MUS421 Lecture 1
Introduction and Overview

Julius O. Smith III (jos@ccrma.stanford.edu)
Center for Computer Research in Music and Acoustics (CCRMA)
Department of Music, Stanford University
Stanford, California 94305

June 27, 2020

Course Overview

- Spectrum analysis, processing, and synthesis using Short-Time Fourier Transforms (STFT)
- Processing motivated by the mechanics of *hearing*
- Applications include musical sound synthesis and audio signal processing

Main Pointers

- First Handout[1]
- Course Schedule and Outline[2]
  - *Assignments*
  - Weekly class schedule
  - Pointers to all lecture overheads and reading/viewing materials
- Class home page[3]

[1] http://ccrma.stanford.edu/~jos/intro421/
[2] http://ccrma.stanford.edu/~jos/intro421/Schedule_Assignments.html
[3] http://ccrma.stanford.edu/courses/421/

Why The Fourier Transform

- Natural for *visualizing* audio signals:
  The *ear* performs a kind of Fourier analysis
- Spectral models can be very compact and flexible:
  - MPEG audio coding
  - Sinusoidal modeling ("additive synthesis")
  - Sparse modeling elements for
    - *Machine listening*
    - *Music Information Retrieval* (MIR)
  - AES talk[4] on some history of audio spectral modeling at CCRMA and elsewhere.
- Any Linear Time Invariant (LTI) system can be implemented in the frequency domain by means of the Fourier Transform ("FFT convolution")

[4] http://ccrma.stanford.edu/~jos/pdf/AES-Heyser.pdf

Audio Applications of the Short-Time Fourier Transform (STFT)

- Frequency-domain display of audio signals
- Fast (FFT) convolution
- Robust, time-varying, linear filtering
- Fourier analysis, modification, and resynthesis
- Musical sound synthesis via spectral modeling:
  - Additive synthesis using sinusoids
  - Sines + Noise modeling
  - Sines + Noise + Transients modeling
- Speech analysis and synthesis
- Vocoders
- Time scaling
- Pitch shifting (frequency scaling)
- Pitch (fundamental frequency) detection
- Noise reduction
- Audio compression (MPEG audio: .mp3, .m4a)
- Signal source separation in the frequency domain

- Computational Auditory Scene Analysis (CASA)
- Machine listening
- Music Information Retrieval (MIR)
- Music identification (Shazam)

Audio Compression

Spectral audio processing is used in transform coders for audio compression, such as

- MPEG AAC (10X common), and
- "MP3" (MPEG-II, Layer III — $\approx$ 10X-AAC at 8X)

Music 422 (EE 367C) is an entire CCRMA course devoted to this topic (offered winter quarters).

Main Pointer

The course schedule and outline[5] (reachable from the class home page[6]) lists the following information:

- *Assignments*
- Weekly class schedule
- Pointers to all lecture overheads
- Pointers to supplementary reading/listening

[5] http://ccrma.stanford.edu/~jos/intro421/Schedule_Assignments.html

[6] http://ccrma.stanford.edu/courses/421/

Notation

Frequency and Time:

ω denotes continuous *radian frequency* (rad/sec)

f denotes continuous *frequency* in Hertz (Hz)

$\omega = 2\pi f$

ω_k denotes discrete frequency, $\omega_k = 2\pi(k/N)f_s$

$\omega, \omega_k \in \mathbb{R}$ (frequencies are always real)

$T =$ sampling interval (sec) (typically $T = 1$)

$f_s =$ sampling rate, $f_s = \frac{1}{T}$

$t_n = nT$ (discrete time)

$n, k \in \mathbb{Z}$ (integers)

$t, t_n \in \mathbb{R}$ (times are always real)

Introduction to Audio Spectrum Analysis

Spectrum analysis of real-world signals typically occurs over short time *segments*. We are therefore most interested in *short-time spectrum analysis*:

- Spectral content typically varies over time.
- The human ear uses less than one second of past sound to form a spectrum.
- There is a limit to the length of signal we can analyze at once.

To extract and analyze a sound segment, it is necessary to apply a *window function*. An unmodified segment extraction corresponds to a "rectangular window".

Everything we 'look at' will be through a 'window', hence it is important to realize what the window is doing to our underlying signal.

Applications we'll discuss first:

- Spectral Analysis for Display
- FIR Filter Design by Window Method

Example of Windowing

Let's look at a simple example of windowing to demonstrate what happens when we turn an infinite duration signal into a finite duration signal through windowing.

Complex Sinusoid:

$$x(n) = e^{j\omega nT}, \qquad 0 \leq \omega T < \pi$$

Notes:

- real part $= \cos(\omega nT)$
- The frequencies present in our signal are only positive. A fancy name for $x(n)$ is an 'analytic signal'

