MUS421 Lecture 3A
FFT Windows

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

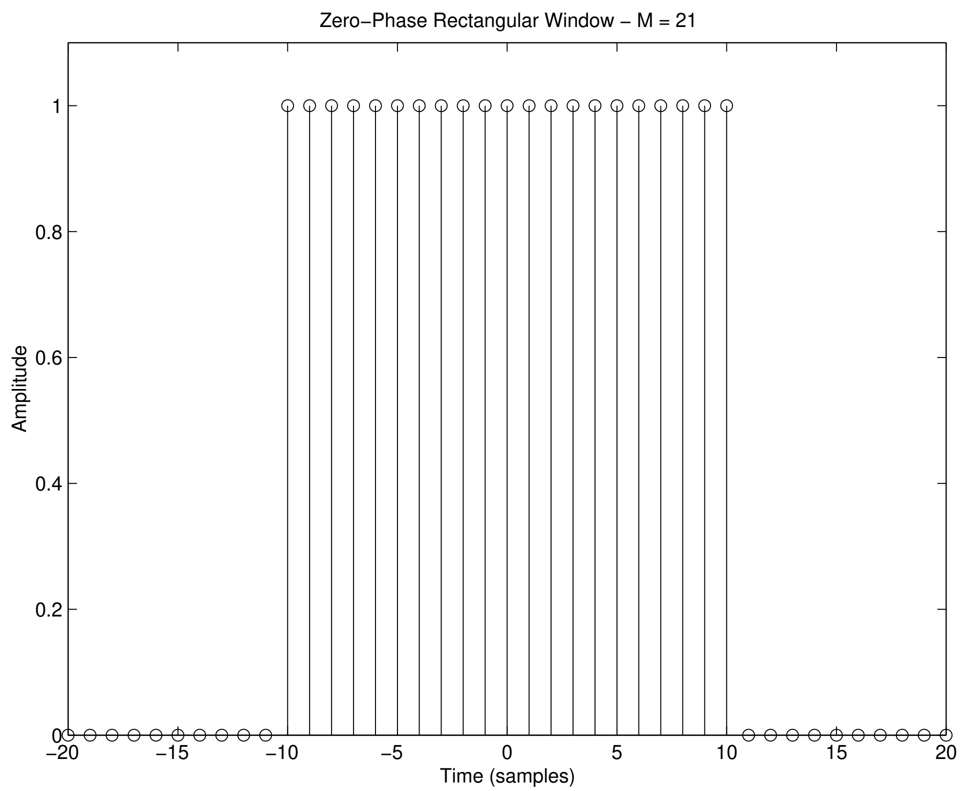
Outline

- Rectangular, Hann, Hamming
- MLT Sine
- Blackman-Harris Window Family
- Bartlett
- Poisson
- Slepian and Kaiser
- Dolph-Chebyshev
- Gaussian
- Optimal Windows

The Rectangular Window

Previously, we looked at the rectangular window:

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



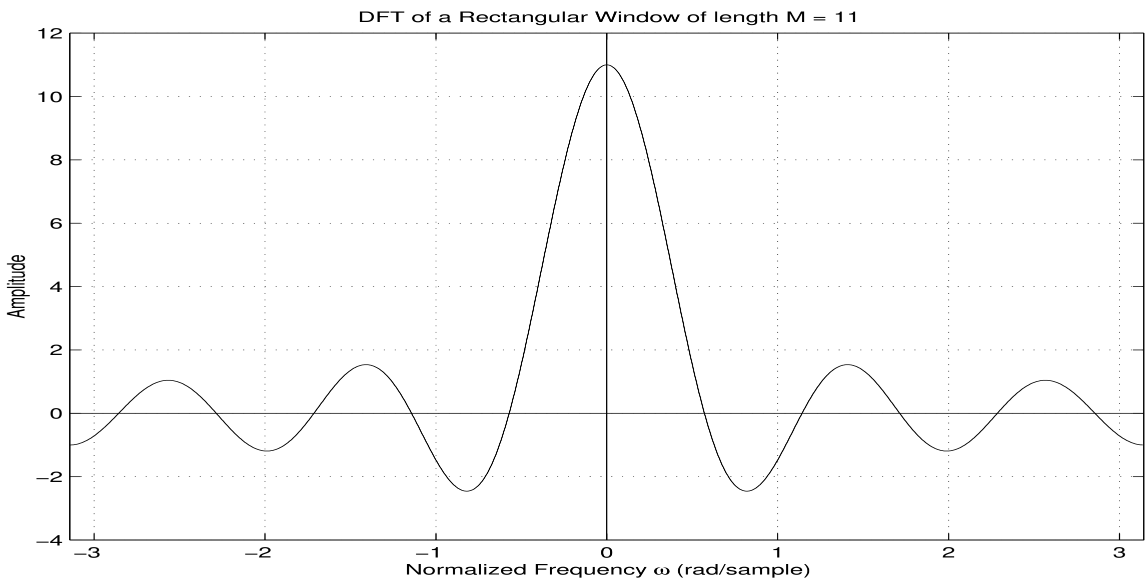
The window transform (DTFT) was found to be

$$W_R(\omega) \ = \ \frac{\sin\left(M\frac{\omega}{2}\right)}{\sin\left(\frac{\omega}{2}\right)} \triangleq M \cdot \text{asinc}_M(\omega) \qquad (1)$$

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

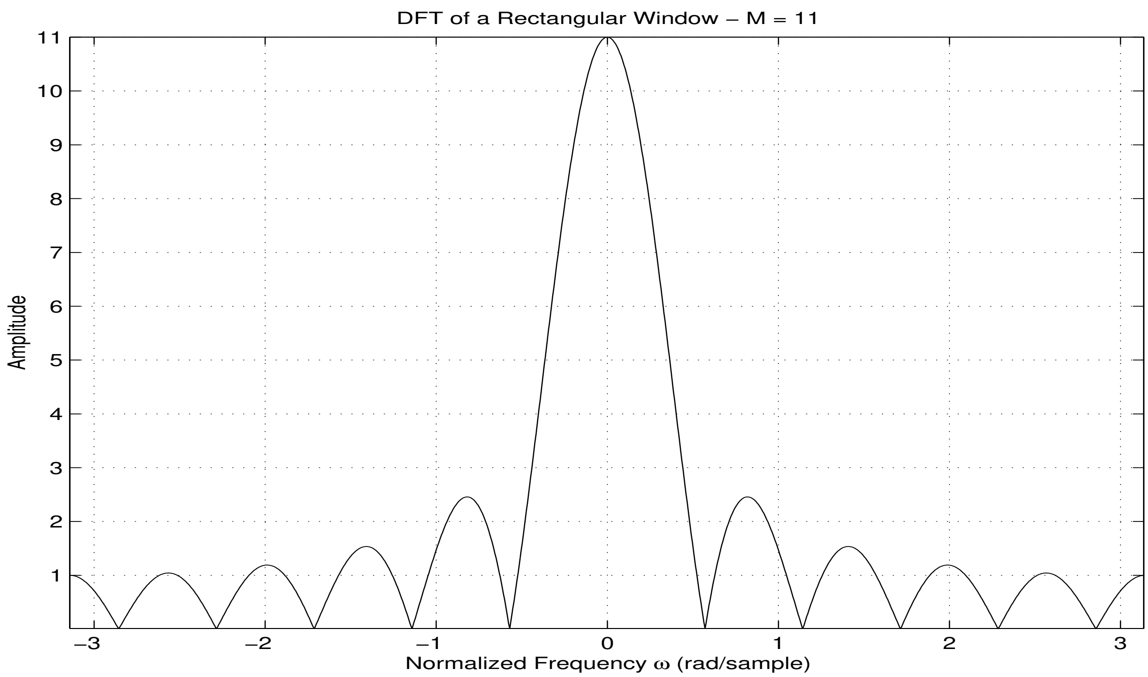
$$\text{asinc}_M(\omega) \triangleq \frac{\sin(M\omega/2)}{M \cdot \sin(\omega/2)}$$

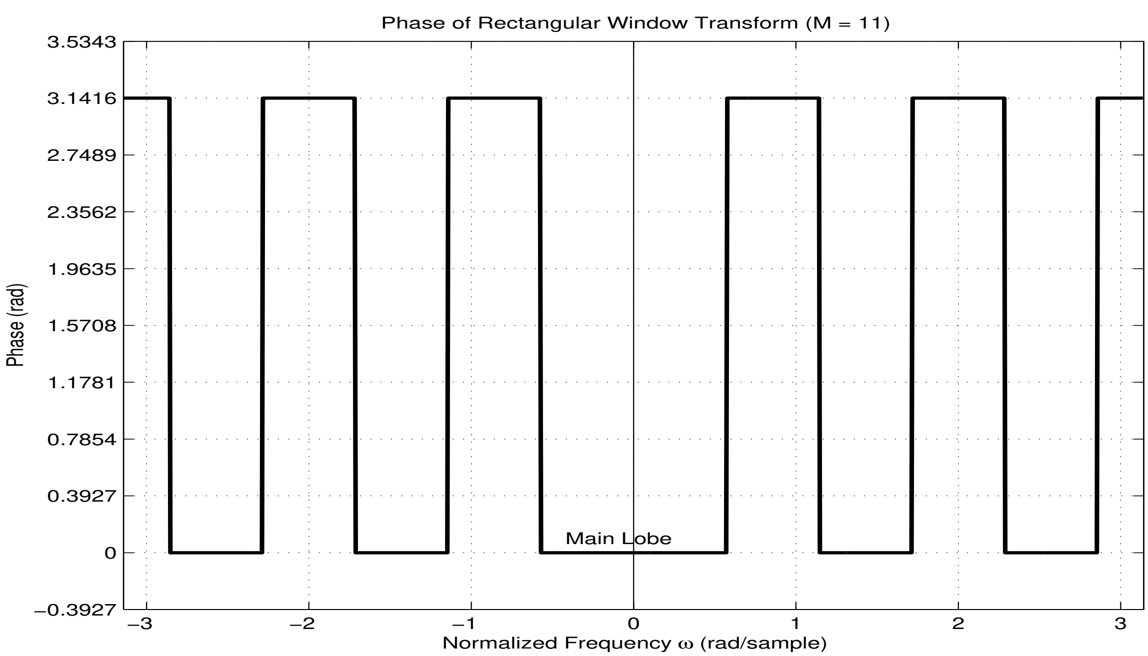
This result is plotted below:



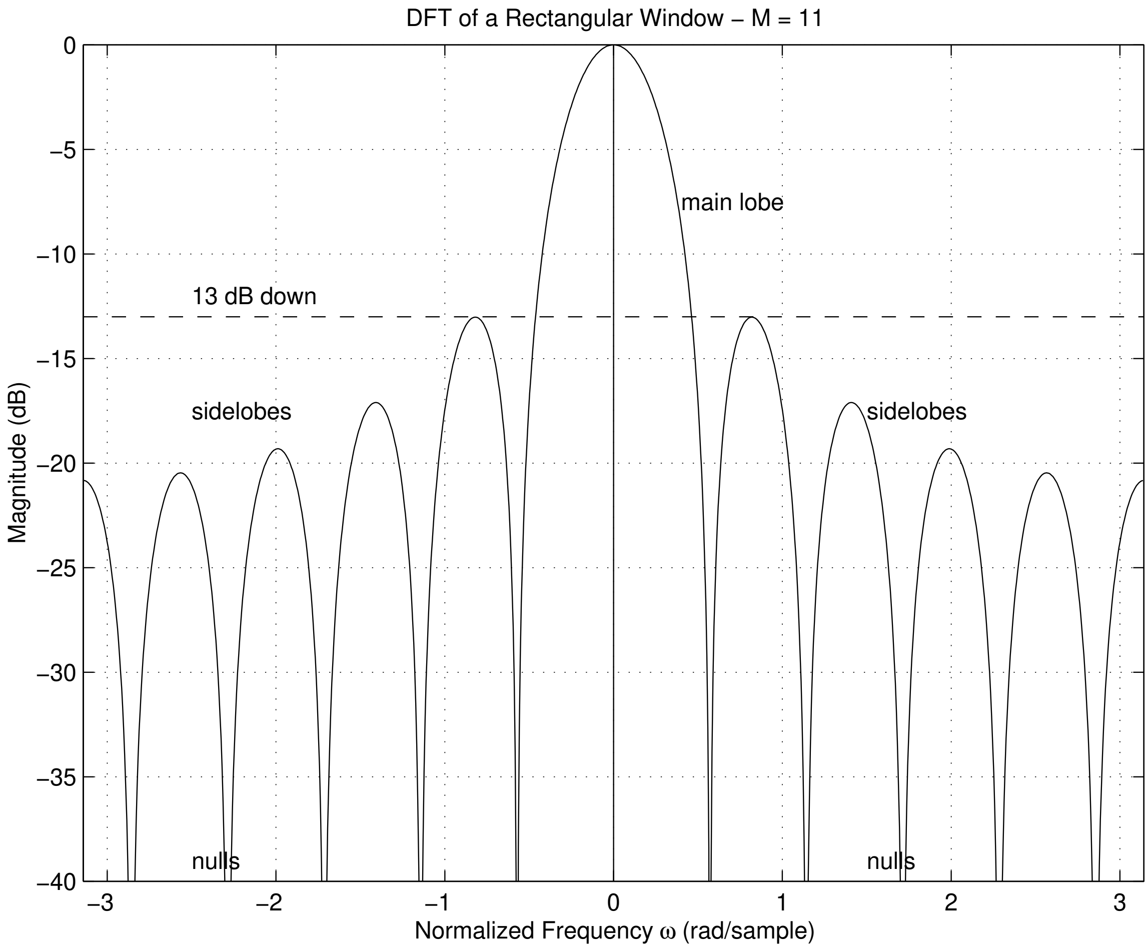
Note that this is the complete window transform, not just its magnitude. We obtain real window transforms like this only for symmetric, zero-centered windows.

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

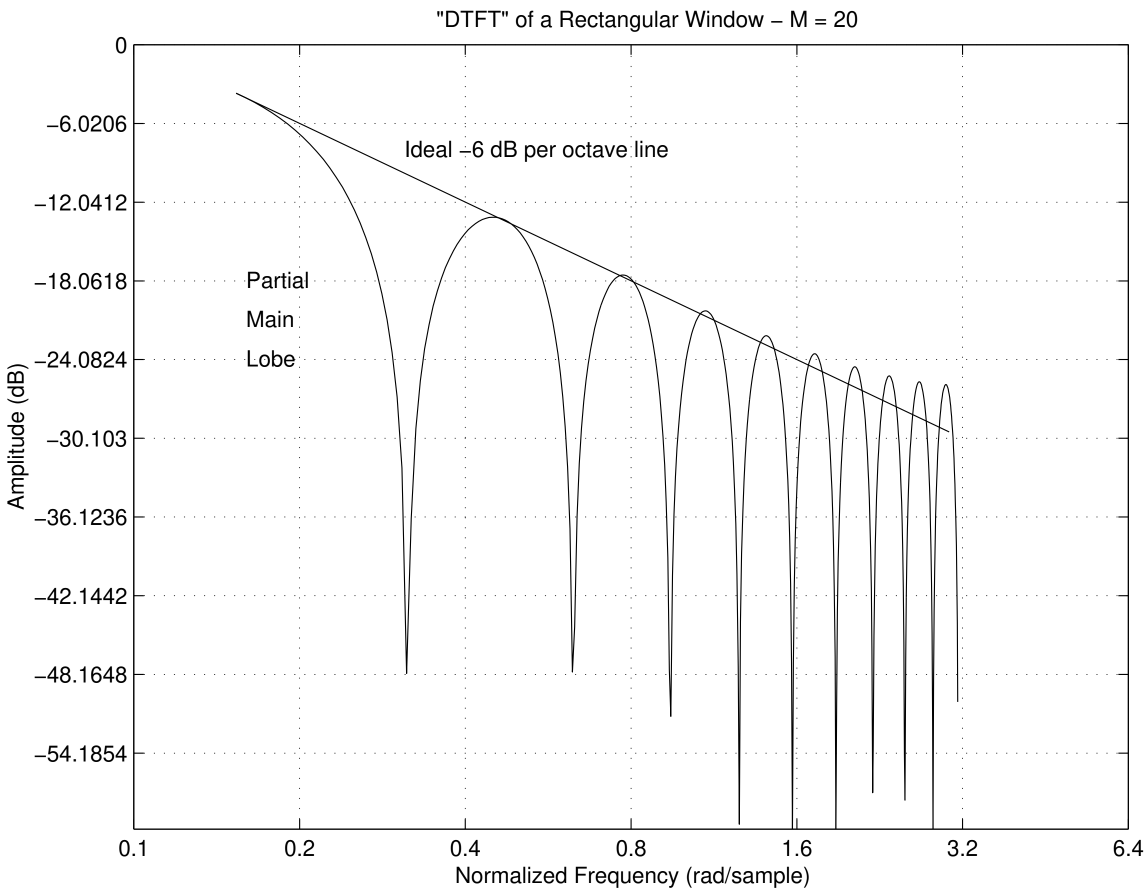




In audio work, we more typically plot the window transform magnitude on a *decibel (dB) scale*:



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 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, when only the first n terms *exist* in the power-series expansion of a continuous function $w(t)$ (*i.e.*, each term is finite), then the Fourier Transform magnitude $|W(\omega)|$ is asymptotically proportional to

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

Proof: Papoulis, **Signal Analysis**, McGraw-Hill, 1977

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

$$\boxed{n \text{ terms} \leftrightarrow -6n \text{ dB per octave roll-off rate}}$$

(since $-20\log_{10}(2) = 6.0205999\ldots$).
This is also $-20n$ 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.

