LECTURE 12: FOURIER METHODS

- We discussed different bases for regression in lecture 13: polynomial, rational, spline/gaussian…
- One of the most important basis expansions is Fourier basis: Fourier (or spectral) transform
- Several reasons for its importance: the basis is complete (any function can be Fourier expanded)
- ability to compute convolutions and spectral densities (power spectrum)
- Ability to convert dome partial differential equations (PDE) into ordinary differential equations (ODE)
- Main reason for these advantages: fast Fourier transform (FFT)

Definition and Properties of Fourier Transforms

- Complete basis

$$\omega \equiv 2\pi f \qquad H(\omega) \equiv [H(f)]_{f=\omega/2\pi}$$

$$H(f) = \int_{-\infty}^{\infty} h(t) e^{2\pi i f t} dt \qquad H(\omega) = \int_{-\infty}^{\infty} h(t) e^{i\omega t} dt$$

$$h(t) = \int_{-\infty}^{\infty} H(f) e^{-2\pi i f t} df \qquad h(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} H(\omega) e^{-i\omega t} d\omega$$

If...	then...
$h(t)$ is real	$H(-f) = [H(f)]^*$
$h(t)$ is imaginary	$H(-f) = -[H(f)]^*$
$h(t)$ is even	$H(-f) = H(f)$ [i.e., $H(f)$ is even]
$h(t)$ is odd	$H(-f) = -H(f)$ [i.e., $H(f)$ is odd]
$h(t)$ is real and even	$H(f)$ is real and even
$h(t)$ is real and odd	$H(f)$ is imaginary and odd
$h(t)$ is imaginary and even	$H(f)$ is imaginary and even
$h(t)$ is imaginary and odd	$H(f)$ is real and odd

Properties of Fourier Transforms

$h(at) \Longleftrightarrow \frac{1}{\lvert a \rvert} H(\frac{f}{a})$		time scaling
$\frac{1}{\lvert b \rvert} h(\frac{t}{b}) \Longleftrightarrow H(bf)$		frequency scaling
$h(t-t_0) \Longleftrightarrow H(f)\, e^{2\pi i f t_0}$		time shifting
$h(t)\, e^{-2\pi i f_0 t} \Longleftrightarrow H(f-f_0)$		frequency shifting

- Convolution theorem: $g * h \equiv \int_{-\infty}^{\infty} g(\tau) h(t-\tau)\, d\tau$

$$\int g(\tau)h(t-\tau)d\tau$$

$$= \int \frac{dw}{2\pi} e^{-iw\tau} g(w) \int \frac{dw'}{2\pi} e^{-iw'(t-\tau)} h(w') d\tau$$

$$\int e^{-i(w-w')\tau} d\tau = 2\pi \delta_D(w-w')$$

$$= \int \frac{dw}{2\pi} e^{iwt} g(w) h(w)$$

$$\int \frac{dw'}{2\pi} 2\pi \delta_D(w-w') = 1$$

Correlation Function and Power Spectrum

$$\mathrm{Corr}(g,h)(t) = \int g(\tau+t)h(\tau)d\tau$$

$$= \int\int \frac{dw}{2\pi} e^{-iw(\tau+t)} g(w) \frac{dw'}{2\pi} e^{-iw'\tau} h(w') d\tau$$

$$\Big\downarrow \quad \int e^{-i\tau(w+w')} d\tau = 2\pi\delta_D(w+w')$$

$$= \int \frac{dw}{2\pi} g(w)h(-w)e^{-iwt} = \mathrm{FT}\Big(g(w)h^*(w)\Big)$$

$g = h, \quad \mathrm{Corr}(g,g)(t) = \mathrm{FT}(gg^*) = \mathrm{FT}(|g|^2)$ Wiener–Khinchin theorem

POWER SPECTRUM $\quad P(w) = |g(w)|^2$ Parseval's theorem

$$t = 0, \quad \int |g(w)|^2 \frac{dw}{2\pi} = \int g(\tau)^2 d\tau$$

Power Spectrum in higher dimensions

$$g(\vec{x}) \quad \mathrm{FT}: g(\vec{k}) = \int d^\mu \vec{x} \cdot g(\vec{x}) e^{i\vec{k}\cdot\vec{x}}$$

$$g(\vec{x}) = \int \frac{d^\mu \vec{k}}{(2\pi)^N} e^{-i\vec{k}\cdot\vec{x}} g(\vec{k})$$

$$\mathrm{Corr}\left[g(\vec{x}_1) g(\vec{x}_2 = \vec{x}_1 + \vec{x})\right] = \int \frac{d^\mu \vec{k}}{(2\pi)^N} P(\vec{k}) e^{-i\vec{k}\cdot\vec{x}}$$

$$P(\vec{k}) = g(\vec{k}) g^*(\vec{k})$$

If isotropic, only depends on *k* length

$$P(\vec{k}) = P(k)$$

$$\mathrm{Corr}(\vec{x}) = \int \frac{d^\mu \vec{k}}{(2\pi)^N} P(\vec{k}) e^{-i\vec{k}\cdot\vec{x}}$$

Discrete Sampling: Sampling Theorem

- We sample interval in points of length Δ: $h_n = h(n\Delta)$
- Nyquist frequency: $f_c = 1/2\Delta$
- Sampling theorem: if the function h(t) does not have frequencies above f_c ($h(f) = 0$ for $f > f_c$) it is bandwidth limited. Then $h(t)$ is completely determined by h_n:

$$h(t) = \Delta \sum_{n=-\infty}^{+\infty} h_n \frac{\sin[2\pi f_c(t - n\Delta)]}{\pi(t - n\Delta)}$$