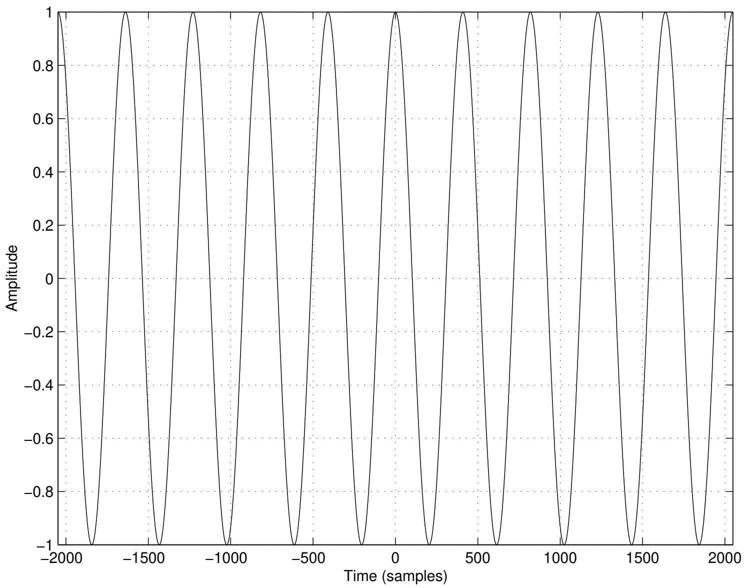
This signal is infinite duration. (It doesn't die out as n increases.) In order to end up with a signal which dies out eventually (so we can use the DFT), we need to multiply our signal by a window (which does die out).

Putting all this together, we get the following:

Our original signal (unwindowed, infinite duration), is

$$x(n) = e^{j\omega_0 nT}, \quad n \in \mathbb{Z}$$

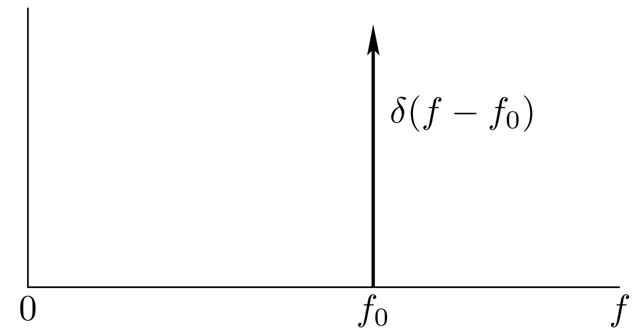
A portion of the real part, $\cos(\omega_0 nT)$, is plotted below:



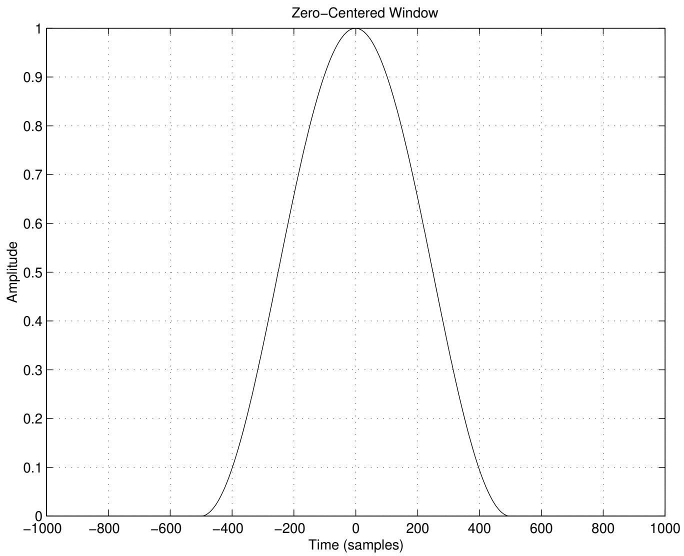
The imaginary part, $\sin(\omega_0 nT)$, is of course identical but for a 90-degree phase-shift to the right.

The Fourier Transform of this infinite duration signal is a delta function at ω_0:

$$X(\omega) = 2\pi\delta(\omega - \omega_0) = \delta(f - f_0)$$

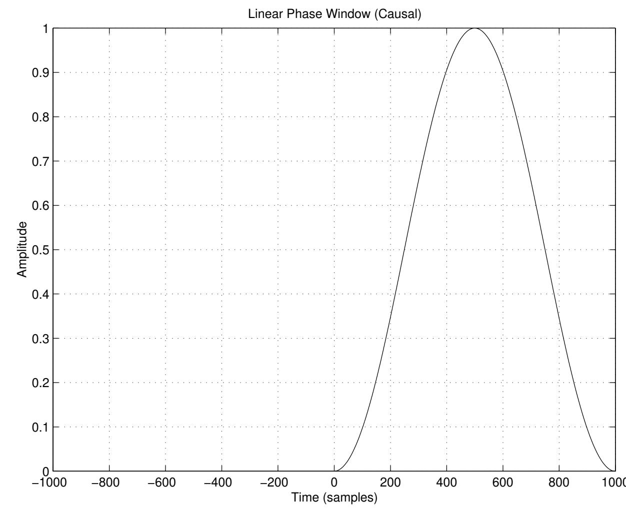


The following is a diagram of a typical window function:



This may be called a “zero-centered” (or “zero phase”, or “even”) window function, which means its phase in the frequency domain is either zero or π, as we will see in detail later. (Recall that a real and even function has a real and even Fourier transform.) The window is also nonnegative, as is typical.

