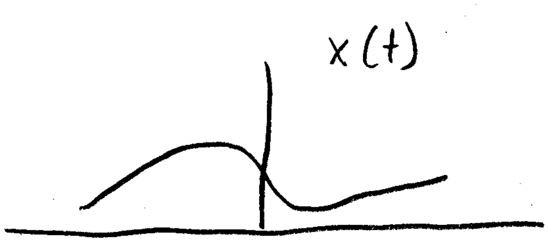
The DFT

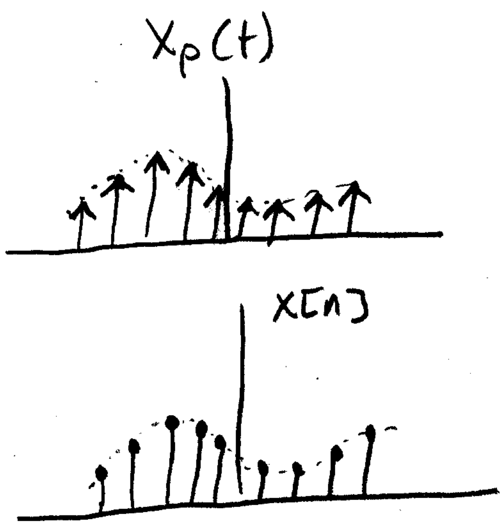
- For a discrete-time signal $x[n]$, we have seen that the Fourier Transform $X(e^{j\omega})$ is always periodic (with period 2π).

- For periodic continuous-time signals like $e^{j\omega_0 t}$, $\cos \omega_0 t$, and $\sin \omega_0 t$, we saw that the Fourier transform $X(\omega)$ consisted exclusively of impulses.

  $\Rightarrow$ In a sense, a continuous-time signal that contains only impulses is discrete, because it is nonzero only at a countable number of places.

  $\Rightarrow$ We used this concept in sampling when we "picked off" the weights of the impulses in $x_p(t)$ to get $x[n]$:





- So we have seen that

  Discrete in time $\longrightarrow$ periodic in frequency

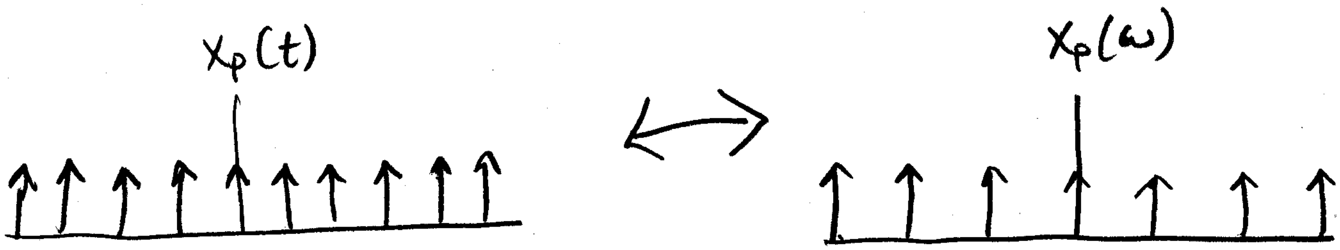
  Periodic in time $\longrightarrow$ discrete in frequency

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

  $\Rightarrow$ A signal that is periodic in one domain is discrete in the other.

<u>NOTE</u>: This implies that a signal that is <u>both</u> periodic <u>and</u> discrete in one domain <u>must</u> be periodic <u>and</u> discrete in the other domain.

<u>EX</u>: the periodic impulse train $x_p(t)$:



- This concept that

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

is the main idea behind the Discrete Fourier Transform, or "DFT".

Preliminaries: the Discrete Time Fourier Series (DFS)

- Suppose $x[n]$ is a discrete-time periodic signal with period N.

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

$$\boxed{\omega_0 = \frac{2\pi}{N}}$$

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} A_k e^{jk\omega_0 n} \qquad (3.1)$$

- Since the signal 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 $e^{j(k\omega_0)n}$, $0 \le k \le N-1$ are needed to represent the signal [just like three vectors $\vec{i}, \vec{j}, \vec{k}$, are sufficient in $\mathbb{R}^3$].

- So, whereas the continuous-time Fourier series generally required an infinite number of terms in the sum,

→ The DFS sum (3.1) has only $\underline{\underline{N}}$ terms.

- The DFS coefficients A_k are given by

$$A_k = \sum_{n=0}^{N-1} x[n]\, e^{-jk\omega_0 n} \qquad (4.1)$$

→ A <u>dot product</u> (as always) between the signal and (the conjugates of) the basis functions.