- This says that the information content is limited. If we know the maximum bandwidth frequency then we know how to sample the function using $f_c = 1/2D$ to get the full information content

If we are not bandwidth limited, we get Aliasing

- All the power outside $-f_c < f < f_c$ is moved into this bandwidth by the process of discrete sampling
- Take $\exp(2\pi i f_1 t)$ and $\exp(2\pi i f_2 t)$: if $f_1 - f_2 = m/\Delta$ then the values at $h_n = h(n\Delta)$ differ by $\exp(2\pi i n m) = 1$: we cannot distinguish between these frequencies using discrete sampling
- In practice we thus want to look at the power spectrum as a function of f: if it goes to 0 as f approaches f_c then aliasing is small, otherwise we need to decrease the sampling rate Δ, ie increase $f_c = 1/2\Delta$

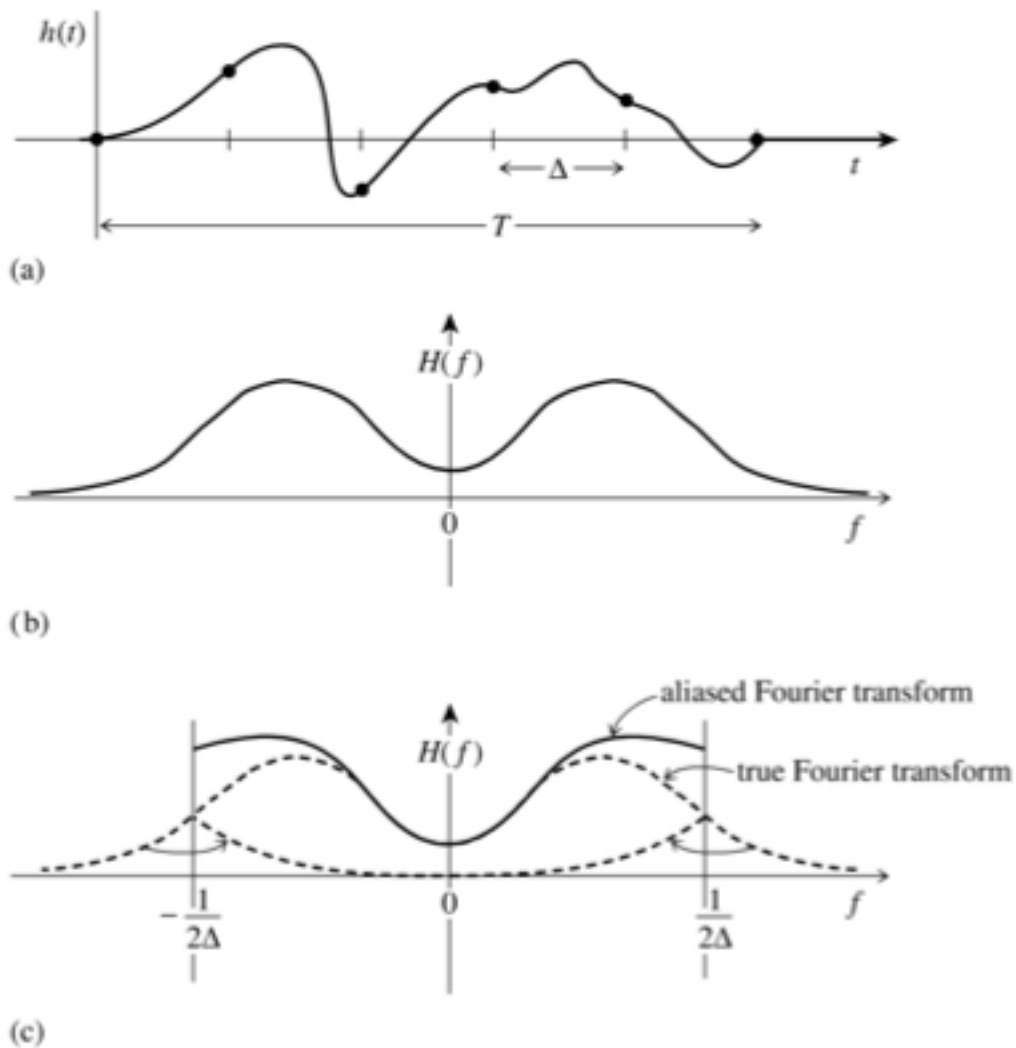


Figure 12.1.1. The continuous function shown in (a) is nonzero only for a finite interval of time T. It follows that its Fourier transform, whose modulus is shown schematically in (b), is not bandwidth limited but has finite amplitude for all frequencies. If the original function is sampled with a sampling interval Δ, as in (a), then the Fourier transform (c) is defined only between plus and minus the Nyquist critical frequency. Power outside that range is folded over or "aliased" into the range. The effect can be eliminated only by low-pass filtering the original function *before sampling*.

Discrete Fourier Transform (DFT)

- We measure a function on an interval $N\Delta$. Maybe the function is 0 outside the interval, otherwise we assume periodicity (periodic boundary conditions), because sin and cos are periodic

$$h_k \equiv h(t_k), \qquad t_k \equiv k\Delta, \qquad k = 0, 1, 2, \ldots, N-1$$

- Frequency range from $-f_c$ to f_c $\quad f_n \equiv \frac{n}{N\Delta}, \qquad n = -\frac{N}{2}, \ldots, \frac{N}{2}$

- DFT: $\quad H(f_n) = \int_{-\infty}^{\infty} h(t) e^{2\pi i f_n t} dt \approx \sum_{k=0}^{N-1} h_k \, e^{2\pi i f_n t_k} \Delta = \Delta \sum_{k=0}^{N-1} h_k \, e^{2\pi i k n/N}$