We might also require that our window be zero for negative time. Such a window is said to be ‘causal’. Causal windows are necessary for real-time processing:

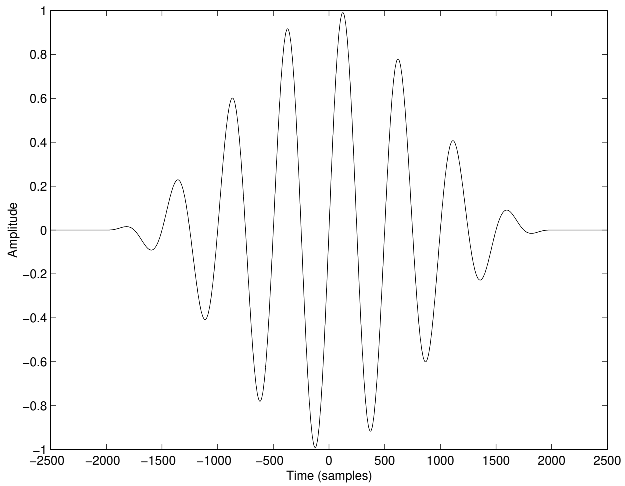


By shifting the original window in time by half its length, we have turned the original non-causal window into a causal window. The Shift property of the Fourier Transform tells us that we have introduced a linear phase term.

The windowed complex sinusoid is:

$$x_w(n) \;=\; w(n)x(n) \;\stackrel{\Delta}{=}\; w(n)e^{-j\omega_0 nT} \qquad n \in \mathbb{Z}$$

(Note carefully the difference between w and ω.)

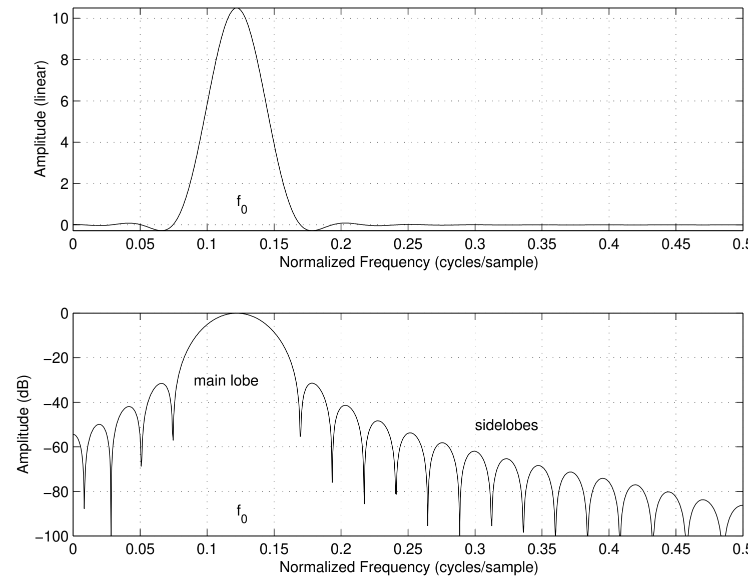


The Convolution Theorem tells us that our multiplication in the time domain results in a convolution in the frequency domain. Hence, in our case, we will obtain the convolution of a delta function at frequency ω_0, and the transform of the window:

$$X_w(\omega) \;=\; (W * X)(\omega) \;=\; W(\omega - \omega_0)$$

The result of convolution with a delta function is the original function, shifted to the location of the delta

function. (The delta function is the identity element for convolution.)



Summary

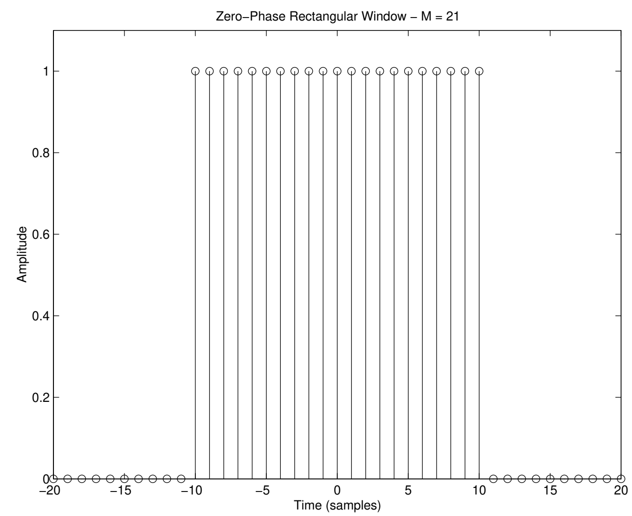
- We saw that a sinusoid at amplitude A, frequency ω_0, and phase ϕ becomes a *window transform* shifted out to frequency ω_0, and scaled by $Ae^{j\phi}$.
- Windowing in the time domain resulted in a *'smearing'* or *'smoothing'* in the frequency domain. We need to be aware of this if we are trying to *resolve* sinusoids which are close together in frequency.
- Windowing also introduced *side lobes*. This is important when we are trying to resolve low amplitude sinusoids in the presence of higher amplitude signals. When we look at specific windows, we will be looking at this behavior.
- The window $w(n)$ can be thought of as the *time-domain sampling kernel* at time 0
- The window transform $W(\omega)$ is the corresponding *frequency-domain sampling kernel* at dc
- In ordinary sampling, we have sinc$(t/T)/T$ and its (rectangular) transform as the sampling kernels

There are many type of windows which serve various purposes and exhibit various properties, as we shall see.

