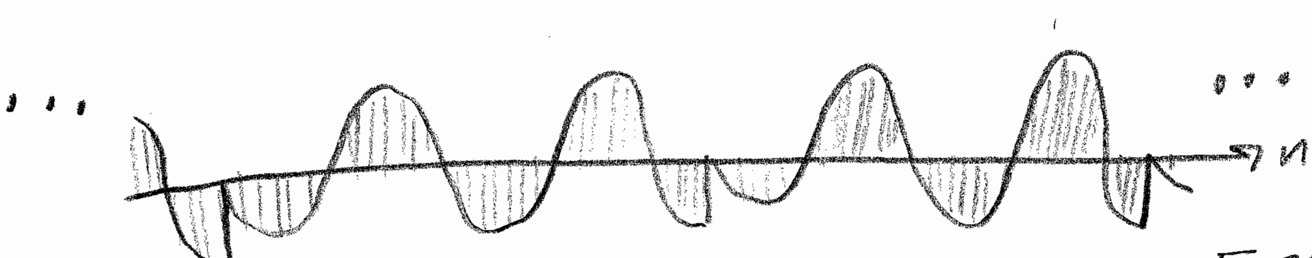
- But instead you get boatloads of nonzero DFT coefficients.

→ Why?

→ Because you computed a frequency representation <u>not</u> for a sinusoid, but for the periodic extension $\tilde{x}[n]$ of your samples, which looks like this:

(more on this much later if we have time)

$\tilde{x}[n]$



- The basis functions $W_N^{-nk} = e^{+j2\pi nk/N}$ are orthogonal.
- This leads to a formula that can be quite useful in working out DFT's by hand.
  → Taking the dot product between the basis functions $e^{j2\pi nk/N}$ and $e^{j2\pi nm/N}$, we get

$$\left\langle e^{j2\pi nk/N}, e^{j2\pi nm/N} \right\rangle = \sum_{n=0}^{N-1} e^{j2\pi nk/N} e^{-j2\pi nm/N}$$

$$\boxed{\sum_{n=A}^{B} a^n = \frac{a^A - a^{B+1}}{1-a}}$$

$$= \sum_{n=0}^{N-1} e^{j2\pi n(k-m)/N} \quad (☺)$$

$$= \frac{\left[e^{j2\pi(k-m)/N}\right]^0 - \left[e^{j2\pi(k-m)/N}\right]^N}{1-e^{j2\pi(k-m)/N}}$$

$$= \frac{1-e^{j2\pi(k-m)}}{1-e^{j2\pi(k-m)/N}} \quad (☹)$$

→ if $k=m$, then

$$(☺) = \sum_{n=0}^{N-1} 1 = N$$

→ if $k \neq m$, then

$$(☹) = \frac{1-1}{1-(\text{not } 1)} = \frac{\text{zero}}{\text{not zero}} = 0.$$

→ Thus, we get the useful formula

$$☆ \quad \sum_{n=0}^{N-1} e^{j2\pi n(k-m)/N} = \begin{cases} N, & k=m \\ 0, & \text{other} \end{cases}$$

Valid for $n, k, m \in \mathbb{Z}$ and $N \in \mathbb{N}$.

Computational Complexity of the DFT

- The DFT equation (8.67) from page 5.26 is:

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{nk}$$

→ There are N coefficients $X[k]$ for $0 \leq k \leq N-1$.

→ Each one requires a sum involving N multiply-adds.

→ So the complexity is N^2 complex multiply adds.

- The term "Fast Fourier Transform" (FFT) refers generically to tricky methods for speeding up the DFT computation.

- The general strategy is to factor the DFT calculation in a way that maximizes the opportunity to re-use partial products.

- The two classic algorithms are called "decimation in time," where the factoring is done in the time domain, and "decimation in frequency," where the factoring is done in the frequency domain.

- If N is a power of 2, these FFT algorithms lower the computational complexity from N^2 to $N \log_2 N$.

  → If $N = 2^{10} = 1024$,

$$\left.\begin{array}{r} \text{then } N^2 = 1{,}048{,}576 \\ N \log_2 N = 10{,}240 \end{array}\right\} \text{ wow!!}$$

- FFT algorithms are discussed in some detail in Chapter 9 of the book.

Matrix Formulation

Let

$$D_N = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N & W_N^2 & \cdots & W_N^{N-1} \\ 1 & W_N^2 & W_N^4 & \cdots & W_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W_N^{N-1} & W_N^{2(N-1)} & \cdots & W_N^{(N-1)^2} \end{bmatrix}.$$

Then

$$D_N^{-1} = \frac{1}{N} D_N^{*} = \frac{1}{N} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N^{-1} & W_N^{-2} & \cdots & W_N^{-(N-1)} \\ 1 & W_N^{-2} & W_N^{-4} & \cdots & W_N^{-2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W_N^{-(N-1)} & W_N^{-2(N-1)} & \cdots & W_N^{-(N-1)^2} \end{bmatrix}$$

$\longrightarrow$

- In terms of these matrices, the DFT is

$$\begin{bmatrix} X[0] \\ X[1] \\ \vdots \\ X[N-1] \end{bmatrix} = D_N \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix}$$

and the IDFT is

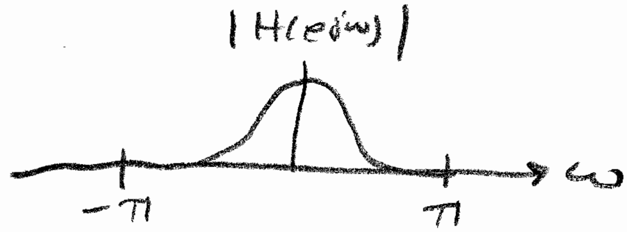
$$\begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix} = D_N^{-1} \begin{bmatrix} X[0] \\ X[1] \\ \vdots \\ X[N-1] \end{bmatrix}.$$

Interpretation of the Frequency Content

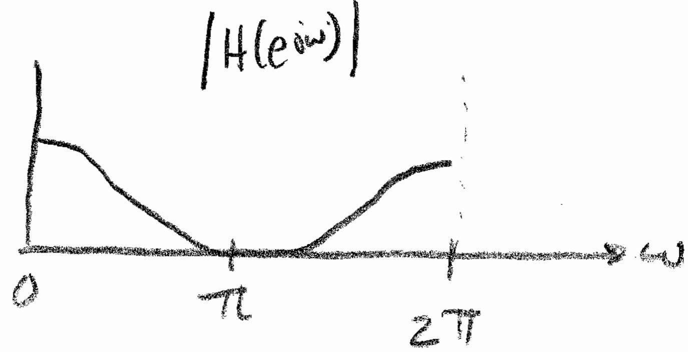
- Recall that the DTFT of any discrete-time signal is 2π-periodic.

  → For example, you could have a lowpass filter with impulse response $h[n]$ and frequency response $H(e^{j\omega})$.

  → If you graph $H(e^{j\omega})$ from $-\pi$ to π, it's clear that it's a lowpass filter, e.g.



  → But if you graph it from 0 to π, it's not so clear unless you remember that $H(e^{j\omega})$ is periodic:



- Because the DFT coefficients $X[k]$, $0 \le k \le N-1$, are one period of a periodic DFS spectrum, similar comments apply to them.

- Suppose you have a length-8 signal $x[n]$ and you take its 8-point DFT $X[k]$.

- The coefficients $X[k]$ are inner products between $x[n]$ and the basis functions

$$W_N^{-nk} = e^{j\frac{2\pi k}{8}n} \quad \text{for } 0 \le k \le N-1$$

→ These are complex sinusoids

$$\cos\left[\frac{2\pi k}{8}n\right] + j\sin\left[\frac{2\pi k}{8}n\right]$$

$$\text{for } 0 \le k \le N-1.$$

→ So the coefficient $X[k]$ goes with a sinusoid of frequency $\frac{2\pi k}{8}$ radians per sample.

→ In other words,

k	0	1	2	3	4	5	6	7
ω_k, rad/sample	0	$\frac{\pi}{4}$	$\frac{\pi}{2}$	$\frac{3\pi}{4}$	π	$\frac{5\pi}{4}$	$\frac{3\pi}{2}$	$\frac{7\pi}{4}$
Matlab array Index	1	2	3	4	5	6	7	8

⇒ Don't forget: for the DFT math, the index "k" starts at <u>zero</u>. For matlab arrays, the index starts at <u>one</u>.