- $H(-f_c) = H(f_c)$ $\quad \underline{H(f_n) \approx \Delta H_n} \qquad H_n \equiv \sum_{k=0}^{N-1} h_k \, e^{2\pi i k n/N}$

- Inverse DFT $\quad h_k = \frac{1}{N} \sum_{n=0}^{N-1} H_n \, e^{-2\pi i k n/N}$

- Parseval's theorem $\quad \sum_{k=0}^{N-1} |h_k|^2 = \frac{1}{N} \sum_{n=0}^{N-1} |H_n|^2$

Fast Fourier Transform (FFT)

- How expensive is to do FT? Naively it appears to be a matrix multiplication, hence $O(N^2)$

$$W \equiv e^{2\pi i/N} \qquad H_n = \sum_{k=0}^{N-1} W^{nk} h_k$$

- FFT: $O(N\log_2 N)$
- The difference is enormous: these days we can have $N > 10^{10}$. FFT is one of the most important algorithms of numerical analysis
- FFT existence became widely known in 1960s (Cooley & Tukey), but known (and forgotten) since Gauss (1805)

How FFT works?

- Assume for now $N = 2^M$: one can show that FT of length N can be rewritten as the sum of 2 FTs of length N/2 (Danielson & Lanczos 1942): even (e) and odd (o)

$$\begin{aligned} F_k &= \sum_{j=0}^{N-1} e^{2\pi ijk/N} f_j \\ &= \sum_{j=0}^{N/2-1} e^{2\pi ik(2j)/N} f_{2j} + \sum_{j=0}^{N/2-1} e^{2\pi ik(2j+1)/N} f_{2j+1} \\ &= \sum_{j=0}^{N/2-1} e^{2\pi ikj/(N/2)} f_{2j} + W^k \sum_{j=0}^{N/2-1} e^{2\pi ikj/(N/2)} f_{2j+1} \\ &= F_k^e + W^k F_k^o \end{aligned}$$

- This can be recursively repeated, until we reach the length of 1, at which point we have for some n $F_k^{eoeeoeo\cdots oee} = f_n$

Bit Reversal

- To determine n we take the sequence eoeeoo…oee, reverse the pattern of e and o, assign e=0 and o=1, and we get n in binary form.

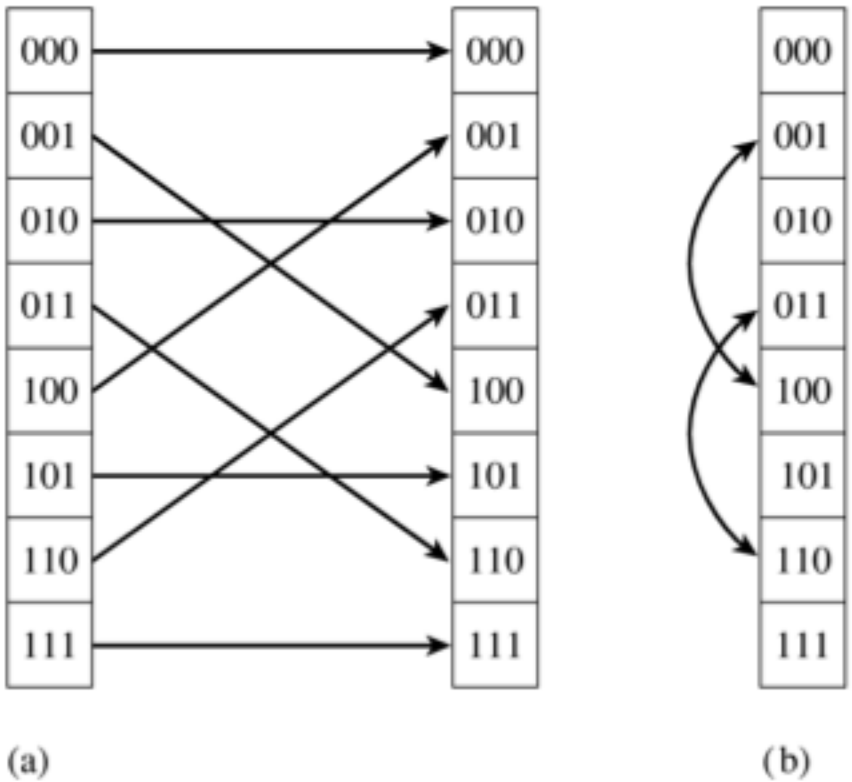
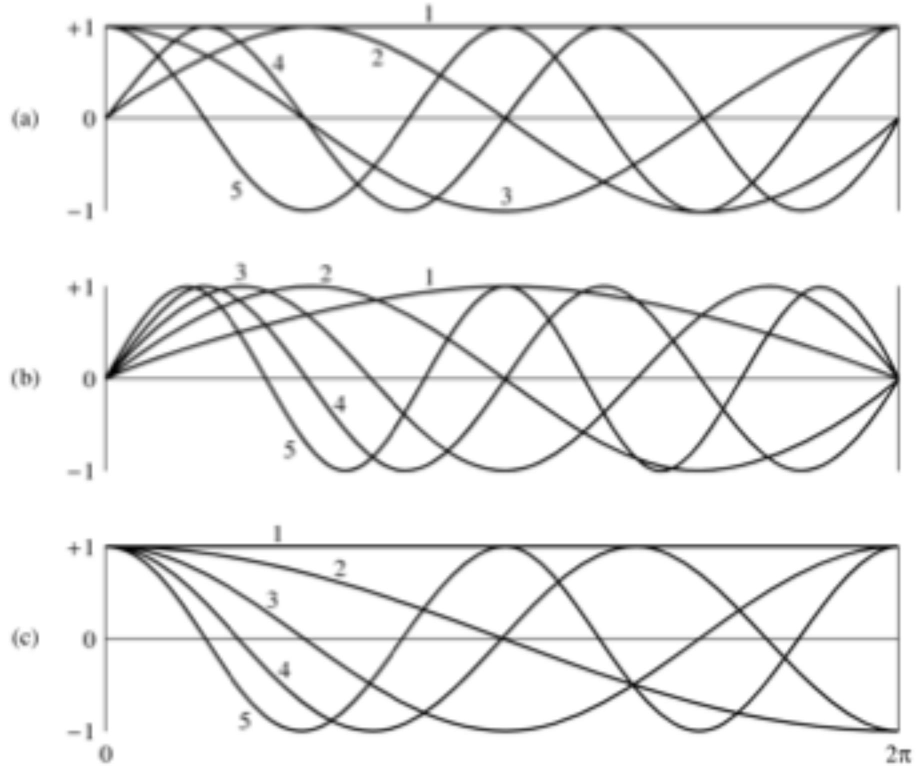