The Rectangular Window

The rectangular window may be defined as:

$$w_R(n) \triangleq \begin{cases} 1, & |n| \leq \frac{M-1}{2} \\ 0, & \text{otherwise} \end{cases}$$



- "Zero centered" definition (*even* in time domain)
- Need M odd in zero-centered case
- Scale window by $1/M$ to obtain unity dc gain

To see what happens in the frequency domain, we need to look at the DTFT of the window:

$$\begin{aligned} W_R(\omega) &= \mathrm{DTFT}_\omega(w_R) \stackrel{\Delta}{=} \sum_{n=-\infty}^{\infty} w_R(n)e^{-j\omega n} \\ &= \sum_{n=-\frac{M-1}{2}}^{\frac{M-1}{2}} e^{-j\omega n} = \frac{e^{j\omega\frac{M-1}{2}} - e^{-j\omega\frac{M+1}{2}}}{1-e^{-j\omega}} \end{aligned}$$

where we used the closed form of a geometric series:

$$\sum_{n=L}^{U} r^n = \frac{r^L - r^{U+1}}{1-r}$$

We can factor out linear phase terms from the numerator and denominator of the above expression to get

$$\begin{aligned} W_R(\omega) &= \frac{e^{-j\omega\frac{1}{2}}}{e^{-j\omega\frac{1}{2}}}\left[\frac{e^{j\omega\frac{M}{2}} - e^{-j\omega\frac{M}{2}}}{e^{j\omega\frac{1}{2}} - e^{-j\omega\frac{1}{2}}}\right] \\ &= \frac{\sin\left(M\frac{\omega}{2}\right)}{\sin\left(\frac{\omega}{2}\right)} \stackrel{\Delta}{=} M \cdot \mathsf{asinc}_M(\omega) \end{aligned}$$

where $\mathsf{asinc}_M(\omega)$ denotes the *aliased sinc function*.

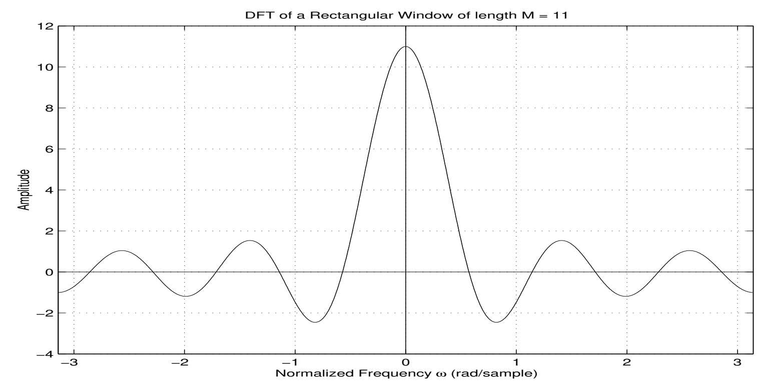
$$\mathsf{asinc}_M(\omega) \stackrel{\Delta}{=} \frac{\sin(M\omega/2)}{M \cdot \sin(\omega/2)}$$

(also called the *Dirichlet function*)

Rectangular Window Transform (Cont'd)

Above, we found the rectangular window transform to be the aliased sinc function:

$$W_R(\omega) = M \cdot \mathsf{asinc}_M(\omega) \stackrel{\Delta}{=} \frac{\sin\left(M\frac{\omega}{2}\right)}{\sin\left(\frac{\omega}{2}\right)}$$



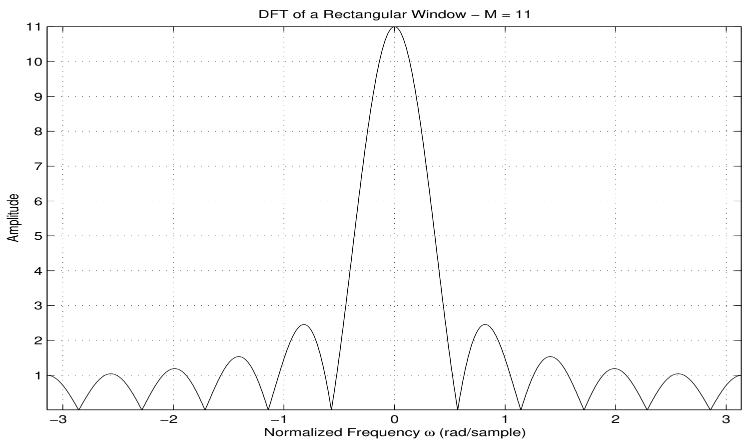
This (real) result is for the *zero-centered* rectangular window. For the *causal* case, a linear phase term appears:

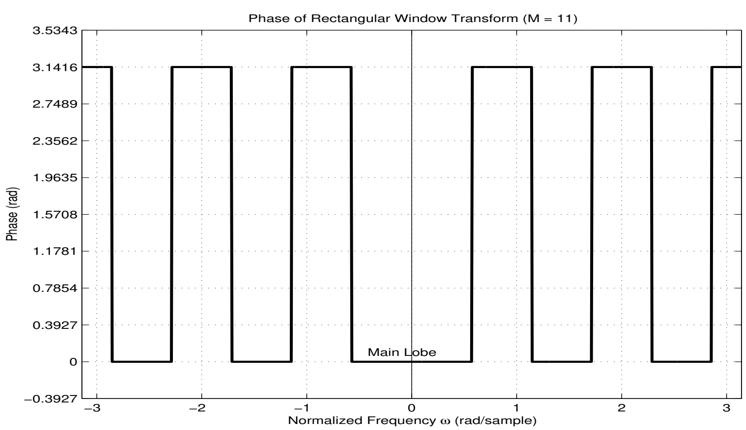
$$W_R^c(\omega) = e^{-j\frac{M-1}{2}\omega} M \mathsf{asinc}_M(\omega)$$