<u>NOTE</u>: since $\omega_0 = \frac{2\pi}{N}$, we have that

$$e^{j(k+N)\omega_0 n} = e^{j(k+N)\frac{2\pi}{N}n}$$

$$= e^{jk\frac{2\pi}{N}n}\underbrace{e^{j2\pi n}}_{1}$$

$$= e^{jk\frac{2\pi}{N}n}$$

$$= e^{jk\omega_0 n}$$

- Thus, even though the DFS sum (3.1) has only N terms and N coefficients A_k, one may speak of a coefficient A_{k+N} that would be multiplied times $e^{j(k+N)\omega_0 n} = e^{jk\omega_0 n}$, and so <u>must</u> be the same as A_k.

- So we have the following:
  - → $x[n]$ periodic with period N, specified completely by any N consecutive samples.
  - → A_k, periodic with period N, specified completely by any N consecutive samples.
  - → To find the fundamental period of the A_k's, take dot products of the basis functions with any one period of $x[n]$, as in (4.1).
  - → To find the fundamental period of $x[n]$, add up any one period of the A_k's times their respective basis functions, as in (3.1).
- It is <u>customary</u> to write

$$X(k) = A_k$$
$$W_N = e^{-j2\pi/N} = e^{-j\omega_0}$$

- The DFS equations (3.1) and (4.1) then become:

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X(k) W_N^{-kn} \qquad (5.1)$$

$$X(k) = \sum_{n=0}^{N-1} x[n] W_N^{kn} \qquad (5.2)$$

DFT

- Now suppose we sample a continuous-time signal and acquire N samples <u>only</u>. (This will <u>always</u> be the case in real life, since you will probably not live long enough to sample <u>forever</u>.)

- This gives us a discrete-time signal $x[n]$ defined for $n = 0, 1, 2, \ldots, N-1$ <u>only</u>.

- Suppose that
  1. We want a frequency representation of $x[n]$.
  2. We want both $x[n]$ and the frequency representation to be processed in a computer (DSP).

$\Rightarrow$ The discrete-time Fourier transform will <u>not</u> do, because ω is a continuous variable in $X(e^{j\omega})$, so this <u>cannot</u> be handled by the DSP chip.

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

- writing a DFS for $\tilde{x}[n]$ (the periodic signal) then gives us a <u>discrete</u> <u>and</u> <u>periodic</u> frequency representation for $x[n]$:

$$X(k) = \sum_{n=0}^{N-1} x[n] W_N^{kn} \qquad (7.1)$$

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-kn} \qquad (7.2)$$

$\left.\right\} \begin{array}{l} 0 \le n \le N-1 \\ 0 \le k \le N-1 \end{array}$

- Eq. (7.1) is the <u>DFT</u> of $x[n]$.
- Eq (7.2) is the <u>IDFT</u> of $X(k)$.

⇒ We consider that $x[n]$ is <u>one</u> <u>period</u> of $\tilde{x}[n]$ and that $X(k)$ is <u>one</u> <u>period</u> of the associated $A_k^{s'}$.

⇒ The DFT $X(k)$ and IDFT $x[n]$ are both of <u>finite</u> length N.

→ We often write

$$x[n] = \text{IDFT}\{X(k)\}$$

$$X(k) = \text{DFT}\{x[n]\}$$

$$x[n] \xleftrightarrow{\text{DFT}} X(k).$$

DFT Properties

- Relationship to discrete-time Fourier transform:
  - INSTEAD of considering the N samples $x[n]$, $0 \le n \le N-1$, to be one period of a periodic signal $\tilde{x}[n]$, we could consider them to be the only nonzero samples of a signal $\hat{x}[n]$ given by

$$\hat{x}[n] = \begin{cases} x[n], & 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases} \qquad (8.1)$$

  - In this case, $\hat{x}[n]$ has a discrete-time Fourier transform $\hat{X}(e^{j\omega})$.
  - The relationship between $\hat{X}(e^{j\omega})$ and the DFT $X(k)$ is

$$X(k) = \hat{X}(e^{j\omega})\Big|_{\omega = 2\pi k/N}, \quad 0 \le k \le N-1 \qquad (8.2)$$

$\Rightarrow$ $X(k)$ is N equally spaced samples of the Fourier transform $\hat{X}(e^{j\omega})$ placed in $-\pi \le \omega \le \pi$

$\Rightarrow$ $X(k)$ is N equally spaced samples along the unit circle of $\hat{X}(z)$.