Figure 12.2.1. Reordering an array (here of length 8) by bit reversal, (a) between two arrays, versus (b) in place. Bit-reversal reordering is a necessary part of the fast Fourier transform (FFT) algorithm.

Extensions

- So far this was FFT of a complex function. FFT of real function can be done as FFT of complex of length *N/2*.
- FFT of sine and cosine: it is common to use FFT to solve differential equations. The classic problem in physics are oscillations on a string. If the string ends are fixed the values are 0 there, hence we use sine. If they are open we often use derivative = 0, for which we use cosine expansion.



FFT in higher dimensions

- One can write it as a sequence of 1d FFTs

$$H(n_1, n_2) \equiv \sum_{k_2=0}^{N_2-1} \sum_{k_1=0}^{N_1-1} \exp(2\pi i k_2 n_2 / N_2) \exp(2\pi i k_1 n_1 / N_1) \, h(k_1, k_2)$$

$$\begin{aligned} H(n_1, n_2) &= \text{FFT-on-index-1} \left(\text{FFT-on-index-2} \left[h(k_1, k_2) \right] \right) \\ &= \text{FFT-on-index-2} \left(\text{FFT-on-index-1} \left[h(k_1, k_2) \right] \right) \end{aligned}$$

- 2d FFT of real function used in image processing: high pass, low pass filters, convolutions, deconvolutions…
- 3d FFT: solving Poisson’s equation…

Literature

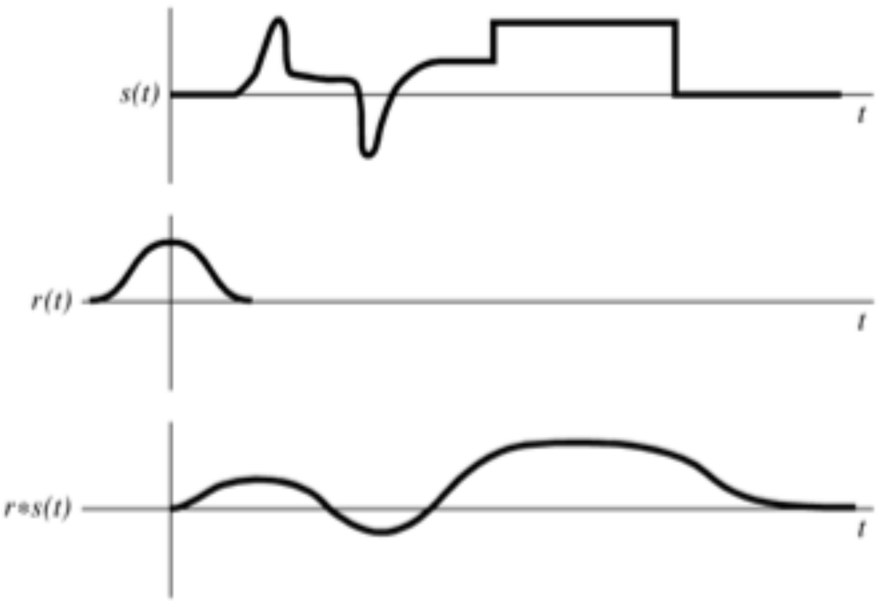
Figure 12.6.2. Fourier processing of an image. Upper left: Original image. Upper right: Blurred by low-pass filtering. Lower left: Sharpened by enhancing high frequency components. Lower right: Magnitude of the derivative operator as computed in Fourier space.

Low pass filter: smoothly set high f/ω to 0
High frequency sharpening: increase high f/ω

Derivative operator in FT: multiply Fourier modes with $i\omega$

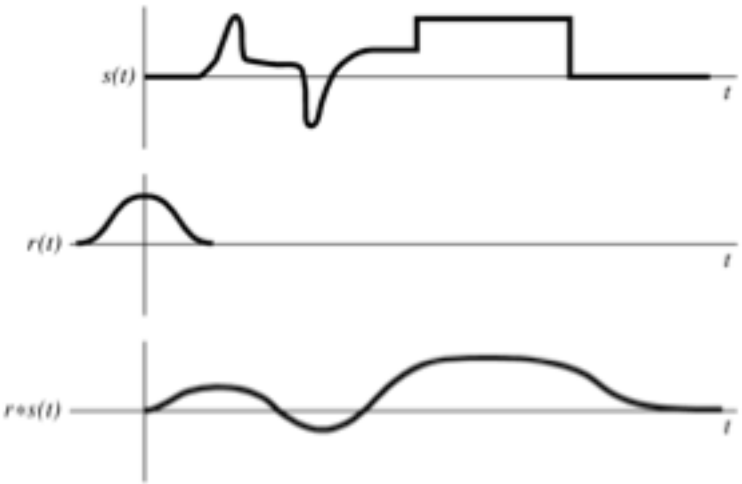
(De)convolutions with FFT

- We have data that have been convolved with some response. For example, we observe the sky through the telescope which does not resolve angles below λ/R, where R is its diameter. Here response is Airy function. To simulate the process we convolve data $s(t)$ with $r(t)$, by multiplying their FTs.
- If the data are perfect (no noise) we can deconvolve the process: we FT the data $r*s(t)$ to get $r(f)s(f)$ and divide by the convolution term $r(f)$ to get $s(f)$, then inverse FT to get $s(t)$.