- For spectral analysis, it is often more convenient to work with a zero-centered spectrum.
- Since the DFT is inherently periodic with period N,

  ($N=8$ here)

  $X[4]$ is the frequency coefficient for $k=4$ and $\omega = \frac{2\pi \cdot 4}{8} = \pi$ rad/sample,
  → but also for $k = 4-8 = -4$ and $\omega = -\pi \frac{\text{rad}}{\text{sample}}$.

  $X[5]$ is for $k=5$ and $\omega = \frac{5\pi}{4}$ rad/sample
  → but also for $k = 5-8 = -3$ and $\omega = \frac{-3\pi}{4} \frac{\text{rad}}{\text{sample}}$.

  $X[6]$ is for $k=6$ and $\omega = \frac{3\pi}{2}$ rad/sample,
  → but also for $k = 6-8 = -2$ and $\omega = -\frac{\pi}{2}$ rad/sample.

  $X[7]$ is for $k=7$ and $\omega = \frac{7\pi}{4}$ rad/sample,
  → but also for $k = 7-8 = -1$ and $\omega = -\frac{\pi}{4}$ rad/sample.

⇒ For this reason, people often think of:

- $X[0]$ as the "DC coefficient."
- $X[1]$, $X[2]$, $X[3]$ as the "positive frequency coefficients."
- $X[5]$, $X[6]$, $X[7]$ as the "negative frequency coefficients."
- $X[4]$ as being for frequencies $\pm\pi$, i.e., "both positive and negative."

- So, by circularly shifting the DFT array right by $\frac{N}{2}$ samples, one obtains the so-called "centered spectrum":

new k	0	1	2	3	4	5	6	7
original k	4	5	6	7	0	1	2	3
ω_k, rad/sample	$-\pi$	$-\frac{3\pi}{4}$	$-\frac{\pi}{2}$	$-\frac{\pi}{4}$	0	$\frac{\pi}{4}$	$\frac{\pi}{2}$	$\frac{3\pi}{4}$
Matlab Index	1	2	3	4	5	6	7	8

- Matlab provides a function "fftshift" for doing this.

EX: suppose xn is an 8-element Matlab array that holds the time domain samples of a length-8 discrete-time sequence.

Then Xk = fft(xn) will put the DFT coefficients into the array Xk as shown in the table on page 5.36.

And Xk2 = fftshift (fft(xn)) will put the centered DFT coefficients into the array Xk2 as shown in the table above on this page.

- If the finite-length sequence $x[n]$ was obtained by sampling a continuous-time signal $x(t)$, then you may want to relate the digital frequencies $\omega_k = \frac{2\pi k}{N}$ (rad/sample) to analog frequencies expressed in rad/sec.

- To do this, use the formula $\Omega = \frac{\omega}{T}$ from page 3.8-16 of the notes;

e.g., for a sampling interval of

$$T = 1 \text{ msec/sample} = 10^{-3} \frac{\text{sec}}{\text{sample}}$$

with $N = 8$,

$X[3]$ is the DFT coefficient for:

$$k = 3$$

$$\omega_3 = \frac{2\pi \cdot 3}{8} = \frac{3\pi}{4} \text{ rad/sample}$$

$$\Omega_3 = \omega_3 \cdot \frac{1}{T} = \frac{3\pi}{4} \frac{\text{rad}}{\text{sample}} \cdot \frac{1}{10^{-3} \text{ sec/sample}}$$

$$= \frac{3\pi}{4} \frac{\text{rad}}{\text{sample}} \cdot 10^{3} \frac{\text{sample}}{\text{sec}}$$

$$= \frac{3000\pi}{4} \frac{\text{rad}}{\text{sec}}$$

$$= 750\pi \text{ rad/sec.}$$

Relationship Between DFT, DTFT, and z-transform

- Recall that we have:

  → $x[n]$, a length-N sequence defined for $0 \leq n \leq N-1$. It has an N-point DFT $X[k]$ defined for $0 \leq k \leq N-1$. (we interpret $x[n]$ as one period of a periodic signal and $X[k]$ as one period of the periodic spectrum).

  → $\tilde{x}[n]$, the periodic extension of $x[n]$. It is defined on all of $\mathbb{Z}$. It has periodic DFS coefficients A_k with fundamental period given by $X[k]$.

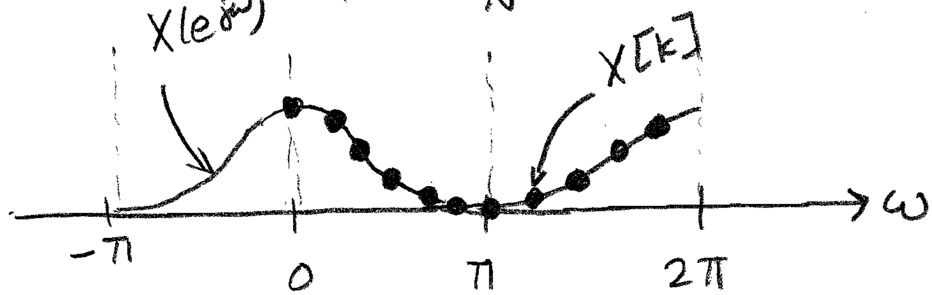
  → $\hat{x}[n]$, the zero padded infinite-length version of $x[n]$. It is defined by

$$\hat{x}[n] = \begin{cases} x[n], & 0 \leq n \leq N-1 \\ 0, & \text{other.} \end{cases}$$

  It has a DTFT $\hat{X}(e^{j\omega})$ and a z-transform $\hat{X}(z)$.

- The relationship between the DFT $X[k]$ and the DTFT $\hat{X}(e^{j\omega})$ is this:
  - → $X[k]$ is given by N samples of $\hat{X}(e^{j\omega})$ uniformly spaced between $-\pi$ and π.
  - → In other words,

$$X[k] = \hat{X}(e^{j\omega})\Big|_{\omega = \frac{2\pi k}{N}}, \quad 0 \leq k \leq N-1.$$

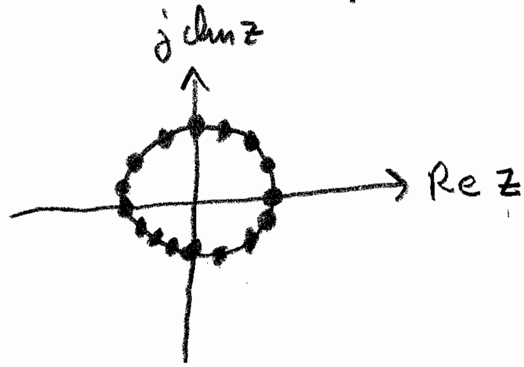


NOTE: given N samples $x[n]$, this implies that you can approximate $\hat{X}(e^{j\omega})$ with arbitrarily fine frequency resolution by zero padding $x[n]$ on the right and taking the DFT.

- For example, suppose you have 512 samples $x[n]$, and you want to plot $\hat{X}(e^{j\omega})$ with 2048 frequency samples uniformly placed between 0 and 2π.
  - → You simply add 1536 zeros to the right side of $x[n]$ and take a 2048-point DFT.
  - → The Matlab function "freqz" does this.

- Since $\hat{X}(e^{j\omega})$ is exactly $\hat{X}(z)$ evaluated on the unit circle of the z-plane, the relationship between the DFT $X[k]$ and the z-transform $\hat{X}(z)$ is that $X[k]$ is given by N samples of $\hat{X}(z)$ spaced uniformly around the unit circle.

$$\rightarrow X[k] = \hat{X}(z)\Big|_{z=e^{j2\pi k/N}}, \quad 0 \leq k \leq N-1.$$



DFT Properties

Linearity: if $x_1[n] \xleftrightarrow{DFT_N} X_1[k]$ and $x_2[n] \xleftrightarrow{DFT_N} X_2[k]$ and α and β are complex constants, then

$$\alpha x_1[n] + \beta x_2[n] \xleftrightarrow{DFT_N} \alpha X_1[k] + \beta X_2[k]$$

Note: $x_1[n]$ and $x_2[n]$ have to be the same length. If they aren't, then zero pad the shorter sequence.