- Linearity:

$$a x_1[n] + b x_2[n] \overset{DFT}{\longleftrightarrow} a X_1(k) + b X_2(k)$$

- Even Symmetry: (about the middle of the signal)

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

then $X(k) = X(N-k)$, $0 \le k \le N-1$.

- Odd Symmetry: (about the middle of the signal)

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

then $X(k) = -X(N-k)$, $0 \le k \le N-1$.

- Conjugate Symmetry for Real signals:

if $x[n]$ is real, then

$$|X(k)| = |X(N-k)|,$$

$$\arg X(k) = -\arg X(N-k),$$

$$\mathrm{Re}[X(k)] = \mathrm{Re}[X(N-k)],$$

$$\mathrm{Im}[X(k)] = -\mathrm{Im}[X(N-k)].$$

$0 \le k \le N-1$

- Time Shifting:
    - Let $\acute{x}[n] = \tilde{x}[n-\ell]$, where $\tilde{x}[n]$, as before, is the periodic extension of $x[n]$.
    - Then

$$\acute{x}[n] \overset{DFT}{\longleftrightarrow} W_N^{\ell k} X(k)$$

- Duality:

$$DFT\{X(k)\} = N x[n]$$

- Parseval's Theorem:

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

<u>NOTE</u>: When we wish to emphasize the fact that $X(k)$ is an <u>N-point</u> DFT, we can write

$$X(k) = DFT_N\{x[n]\}$$

$$x[n] = IDFT_N\{X(k)\}$$

Circular Convolution

- Suppose $x[n]$ is an N_1-point finite length signal, defined for $0 \le n \le N_1 - 1$.
- Suppose H is an LTI system with impulse response $h[n]$ that is nonzero only for $0 \le n \le N_2 - 1$.
- Then we can consider $h[n]$ to be a finite length signal of length N_2, defined only for $0 \le n \le N_2 - 1$.
- We <u>cannot</u> directly multiply the DFT's of $x[n]$ and $h[n]$, because they have different lengths, so the samples $X(k)$ and $H(k)$ aren't taken at the same frequencies.
- Suppose we zero-pad $x[n]$ and $h[n]$ to length $\max(N_1, N_2)$. Then we can multiply the DFT's.
- Then $Y(k) = X(k)H(k)$ is the $\max(N_1, N_2)$-point DFT of

$$y[n] = \tilde{x}[n] * h[n]$$

$\Rightarrow$ This is called the <u>circular</u> <u>convolution</u> of $x[n]$ and $h[n]$, written

$$y[n] = x[n] \overset{\max(N_1,N_2)}{\circledast} h[n].$$

$\Rightarrow$ $y[n]$ is the <u>plain</u> <u>old</u> convolution ("linear convolution") of $h[n]$ with the periodic extension of $x[n]$ with period $\max(N_1, N_2)$.

$\Rightarrow$ This is not the same as the linear convolution of $h[n]$ with $\hat{x}[n]$ in (8.1), which is <u>usually</u> what we want.

- So, when we multiply DFT s/, we get <u>circular convolution</u> in time, not linear convolution.

<u>NOTE</u>: "zero padding" means adding zeros to the right side of a signal to extend it's length.

<u>NOTE</u>: zero padding $x[n]$ changes the length of the DFT $X(k)$, which gives us a different set of samples of $\hat{X}(e^{j\omega})$ with increased frequency resolution.

- The important question is,

  → Since multiplying DFT's gives us circular convolution

  → Can we somehow <u>implement</u> the linear convolution

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

  using circular convolution?

⇒ The answer is <u>yes</u>.

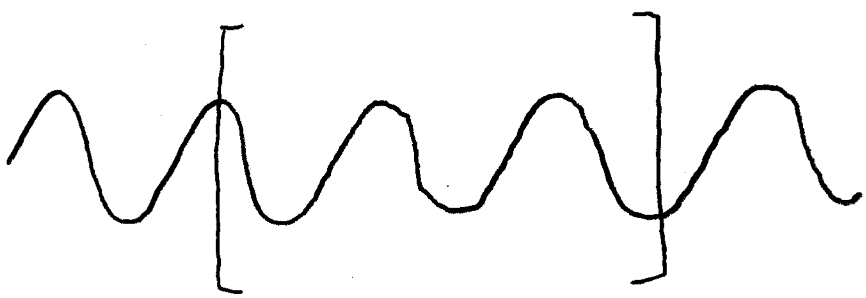
- All we have to do is zero pad <u>both</u> $x[n]$ and $h[n]$ to length $N_1 + N_2 - 1$.