Example Roll-Off Rates:

- Amplitude discontinuity $(n = 1) \leftrightarrow -6$ dB/octave
- Slope discontinuity $(n = 2) \leftrightarrow -12$ dB/octave
- Curvature discontinuity $(n = 3) \leftrightarrow -18$ dB/octave

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':

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

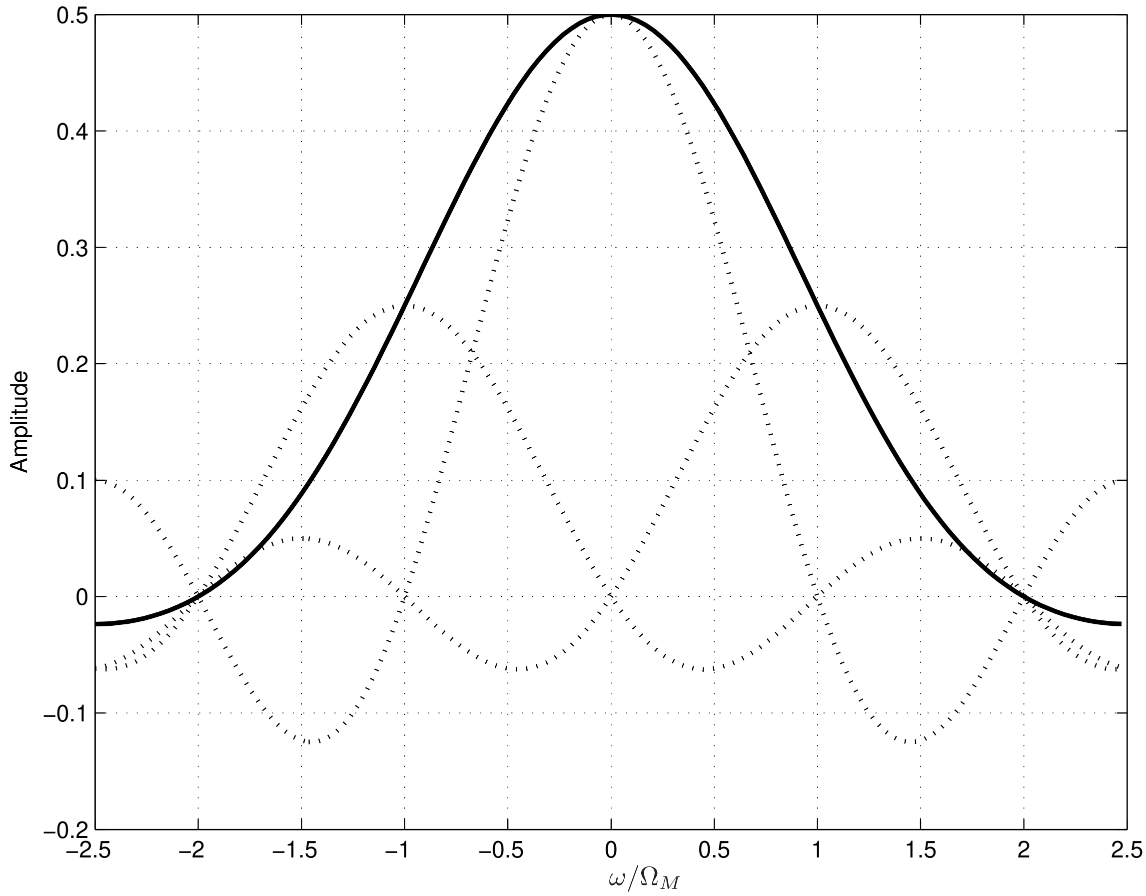
$$\approx \frac{\sin(\pi f MT)}{M\pi f T} \triangleq \text{sinc}(fMT)$$

Points to note:

- Zero crossings at integer multiples of $\Omega_M \triangleq \frac{2\pi}{M}$ where $\Omega_M \triangleq \frac{2\pi}{M} =$ frequency sampling interval for a length M DFT
- Main lobe width is $2\Omega_M = \frac{4\pi}{M}$
- As M gets bigger, the mainlobe narrows (better frequency resolution)
- M has *no effect on the height of the side lobes* (Same as the "Gibbs phenomenon" for Fourier series)
- First sidelobe only 13 dB down from main-lobe peak
- Side lobes roll off at approximately 6 dB per octave
- A *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., centered about time 0)

Generalized Hamming Window Family

Consider the following picture in the frequency domain:



https://ccrma.stanford.edu/~jos/Windows/Generalized_Hamming_Window_Family.html

We have added 2 extra aliased sinc functions (shifted), which results in the following behavior:

- There is some cancellation of the side lobes
- The width of the main lobe is doubled

In terms of the rectangular window transform $W_R(\omega) = M \cdot \text{asinc}_M(\omega)$ (zero-centered, unit-amplitude case), this can be written as:

$$W_H(\omega) \stackrel{\Delta}{=} \alpha W_R(\omega) + \beta W_R(\omega - \Omega_M) + \beta W_R(\omega + \Omega_M)$$

Using the Shift Theorem dual, we can take the inverse transform of the above equation:

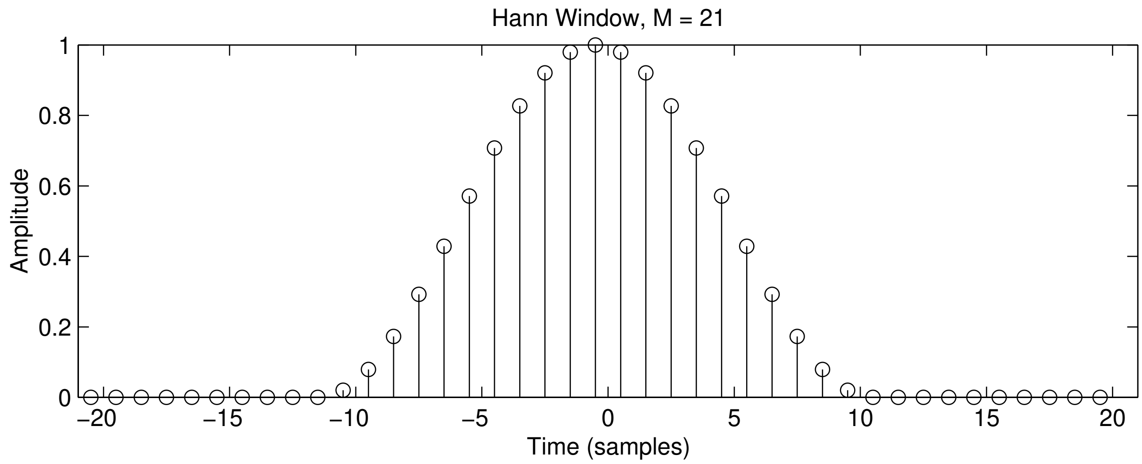
$$\begin{aligned} w_H &= \alpha w_R(n) + \beta e^{-j\Omega_M n} w_R(n) + \beta e^{j\Omega_M n} w_R(n) \\ &= w_R(n) \left[\alpha + 2\beta \cos\left(\frac{2\pi n}{M}\right) \right] \end{aligned}$$

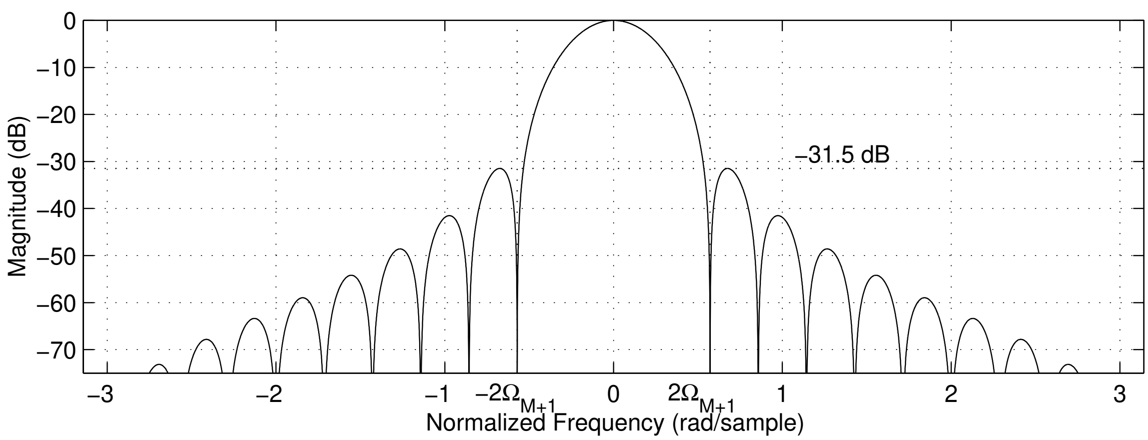
Choosing various parameters for α and β result in different windows in the generalized Hamming family, some of which have names.

Hann or Hanning or Raised Cosine

The Hann window is defined by the settings $\alpha = 1/2$ and $\beta = 1/4$:

$$\boxed{w_H(n) = w_R(n)\left[\frac{1}{2} + \frac{1}{2}\cos(\Omega_M n)\right] = w_R(n)\cos^2\left(\frac{\Omega_M}{2}n\right)}$$





Hann window properties:

- Main lobe is $4\Omega_M$ wide
- First side lobe is at -31 dB
- Side-lobes roll off at ≈ 18 dB / octave

Compare to the Rectangular window:

- Main lobe is $2\Omega_M$ wide
- First side lobe at -13 dB
- Side-lobes roll off at ≈ 6 dB / octave

Hamming

This window is determined by choosing α to cancel the first side lobe and β to normalize peak amplitude to 1 in the time domain:

$$\begin{aligned}\alpha &= \frac{25}{46} \approx 0.54\\ \beta &= (1-\alpha)/2\end{aligned}$$

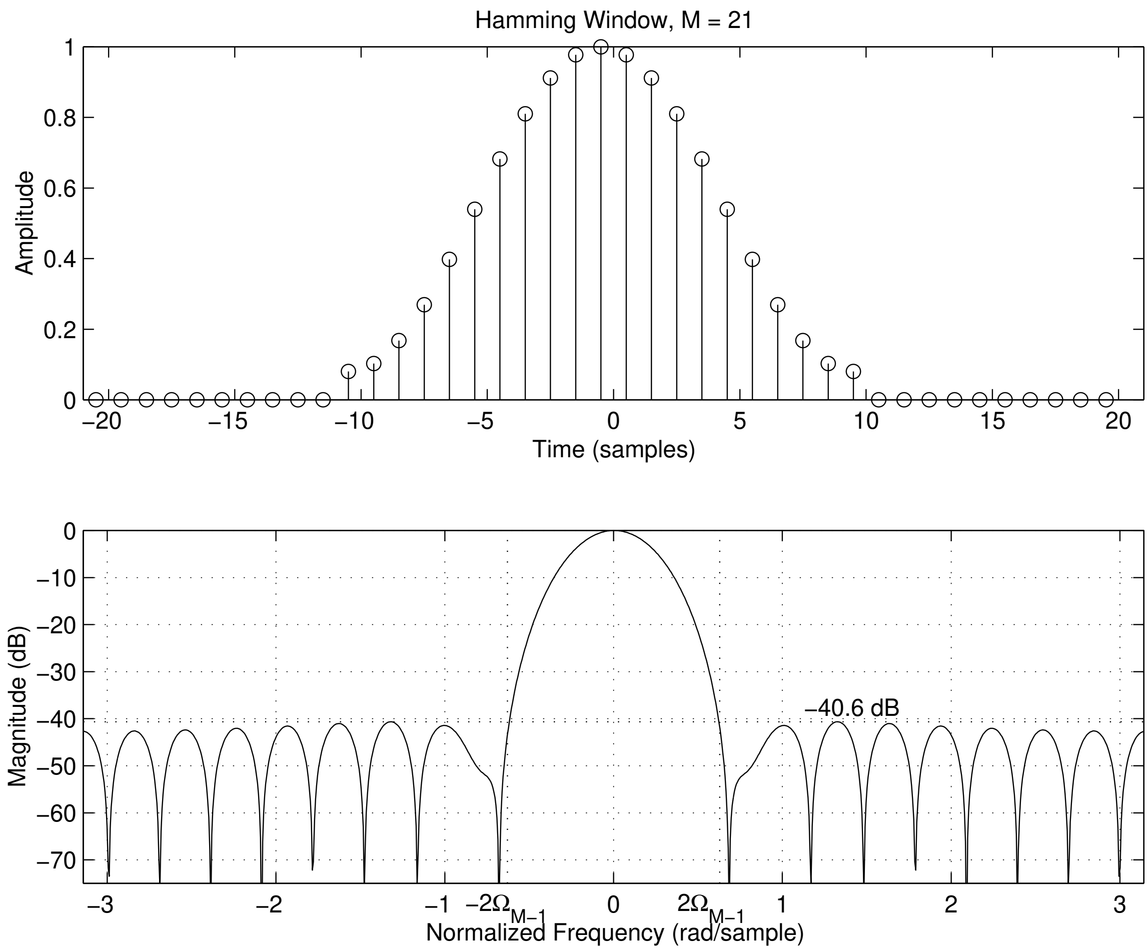
Note: The Hamming window is very close to the generalized Hamming window which *minimizes sidelobe level* within the family:

$\alpha = 0.53836$ (minimum peak side-lobe magnitude)

Thus, the Hamming window is the "Chebyshev Generalized Hamming Window" rounded to two significant digits.

Chebyshev-type designs generally exhibit *equiripple* error behavior, since the worst-case error (sidelobe level in this case) is minimized (see Dolph-Chebyshev window below)

Hamming Window

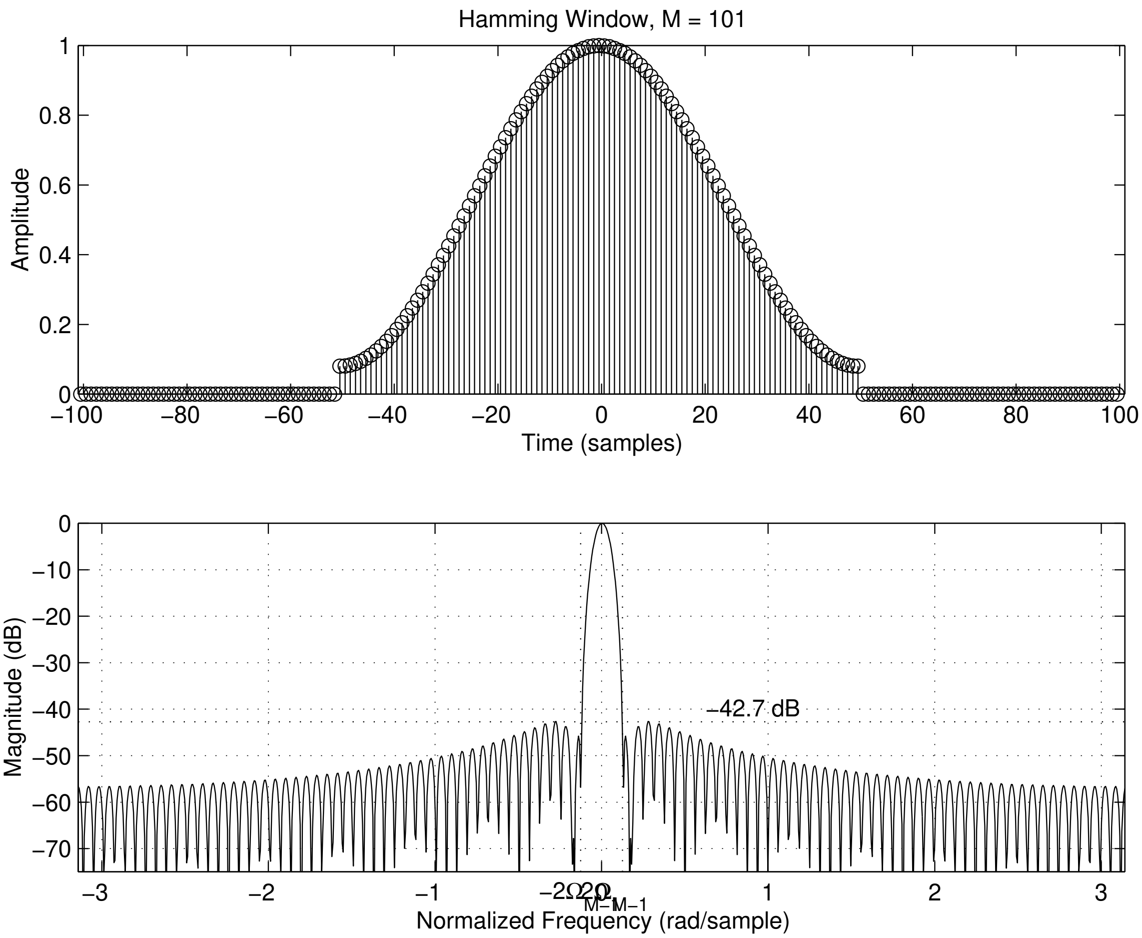


Hamming Window Properties

- Discontinuous "slam to zero" at endpoints
- main lobe is $4\Omega_M$ (like Hann)
- Roll off is approx. 6 dB/octave (but aliased)
- 1st side lobe is improved over Hann
- side lobes closer to "equal ripple"