Time Shift: if $x[n] \xleftrightarrow{DFT} X[k]$, then

$$x[\langle n-n_0\rangle_N] \xleftrightarrow{DFT} W_N^{kn_0} X[k].$$

Frequency Shift: If $x[n] \xleftrightarrow{DFT} X[k]$, then

$$W_N^{-k_0 n} x[n] \xleftrightarrow{DFT} X[\langle k-k_0\rangle_N].$$

Duality: if $x[n] \xleftrightarrow{DFT} X[k]$, then

$$X[n] \xleftrightarrow{DFT} N x[\langle -k\rangle_N].$$

Parseval Relation: if $x[n] \xleftrightarrow{DFT_N} X[k]$ and $y[n] \xleftrightarrow{DFT_N} Y[k]$, then

$$\sum_{n=0}^{N-1} |x[n]|^2 = \frac{1}{N}\sum_{k=0}^{N-1} |X[k]|^2$$

and

$$\sum_{n=0}^{N-1} x[n]\, y^*[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]\, Y^*[k].$$

Circular Convolution

(Also known as "periodic convolution")

- If $x[n]$ and $h[n]$ are two discrete-time signals with domain $\mathbb{Z}$, then the "plain old" convolution operation is

$$x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k]\,h[n-k]$$

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

- In this case, we have

$$x[n] * h[n] \xleftrightarrow{\text{DTFT}} X(e^{j\omega})H(e^{j\omega}).$$

- This "plain old" convolution is also called <u>linear</u> <u>convolution</u>.

- Linear convolution is useful because the output of an LTI system is given by the linear convolution of the input with the impulse response, i.e.

$$x[n] \longrightarrow \boxed{\overset{\text{LTI}}{H}} \longrightarrow y[n]$$

$$y[n] = x[n] * h[n]$$

$$Y(e^{j\omega}) = X(e^{j\omega})H(e^{j\omega})$$

- for two length-N discrete-time signals

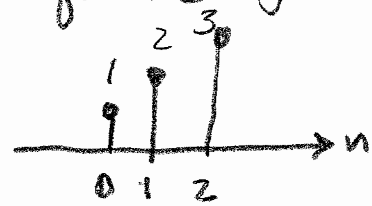
$$x[n] \xleftrightarrow{DFT} X[k] \quad \text{and} \quad h[n] \xleftrightarrow{DFT} H[k],$$

let $Y[k] = X[k]H[k]$

- Then $y[n] = IDFT_N\{Y[k]\}$ is a length-N discrete-time signal, but it is <u>not</u> the linear convolution of $x[n]$ with $y[n]$.
  - → It is a different kind of convolution called "<u>circular convolution</u>" or "<u>periodic convolution</u>".
- In the book,
  - → Circular convolution is denoted by the symbol Ⓝ, e.g. Ⓝ, ⑧, (256), etc., where it is <u>understood</u> that the two signals being convolved both have length N.
    - ⇒ in case the two signals have different lengths, it is customary to zero pad the shorter signal (on the right) to make its length equal to that of the longer signal.
  - → The symbols $\circledast$ and $\overset{N}{\circledast}$ are also <u>widely</u> <u>used</u> to indicate circular convolution.
- In order to understand the difference between linear convolution and circular convolution, let us first take a look at how the linear convolution of two finite length sequences should be defined.

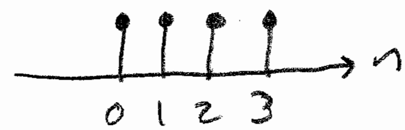
- Suppose $x[n]$ is a 3-point sequence given by

$$x[n] = [1 \;\; 2 \;\; 3]$$



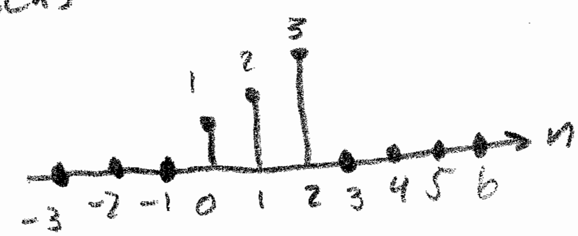
- Let $h[n]$ be a 4-point sequence given by

$$h[n] = [1 \;\; 1 \;\; 1 \;\; 1]$$



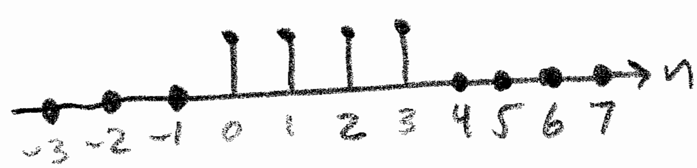
- The zero padded extension of $x[n]$ to all of $\mathbb{Z}$ is given by

$$\hat{x}[n] = \delta[n] + 2\delta[n-1] + 3\delta[n-2]$$



- The zero padded extension of $h[n]$ is

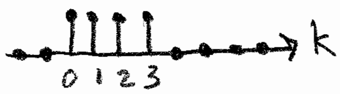
$$\hat{h}[n] = u[n] - u[n-4]$$



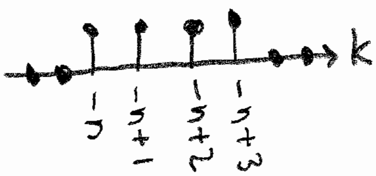
- Let $\hat{y}[n] = \hat{x}[n] * \hat{h}[n]$

$$= \sum_{k \in \mathbb{Z}} \hat{x}[k]\hat{h}[n-k]$$

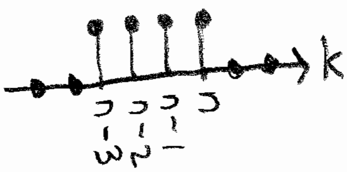
$\hat{h}[k]$



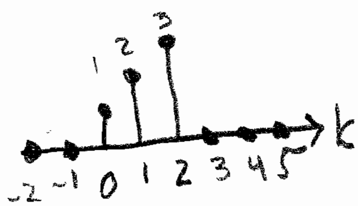
$\hat{h}[k - -n] = \hat{h}[n+k]$



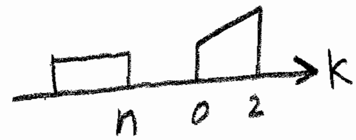
$\hat{h}[n-k]$



$\hat{x}[k]$



- For $n<0$, we have



So $\hat{x}[k]\hat{h}[n-k] = 0 \quad \forall k.$

So $\hat{y}[n] = 0$ for $n<0$.

- For $n=0$,

$$\hat{y}[n] = \sum_{k \in \mathbb{Z}} \hat{x}[k]\hat{h}[n-k]$$

$$= \sum_{k=0}^{0} \hat{x}[k]\hat{h}[0-k]$$

$$= \hat{x}[0]\hat{h}[0] = 1 \cdot 1 = 1$$

- For $n=1$,

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

$$= \hat{x}[0]\hat{h}[1] + \hat{x}[1]\hat{h}[0]$$

$$= 1 \cdot 1 + 2 \cdot 1 = 3$$

- For $n=2$,

$$\hat{y}[n] = \sum_{k=0}^{2} \hat{x}[k]\hat{h}[2-k]$$

$$= \hat{x}[0]\hat{h}[2] + \hat{x}[1]\hat{h}[1] + \hat{x}[2]\hat{h}[0]$$

$$= 1 \cdot 1 + 2 \cdot 1 + 3 \cdot 1 = 6$$

5.47

- For $n=3$,

$$\hat{y}[n] = \sum_{k=0}^{2} \hat{x}[k]\,\hat{h}[3-k]$$

$$= \hat{x}[0]\,\hat{h}[3] + \hat{x}[1]\,\hat{h}[2] + \hat{x}[2]\,\hat{h}[1]$$

$$= 1\cdot 1 + 2\cdot 1 + 3\cdot 1 = 6$$

- For $n=4$,

$$\hat{y}[n] = \sum_{k=1}^{2} \hat{x}[k]\,\hat{h}[4-k]$$

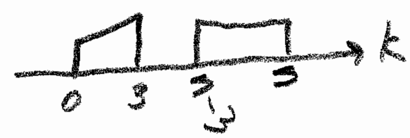
$$= \hat{x}[1]\,\hat{h}[3] + \hat{x}[2]\,\hat{h}[2]$$

$$= 2\cdot 1 + 3\cdot 1 = 5$$

- For $n=5$,

$$\hat{y}[n] = \sum_{k=2}^{2} \hat{x}[k]\,\hat{h}[5-k] = \hat{x}[2]\,\hat{h}[3]$$