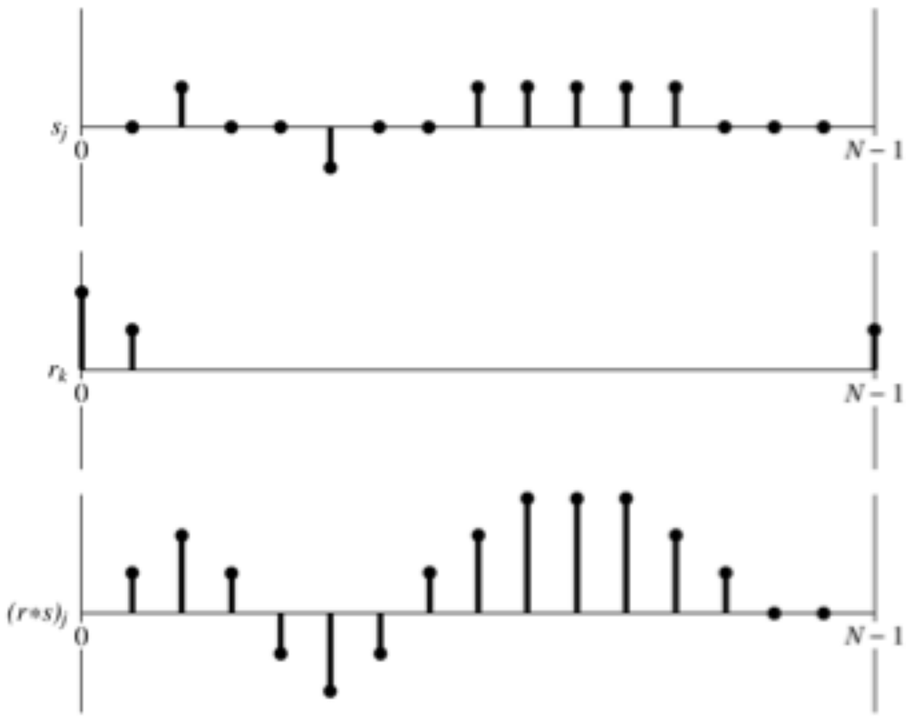


Discrete Version

- Discrete convolution assumes periodic signal on the interval

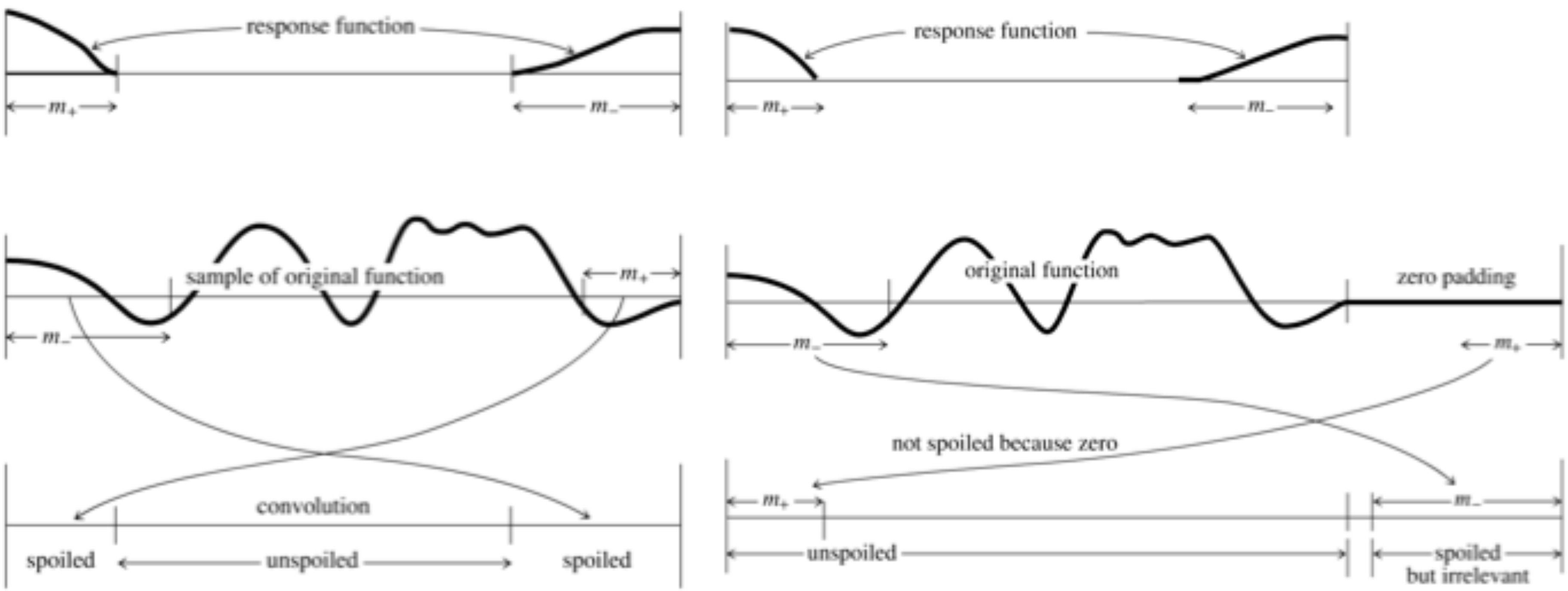


$$(r * s)_j \equiv \sum_{k=-M/2+1}^{M/2} s_{j-k}\, r_k$$



Zero Padding

- Because discrete FFTs assume periodicity we need to zero pad the data. If r has non-zero support over M and is symmetric then zero pad $M/2$ pixels beyond the data.