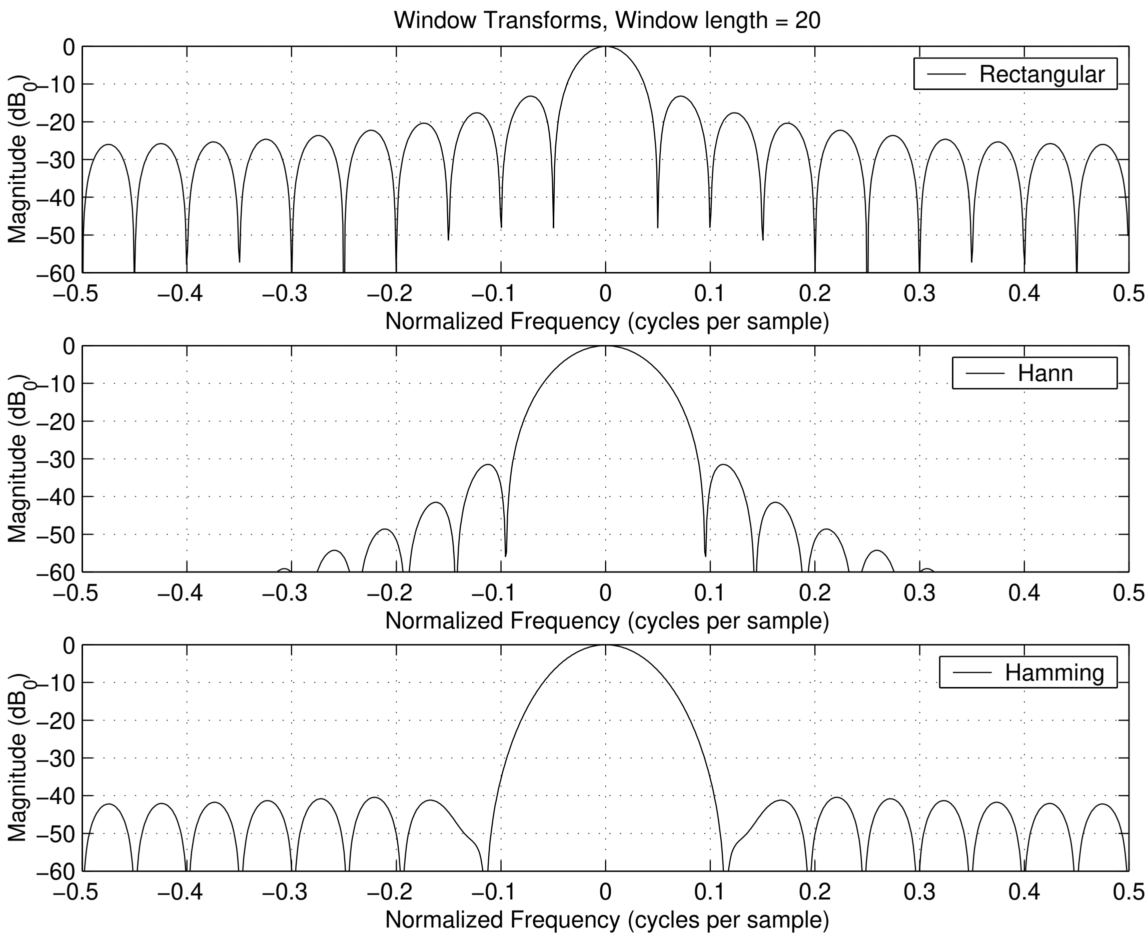
Question: How can side-lobes approximate "equal ripple" when they cannot roll-off slower than 6 dB / octave?

Longer Hamming Window



- Since the side-lobes nearest the main lobe are most affected by the Hamming optimization, we now have a larger frequency region over which the spectral envelope looks like that of the asinc function (an "aliased -6 dB/octave roll-off").
- The side-lobe level (-42.7 dB) is also improved over that of the shorter window (-40.6 dB).

Window Transform Summary



The MLT Sine Window

The *Modulated Lapped Transform* (MLT) uses the *sine window*:

$$w(n) = \sin\left[\left(n + \frac{1}{2}\right)\frac{\pi}{2M}\right], \quad n = 0, 1, 2, \ldots, 2M - 1\,.$$

- Used in MPEG-1, Layer 3 (MP3 format), MPEG-2 AAC, MPEG-4
- Sidelobes 24 dB down
- Asymptotically optimal coding gain
- Zero-phase-window transform ("truncated cosine window") has smallest *moment of inertia* over all windows:

$$\int_{-\pi}^{\pi} \omega^2 W(\omega) d\omega = \text{min}$$

Blackman-Harris Window Family

- The Blackman-Harris family of windows is basically a generalization of the Hamming family.
- In the case of the Hamming family, we constructed a summation of 3 shifted sinc functions.
- The Blackman-Harris family is derived by considering a more general summation of shifted sinc functions:

$$\boxed{w_B(n) = w_R(n) \sum_{l=0}^{L-1} \alpha_l \cos(l\Omega_M n)}$$

where $\Omega_M \stackrel{\Delta}{=} 2\pi/M$,
$n = -(M-1)/2, \ldots (M-1)/2,$ (M odd).

Special Cases:

- $L = 1 \;\Rightarrow$ Rectangular
- $L = 2 \;\Rightarrow$ Generalized Hamming
- $L = 3 \;\Rightarrow$ Blackman Family
- $L > 3 \;\Rightarrow$ Blackman-Harris Family

Frequency-Domain Implementation

The Blackman-Harris window family can be very efficiently implemented in the frequency domain as a $(2L-1)$-point convolution with the spectrum of the unwindowed data. Example:

1. Hann Window $=$ 3-Point DFT_M Smoother:
   - Start with a length M rectangular window
   - Take an M-point DFT
   - Convolve the DFT data with the 3-point smoother $[1/4, 1/2, 1/4]$ to implement a Hann window
   - Note that the Hann window requires *no multiplies* in linear fixed-point data formats
2. Any Blackman window is a 5-point smoother for a Length M (critically sampled) DFT

Classic Blackman

The so-called "Blackman Window" is the specific case in which $\alpha_0 = 0.42$ $\alpha_1 = 0.5$, and $\alpha_2 = 0.08$

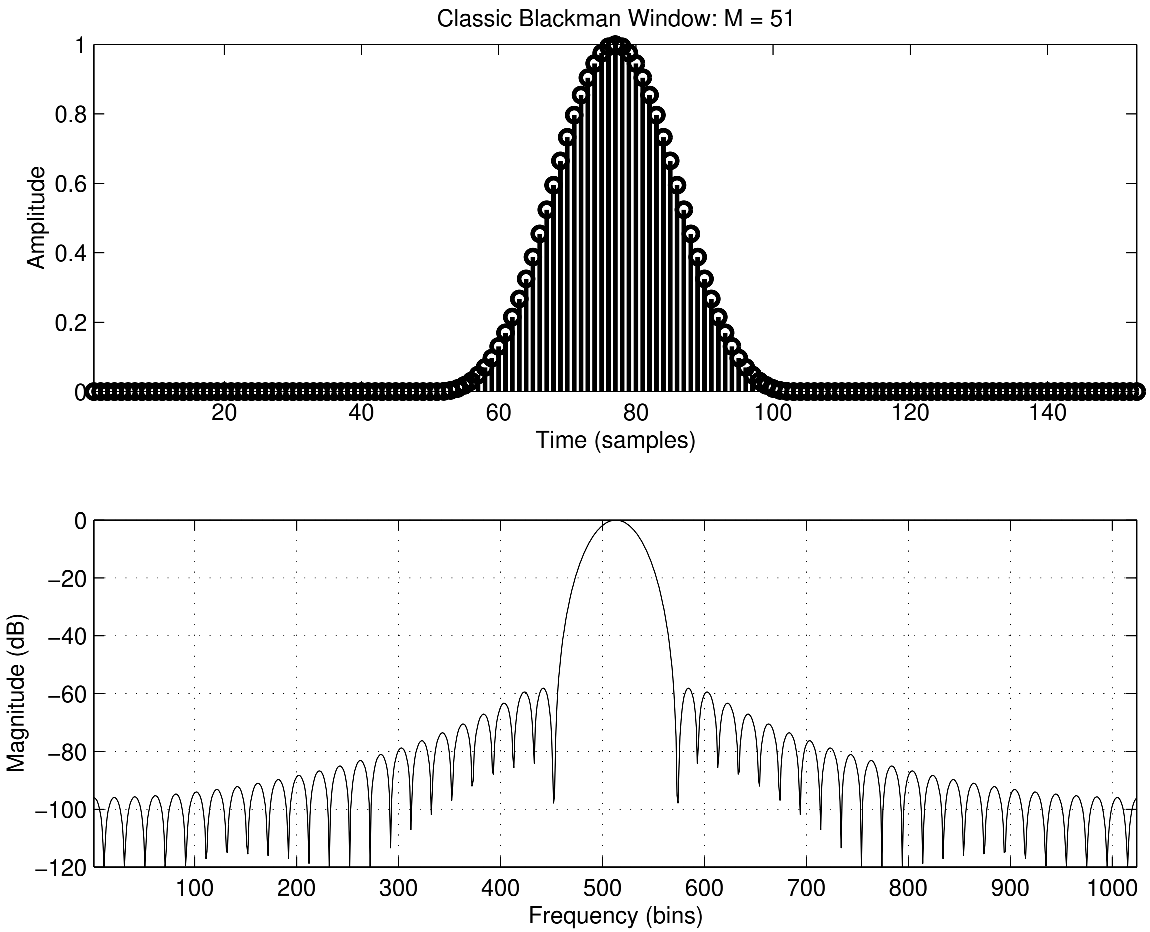
Properties:

- Sidelobes roll off about 18 dB per octave (as $T \to 0$)
- -58 dB sidelobe level (worst case)
- One degree of freedom used to increase the roll-off rate from 6 dB/octave to 18 dB per octave
- One degree of freedom used to minimize sidelobes
- One degree of freedom used to scale the window

Matlab:

```
N = 101; L = 3; No2 = (N-1)/2; n=-No2:No2;
ws = zeros(L,3*N); z = zeros(1,N);
for l=0:L-1
  ws(l+1,:) = [z,cos(l*2*pi*n/N),z];
end
alpha = [0.42,0.5,0.08]; % Classic Blackman
w = alpha * ws;
```

Classic Blackman Window and Transform



Three-Term Blackman-Harris

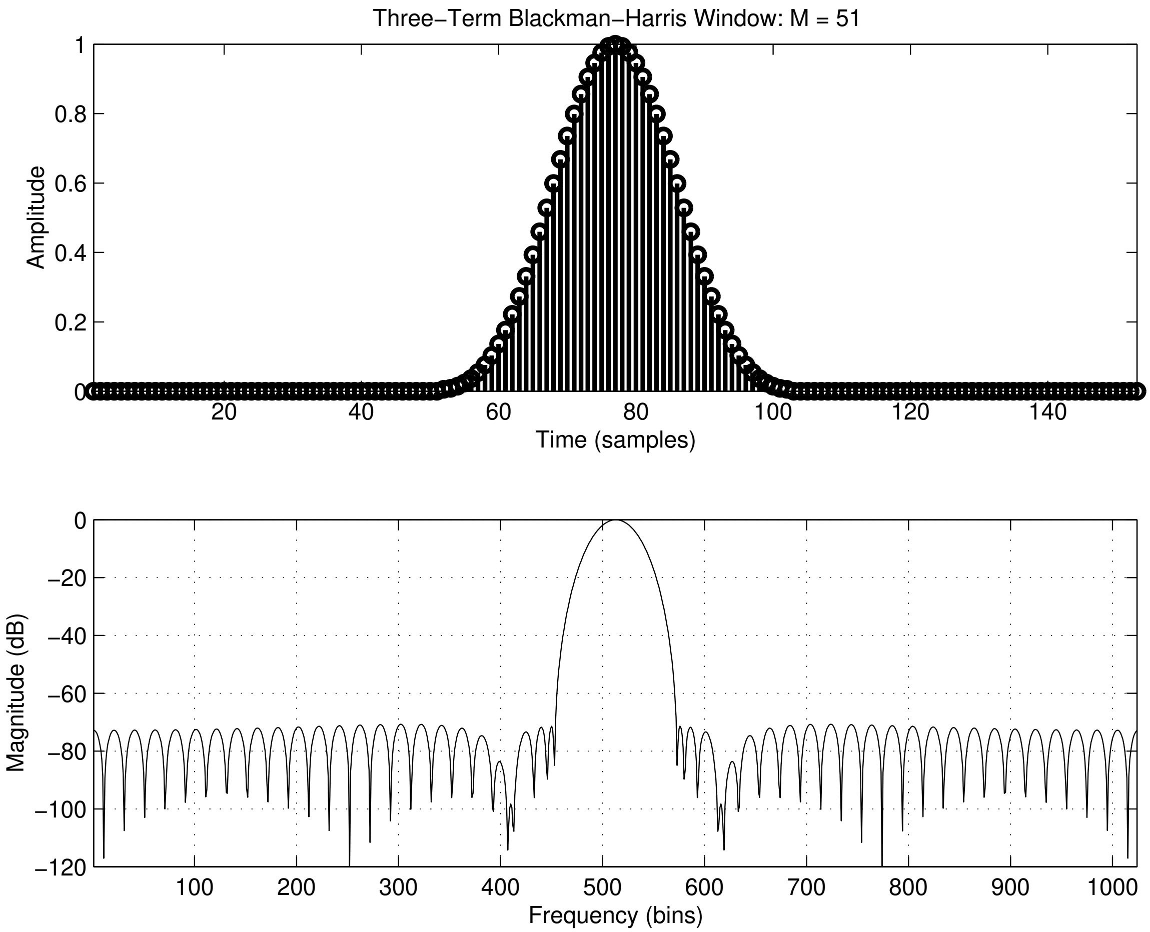
Properties:

- $\alpha_0 = 0.4243801$ $\alpha_1 = 0.4973406$, and $\alpha_2 = 0.0782793$.
- Side-lobe level -71.5 dB.
- Side lobes roll off ≈ 6 dB per octave in the absence of aliasing (like rectangular and Hamming).
- *All degrees of freedom (scaling aside) are used to minimize side lobes* (like Chebyshev-Hamming $\approx$ Hamming).

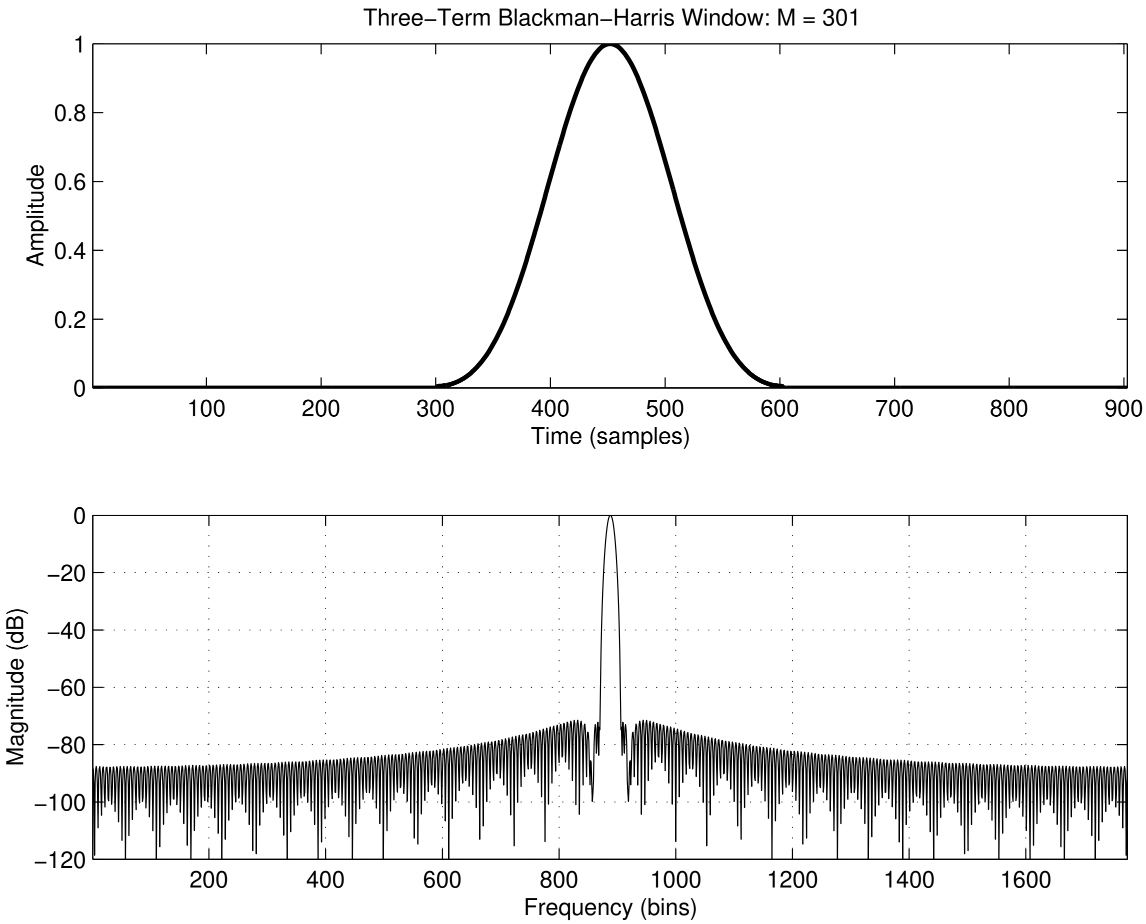
Matlab:

```
N = 101; L = 3; No2 = (N-1)/2; n=-No2:No2;
ws = zeros(L,3*N); z = zeros(1,N);
for l=0:L-1
  ws(l+1,:) = [z,cos(l*2*pi*n/N),z];
end
% 3-term Blackman-Harris(-Nuttall):
alpha = [0.4243801, 0.4973406, 0.0782793];
w = alpha * ws;
```