As the sampling rate goes to infinity, the aliased sinc function approaches the regular *sinc function*

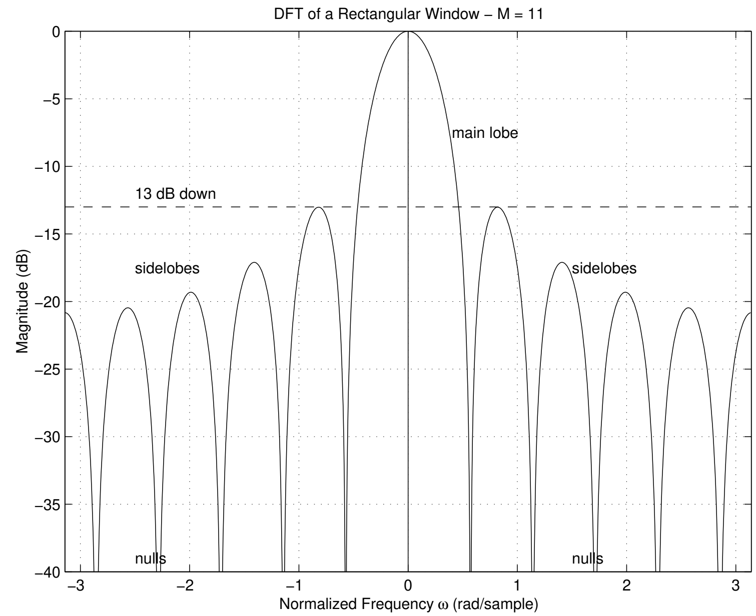
$$\mathsf{sinc}(x) \stackrel{\Delta}{=} \frac{\sin(\pi x)}{\pi x}$$

More generally, we may plot both the *magnitude* and *phase* of the window transform versus frequency:

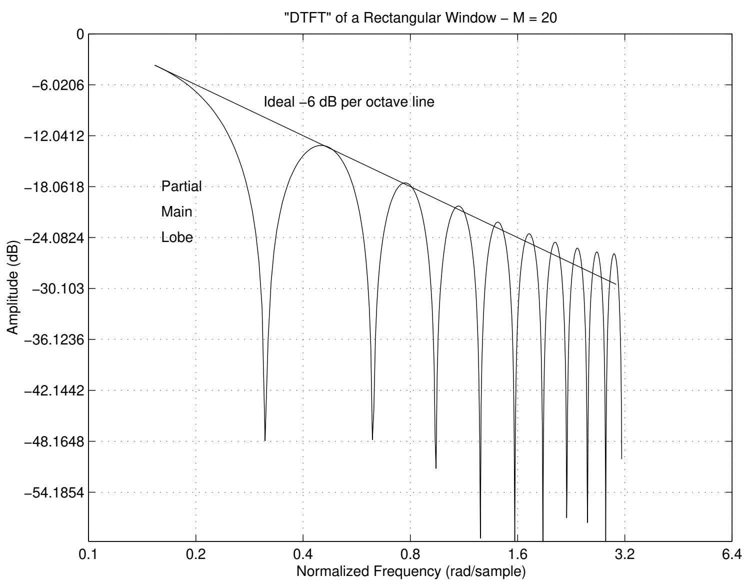




In audio work, we more typically plot the window transform magnitude on a *decibel (dB) scale*:



Since the DTFT of the rectangular window approximates the sinc function, it should "roll off" at approximately 6 dB per octave, as verified in the log-log plot below:



As the sampling rate approaches infinity, the rectangular-window transform (asinc) converges exactly to the sinc function. Therefore, the departure of the roll-off from that of the sinc function can be ascribed to *aliasing* in the frequency domain, due to sampling in the time domain.

Sidelobe Roll-Off Rate

In general, if the first n derivatives of a continuous function $w(t)$ exist (*i.e.*, they are finite and uniquely defined), then its Fourier Transform magnitude is asymptotically proportional to

$$|W(\omega)| \to \frac{\text{constant}}{\omega^{n+1}} \quad (\text{as } \omega \to \infty)$$

Proof: Look up "roll-off rate" in text index.

- Thus, we have the following rule-of-thumb:

  $\boxed{n \text{ derivatives } \longleftrightarrow\ -6(n+1) \text{ dB per octave roll-off rate}}$

  (since $20\log_{10}(2) = 6.0205999\ldots$).
- This is also $-20(n+1)$ dB per *decade*.
- To apply this result, we normally only need to look at the window's *endpoints*. The interior of the window is usually differentiable of all orders.

Examples:

- Amplitude discontinuity $\longleftrightarrow$ -6 dB/octave roll-off
- Slope discontinuity $\longleftrightarrow$ -12 dB/octave roll-off
- Curvature discontinuity $\longleftrightarrow$ -18 dB/octave roll-off

For discrete-time windows, the roll-off rate slows down at high frequencies due to aliasing.