$$= 3\cdot 1 = 3$$

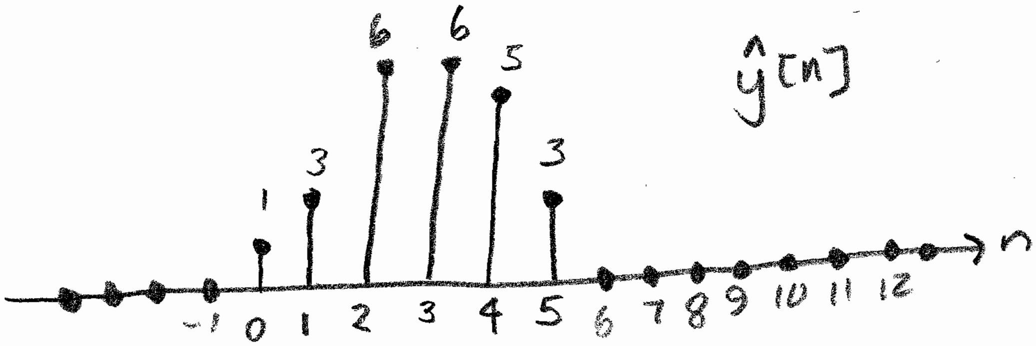
- And finally for $n \geq 6$, we have 

so $\hat{x}[k]\,\hat{h}[n-k] = 0 \quad \forall k.$

so $\hat{y}[n] = 0, \quad n \geq 6.$

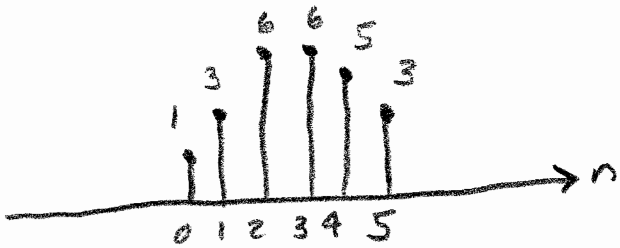
- All together, we have

$$\hat{y}[n] = \delta[n] + 3\delta[n-1] + 6\delta[n-2] + 6\delta[n-3] + 5\delta[n-4] + 3\delta[n-5].$$



- Throwing away the zeros, we have

$$y[n] = [1\ 3\ 6\ 6\ 5\ 3]$$



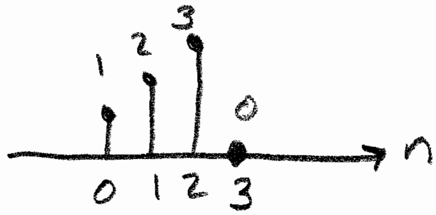
- Note that $y[n]$ has length $6 = 3+4-1$.

- In general, the linear convolution of an N_1-point finite length signal with an N_2-point finite length signal is a signal of length $N_1 + N_2 - 1$.

- In matlab, linear convolution of finite length sequences is done by calling "conv."

- Now let's compute the circular convolution of the sequences $x[n]$ and $h[n]$.

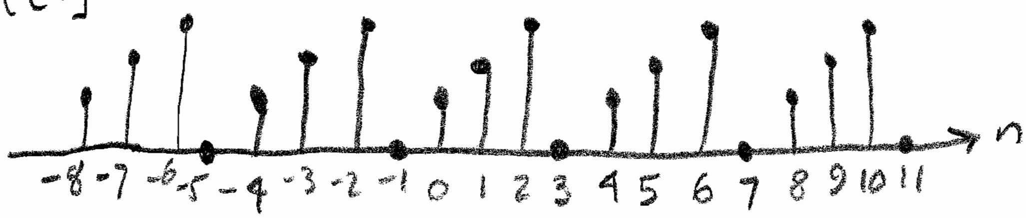
- Since they do not have the same length, we must first zero pad $x[n]$ to length 4 by adding zero(s) to the right. This gives us the new 4-point version of $x[n] = [1\ 2\ 3\ 0]$



- The periodic extension of the 4-point version of $x[n]$ is given by

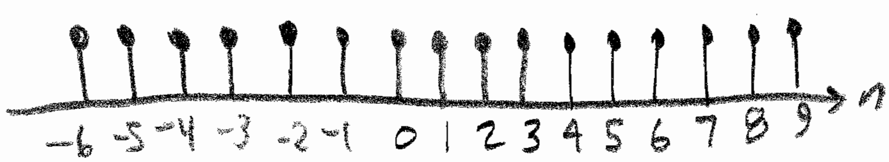
$$\tilde{x}[n] = \sum_{\ell=-\infty}^{\infty} \delta[n-4\ell] + 2\delta[n-1-4\ell] + 3\delta[n-2-4\ell]$$

$\tilde{x}[n]$



- The periodic extension of $h[n] = [1\ 1\ 1\ 1]$ is given by

$$\tilde{h}[n] = \sum_{\ell=-\infty}^{\infty} u[n-4\ell] - u[n-4-4\ell]$$



- The circular convolution of the two 4-point sequences $x[n]$ and $h[n]$ is given by

$$y_c[n] = \sum_{k=0}^{3} \tilde{x}[k]\tilde{h}[n-k], \quad 0 \leq n \leq 3.$$

→ Notice that $y_c[n]$ is a 4-point sequence, just like $x[n]$ and $h[n]$.

→ You can think of $y_c[n]$ as the linear convolution of $\hat{x}[n]$ with $\tilde{h}[n]$, i.e.

$$y_c[n] = \sum_{k=-\infty}^{\infty} \hat{x}[k]\tilde{h}[n-k], \quad 0 \leq n \leq 3.$$

→ By using modular arithmetic, you can write $y_c[n]$ directly in terms of the 4-point sequences $x[n]$ and $h[n]$ as

$$y_c[n] = \sum_{k=0}^{3} x[k]\,h[\langle n-k \rangle_4], \quad 0 \leq n \leq 3.$$

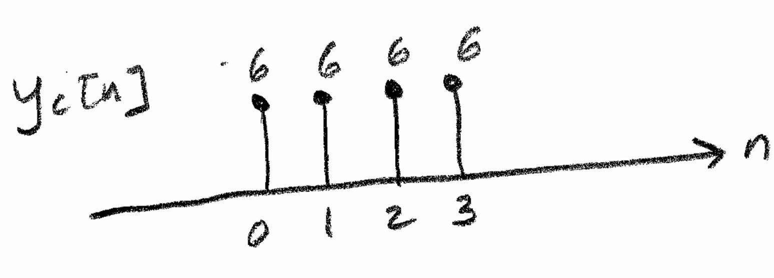
- More generally, the circular convolution of two N-point sequences $x[n]$ and $h[n]$ is an N-point sequence given by

$$y_c[n] = \sum_{k=0}^{N-1} x[k]\,h[\langle n-k \rangle_N], \quad 0 \leq n \leq N-1 \qquad (8.114)$$

- Now, referring back to the graph of $\tilde{h}[n]$ on page 5.50, let's compute the 4-point circular convolution of the two sequences

$$x[n] = [1\ 2\ 3\ 0] \quad \text{and} \quad h[n] = [1\ 1\ 1\ 1].$$

$$\begin{aligned}
y_c[n] &= x[n] \ (4)\ h[n] \\
&= \sum_{k=0}^{3} x[k]\, h[\langle n-k \rangle_4] \\
&= \sum_{k=0}^{3} x[k]\, \tilde{h}[n-k] \\
&= x[0]\tilde{h}[n] + x[1]\tilde{h}[n-1] + x[2]\tilde{h}[n-2] \\
&\qquad + x[3]\tilde{h}[n-3] \\
&= x[0]\cdot 1 + x[1]\cdot 1 + x[2]\cdot 1 + x[3]\cdot 1 \\
&= 1 + 2 + 3 + 0 \\
&= 6, \qquad 0 \le n \le 3
\end{aligned}$$



$\Rightarrow y_c[n] = [6\ 6\ 6\ 6]$

$\Rightarrow$ This is <u>not</u> the same as the linear convolution result given on page 5.50!!

- Another way to think about it: the circular convolution $x[n] \circledN h[n]$ can be written as the sequence $x[n]$ premultiplied by a circulant matrix constructed from $h[n]$:

$$\begin{bmatrix} y_c[0] \\ y_c[1] \\ y_c[2] \\ \vdots \\ y_c[N-1] \end{bmatrix} = \begin{bmatrix} h[0] & h[N-1] & h[N-2] & \cdots & h[1] \\ h[1] & h[0] & h[N-1] & \cdots & h[2] \\ h[2] & h[1] & h[0] & \cdots & h[3] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h[N-1] & h[N-2] & h[N-3] & \cdots & h[0] \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ \vdots \\ x[N-1] \end{bmatrix}$$

<u>FACT</u>: Circular convolution is commutative. That is, for two N-point sequences $x[n]$ and $h[n]$,

$$x[n] \circledN h[n] = h[n] \circledN x[n].$$

<u>Proof</u>: We will show on page 5.54 that

$$\mathrm{DFT}_N\{x[n] \circledN h[n]\} = X[k]H[k].$$

It follows immediately that

$$\begin{aligned} x[n] \circledN h[n] &= \mathrm{IDFT}_N\{X[k]H[k]\} \\ &= \mathrm{IDFT}_N\{H[k]X[k]\} \\ &= h[n] \circledN x[n]. \quad \text{QED.} \end{aligned}$$

More DFT Properties

Circular Convolution Property:

if $x[n] \overset{DFT_N}{\longleftrightarrow} X[k]$ and $h[n] \overset{DFT_N}{\longleftrightarrow} H[k]$,

then $y[n] = x[n] \, \text{Ⓝ} \, h[n] \overset{DFT_N}{\longleftrightarrow} X[k]H[k]$.