Three-Term Blackman-Harris Window and Transform



Longer Three-Term Blackman-Harris Window and Transform



Power-of-Cosine

$$w(n) = w_R(n) \cos^P\left(\frac{\pi n}{M}\right), \qquad n \in \left[-\frac{M-1}{2}, \frac{M-1}{2}\right]$$

- $P = 0, 1, 2, \ldots$
- first P terms of its Taylor expansion, evaluated at the endpoints (1/2 sample beyond last sample) are 0
- roll-off rate $\approx 6(P+1)$ dB/octave
- $P = 0 \Rightarrow$ Rectangular window
- $P = 1 \Rightarrow$ MLT sine window (shifted to zero-phase)
- $P = 2 \Rightarrow$ Hann window ("raised cosine" = "$\cos^2$")
- $P = 4 \Rightarrow$ Alternate Blackman (max roll-off rate in Blackman family)
- $\cdots$

Thus, $\cos^P$ windows parametrize Lth-order Blackman-Harris windows configured to use all degrees of freedom to maximize roll-off rate ($L = P/2 + 1$)

Spline Windows

A spline window of order N is a repeated convolution of rectangular windows:

$$\begin{aligned} w_{\mathsf{Spline}(N)}(n) &= (\underbrace{w_R * w_R * \cdots * w_R}_{N+1})(n) \\ &\leftrightarrow\ W_{\mathsf{Spline}(N)}(\omega) = \operatorname{asinc}^{N+1} \end{aligned}$$

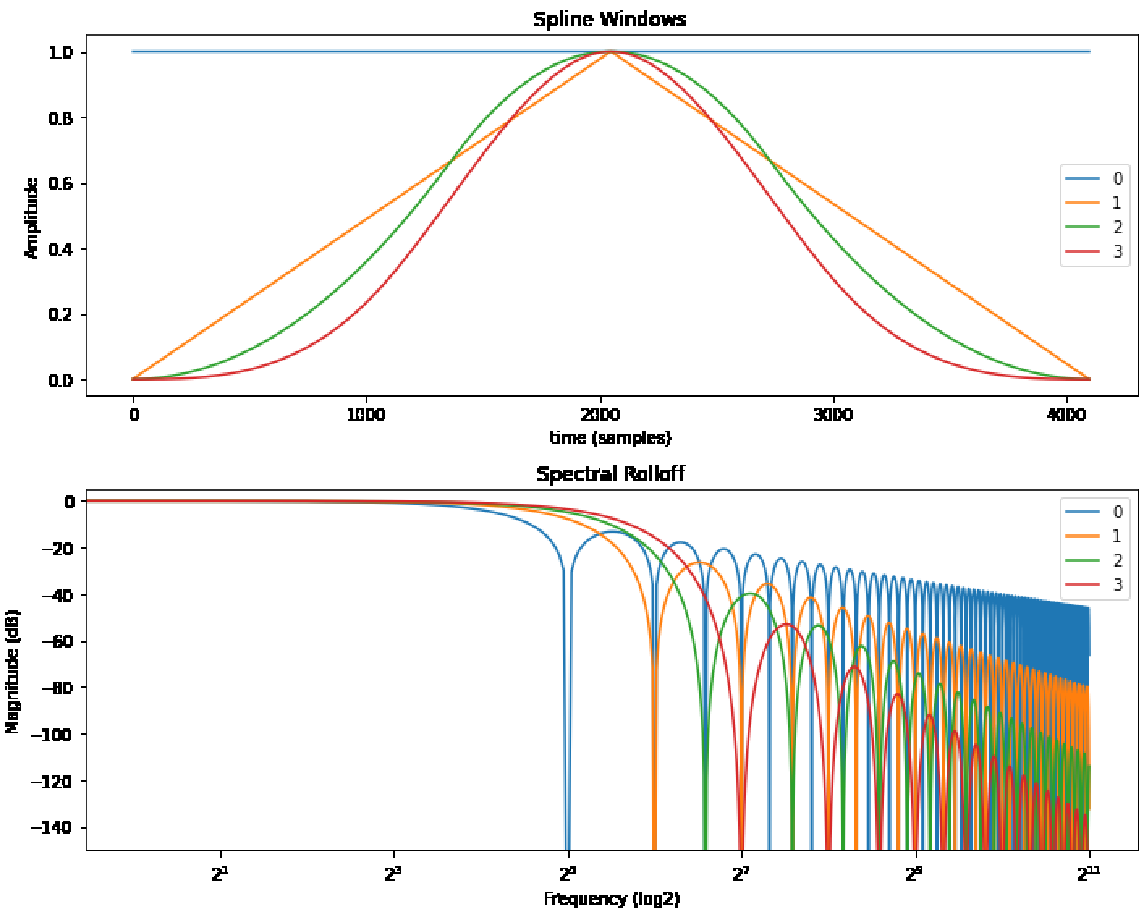
Special Cases:

- $N = 0 \ \Rightarrow$ Rectangular (constant)
- $N = 1 \ \Rightarrow$ Triangular (linear)
- $N = 2 \ \Rightarrow$ Quadratic
- $N = 3 \ \Rightarrow$ Cubic

Roll-Off Rate:

As N increases, the window becomes smoother. $w_{\mathsf{Spline}(N)}$ is $(N-1)$-times continuously differentiable, and has roll-off rate $6(N+1)$ dB per octave.

Spline Window Examples



Miscellaneous Windows

Bartlett ("Triangular")

$$w(n) = w_R(n)\left[1 - \frac{|n|}{(M-1)/2}\right]$$

- Convolution of two half-length rectangular windows
- Window transform is sinc^2 $\Longrightarrow$
  - First sidelobe twice as far down as rect (-26 dB)
  - Main lobe twice as wide as that of a rectangular window *having the same length*
    (same as that of a half-length rect used to make it)
- Often applied to *sample correlations* of finite data
- Also called the "tent function"
- $M-1$ often replaced by M or $M+1$ to avoid including endpoint zeros

Using Any Window as a Tapering Function

Sometimes we need a wide rectangular window with *tapered edges*:

1. *Split* any window into halves, inserting the rectangle between
2. *Convolve* the rectangular window with any desired window

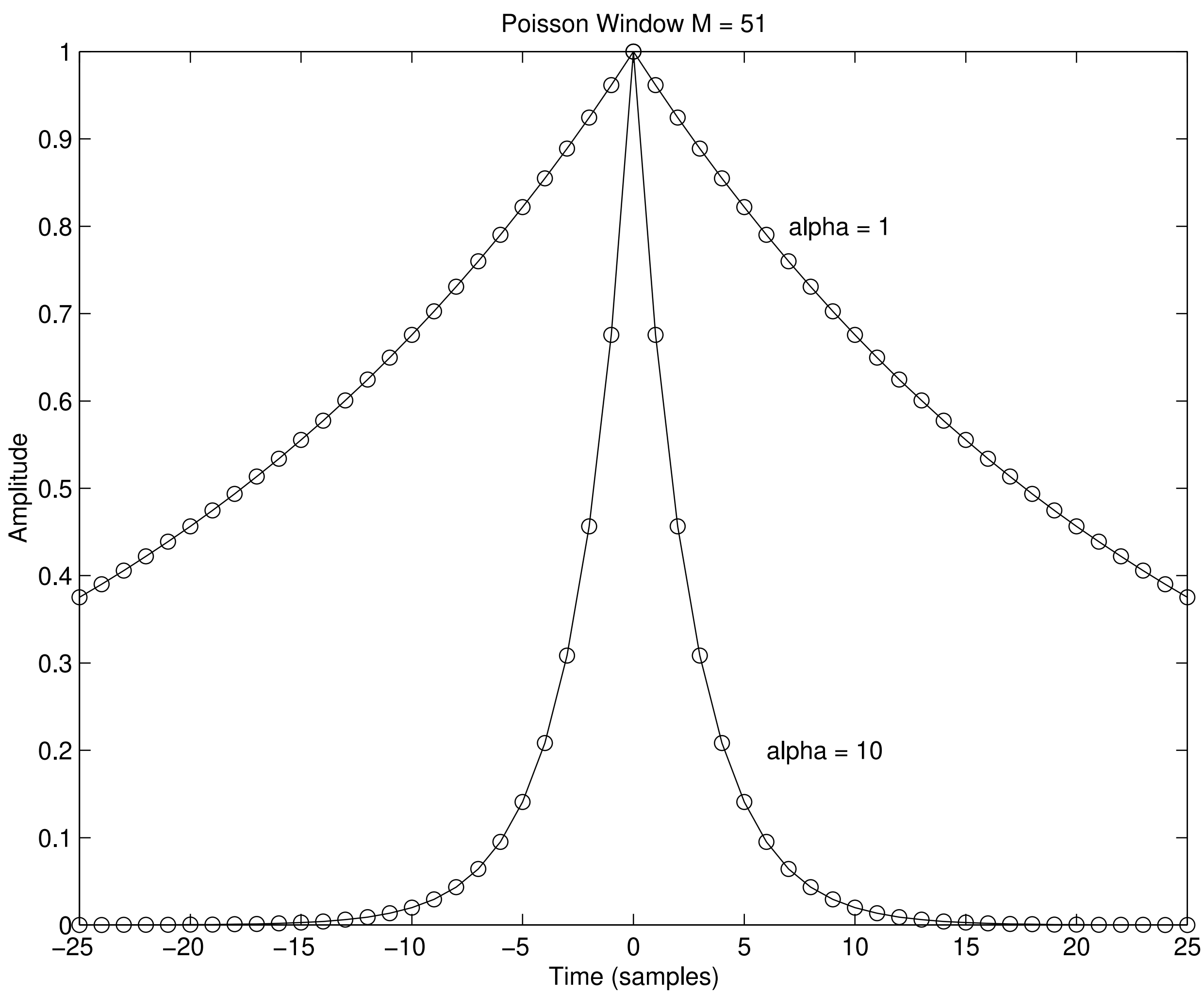
Method 2 preserves *smoothness* of the tapering window and hence its *roll-off rate*.

Poisson ("Exponential")

$$w_P(n) = w_R(n) e^{-\alpha \frac{|n|}{(M-1)/2}}$$

where α determines the time constant τ:

$$\frac{\tau}{T} = \frac{M-1}{2\alpha} \quad \text{samples}$$

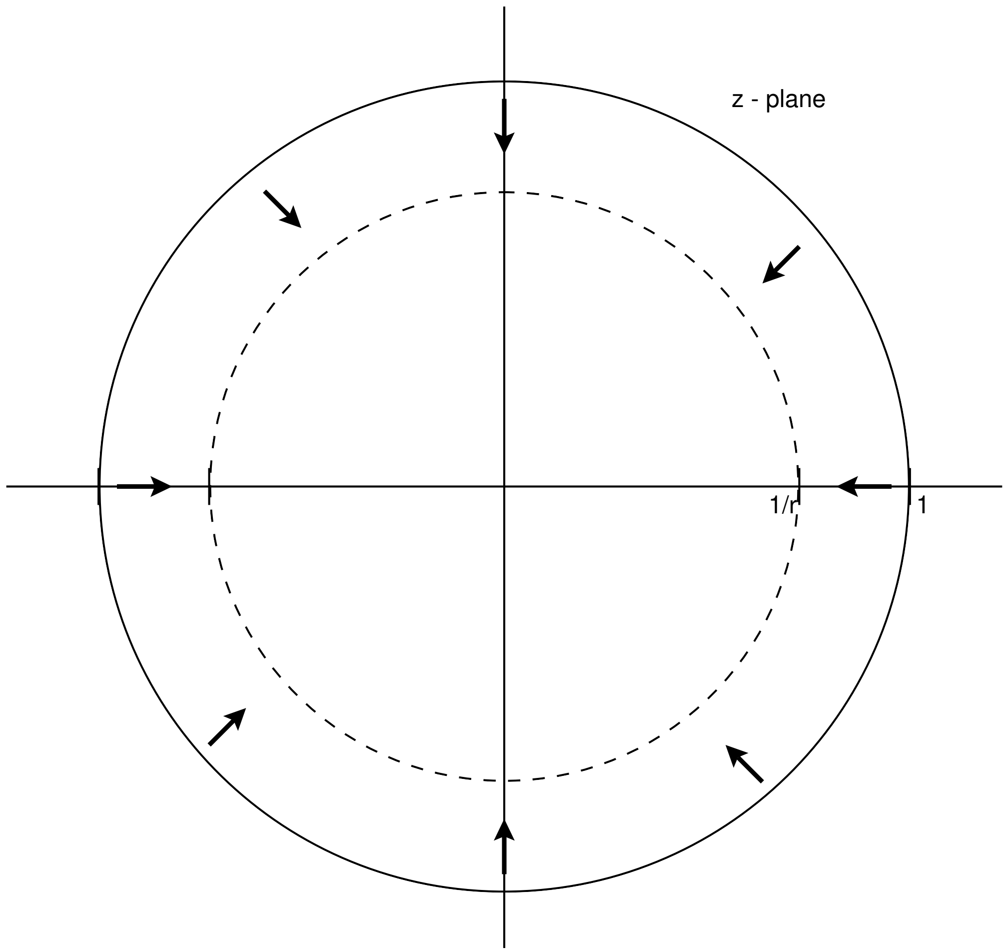


Poisson Window in System Identification

In the z-plane, the Poisson window has the effect of contracting the spectrum toward zero inside unit circle. Consider an *infinitely long* Poisson window (no truncation by a rectangular window w_R) applied to a causal signal $h(n)$ having z transform $H(z)$:

$$
\begin{aligned}
H_P(z) &= \sum_{n=0}^{\infty} [w(n)h(n)] z^{-n} \\
&= \sum_{n=0}^{\infty} \left[h(n) e^{-\frac{\alpha n}{M/2}} \right] z^{-n} \qquad \text{(let } r \stackrel{\Delta}{=} e^{\frac{\alpha}{M/2}}\text{)} \\
&= \sum_{n=0}^{\infty} h(n) z^{-n} r^{-n} = \sum_{n=0}^{\infty} h(n) (zr)^{-n} \\
&= H(zr)
\end{aligned}
$$

- Unit-circle response moved to $|z| = 1/r < 1$
- Marginally stable poles now decay as $r^{-n} = e^{-\alpha n/(M/2)}$



The Poisson window can be useful for impulse-response modeling by poles and/or zeros ("system identification"). In such applications, the window length is best chosen to include substantially all of the impulse-response data.

Hann-Poisson ("No Sidelobes")

$$w(n) = \frac{1}{2}\left[1 + \cos\left(\pi \frac{n}{(M-1)/2}\right)\right] e^{-\alpha \frac{|n|}{(M-1)/2}}$$

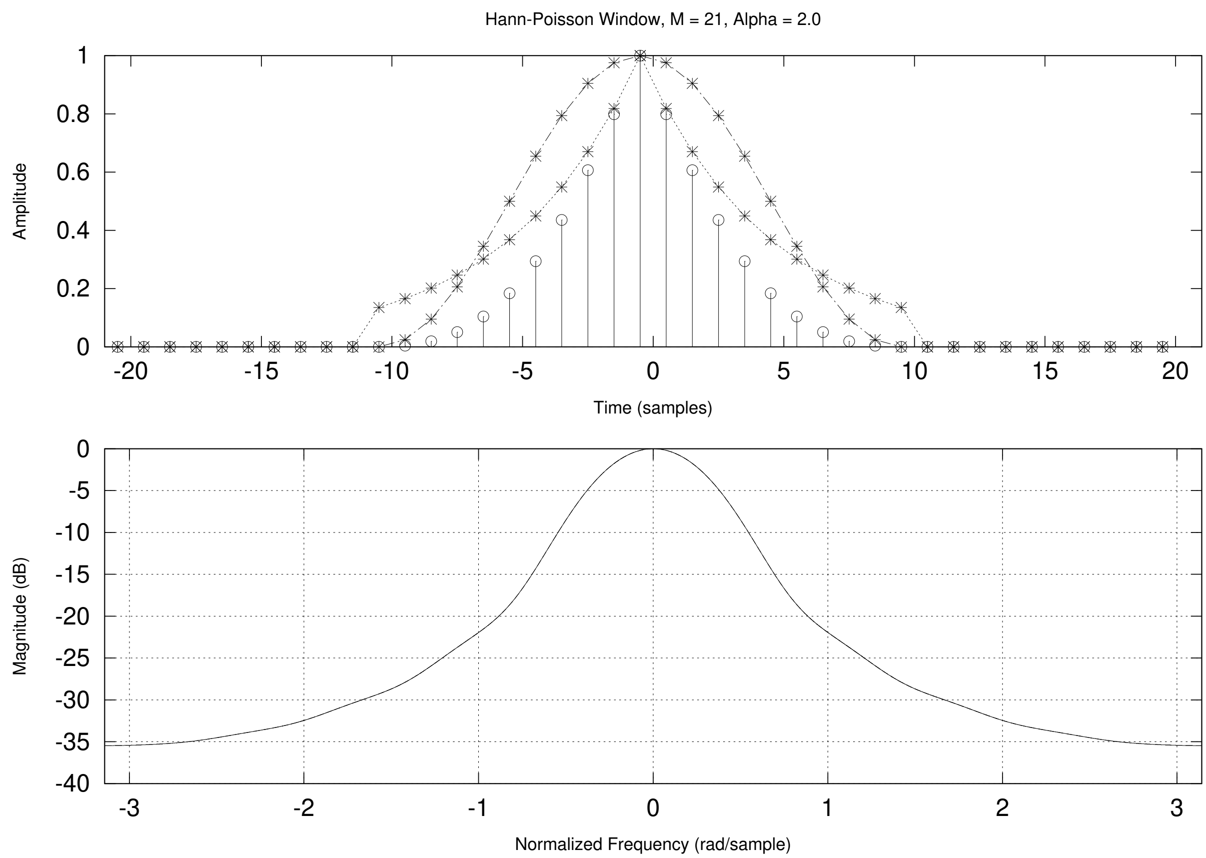
- Poisson window times Hann window
  (exponential times raised cosine)
- "No sidelobes" for $\alpha \geq 2$
- Valuable for "hill climbing" optimization methods
  (gradient-based)