In summary, the DTFT of the M-sample **rectangular window** is proportional to the 'aliased sinc function':

$$\text{asinc}_M(\omega T) \triangleq \frac{\sin(\omega M T/2)}{M \sin(\omega T/2)}$$

$$\approx \frac{\sin(\pi f M T)}{M \pi f T} \triangleq \text{sinc}(f M T)$$

Some important points **(rect window transform)**:

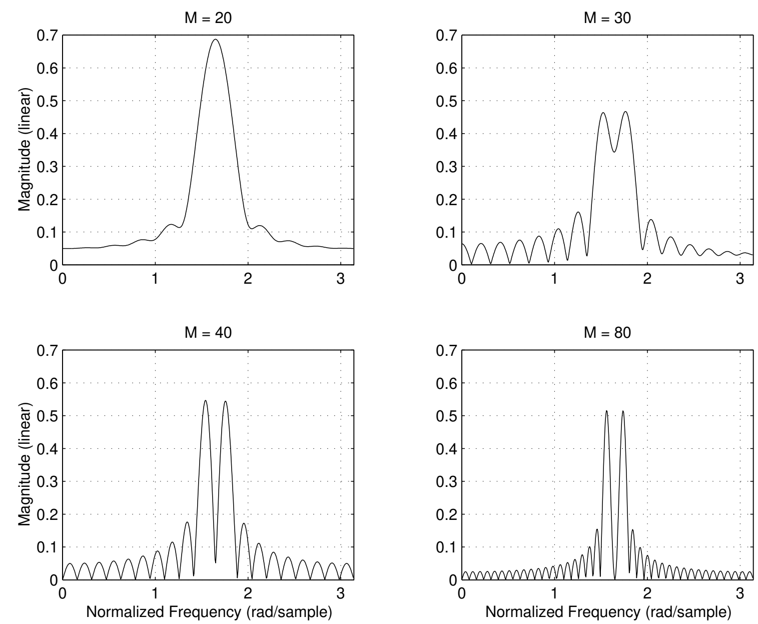
- Zero crossings at integer multiples of $\boxed{\Omega_M \triangleq \frac{2\pi}{M}}$ (= freq. sampling interval used by a length M DFT)
- Main lobe width is $2\Omega_M = \frac{4\pi}{M}$
- As M gets bigger, the main-lobe narrows (better frequency resolution)
- M has *no effect on the height of the side lobes* (Same as the "Gibbs phenomenon" for Fourier series)
- First side lobe only 13 dB down from main-lobe peak
- Side lobes roll off at approximately 6dB per octave
- A *linear phase term* arises when we shift the window to make it *causal*, while the window transform is real in the zero-centered case (i.e., when the window $w(n)$ is an *even function* of n)

Frequency Resolution

The next series of plots shows the effect that an increased window length has on our ability to resolve two sinusoids.

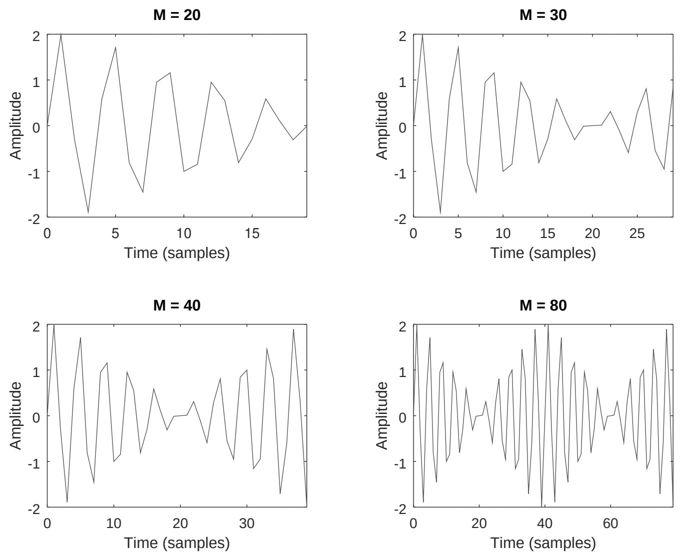
Two Cosines ("In-Phase" Case)

- 2 cosines separated by $\Delta\omega = \frac{2\pi}{40}$
- Rectangular Windows of lengths: $M = 20, 30, 40, 80$ ($\Delta\omega = \frac{1}{2}\Omega_M, \frac{3}{4}\Omega_M, \Omega_M, 2\Omega_M$, where $\Omega_M \triangleq 2\pi/M$)



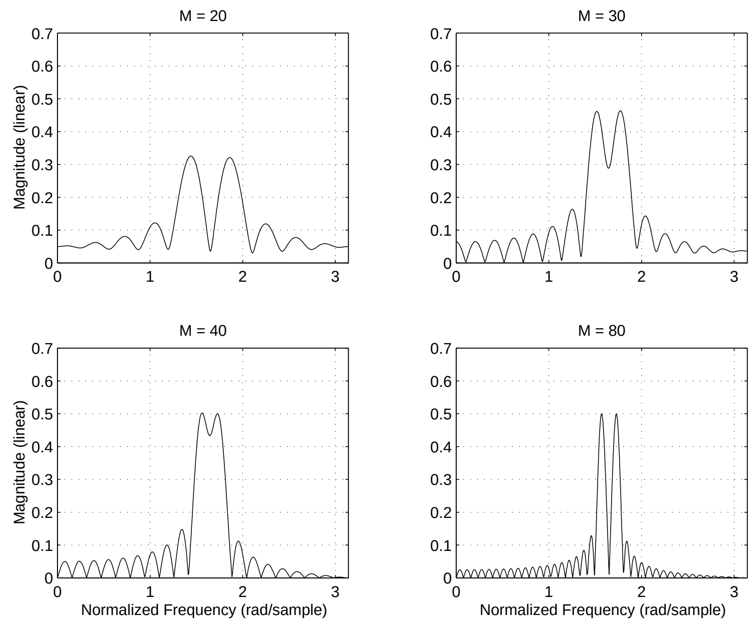
Two Cosines ("In-Phase" Case) in Time Domain