Proof:

$$Y[k] = \sum_{n=0}^{N-1} y[n] W_N^{kn}$$

$$= \sum_{n=0}^{N-1} \left(\sum_{m=0}^{N-1} x[m] h[\langle n-m \rangle_N] \right) W_N^{kn}. \quad (*)$$

Now, $\langle n-m \rangle_N = (n-m) \bmod N$. So we can write

$\ell = \langle n-m \rangle_N$ which implies $0 \leq \ell \leq N-1$ and

$n-m = \ell + Nr$ for some integer r. Then $n = m+\ell+Nr$.

Then $$(*) = \sum_{\ell=0}^{N-1} \sum_{m=0}^{N-1} x[m] h[\ell] \; W_N^{k(m+\ell+Nr)}$$

$$= \sum_{m=0}^{N-1} x[m] \sum_{\ell=0}^{N-1} h[\ell] W_N^{k\ell} \, W_N^{km} \, W_N^{rkN}. \quad (**)$$

But $W_N^{rkN} = e^{-j2\pi rkN/N} = e^{-j2\pi rk} = 1$.

So $$(**) = \sum_{m=0}^{N-1} x[m] W_N^{km} \sum_{\ell=0}^{N-1} h[\ell] W_N^{k\ell}$$

$$= X[k]H[k]. \quad \text{QED.}$$

Modulation Property (Frequency Convolution):

if $x[n] \overset{DFT_N}{\longleftrightarrow} X[k]$ and $h[n] \overset{DFT_N}{\longleftrightarrow} H[k]$, then

$$x[n]h[n] \overset{DFT_N}{\longleftrightarrow} \frac{1}{N}\sum_{m=0}^{N-1} X[m]\,H[\langle k-m\rangle_N]$$

$$= \frac{1}{N} X[k] \, Ⓝ \, H[k].$$

DFT Symmetry Properties

① Conjugation: $x^*[n] \overset{DFT}{\longleftrightarrow} X^*[\langle -k\rangle_N]$

$x^*[\langle -n\rangle_N] \overset{DFT}{\longleftrightarrow} X^*[k]$

② If $x[n]$ is real, then $X[k]$ is periodically conjugate symmetric. This means:

$X[k] = X^*[\langle -k\rangle_N]$ ($X[k]$ periodically conjugate symmetric)

$\mathrm{Re}\{X[k]\} = \mathrm{Re}\{X[\langle -k\rangle_N]\}$ (Real part periodically even)

$\mathrm{Im}\{X[k]\} = -\mathrm{Im}\{X[\langle -k\rangle_N]\}$ (Imaginary part periodically odd)

$|X[k]| = |X[\langle -k\rangle_N]|$ (Magnitude periodically even)

$\arg X[k] = -\arg X[\langle -k\rangle_N]$ (Phase periodically odd)

③ If $x[n] = x_{cs}[n] + x_{ca}[n]$ where

$$x_{cs}[n] = \frac{1}{2}\left\{x[n] + x^*[\langle -n \rangle_N]\right\}$$ is periodically conjugate symm.

and $$x_{ca}[n] = \frac{1}{2}\left\{x[n] - x^*[\langle -n \rangle_N]\right\}$$ is periodically conjugate antisymmetric,

then
$$x_{cs}[n] \overset{DFT}{\longleftrightarrow} \mathrm{Re}\left\{X[k]\right\}$$
$$x_{ca}[n] \overset{DFT}{\longleftrightarrow} j\,\mathrm{Im}\left\{X[k]\right\}.$$

For $x[n]$ real, ③ becomes;

④ if $x[n]$ is real and $x[n] = x_{pe}[n] + x_{po}[n]$

where $$x_{pe}[n] = \frac{1}{2}\left\{x[n] + x[\langle -n \rangle_N]\right\}$$ is periodically even

and $$x_{po}[n] = \frac{1}{2}\left\{x[n] + x[\langle -n \rangle_N]\right\}$$ is periodically odd,

then
$$x_{pe}[n] \overset{DFT}{\longleftrightarrow} \mathrm{Re}\left\{X[k]\right\}$$
$$x_{po}[n] \overset{DFT}{\longleftrightarrow} j\,\mathrm{Im}\left\{X[k]\right\}.$$

It follows from ③, ④, and duality that

⑤ If $x[n]$ is complex, then

$$\mathrm{Re}\{x[n]\} \overset{DFT}{\longleftrightarrow} X_{cs}[k] = \frac{1}{2}\left\{X[k] + X^*[\langle -k\rangle_N]\right\}$$

$$j\,\mathrm{Im}\{x[n]\} \overset{DFT}{\longleftrightarrow} X_{ca}[k] = \frac{1}{2}\left\{X[k] - X^*[\langle -k\rangle_N]\right\}$$

⑥ If $x[n]$ is real and periodically even, then $X[k]$ is real and periodically even.

⑦ If $x[n]$ is pure imaginary and periodically odd, then $X[k]$ is pure imaginary and periodically odd.

FFT Note: FFT algorithms give the best speedup when N, the length of the signal, is a power of 2. Thus, historically it was common practice to zero pad the signal to make the length a power of 2, e.g. 128, 256, 512, 1024, 2048, etc.

But most modern FFT algorithms (like FFTW, which is in the most recent version of matlab) have good odd radix and prime factor FFT implementations that are fast.

- If you are only going to do a few FFT's, you can zero pad to a power of two or not... it won't make a lot of difference.

- If you need to do a large number of N-point DFT's, it's worth analyzing the tradeoff between using N-point odd radix or prime factor FFT's or zero padding to the nearest power of two.

Using the DFT for Linear Convolution

- Pointwise multiplication of DFT's corresponds to circular convolution in the time domain.

- But circular convolution is not what we want usually.

- Most of the time, what we really want is linear convolution:

$$x[n] \longrightarrow \boxed{\begin{matrix}\text{LTI}\\ H\end{matrix}} \longrightarrow y[n] = x[n] * h[n]$$

- So an important question is: can we implement linear convolution using the DFT?

- In other words, can we use the circular convolution $x[n] \, \text{Ⓝ} \, h[n]$ to implement the linear convolution $x[n] * h[n]$?

→ The answer is YES.

- The circular convolution is

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

- Here, $h[n]$ is circularly shifted and periodically time reversed.

- Values that are shifted out of the right side of $h[n]$ get shifted back in on the left side... and that's the main difference between circular convolution and linear convolution.

→ This can be fixed up by zero padding $h[n]$ so that only zeros get shifted back in.

→ Since the two sequences must have the same lengths to multiply their DFT's, this implies that $x[n]$ must also be zero padded.

→ If a sufficient number of zeros are added, then the circular convolution of the padded sequences will be the same as the linear convolution of the original sequences.

- Here's how it works:

  - Suppose $x[n]$ has length N_1 and $h[n]$ has length N_2.
  - To compute the linear convolution

$$y[n] = x[n] * h[n],$$

    follow these steps:

    1. Zero pad both $x[n]$ and $h[n]$ on the right out to a length $N \geq N_1 + N_2 - 1$.
    2. Compute the N-point DFT's $X[k]$ and $H[k]$ of the zero-padded sequences.
    3. Set $Y[k] = X[k]H[k]$
    4. $y[n] = \text{IDFT}_N\{Y[k]\}$.

- For speed, N is often rounded up to the nearest power of two.

Online LTI Filtering with the DFT

- For real-time LTI signal processing, it's often the case that we want to compute the signal $y[n] = x[n] * h[n]$

$$x[n] \rightarrow \boxed{H} \rightarrow y[n] = x[n] * h[n]$$

where H is an FIR filter and the length of the input $x[n]$ is much much larger than the length of the impulse response $h[n]$...

so that by comparison the length of $x[n]$ is essentially infinite.

- In such cases, it's not practical to take the DFT of the whole signal $x[n]$.
- Instead, it's preferable to break $x[n]$ up into blocks and take the DFT of each block independently.
- Then we can multiply zero padded DFT's to linearly convolve each block with $h[n]$.
- But some method is needed to glue the blocks back together to find the "true" $y[n]$.