Matlab:

```
function [w,h,p] = hannpoisson(M,alpha)
%HANNPOISSON  - Length M Hann-Poisson window

Mo2 = (M-1)/2; n=(-Mo2:Mo2)';
scl = alpha / Mo2;
p = exp(-scl*abs(n));
scl2 = pi / Mo2;
h = 0.5*(1+cos(scl2*n));
w = p.*h;
```

Hann-Poisson Window and Transform

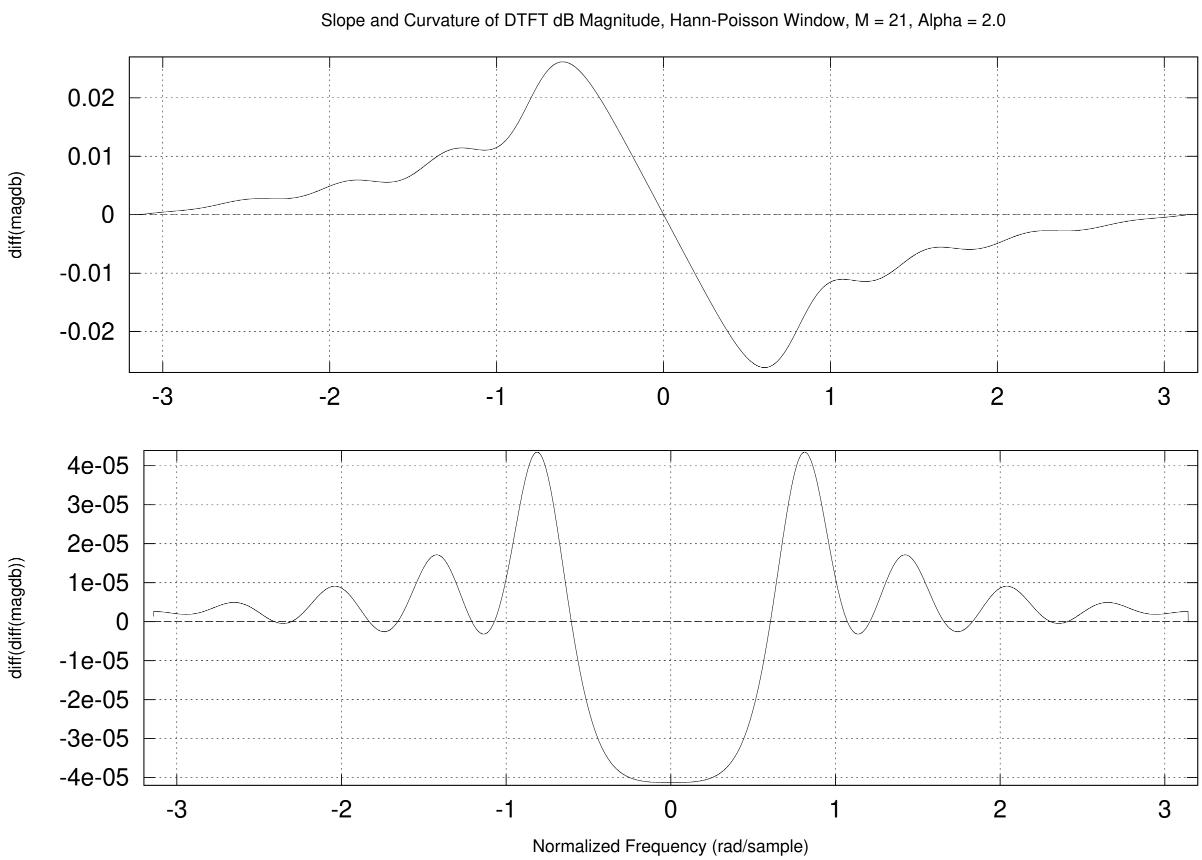


https://ccrma.stanford.edu/~jos/Windows/Hann_Poisson_Window_Transform.html

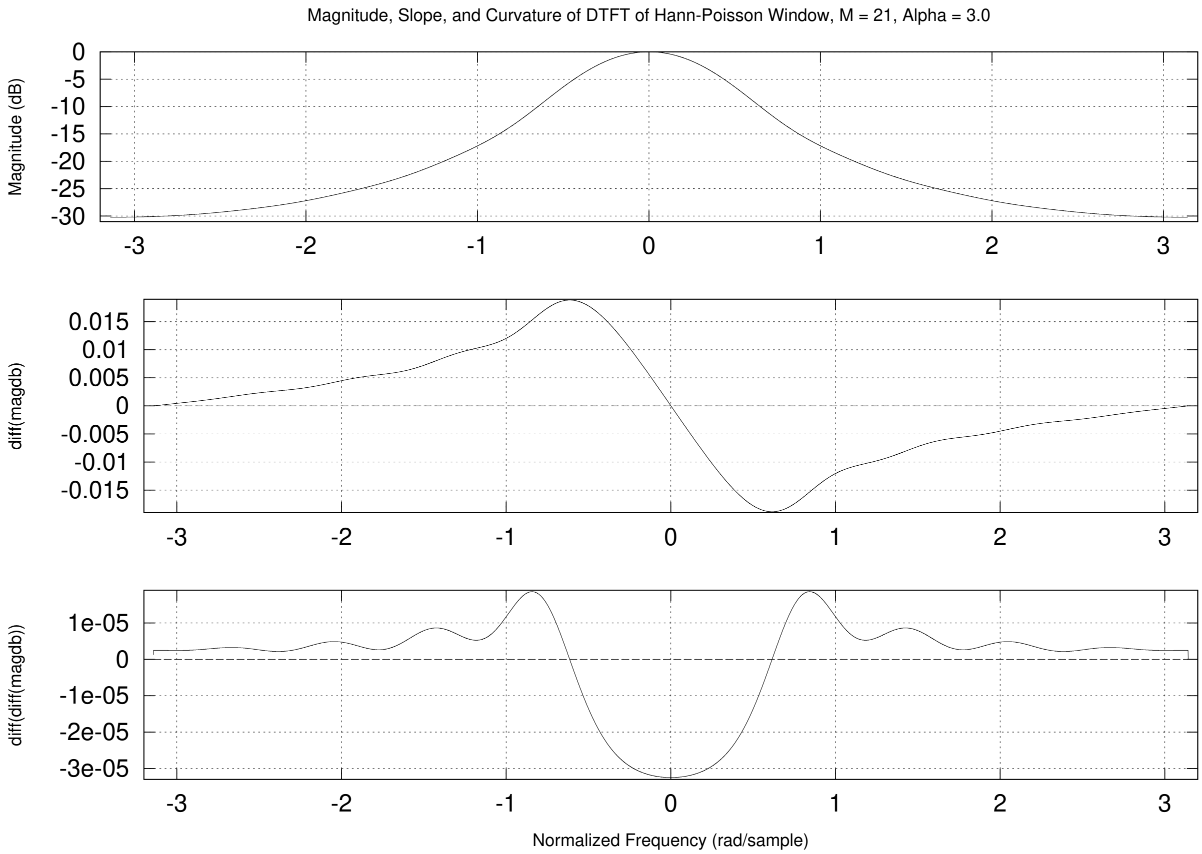
Question:

How can a truncated *anything* avoid having ripples in its spectrum? ("Gibbs ripples")
Let's look at the *derivatives* of the window:

Hann-Poisson Slope and Curvature



Slope and Curvature for Larger Alpha



Maximum Main-Lobe Energy Window: DPSS

Question: How do we use *all* M degrees of freedom (sample values) in an M-point window $w(n)$ to obtain $W(\omega) \approx \delta(\omega)$ in some optimal sense?

That is, we wish to perform the following optimization:

$$\max_{w} \left[\frac{\text{main lobe energy}}{\text{total energy}} \right]$$

In the continuous-time case $[\omega \in (-\infty, \infty)]$, this problem is solved by a *prolate spheroidal wave function*, an eigenfunction of the integral equation

$$\int_{-\omega_c}^{\omega_c} W(\nu) \frac{\sin[\pi D \cdot (\omega - \nu)]}{\pi(\omega - \nu)} \, d\omega = \lambda W(\omega), \;\; |\omega| \leq \omega_c$$

where D is the nonzero time-duration of $w(t)$ in seconds.

Interpretation:

$$\begin{aligned} & [\textsc{Chop}_{2\omega_c}(W)] * [D \, \mathsf{sinc}(D\omega)] \\ = \;& \mathrm{FT}(\textsc{Chop}_D(\mathrm{IFT}(\textsc{Chop}_{2\omega_c}(W)))) \; = \; \lambda W \end{aligned}$$

where $\textsc{Chop}_D(w)$ is a rectangular windowing operation which zeros w outside the interval $t \in [-D/2, D/2]$.

W is thus the *bandlimited extrapolation* of its main lobe ($\omega \in [-\omega_c, \omega_c]$)

The optimal window transform W is an eigenfunction of this operation sequence corresponding to the *largest eigenvalue*.

The resulting optimal window w has maximum main-lobe energy as a fraction of total energy.

It may be called the *Slepian window*, or *prolate spheroidal window* in the continuous-time case.

In discrete time, we need *Discrete Prolate Spheroidal Sequences (DPSS)*, eigenvectors of the following symmetric Toeplitz matrix constructed from a sampled sinc function:

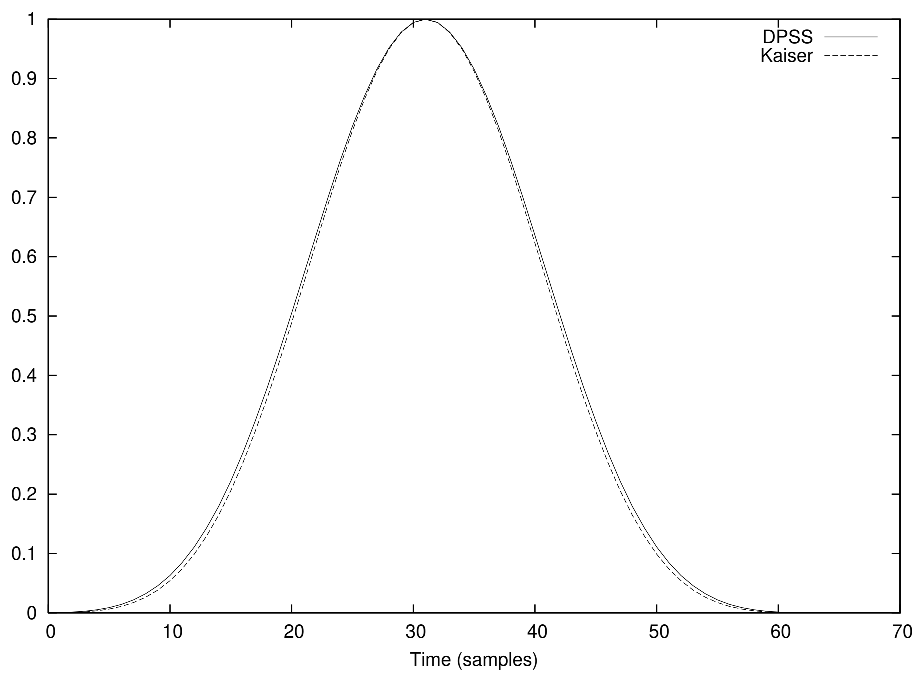
$$S[k,l] = \frac{\sin[\omega_c T(k-l)]}{k-l}, \quad k,l = 0,1,2,\ldots,M-1$$

- $M =$ window length in samples
- $\omega_c =$ main-lobe cut-off frequency (rad/sec)
- $T =$ sampling period in seconds.

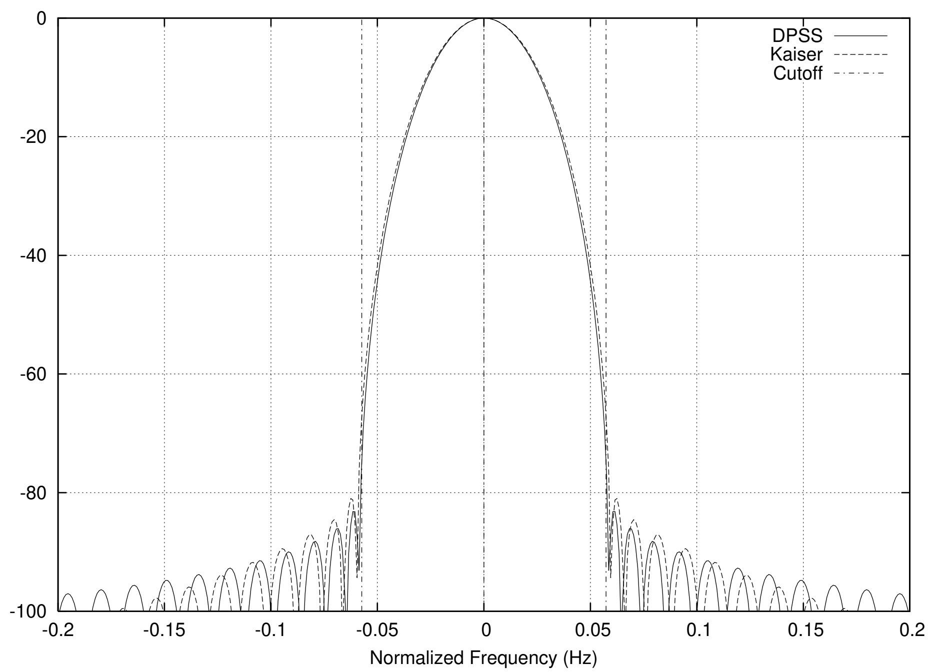
The DPSS window (digital Slepian window) is then given by the eigenvector corresponding to the largest eigenvalue.

Matlab for the DPSS Window

```
function [w,A,V] = dpssw(M,Wc);
% DPSSW - Digital Prolate Spheroidal Sequence windo
%         (Slepian window) of length M, having
%         cut-off frequency Wc in (0,pi).
k = (1:M-1);
s = sin(Wc*k)./ k;
c0 = [Wc,s];
A = toeplitz(c0);   % c0=1st col of symm. Toeplitz
[V,evals] = eig(A); % Only need principal eigenvect
[emax,imax] = max(abs(diag(evals)));
w = V(:,imax);
w = w / max(w);
```



Kaiser and DPSS Window Transforms Compared



- Kaiser $\approx$ DPSS
- DPSS window has a slightly narrower main lobe
- DPSS window has lower overall side-lobe levels
- Kaiser window side lobes roll off faster
- Otherwise they are very similar

Kaiser (Kaiser-Bessel)

Kaiser discovered a very good approximation to prolate spheroidal wave functions using Bessel functions:

$$w_K(n) \stackrel{\Delta}{=} \begin{cases} \frac{I_0\left(\beta\sqrt{1-\left(\frac{n}{M/2}\right)^2}\right)}{I_0(\beta)}, & -\frac{M-1}{2} \leq n \leq \frac{M-1}{2} \\ 0, & \text{elsewhere} \end{cases}$$

$$w_K(n) \stackrel{\Delta}{=} w_R(n)\frac{I_0\left(\beta\sqrt{1-\left(\frac{n}{M/2}\right)^2}\right)}{I_0(\beta)}$$

This is called the *Kaiser* (or *Kaiser-Bessel*) window.

The Fourier transform of the Kaiser window $w_K(t)$ (where t is treated as continuous) is given by

$$\begin{aligned} W(\omega) &= \frac{M}{I_0(\beta)}\frac{\sinh\left(\sqrt{\beta^2-\left(\frac{M\omega}{2}\right)^2}\right)}{\sqrt{\beta^2-\left(\frac{M\omega}{2}\right)^2}} \\ &= \frac{M}{I_0(\beta)}\frac{\sin\left(\sqrt{\left(\frac{M\omega}{2}\right)^2-\beta^2}\right)}{\sqrt{\left(\frac{M\omega}{2}\right)^2-\beta^2}} \end{aligned}$$

where I_0 is the zero-order modified Bessel function of the first kind.