Power Spectrum and Correlation Function

- Cross correlation function Corr(g,h)=FT^{-1}[FT(g)FT(h)*]
- Cross-power spectrum FT(g)FT(h)*
- Auto-power spectrum (power spectrum density PSD, or periodogram) $P_g(f_k)$ = FT(g)FT(g)*
- $P_g(f)$ has dimensions of D so insert factors to get right units

$$C_k = \sum_{j=0}^{N-1} c_j \, e^{2\pi i j k/N} \qquad k = 0, \ldots, N-1$$

$$P(0) = P(f_0) = \frac{1}{N^2} |C_0|^2$$

$$P(f_k) = \frac{1}{N^2} \left[|C_k|^2 + |C_{N-k}|^2 \right] \qquad k = 1, 2, \ldots, \left(\frac{N}{2} - 1 \right)$$

$$P(f_c) = P(f_{N/2}) = \frac{1}{N^2} |C_{N/2}|^2$$

where f_k is defined only for the zero and positive frequencies

$$f_k \equiv \frac{k}{N\Delta} = 2 f_c \frac{k}{N} \qquad k = 0, 1, \ldots, \frac{N}{2}$$

Window Effects

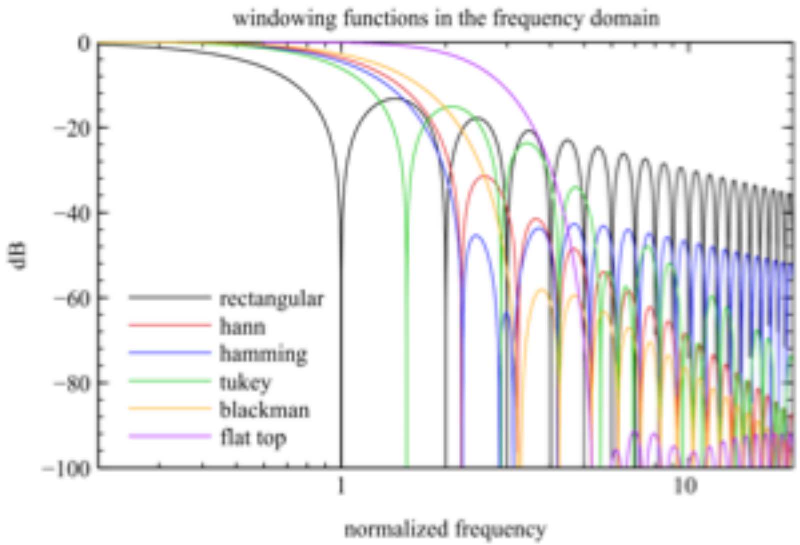
- Because we observe on a finite interval the data are multiplied with the top-hat window, and the power spectrum is convolved with FT of top hat window
- Top hat window has large side-lobes leaking from one *f* to another: mode mixing

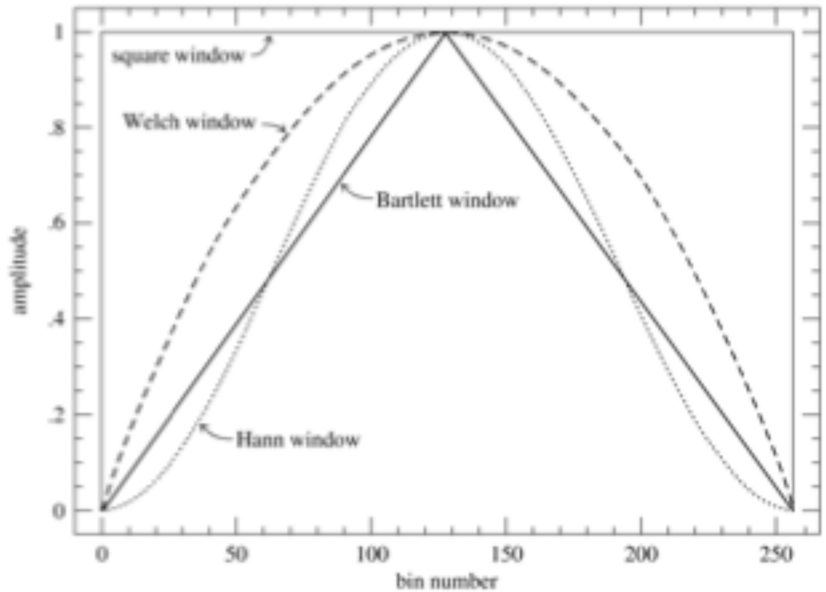
$$W(s) = \frac{1}{N^2}\left[\frac{\sin(\pi s)}{\sin(\pi s/N)}\right]^2$$