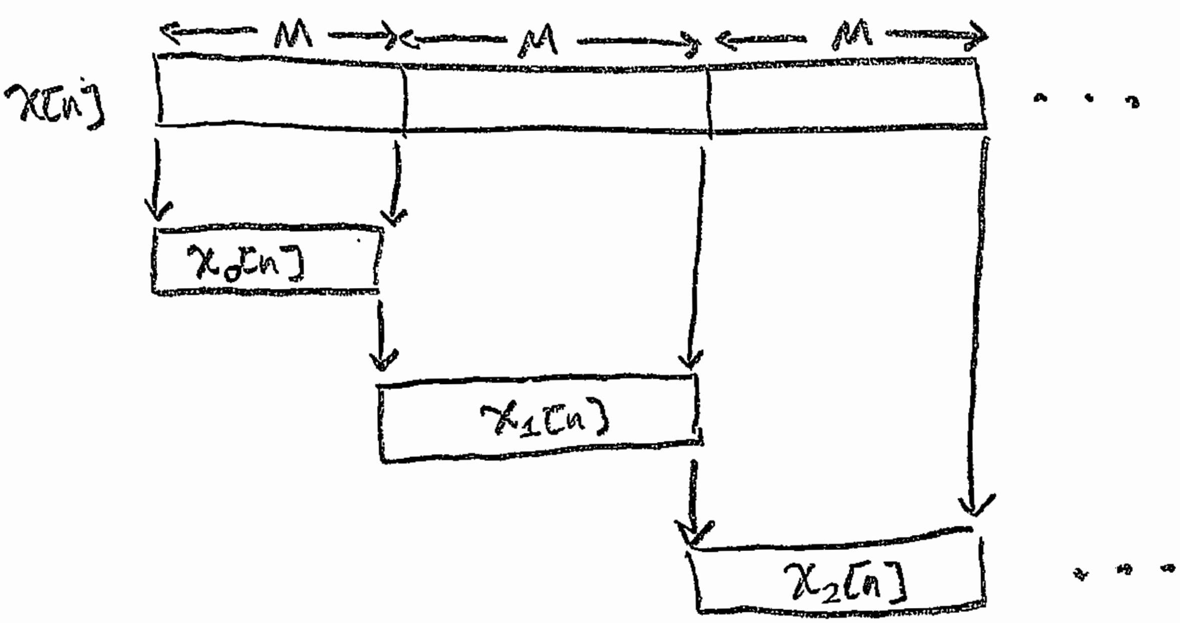
- Two popular techniques for doing this are the "overlap-add method" and the "overlap-save method".

Overlap-Add Method

- Suppose that $x[n]$ is very long and $h[n]$ has length L.
- We need to compute the linear convolution

$$y[n] = x[n] * h[n].$$

- With the overlap-add method, we break up $x[n]$ into non-overlapping blocks of length M each:



- For the i^{th} block, we have

$$x_i[n] = x[n+Mi], \quad 0 \le n \le M-1.$$

- In other words,

$$x[n] = \sum_{i=0}^{\infty} x_i[n-Mi].$$

- The output is given by

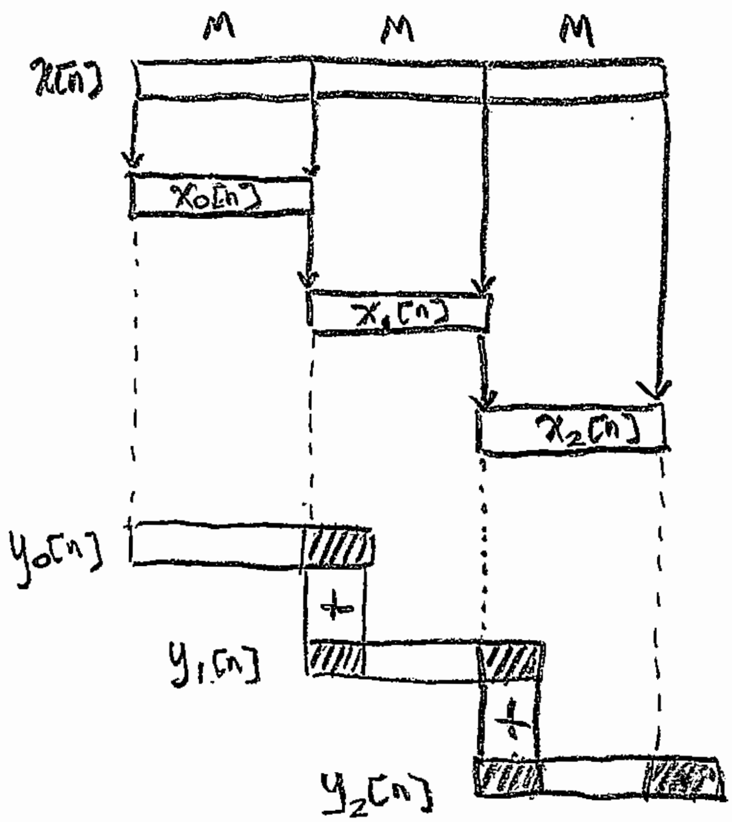
$$\begin{aligned} y[n] &= x[n] * h[n] \\ &= \sum_{i=0}^{\infty} x_i[n-Mi] * h[n] \\ &= \sum_{i=0}^{\infty} y_i[n-Mi], \end{aligned}$$

where $y_i[n] = x_i[n] * h[n]$.

- We compute each output block $y_i[n]$ by multiplying the DFT's of $x_i[n]$ and $h[n]$.
   → To compute linear convolution with DFT's, this means that $x_i[n]$ and $h[n]$ must be zero padded to length $N = M+L-1$

- So each output block $y_i[n]$ has length $N > M$.
- So each pair of output blocks has an overlap of $N - M = L - 1$ samples.

- The overlapping samples must be added to find $y[n]$ correctly.



- This implies that $y_1[n]$ must be computed by inverse DFT before the last $L-1$ samples of the first output block can be calculated.

Overlap-Save Method

- The overlap-save method differs from the overlap-add method in these ways:

  A. The input blocks <u>are</u> overlapping

  B. Each input block is <u>circularly</u> convolved with $h[n]$ by multiplication of DFTs... i.e., zero padding is <u>not</u> done to implement linear convolution of each input block with $h[n]$.

  C. The portion of each output block that is distorted by the "wraparound" effect of circular convolution is <u>discarded</u>. The rest of the output block is <u>saved</u>.

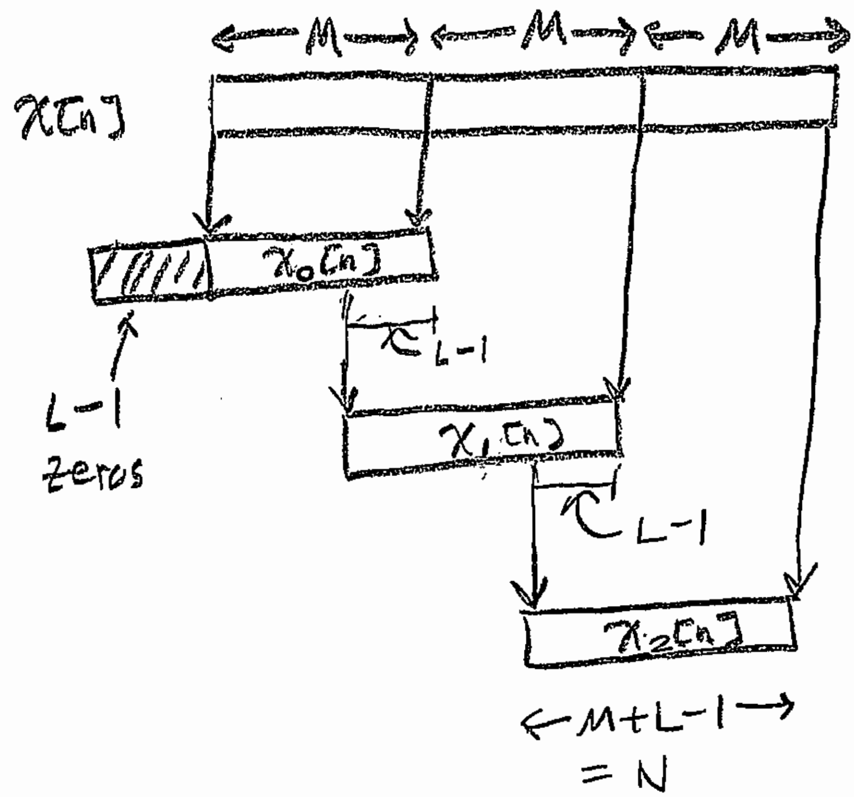
  D. The saved portions of the output blocks do <u>not</u> overlap. They are concatenated to get $y[n]$.