- Then the circular convolution and linear convolution will be the same.

- This is how convolution by spectral multiplication is actually implemented in most modern applications.

Note on the DFT

- Suppose $x(t) = \cos\omega_0 t \overset{\mathcal{F}}{\longleftrightarrow} X(\omega) = 2\pi[\delta(\omega-\omega_0) + \delta(\omega+\omega_0)]$
- Suppose we sample N points to get $x[n]$, $0 \le n \le N-1$.
- Then we take the N-point DFT to get $X(k)$.
- What will $X(k)$ look like?
  - If our N samples capture exactly an integer number of periods of $x(t)$, then $X(k)$ will have only two nonzero samples.
- This is not what usually happens.
- Usually, we get some noninteger number of periods:

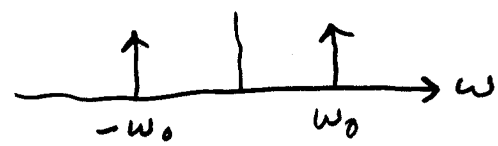


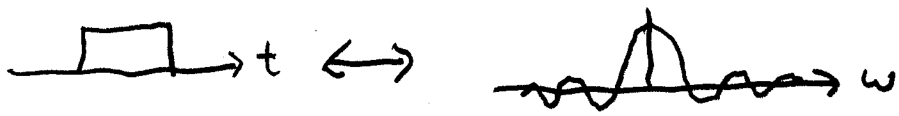
- This means $\tilde{x}[n]$, the periodic extension of $x[n]$, has big discontinuities in it
- $X(k)$ will show a lot of nonzero samples in this case.

- This "effect" is called frequency "leakage", or "edge effects".

- How should we think about this?

- The <u>finite segment</u> of $x(t)$ that we are sampling is $x(t)$ times a boxcar.

$X(\omega) =$ 



- So the spectrum of the <u>finite segment</u> of $x(t)$ that we are sampling is the convolution of $X(\omega)$ with a sync:

- In practice, it is common to "window" $x(t)$ with some window function $w(t)$ that has a "skinnier" spectrum than the boxcar

- This mitigates the frequency leakage (edge effects).
- Usually, the windowing is accomplished by sampling $x(t)$ as before to get $x[n]$, and then multiplying $x[n]$ by $w[n]$ (the samples of the window).

FFT

- Computation of the DFT (5.2)

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

requires N complex multiplies for each k, or N^2 complex multiplies total.

- The "Fast Fourier Transform", or FFT, is a tricky way to implement the DFT that re-uses intermediate calculations to reduce the number of multiplies to $N \log_2 N$.
- The basic FFT algorithms require N to be a power of 2.

- If $N = 2^{10} = 1024$, then

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

Decimation-in-Time FFT

- Let $x[n]$ have length N, $0 \le n \le N-1$.
- Suppose N is a power of 2.
- Let $u[n] = x[2n]$, $0 \le n \le \frac{N}{2}-1$, contain the "even numbered" samples of $x[n]$.
- Let $v[n] = x[2n+1]$, $0 \le n \le \frac{N}{2}-1$, contain the "odd numbered" samples of $x[n]$.
- Then

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

$$= \sum_{n=0}^{\frac{N}{2}-1} \left\{ x[2n] W_N^{2nk} + x[2n+1] W_N^{(2n+1)k} \right\}$$

$$= \sum_{n=0}^{\frac{N}{2}-1} u[n] W_{\frac{N}{2}}^{nk} + W_N^k \sum_{n=0}^{\frac{N}{2}-1} v[n] W_{\frac{N}{2}}^{nk}$$

$$= U(k) + W_N^k V(k), \quad 0 \le k \le \frac{N}{2}-1$$

- So we have written $X(k)$, an N-point DFT, as the sum of two $\frac{N}{2}$-point DFT's.

- This can be repeated until $X(k)$ is finally written as a sum of N one-point DFT's.

- Calculating $X(k)$ in this way is the basic decimation-in-time FFT algorithm, which requires only $N \log_2 N$ multiplies.

- There is also a decimation-in-frequency algorithm that begins by splitting up the even and odd numbered samples of $X(k)$ instead of $x[n]$.

<u>NOTE</u>: Basic FFT algorithms "jumble up" the order of the frequency samples.

  - Some routines fix this up before they return, and some do not.

  - Always read the documentation to find out the order of the frequency samples when you call an FFT routine.