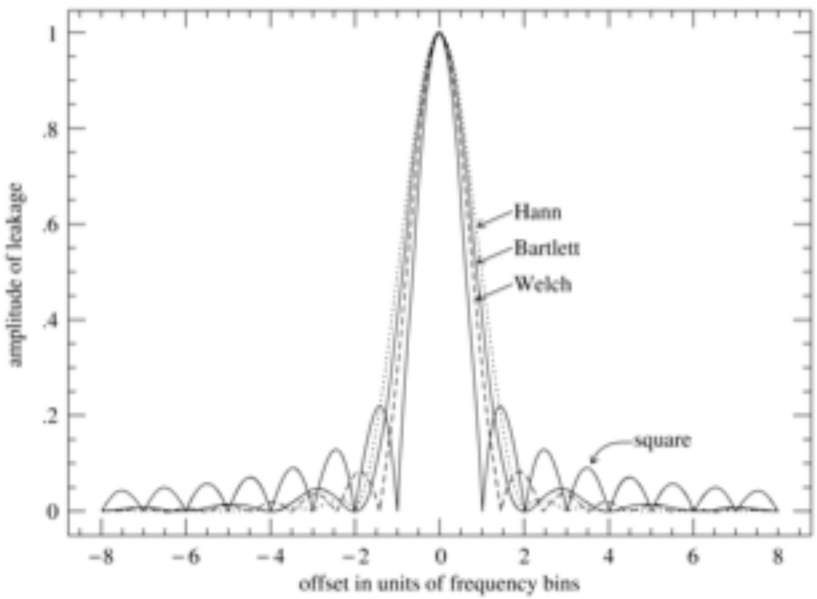
- Top hat has sharp edges: if we smooth it then the side-lobes are reduced and the leakage is reduced (see Project 2)

Apodization or Windowing

- We multiply the signal with a window function, reducing sidelobes
- Note that we are reducing signal at the edges: suboptimal