- As before, suppose that $x[n]$ is very long and $h[n]$ has length L.

- Let M be the number of saved samples in each output block.

- The overlapping input blocks have length $N = M+L-1$.

- For the first input block, $L-1$ zeros are appended on the left.



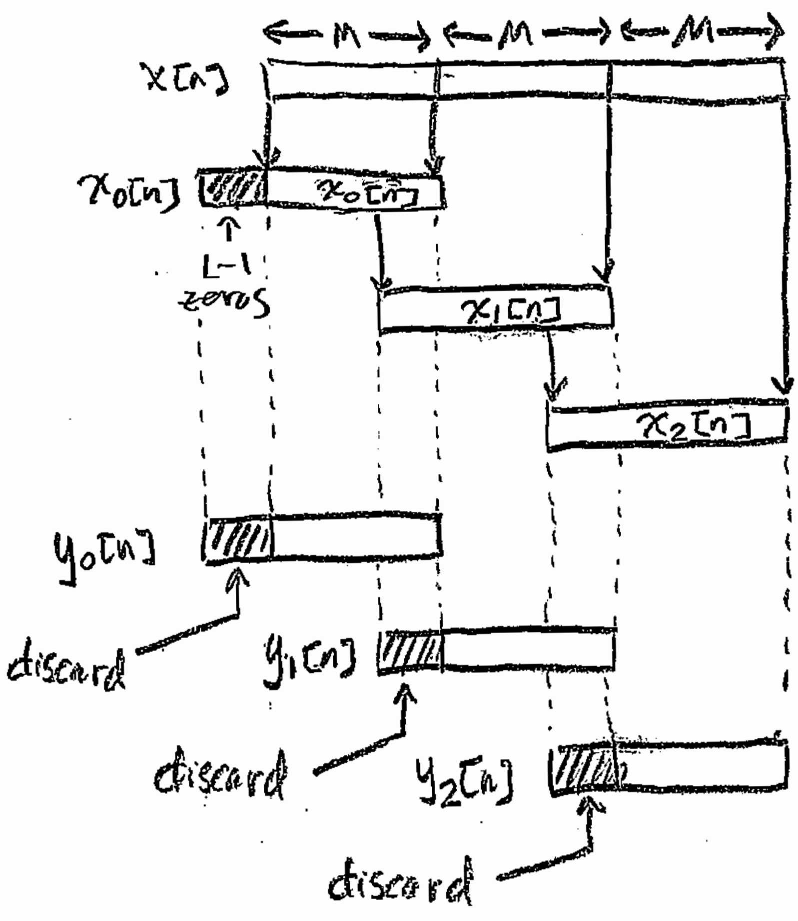
- For each input block, we compute the $N = M+L-1$ point circular convolution $x_i[n] \, (N) \, h[n]$. This is done by zero padding $h[n]$ on the right to length N and multiplying the N-point DFTs $X_i[k]$ and $H[k]$.

- We then compute $\text{IDFT}_N\{X_i[k]H[k]\}$.

  → The first $L-1$ samples are messed up by the wraparound effect of circular convolution... they are <u>discarded</u>.

  → The remaining M samples of the circular convolution are the same as in the linear convolution. They are <u>saved</u> as the output block $y_i[n]$.

  → To get $y[n]$, the output blocks are simply concatenated. They do not overlap.



- The main advantage compared to overlap-add is that the samples $y_1[n]$ can be output without waiting for the IDFT computation of $y_2[n]$.

WORKING DFT PROBLEMS BY HAND USING THE "W_N" NOTATION

- Recall the equations for the DFT and IDFT:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi nk/N}, \quad 0 \le k \le N-1 \qquad (8.67)$$

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j2\pi nk/N}, \quad 0 \le n \le N-1 \qquad (8.68)$$

- Given a sequence $x[n]$, Matlab computes the DFT (8.67) by directly plugging in the numbers.

- Similarly, given a DFT array $X[k]$, Matlab computes the IDFT (8.68) by directly plugging in the numbers.

- Example (circular convolution, $N=4$):

```
xn = [1 2 3 4];
hn = [-2 3 -1 -2];
Xk = fft(xn)

Xk =

  10.0000 + 0.0000i  -2.0000 + 2.0000i  -2.0000 + 0.0000i  -2.0000 - 2.0000i

Hk = fft(hn)

Hk =

  -2.0000 + 0.0000i  -1.0000 - 5.0000i  -4.0000 + 0.0000i  -1.0000 + 5.0000i

Yk = Xk .* Hk

Yk =

 -20.0000 + 0.0000i  12.0000 + 8.0000i   8.0000 + 0.0000i  12.0000 - 8.0000i

yn = ifft(Yk)

yn =

     3   -11    -9    -3
```

- But it is both cumbersome and inconvenient to work DFT problems this way by hand.

- Luckily, there is a much better way to work DFT problems by hand using the "W_N notation."

- Recall from page 5.4:

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

→ It is a constant for any fixed value of N.

→ But in most DFT problems, N is fixed. This is because all the signals in a typical DFT problem need to be the same length.

- otherwise, the DFT coefficients $X[k]$ for the different signals would be at different frequencies and they could not be combined.

- In terms of W_N, the DFT equation is given by

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{nk} \quad , \quad 0 \leq k \leq N-1 \quad (8.67),$$

- as we saw on page 5.26.

- When working DFT problems by hand, this gives us a convenient way to write the DFT $X[k]$ as a polynomial in W_N.

EX: $x[n] = [1 \ 2 \ 3 \ 4]$ $(N=4)$

$= \delta[n] + 2\delta[n-1] + 3\delta[n-2] + 4\delta[n-3], \quad 0 \leq n \leq 3.$

- Using eq. (8.67),

$$X[k] = \sum_{n=0}^{3} x[n] W_4^{nk}$$

$$= 1W_4^0 + 2W_4^k + 3W_4^{2k} + 4W_4^{3k}$$

$$= 1 + 2W_4^k + 3W_4^{2k} + 4W_4^{3k} \quad , \quad 0 \leq k \leq 3$$

- This also makes it easy to compute inverse DFT's using the W_N notation.

- If $N=4$, then for any length-4 signal $x[n]$, the DFT is given by

$$X[k] = \sum_{n=0}^{3} x[n] W_4^{nk}$$

$$= x[0] + x[1] W_4^{k} + x[2] W_4^{2k} + x[3] W_4^{3k} \qquad (*)$$

$$0 \le k \le 3$$

- For any length-4 DFT $X[k]$ that is written as a polynomial in W_N, we can invert the DFT and find $x[n]$ by comparing the polynomial $X[k]$ to $(*)$ above term-by-term and simply picking off the signal values $x[n]$.

<u>EX</u>: at the bottom of page 5.69, we had

$$X[k] = \underbrace{1}_{x[0]} + \underbrace{2}_{x[1]}W_4^{k} + \underbrace{3}_{x[2]} W_4^{2k} + \underbrace{4}_{x[3]} W_4^{3k}, \quad 0 \le k \le 3.$$

- comparing to $(*)$ above, we get:

So, by inspection,

$$x[n] = [1 \quad 2 \quad 3 \quad 4]$$

$$= \delta[n] + 2\delta[n-1] + 3\delta[n-2] + 4\delta[n-3],$$

$$0 \le n \le 3.$$

- Here are two important facts about W_N :

1. For any integer m, $W_N^{mN} = 1$.

Proof : $W_N^{mN} = (e^{-j2\pi/N})^{mN}$

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

$$= e^{-j2\pi m} = e^{j2\pi(\text{integer})} = \underline{\underline{1}}.$$

2. For any integer m, $W_N^{(m+N)} = W_N^m$

Proof : $W_N^{(m+N)} = W_N^m W_N^N$

$$= W_N^m \cdot 1 \quad \text{(using Fact ① above)}$$

$$= \underline{\underline{W_N^m}}.$$

EX : $W_4^6 = W_4^{(2+4)} = W_4^2 W_4^4 = W_4^2$

→ In other words, if $m > N$, then $W_N^m = W_N^{m \bmod N}$

→ You can think of this as a "wrapping" of the powers of W_N when the power is greater than N.

→ It is deeply related to the idea of "wrapping around" in wraparound (or circular) convolution.