Modified Bessel Function of the 1st Kind

A *series expansion* for the order zero, modified Bessel function of the first kind is given by

$$I_0(x) = \sum_{k=0}^{\infty} \left[\frac{(x/2)^k}{k!} \right]^2$$

Compare this with

$$e^{x/2} = \sum_{k=0}^{\infty} \frac{(x/2)^k}{k!}.$$

Kaiser-Bessel Window Notes

$$w_K(n) \;\stackrel{\Delta}{=}\; w_R(n)\frac{I_0\left(\beta\sqrt{1-\left(\frac{n}{M/2}\right)^2}\right)}{I_0(\beta)}$$

- "Closed form" (given I_0 series or table)
- Reduces to rectangular window for $\beta = 0$
- Asymptotic roll-off is 6 dB/octave
- For $\beta \gg 0$, first null in window transform is at $\omega_0 \approx 2\beta/M$
  $\Rightarrow \;\; \beta = M\omega_0/2$
- Sometimes the Kaiser window is parameterized by α:

$$\beta \stackrel{\Delta}{=} \pi\alpha$$

Kaiser Window Time-Bandwidth Product

- Define the main-lobe "cutoff frequency" as half-way to the first zero in $W(\omega)$:

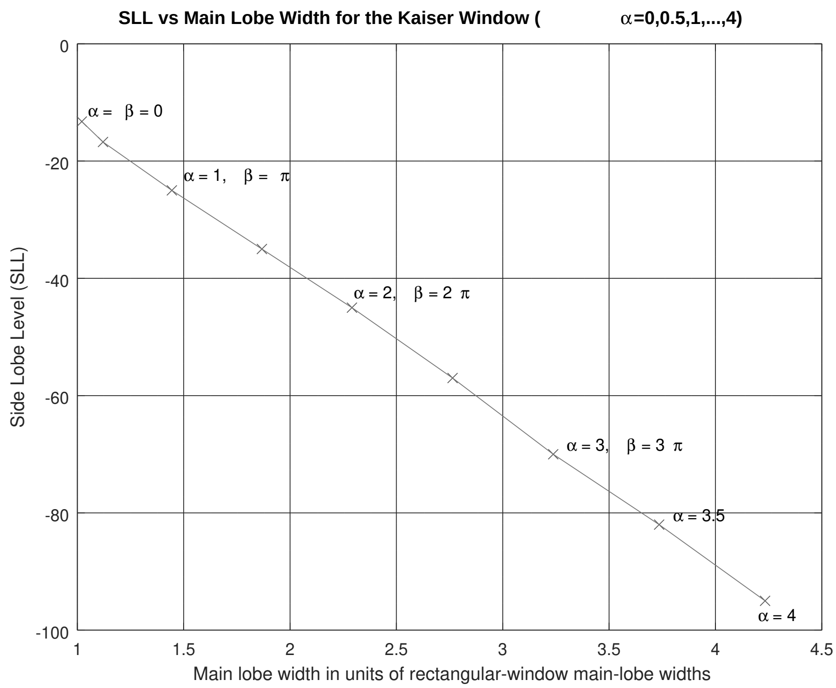
$$\omega_c = \frac{\omega_0}{2} = \frac{\beta}{M} = \frac{\pi\alpha}{M}$$

- Then

$$\beta = M\omega_c = \frac{1}{2}M \cdot (2\omega_c)$$
$$= \frac{1}{2}\,\text{duration (samples)} \times \text{bandwidth (rad/sample)}$$
$$\alpha = \frac{\beta}{\pi} = \frac{2\beta}{2\pi}$$
$$= \text{duration (samples)} \times \text{bandwidth (cycles/sample)}$$

- $\beta = M\omega_c T$ is equal to 1/2 'time-bandwidth product'
$\beta = \frac{1}{2}\Delta t \cdot \Delta\omega \Rightarrow \quad \alpha = \Delta t \cdot \Delta f$
- In this definition of time-bandwidth product, the "cut-off frequency ω_c of the Kaiser-window transform is defined as *half* of the first null frequency, *i.e.*, $\omega_c = \omega_0/2$.
- β trades off side lobe level for main lobe width
larger $\beta \Rightarrow$ lower S.L.L., wider mainlobe

Kaiser Side-Lobe Level vs. Main-Lobe Width

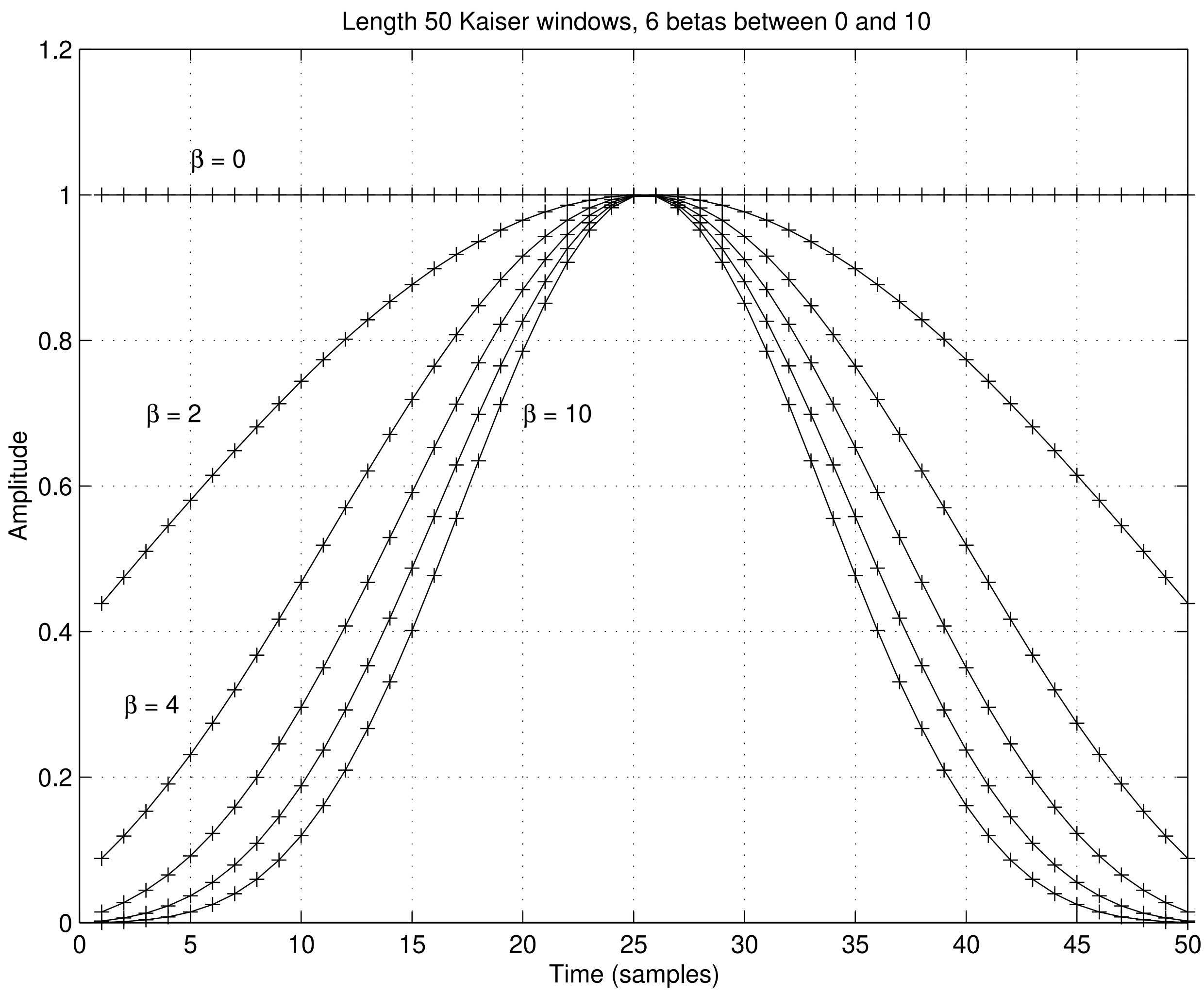


Mathematica Demonstration:

`http://demonstrations.wolfram.com/KaiserWindowTransform/`

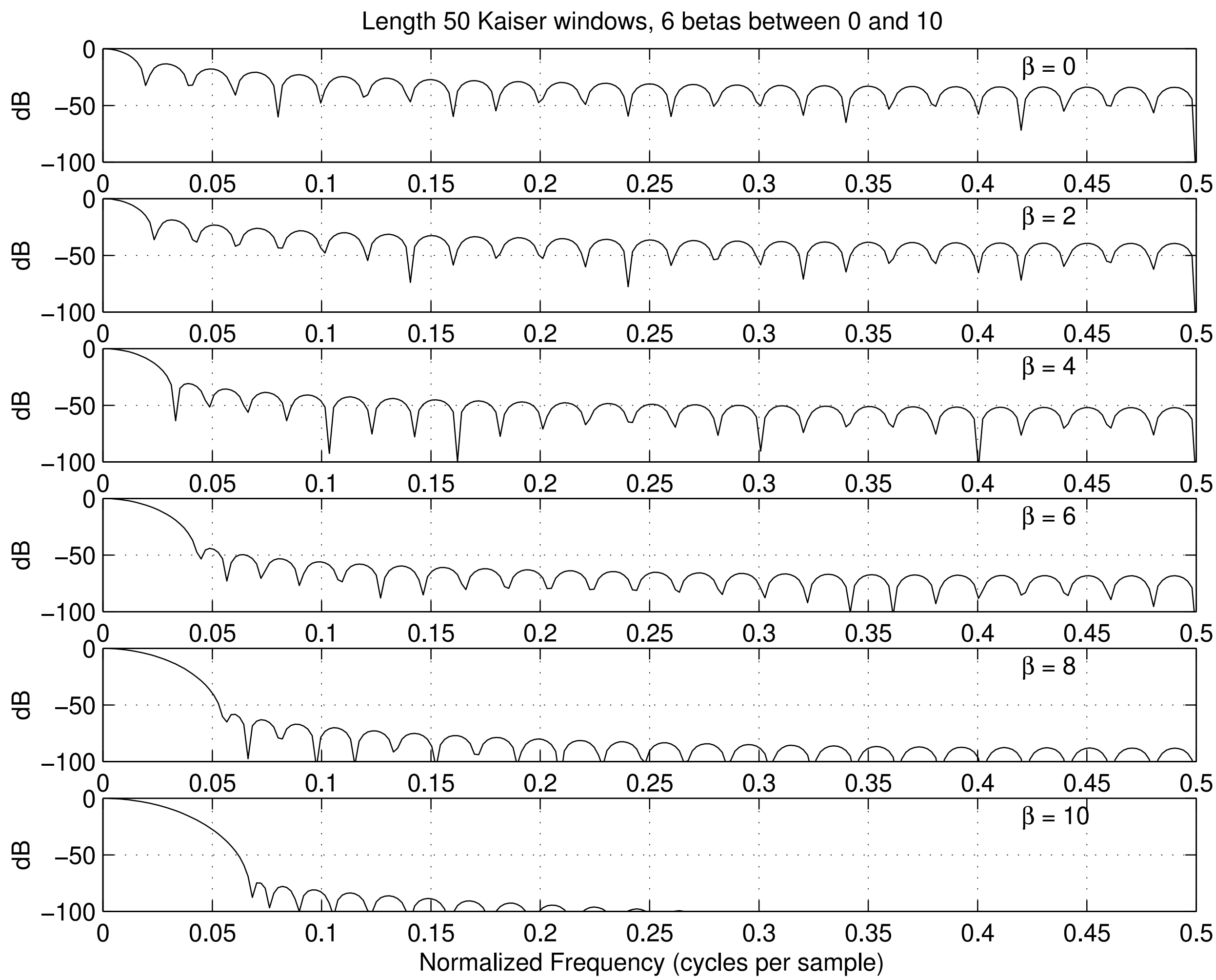
Kaiser Window Examples

$$\beta = [0, 2, 4, 6, 8, 10]$$



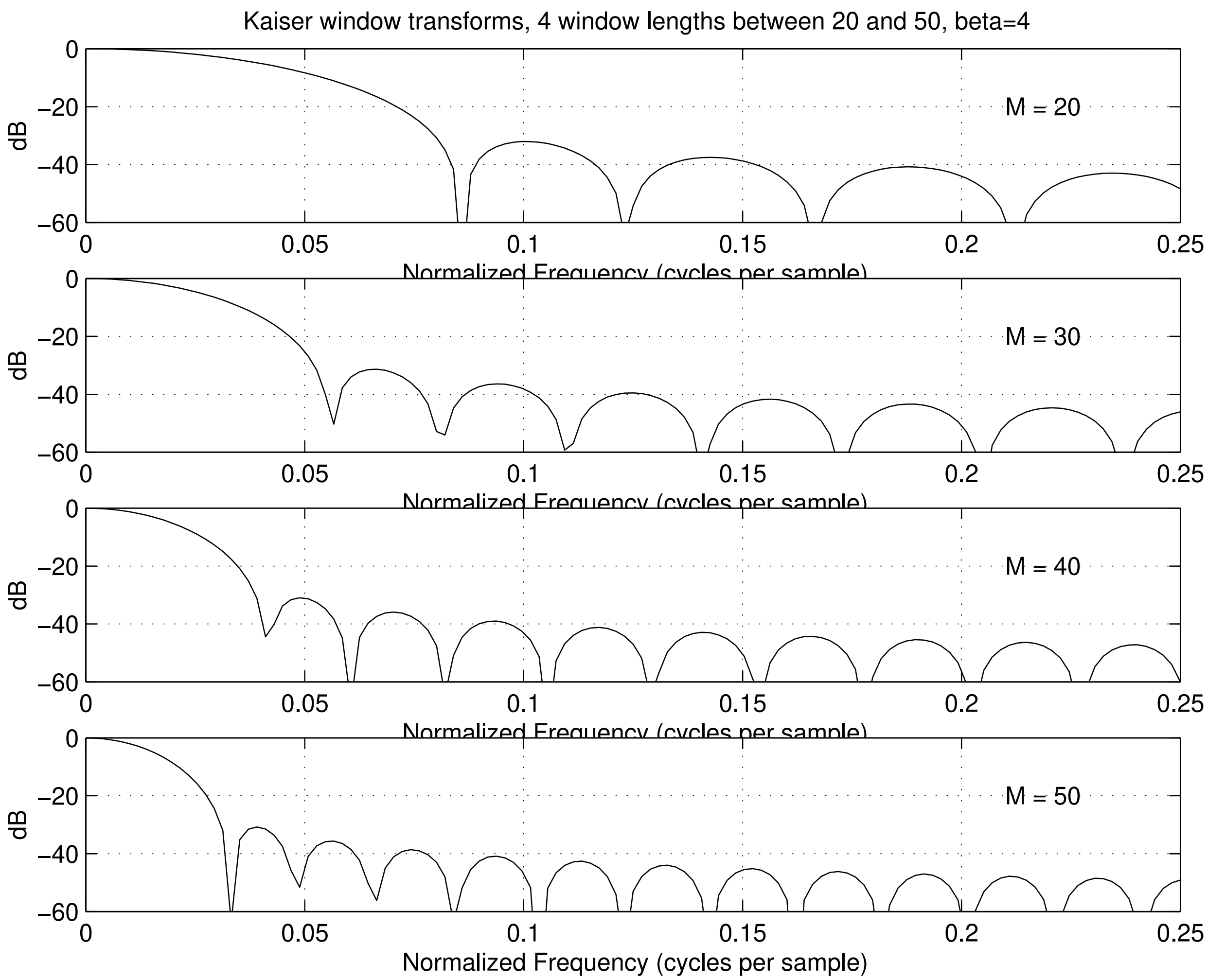
Kaiser Window Transform Examples

$$\beta = [0, 2, 4, 6, 8, 10]$$



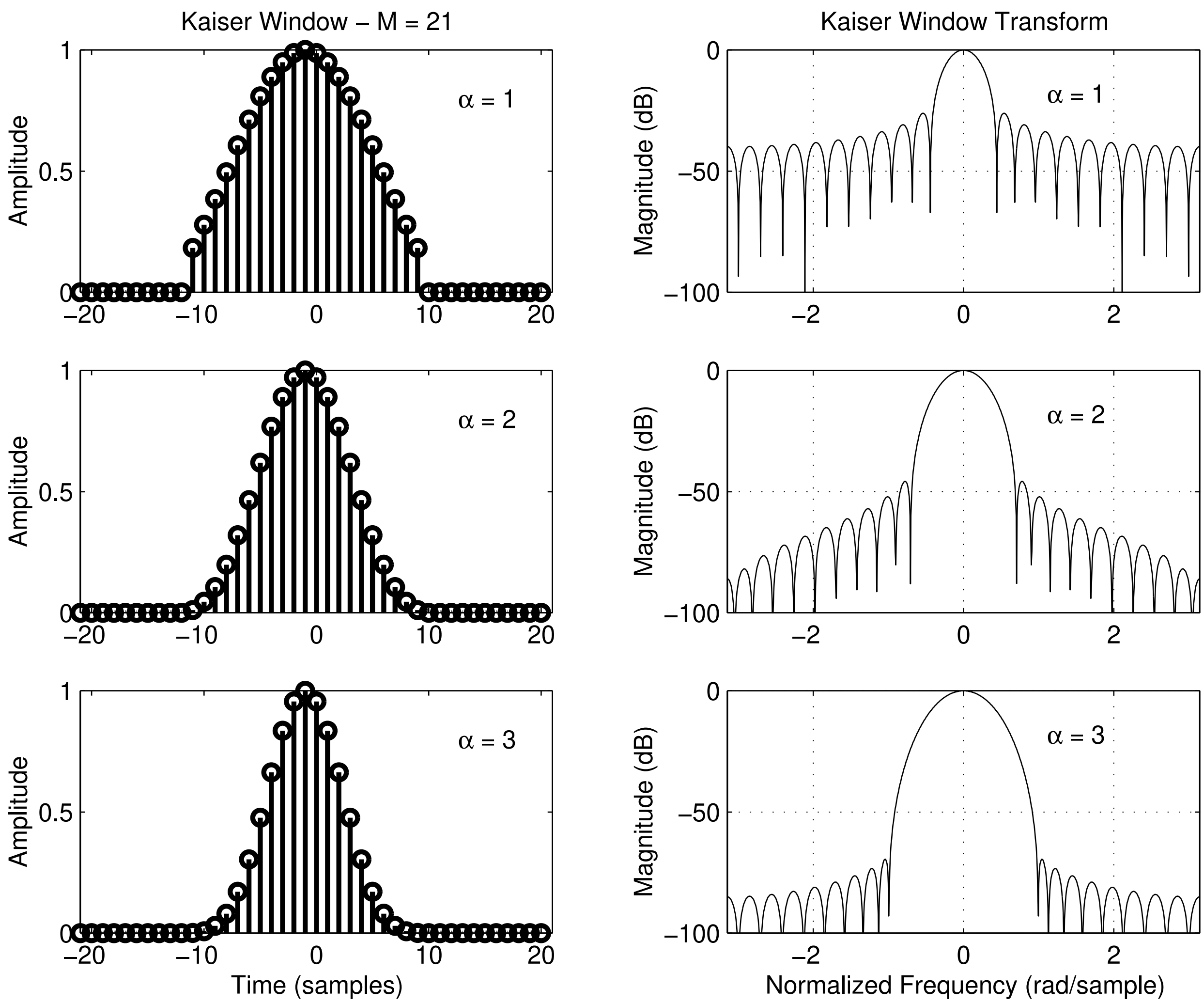
Kaiser Window: Different Lengths M

$$M = [20, 30, 40, 50]$$

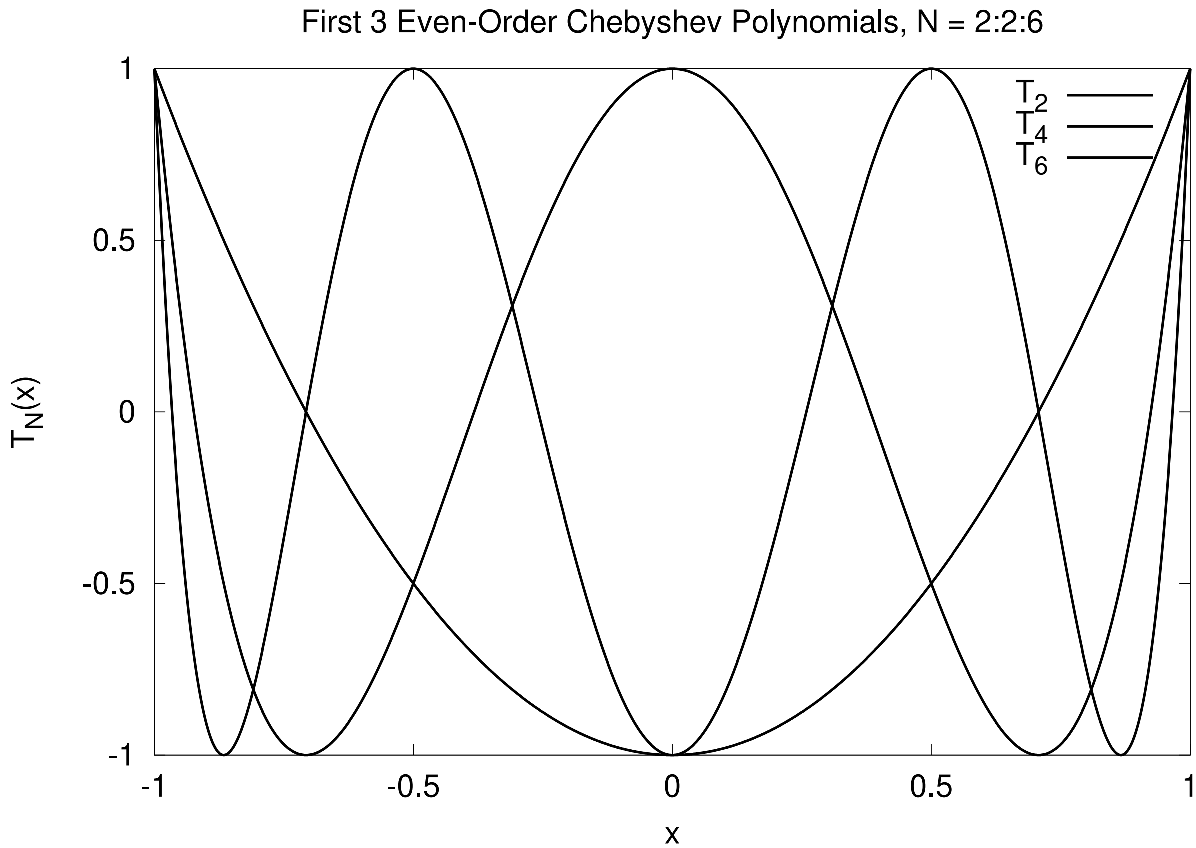


More Kaiser-Window Examples

$$\alpha = [1, 2, 3] \quad (\beta = [\pi, 2\pi, 3\pi])$$



Chebyshev Polynomials



The nth *Chebyshev polynomial* may be defined by

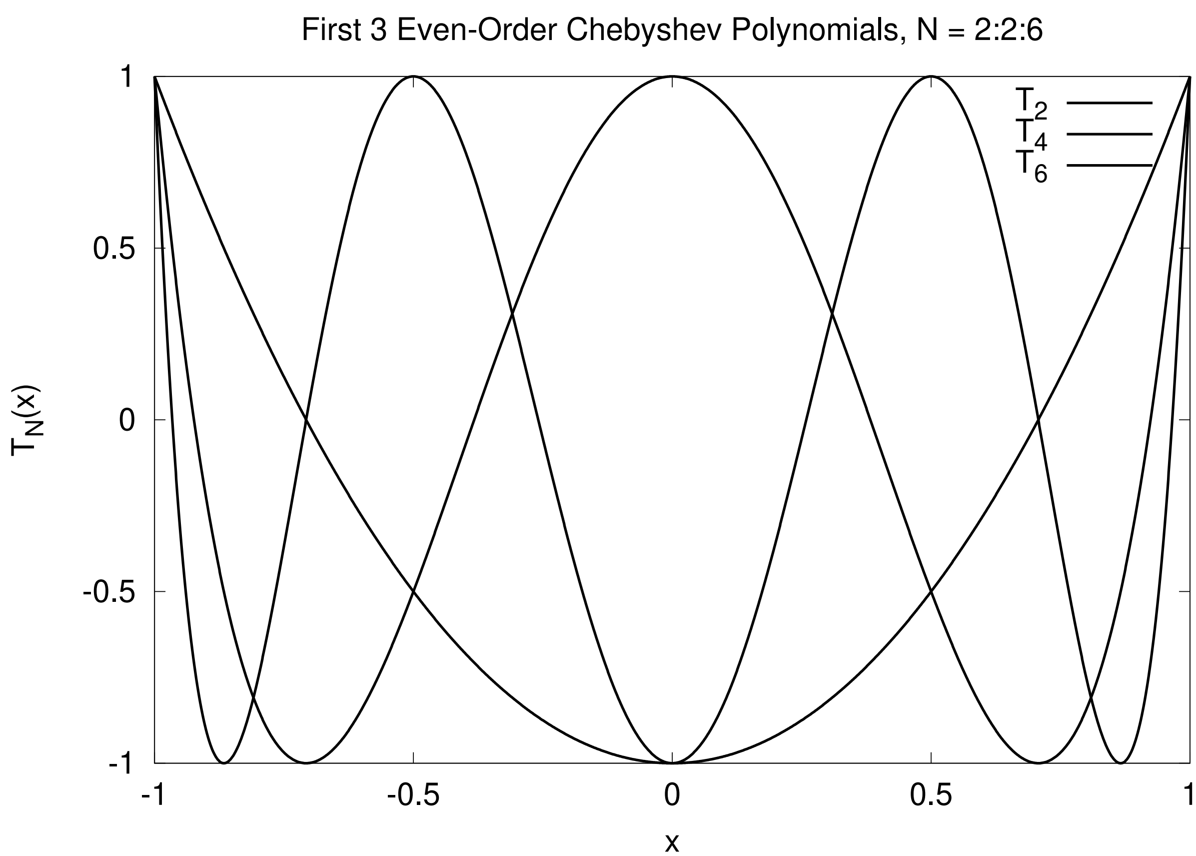
$$T_n(x) = \begin{cases} \cos[n\cos^{-1}(x)], & |x| \le 1 \\ \cosh[n\cosh^{-1}(x)], & |x| > 1 \end{cases}.$$

Clearly, $T_0(x) = 1$ and $T_1(x) = x$.
Using the double-angle trig formula $\cos(2\theta) = 2\cos^2(\theta) - 1$, it can be verified that

$$\boxed{T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x)} \quad (n \ge 2)$$

Chebyshev Polynomial Properties

- $T_n(x)$ is an nth-order polynomial in x
- $T_n(x)$ is an even function when n is an even integer, and odd when n is odd
- $T_n(x)$ has n zeros in the open interval $(-1, 1)$, and $n + 1$ extrema in the closed interval $[-1, 1]$
- $T_n(x) > 1$ for $x > 1$



$$T_n(x) = \cos[n \cos^{-1}(x)], \quad |x| \le 1$$

Dolph-Chebyshev Window

Minimize the *Chebyshev norm* of the side lobes, *e.g.*,

$$\begin{aligned} & \text{min}_{w,\sum w=1} \| \,\text{sidelobes}(W)\, \|_\infty \\ \equiv\ & \text{min}_{w,\sum w=1} \{\text{max}_{\omega>\omega_c} |W(\omega)|\} \end{aligned}$$

Alternatively, *minimize main lobe width* subject to a sidelobe spec:

$$\min_{w,W(0)=1} (\omega_c)\Big|_{|W(\omega)\,|\le c_\alpha,\ \forall|\omega|\ge\omega_c}$$

Closed-Form Window Transform (Dolph):

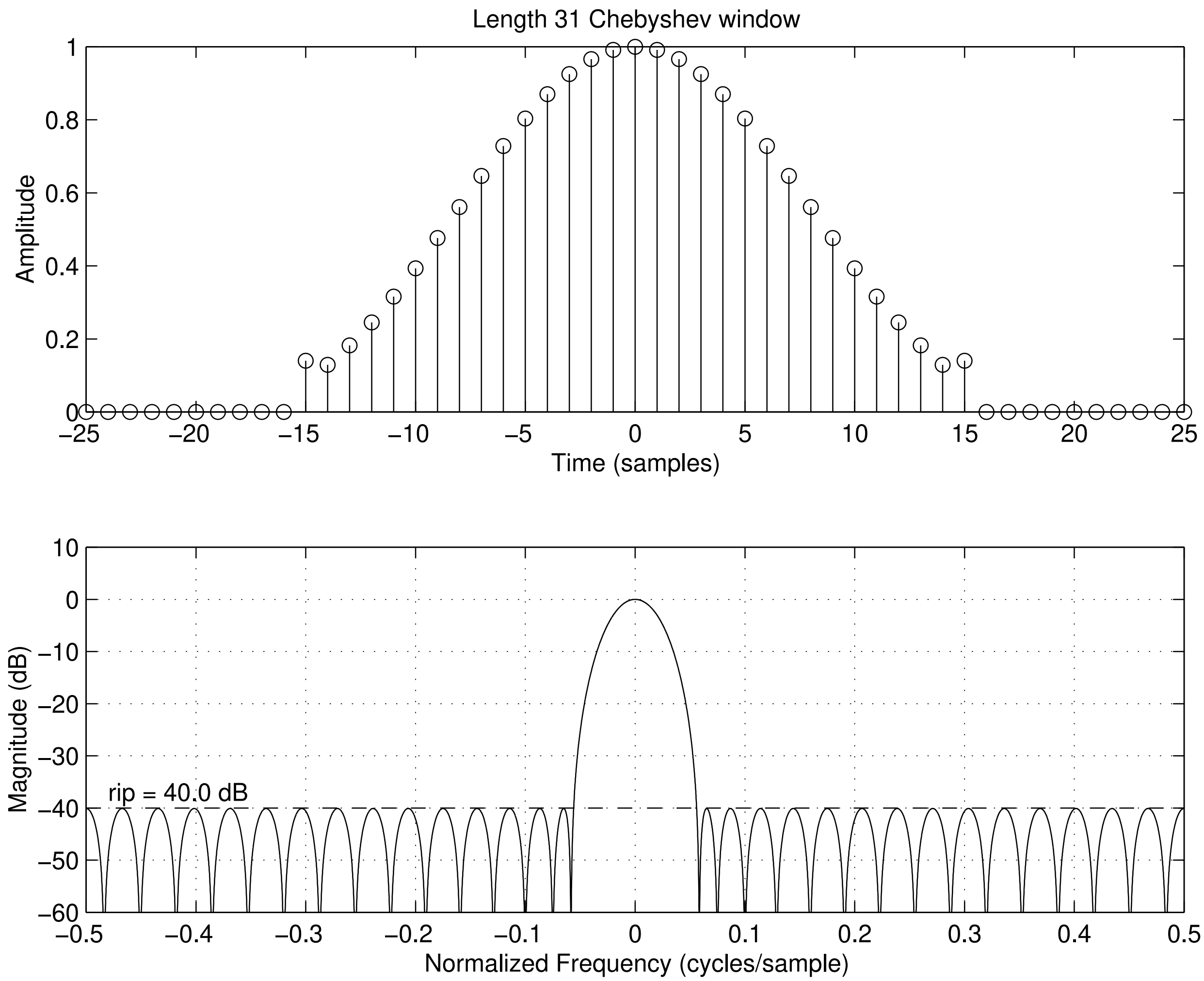
$$\begin{aligned} W(\omega_k) &= \frac{\cos\left\{M\cos^{-1}\left[\Gamma\cos\left(\frac{\pi k}{M}\right)\right]\right\}}{\cosh\left[M\cosh^{-1}(\Gamma)\right]}, \quad (|k|\le M-1) \\ \Gamma &= \cosh\left[\frac{1}{M}\cosh^{-1}(10^\alpha)\right] \ \ge\ 1, \qquad (\alpha\approx 2,3,4) \end{aligned}$$

- Window $w = \mathrm{IDFT}(W)$ [zero-centered case] or IDFT of $(-1)^k W(\omega_k)$ for causal case
- α controls sidelobe level ("stopband ripple"):

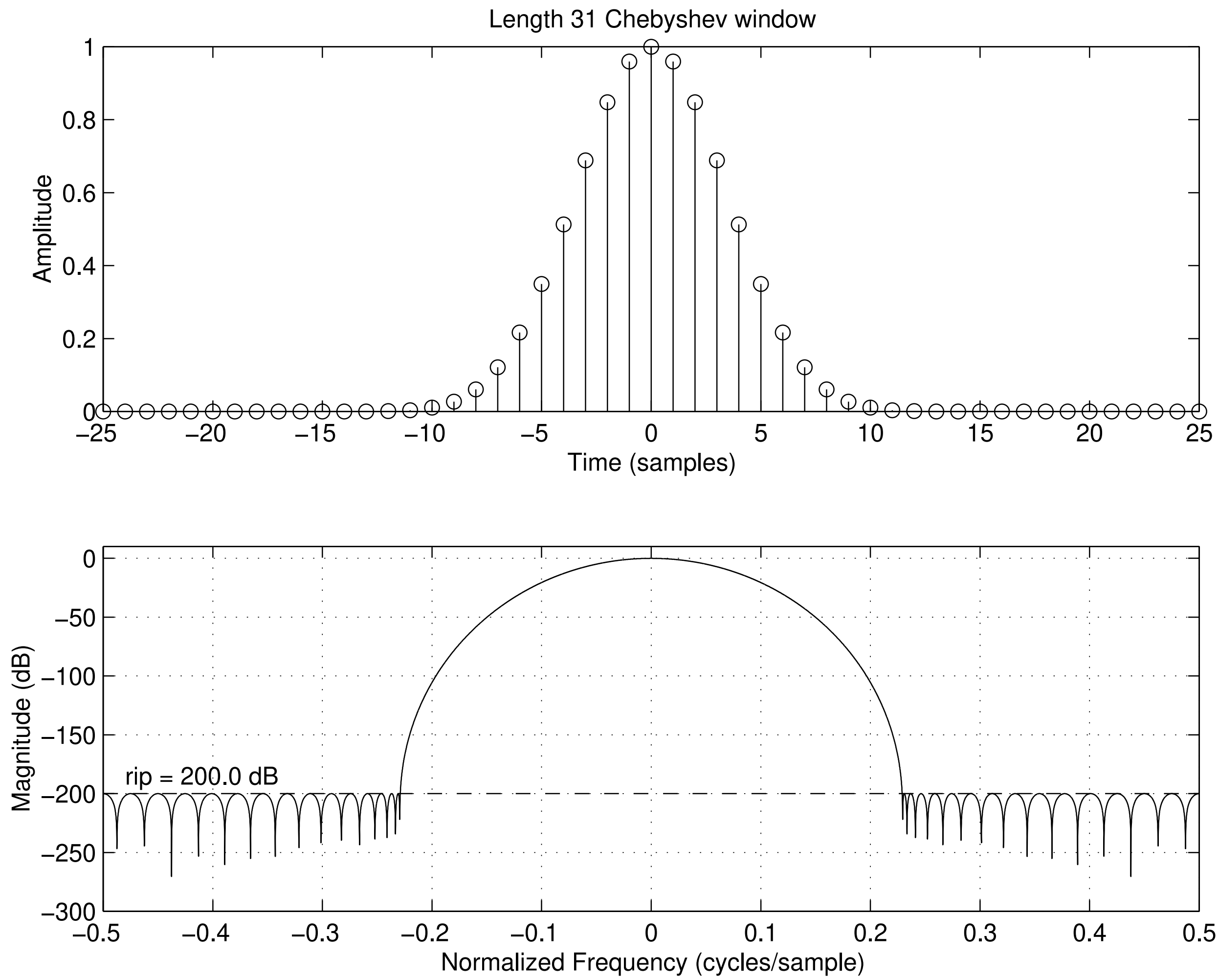
  Side-Lobe Level in dB $= -20\alpha$.