- 2 cosines separated by $\Delta\omega = \frac{2\pi}{40}$
- Rectangular Windows of lengths: $M = 20, 30, 40, 80$ $(\Delta\omega = \frac{1}{2}\Omega_M, \frac{3}{4}\Omega_M, \Omega_M, 2\Omega_M,\ \Omega_M \stackrel{\Delta}{=} 2\pi/M)$



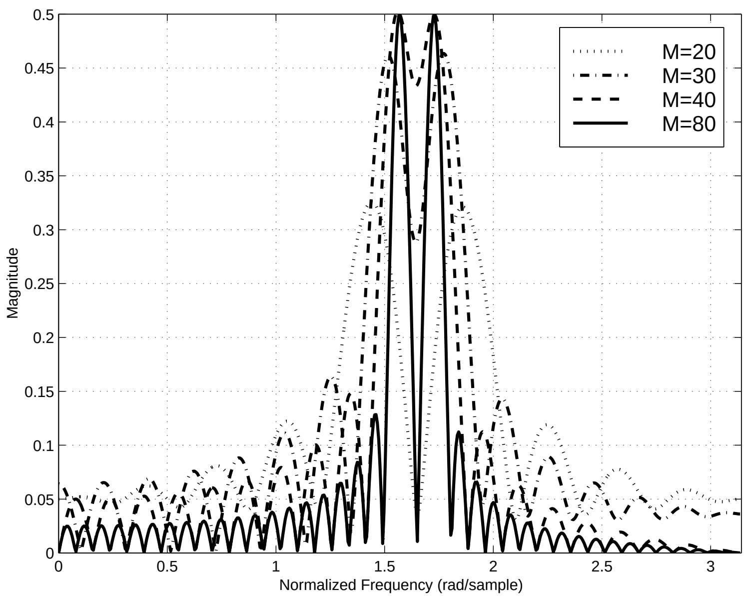
One Sine and One Cosine ("Phase Quadrature" Case)

- As above, but 1 sine and 1 cosine
- Note: least-resolved case appears resolved!
- Note: $M = 40$ case suddenly looks much worse
- Only the $M = 80$ case looks good at all phases



One Sine and One Cosine ("Phase Quadrature" Case) All Four Resolutions Overlaid

- Same plots as on previous page, just overlaid
- Peak locations are *biased* in under-resolved cases, both in amplitude and frequency



The preceding figures suggest that, for a rectangular window of length M, two sinusoids can be most reliably

resolved when they are separated in frequency by a full main-lobe width:

$$\boxed{\Delta\omega \geq 2\Omega_M} \qquad \left(\Omega_M \triangleq \frac{2\pi}{M}\right)$$

This implies there must be at least *two full cycles* of the difference-frequency under the window.

We'll see later that this is an overly conservative requirement—a more careful study reveals that 1.44 cycles is sufficient for the rectangular window.

Sinusoidal Interference as Amplitude Modulation

Resolving two closely spaced sinusoids is equivalent to *AM demodulation*:

$$\cos\left(\omega_c t + \frac{\omega_d}{2}t\right) + \cos\left(\omega_c t - \frac{\omega_d}{2}t\right) = 2\cos\left(\omega_c t\right)\cos\left(\frac{\omega_d}{2}t\right)$$

where ω_d is the *difference frequency* in rad/s.

- Intuitively, it makes sense to require two cycles of the difference-frequency ω_d, since that is *one cycle* of the equivalent AM modulation (two “beats”)

Beating Heisenberg

In principle, arbitrarily small frequency separations can be resolved if

- there is no noise, and
- we are sure we are looking at the sum of two ideal sinusoids under the window

In this case, the *maximum likelihood estimate* (MLE) will reliably find the six sinusoidal parameters (amplitude, frequency, and phase for both sinusoids). We will return to the MLE later in the quarter.

However, in practice, there is almost always some noise and/or interference, so we normally require sinusoidal frequency separation by on the order of a main-lobe width (of the asinc function in this case, or the window transform more generally) whenever possible.

Minimizing Side-Lobe Level

In addition to minimizing main-lobe width to maximize frequency-resolution, we also want *minimum side-lobe level*.

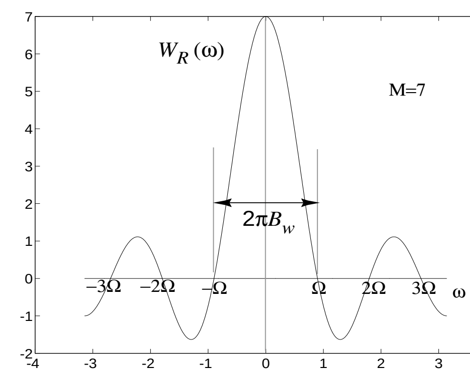
The rectangular window provides an abrupt transition at its edge. This minimizes main-lobe width while *maximizing* side-lobe level among all windows in the normal (monotonically decaying away from time 0) case.