Here is an example of computing a linear convolution by hand using the DFT and the W_N notation:

5. 25/20 pts. Let $h[n]$ and $x[n]$ be finite-length discrete-time signals given by

$$\begin{aligned} h[n] &= [-2 \quad 4 \quad -2] \\ &= -2\delta[n] + 4\delta[n-1] - 2\delta[n-2], \; 0 \le n \le 2, \end{aligned}$$

and

$$\begin{aligned} x[n] &= [3 \quad 2 \quad -1] \\ &= 3\delta[n] + 2\delta[n-1] - \delta[n-2], \; 0 \le n \le 2. \end{aligned}$$

Use the DFT to find the linear convolution $y_\ell[n] = h[n] * x[n]$.

$N_1 = N_2 = 3$. For linear convolution, zero pad to $N = N_1 + N_2 - 1 = 3+3-1 = 5$.

$h_5[n] = [-2 \quad 4 \quad -2 \quad 0 \quad 0]$

$x_5[n] = [3 \quad 2 \quad -1 \quad 0 \quad 0]$

$$H_5[k] = \sum_{n=0}^{4} h_5[n] W_5^{nk} = -2 + 4W_5^k - 2W_5^{2k}$$

$$X_5[k] = \sum_{k=0}^{4} x_5[n] W_5^{nk} = 3 + 2W_5^k - W_5^{2k}$$

$$Y_\ell[k] = H_5[k] X_5[k] = \left(-2 + 4W_5^k - 2W_5^{2k}\right)\left(3 + 2W_5^k - W_5^{2k}\right)$$

$$\begin{aligned} = &-6 - 4W_5^k + 2W_5^{2k} \\ &+ 12W_5^k + 8W_5^{2k} - 4W_5^{3k} \\ &- 6W_5^{2k} - 4W_5^{3k} + 2W_5^{4k} \end{aligned}$$

$$Y_\ell[k] = -6 + 8W_5^k + 4W_5^{2k} - 8W_5^{3k} + 2W_5^{4k}$$

$$Y_\ell[k] = \sum_{n=0}^{4} y_\ell[n] W_5^{nk} = y_\ell[0] + y_\ell[1] W_5^k + y_\ell[2] W_5^{2k} + y_\ell[3] W_5^{3k} + y_\ell[4] W_5^{4k}$$

$$\boxed{\begin{aligned} y_\ell[n] &= [-6 \quad 8 \quad 4 \quad -8 \quad 2] \\ &= -6\delta[n] + 8\delta[n-1] + 4\delta[n-2] - 8\delta[n-3] + 2\delta[n-4], \; 0 \le n \le 4 \end{aligned}}$$

- In the example on page 5.72, we used the DFT to compute the linear convolution of $x[n]$ and $h[n]$.

  → Because pointwise multiplication of DFT's can only give us circular convolution, we had to zero pad both signals to a minimum length of $N_1 + N_2 - 1 = 3+3-1 = 5$.

- Recall from pages 5.58 – 5.60 that the added zeros prevent the "wraparound effect" of circular convolution by ensuring that only zeros get wrapped around.

  → This makes the circular convolution of the zero padded signals equal to the linear convolution of the original (un-padded) signals.

- Here's another way to think about that:

  - In the linear convolution example on page 5.72, the added zeros ensure that the highest power of W_5 that can occur in $X[k]$ and $H[k]$ is W_5^{2k}.

  - So the highest power of W_5 that can occur in $Y[k]$ is $(W_5^{2k})^2 = W_5^{4k} = (W_5^4)^k$

    → And W_5^4 does not wrap.

(This will become clearer on the next couple of pages)

- If we had instead computed the circular convolution of two arbitrary length-5 signals,
  - → Then there would be no guarantee of having two zeros at the end of $x[n]$ and $h[n]$.
  - → This would mean that the highest power of W_5 that could occur in $X[k]$ and $H[k]$ would be W_5^{4k}
  - → It would mean that the highest power of W_5 that could occur in $Y[k]$ would be

$$\left(W_5^{4k}\right)^2 = W_5^{8k} = \left(W_5^{8}\right)^k$$

→ And $W_5^8 = W_5^{(5+3)} = W_5^3$ does wrap.

- And looking back at the linear convolution example on page 5.72, we see that zero padding the length-3 signals $x[n]$ and $h[n]$ to length $N=5$ guarantees that this kind of wrapping of W_5 will not occur when we multiply the DFT's pointwise.
- That is another way to think about how zero padding the signals to length $N_1 + N_2 - 1$ prevents the wraparound effect of circular convolution.

- Now we will work a circular convolution example by hand using the DFT and the W_N notation.

  → We will see that, this time, the W_N wraparound effect <u>will</u> occur (because there is no zero padding).

  → This is a deep way to think about the difference between computing circular convolution and linear convolution with the DFT.

<u>EX</u>: (circular convolution by DFT)

let $x[n]$ and $h[n]$ be 4-point discrete-time signals given by

$$x[n] = [3 \;\; 1 \;\; 4 \;\; 1]$$
$$= 3\delta[n] + \delta[n-1] + 4\delta[n-2] + \delta[n-3], \quad 0 \le n \le 3,$$

$$h[n] = [5 \;\; -9 \;\; 2 \;\; -6]$$
$$= 5\delta[n] - 9\delta[n-1] + 2\delta[n-2] - 6\delta[n-3], \quad 0 \le n \le 3.$$

Use the DFT to find the 4-point <u>circular</u> convolution

$$y_c[n] = x[n] \; 4 \; h[n]$$

<u>$N=4$</u>, so we will be using W_4.

→

$$X[k] = \sum_{n=0}^{3} x[n] W_4^{nk}$$

$$= 3W_4^0 + 1W_4^k + 4W_4^{2k} + 1W_4^{3k}$$

$$= 3 + W_4^k + 4W_4^{2k} + W_4^{3k} \quad , \quad 0 \le k \le 3.$$

$$H[k] = \sum_{n=0}^{3} h[n] W_4^{nk}$$

$$= 5W_4^0 - 9W_4^k + 2W_4^{2k} - 6W_4^{3k}$$

$$= 5 - 9W_4^k + 2W_4^{2k} - 6W_4^{3k} \quad , \quad 0 \le k \le 3.$$

$$Y_c[k] = X[k]H[k]$$

$$= (3 + W_4^k + 4W_4^{2k} + W_4^{3k})(5 - 9W_4^k + 2W_4^{2k} - 6W_4^{3k})$$

$$\begin{aligned} = 15 - 27W_4^k &+ 6W_4^{2k} - 18W_4^{3k} \\ + 5W_4^k &- 9W_4^{2k} + 2W_4^{3k} - 6W_4^{4k} \\ &+ 20W_4^{2k} - 36W_4^{3k} + 8W_4^{4k} - 24W_4^{5k} \\ &\qquad + 5W_4^{3k} - 9W_4^{4k} + 2W_4^{5k} - 6W_4^{6k} \end{aligned}$$

$$Y_c[k] = 15 - 22W_4^k + 17W_4^{2k} - 47W_4^{3k} - 7W_4^{4k} - 22W_4^{5k} - 6W_4^{6k}$$

→ But, using the second fact on page 5.71, we see that there are "W_N wraparound effects" in $Y_c[k]$.

⟶

- This is another way to understand the "wraparound effect" of circular convolution.

- We have:

$$W_4^{4k} = (W_4^4)^k = 1^k = 1$$

$$W_4^{5k} = (W_4^{(4+1)})^k = (W_4^4 W_4^1)^k = W_4^k$$

$$W_4^{6k} = (W_4^{(4+2)})^k = (W_4^4 W_4^2)^k = W_4^{2k}$$

- Plugging these into $Y_c[k]$ from the bottom of page 5.76, we get

$$Y_c[k] = 15 - 22W_4^k + 17W_4^{2k} - 47W_4^{3k} \overbrace{-7 - 22W_4^k - 6W_4^{2k}}^{\text{wrapped}}$$

$$Y_c[k] = 8 - 44W_4^k + 11W_4^{2k} - 47W_4^{3k}, \quad 0 \leq k \leq 3$$

Now, comparing this to the definition (5.13) of the DFT

$$Y_c[k] = \sum_{n=0}^{3} y_c[n] W_4^{nk}$$

$$= y_c[0] + y_c[1] W_4^k + y_c[2] W_4^{2k} + y_c[3] W_4^{3k},$$

we see that $y_c[0] = 8$, $y_c[1] = -44$, $y_c[2] = 11$, and $y_c[3] = -47$.

$$\Rightarrow \quad y_c[n] = [8 \;\; -44 \;\; 11 \;\; -47], \quad 0 \leq n \leq 3$$

$$= 8\delta[n] - 44\delta[n-1] + 11\delta[n-2] - 47\delta[n-3]$$

→ You can use Matlab to verify that this is the correct answer for the circular convolution.