- smaller ripple $\Rightarrow$ larger ω_c
- see matlab function "chebwin(M,ripple)"

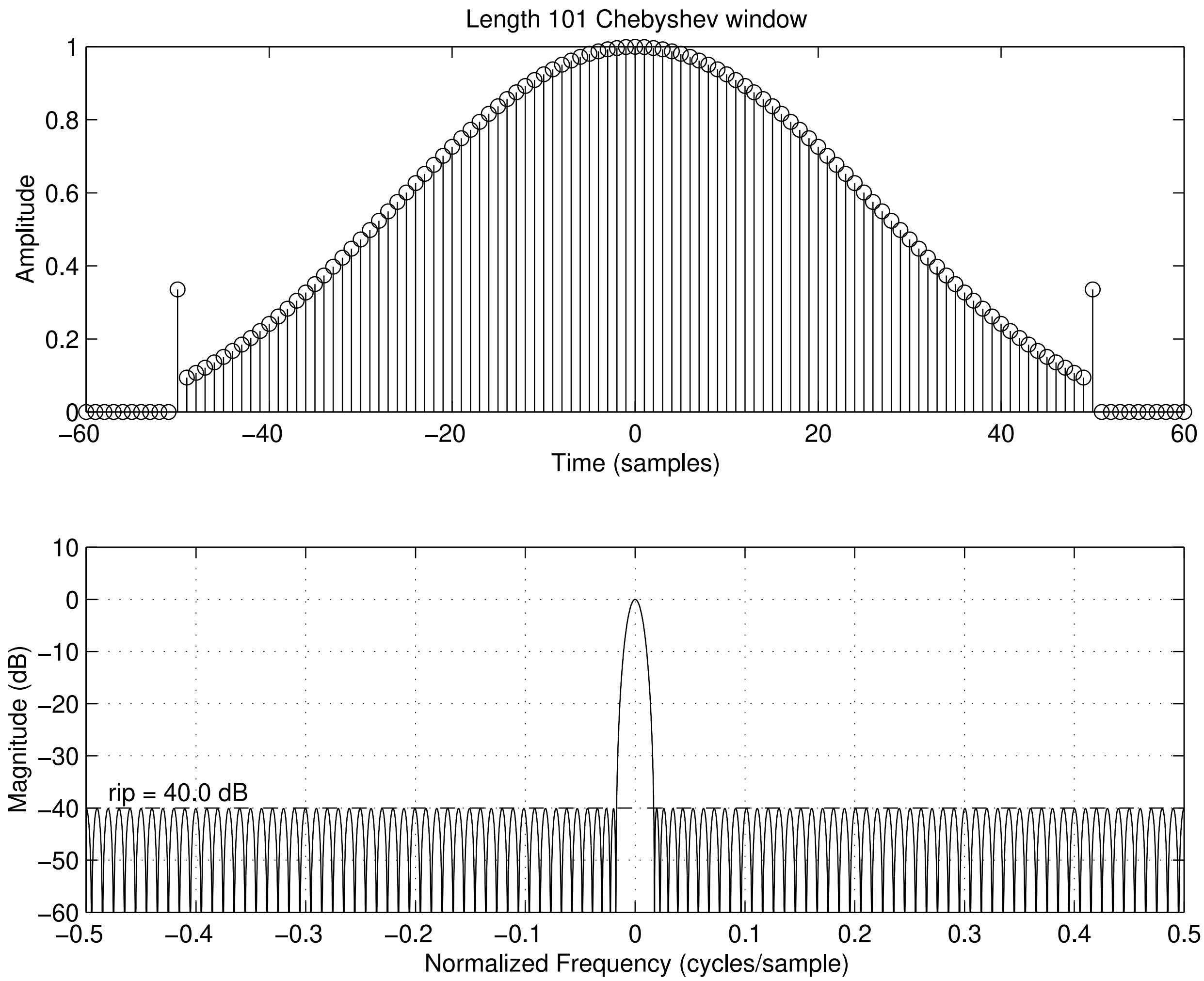
Dolph-Chebyshev Window, Length 31, Ripple -40 dB



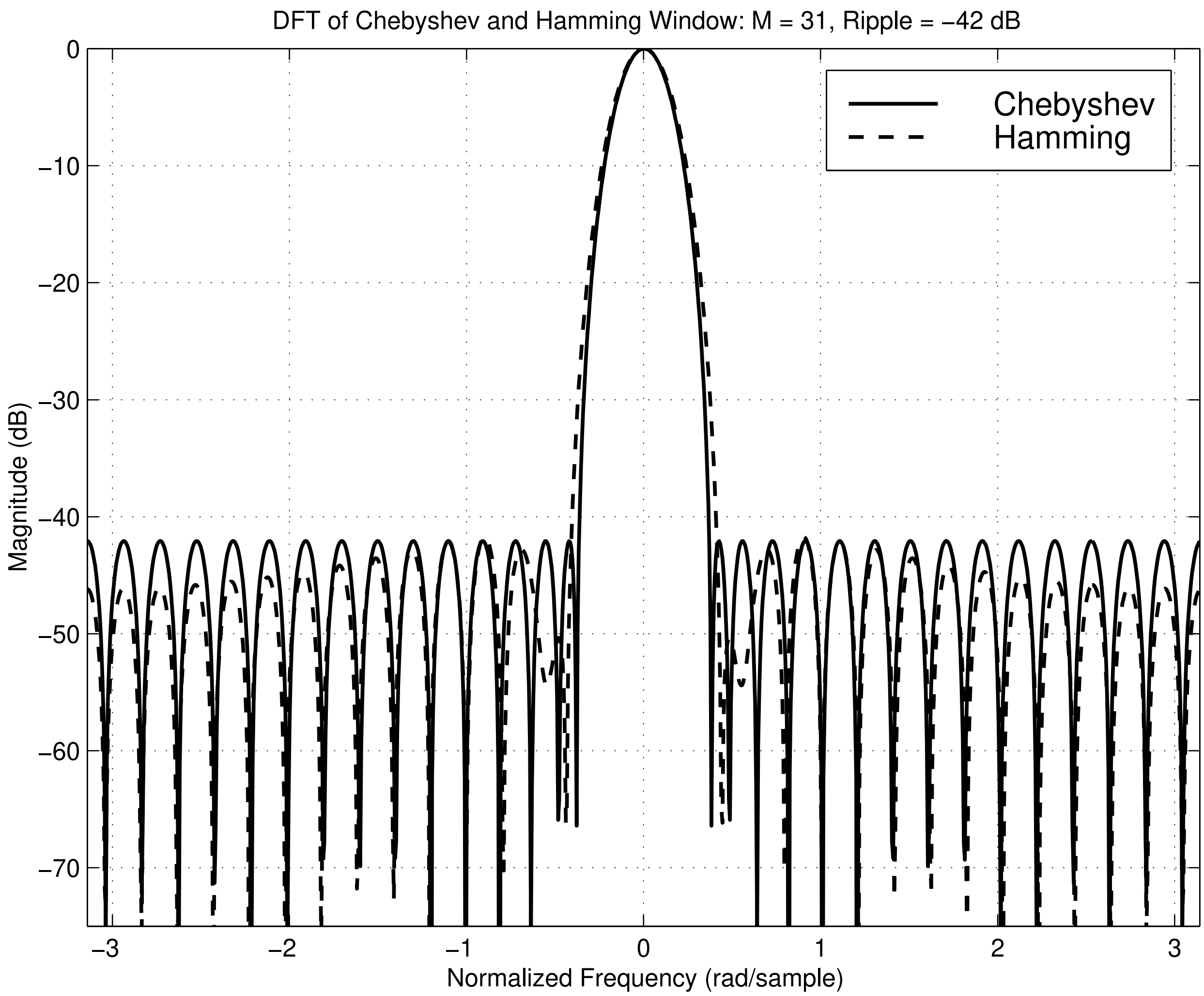
Dolph-Chebyshev Window, Length 31, Ripple -200 dB



Dolph-Chebyshev Window, Length 101, SLL -40 dB



Dolph-Chebyshev and Hamming Windows Compared



For the comparison, we set the ripple parameter for `chebwin` to 42 dB:

```
window = [ chebwin(31,42)' zeros(1,1024-31) ];
```

Gaussian

The Gaussian "bell curve" is the only smooth function that transforms to itself:

$$\frac{1}{\sigma\sqrt{2\pi}} e^{-t^2/2\sigma^2} \leftrightarrow e^{-\omega^2/2(1/\sigma)^2}$$

It also achieves the *minimum time-bandwidth product*

$$\sigma_t \sigma_\omega = \sigma \times (1/\sigma) = 1$$

when "width" of a function is defined as the square root of its second central moment. For even functions $w(t)$,

$$\sigma_t \stackrel{\Delta}{=} \sqrt{\int_{-\infty}^{\infty} t^2 w(t) dt}.$$

- Since the true Gaussian function has infinite duration, in practice we must *window it* with some finite window.
- Philippe Depalle suggests using a *triangular window* raised to some power α for this purpose.
  - This choice *preserves the absence of sidelobes* for sufficiently large α.
  - It also preserves *non-negativity* of the transform

The Gaussian Window in Spectral Modeling

Special Property: On a dB scale, the Gaussian is *quadratic* $\Rightarrow$ *parabolic interpolation of a sampled Gaussian transform is* $exact$.

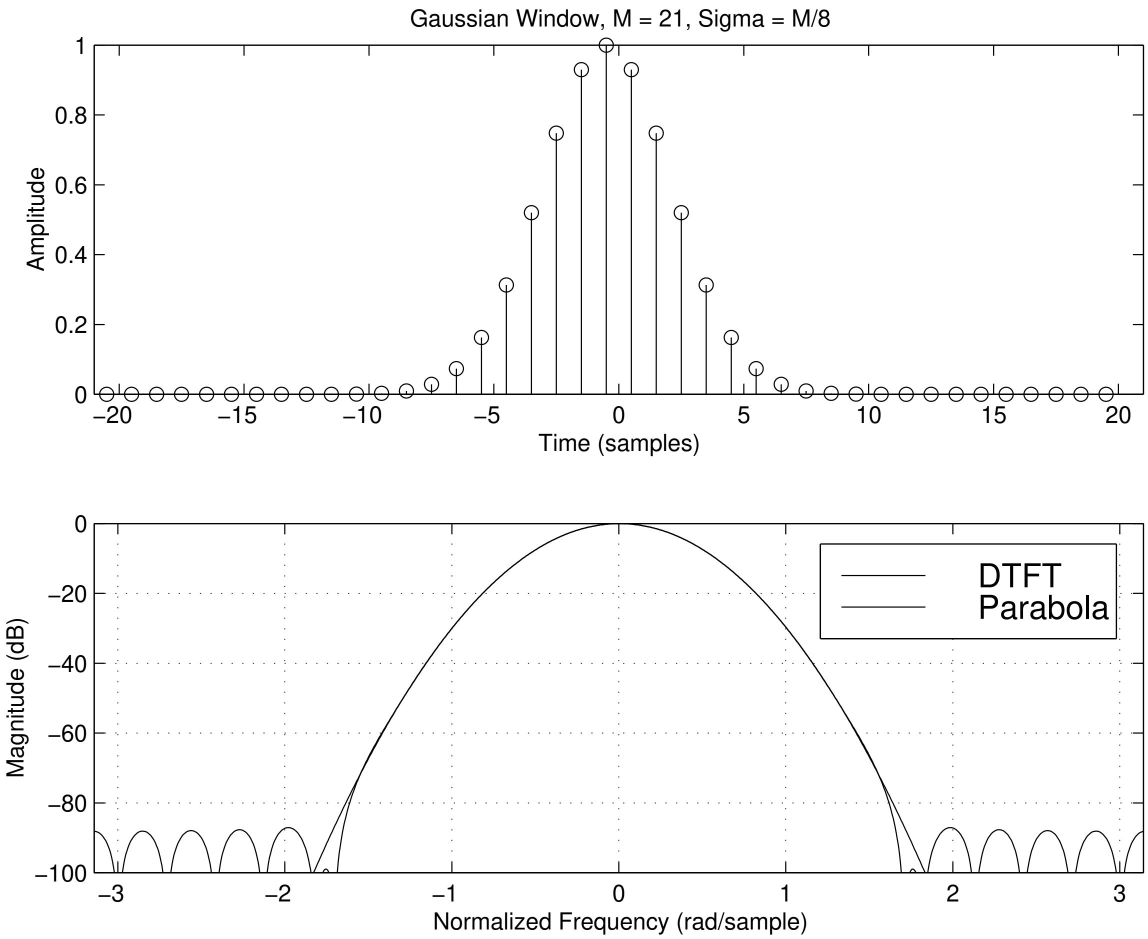
Conjecture: Quadratic interpolation of spectral peaks is generally more accurate on a *log-magnitude scale* (e.g., dB) than on a linear magnitude scale. This has been verified in a number of cases, and no counter-examples are yet known. **Exercise:** Prove this is true for the rectangular window.

Matlab for the Gaussian Window

```
function [w] = gausswin(M,sigma)

n=(-(M-1)/2:(M-1)/2)';
w = exp(-n.*n./(2*sigma.*sigma));
```

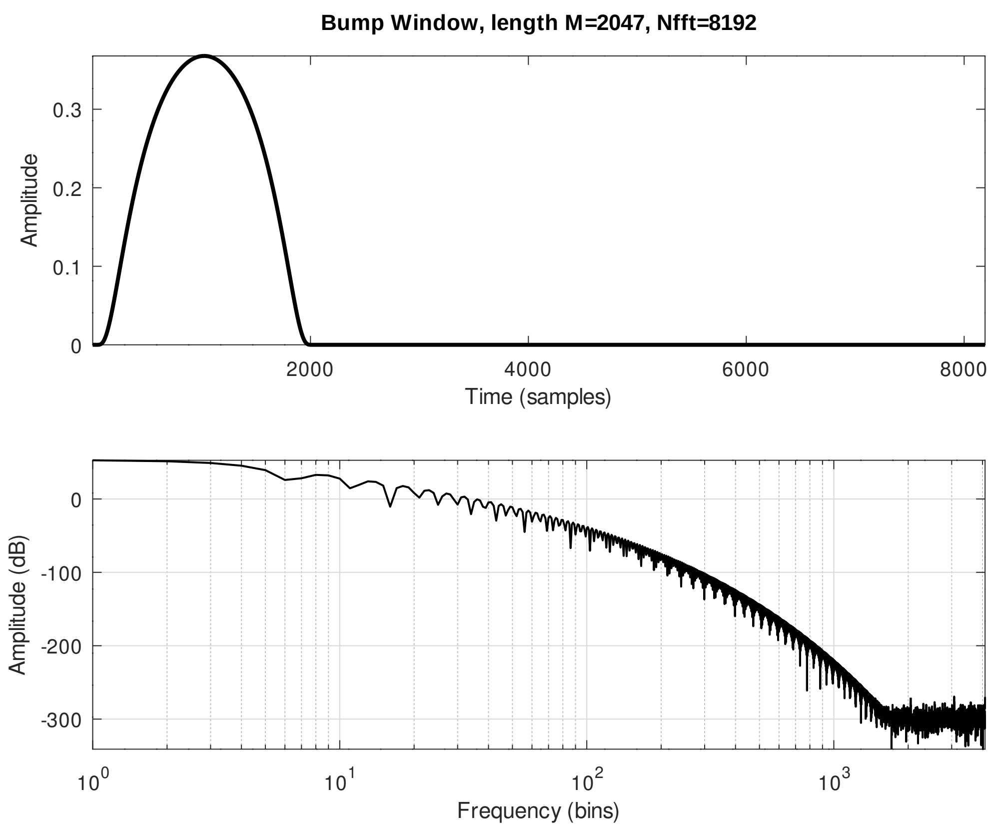
Gaussian Window and Transform



Bump Window

$$w(t) = \begin{cases} e^{-\frac{1}{1-t^2}} & |t| < 1, \\ 0 & \text{otherwise} \end{cases}$$

- Infinitely differentiable everywhere (then sample)
- Roll-off rate unbounded (faster than any polynomial)
- Aliasing progressively slows the decay



More Windows

There is a nice
collection of window definitions and citations on Wikipedia:
`https://en.wikipedia.org/wiki/Window_function`

Optimal Windows

Generally we desire

$$W(\omega) \approx \delta(\omega)$$

- Best results are obtained by formulating this as an *FIR filter design problem*.
- In general, both time-domain and frequency-domain specifications are needed.
- Equivalently, both *magnitude* and *phase* specifications are necessary in the frequency domain.

Optimal Windows for Audio Coding

Recently, numerically optimized windows have been developed by Dolby which achieve the following objectives:

- Narrow the window in time
- Smooth the onset and decay in time
- Reduce sidelobes below the *worst-case masking threshold*

Conclusion

- There is rarely a closed form expression for an optimal window in practice.
- The hardest task is formulating the *ideal error criterion*.
- Given an error criterion, it is usually straightforward to minimize it numerically with respect to the window samples w.