We will soon look at other windows having a more gradual transition to zero, thereby reducing side-lobe level.

Resolution Bandwidth (Resolving Sinusoids)

Our ability to resolve two closely spaced sinusoids is determined by the main-lobe-width and sidelobe-level of our window's Fourier transform.

Let B_w denote the main lobe width in Hz, with the main lobe width defined as the width between zero crossings:



For the Rectangular Window (length M), we have

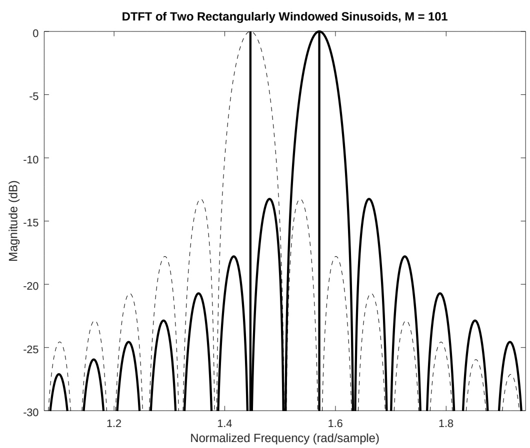
$$W_R(\omega) = \mathsf{asinc}_M(\omega) \triangleq \frac{\sin(M\omega T/2)}{\sin(\omega T/2)} = \frac{\sin(M\pi f T)}{\sin(\pi f T)}$$

Main lobe width is "two sidelobes wide"

$$\Rightarrow \quad \boxed{B_w = 2\frac{\Omega_M}{2\pi} = 2\frac{f_s}{M}} \quad \text{(Hz)}$$

Choosing Window Length to Resolve Sinusoids

A conservative requirement for resolving 2 sinusoids (in noisy conditions) with a spacing of Δf Hz is to choose a window length M long enough so that their main lobes are clearly discernible. For example, we may require that their main lobes meet at the first zero crossings.



To obtain the separation shown above, we must have $B_w \leq \Delta f$, where B_w is the main lobe width in Hz, and Δf is the sinusoidal frequency separation in Hz.

For the rectangular window, B_w can be expressed as

$$B_w = 2\frac{f_s}{M}$$

Hence we need:

$$\begin{aligned} B_w &= 2\frac{f_s}{M} \leq \Delta f \\ &\Rightarrow M \geq 2\frac{f_s}{\Delta f} \end{aligned}$$

or

$$\boxed{M \geq 2\frac{f_s}{|f_2 - f_1|}}$$

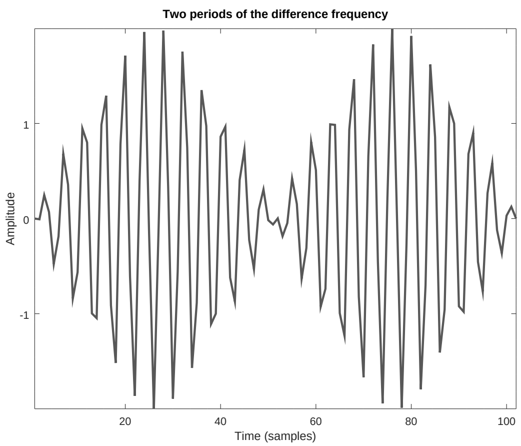
- A length M rectangular window satisfying this inequality is said to *resolve* the sinusoidal frequencies f_1 and f_2
- This is equivalent to our previous observation since

$$M \geq 2\frac{f_s}{\Delta f} \quad \Leftrightarrow \quad \Delta f \geq 2\frac{f_s}{M} \quad \Leftrightarrow \quad \Delta\omega \geq 2\Omega_M$$

- In summary, to resolve sinusoidal frequencies f_1 and f_2 under a rectangular window, it is *sufficient* for the window length M to span at least 2 periods of the *difference frequency* $f_2 - f_1$, where 2 is the width of the main lobe, measured in sidelobe-widths.
- By the Fourier *scaling* theorem, K periods must suffice for a main lobe of width $K\Omega_M$.

Closely Spaced Sinusoids as Amplitude Modulation

The previous example looks like this in the time domain:



- Over one "beat" of the difference frequency, the AM modulation due to "sinusoidal interference" is equivalent to a Hann window
- Modulation envelope is precisely sinusoidal
- In the absence of noise, and under the assumption of sinusoidal modulation (or, equivalenly, interference by one other sinusoid), all parameters can be recovered

Resolving Sinusoidal Components Robustly

We cannot normally assume a sum of precisely two sinusoids with no noise, and so we choose our window length to resolve them robustly:

- FFT window length M spans at least *two periods of the difference frequency* under a *rectangular window* (and *longer* for other windows)
- $\Longleftrightarrow$ Window transform (asinc) separated by a *full main-lobe width* at the *minimum supported peak-frequency separation*
- Any narrower peak spacing is then treated as *amplitude modulation* that plays out over time as spectral-frame amplitude modulation

We are still assuming that sinusoidal signal components are present, at least over the window duration, but this is commonly a good assumption.