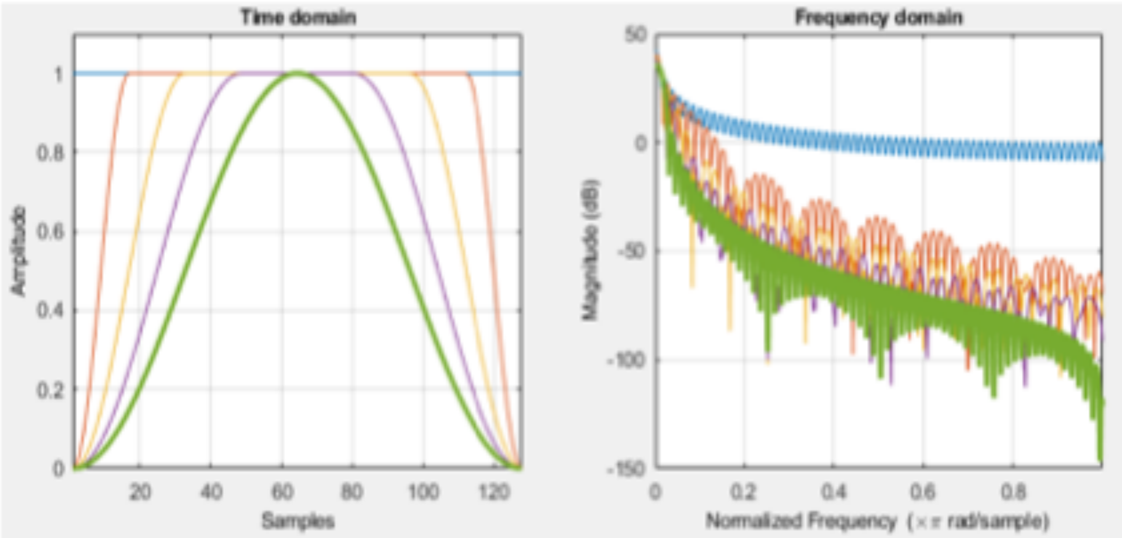




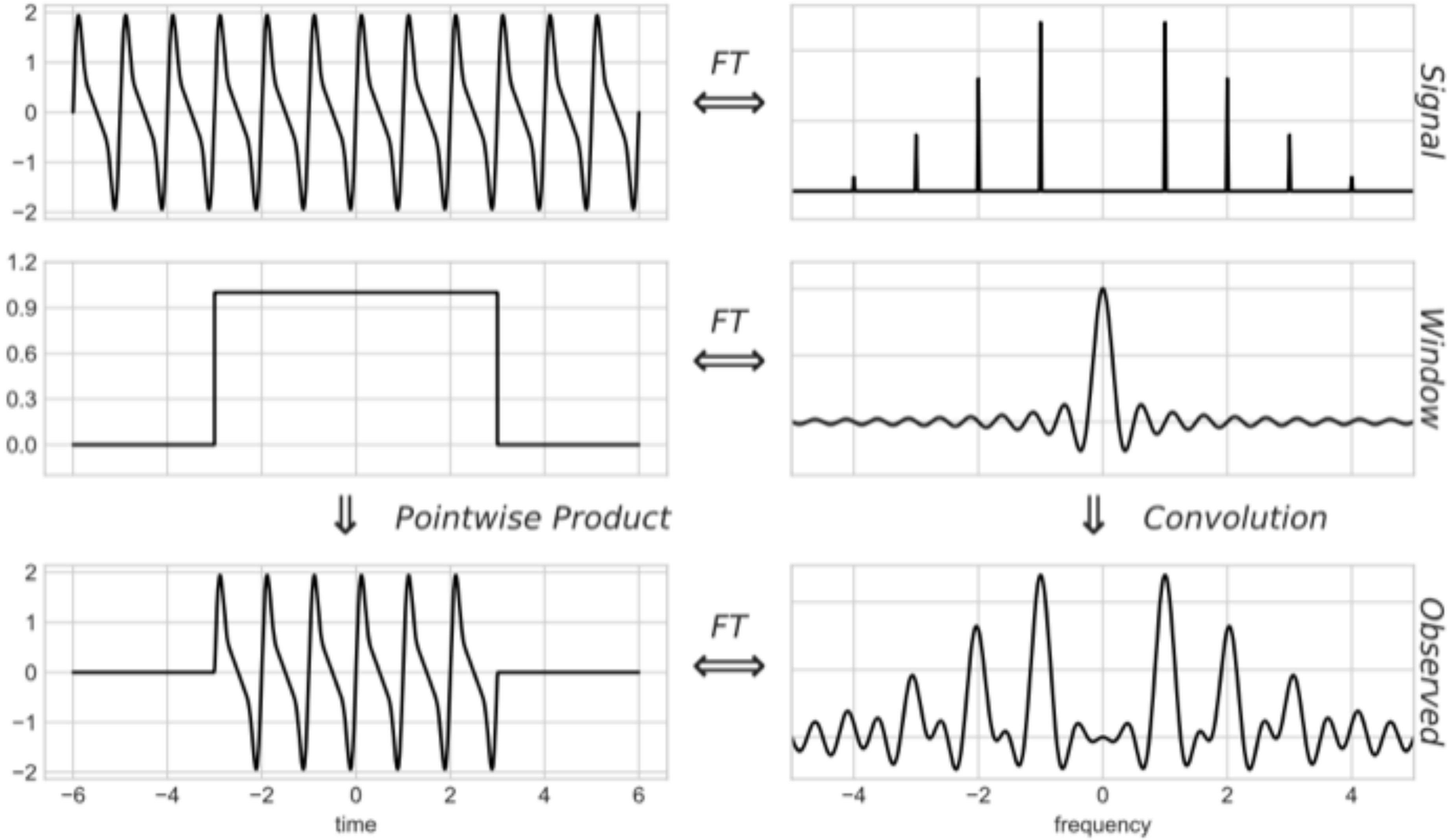


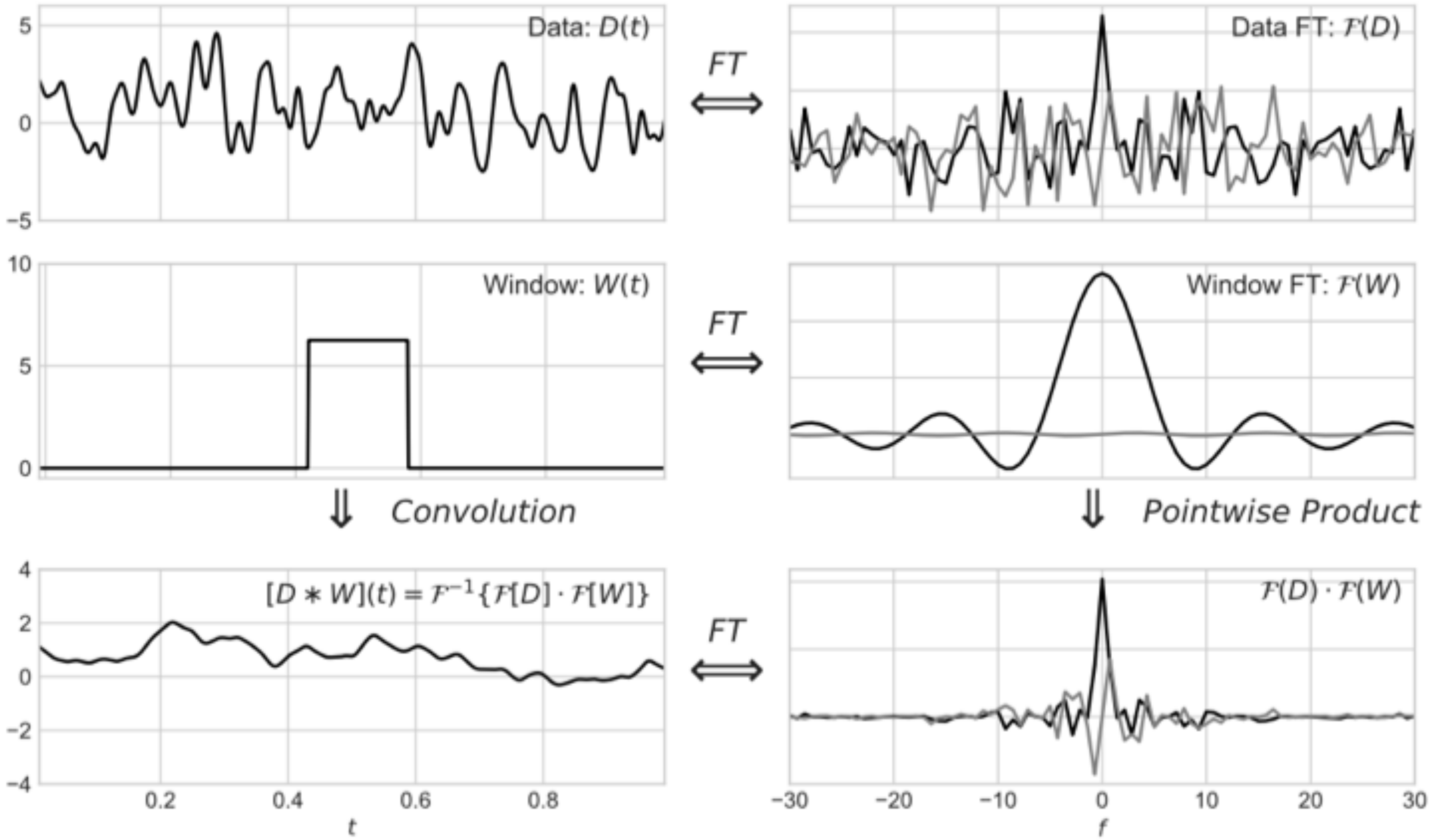
Edge Apodization

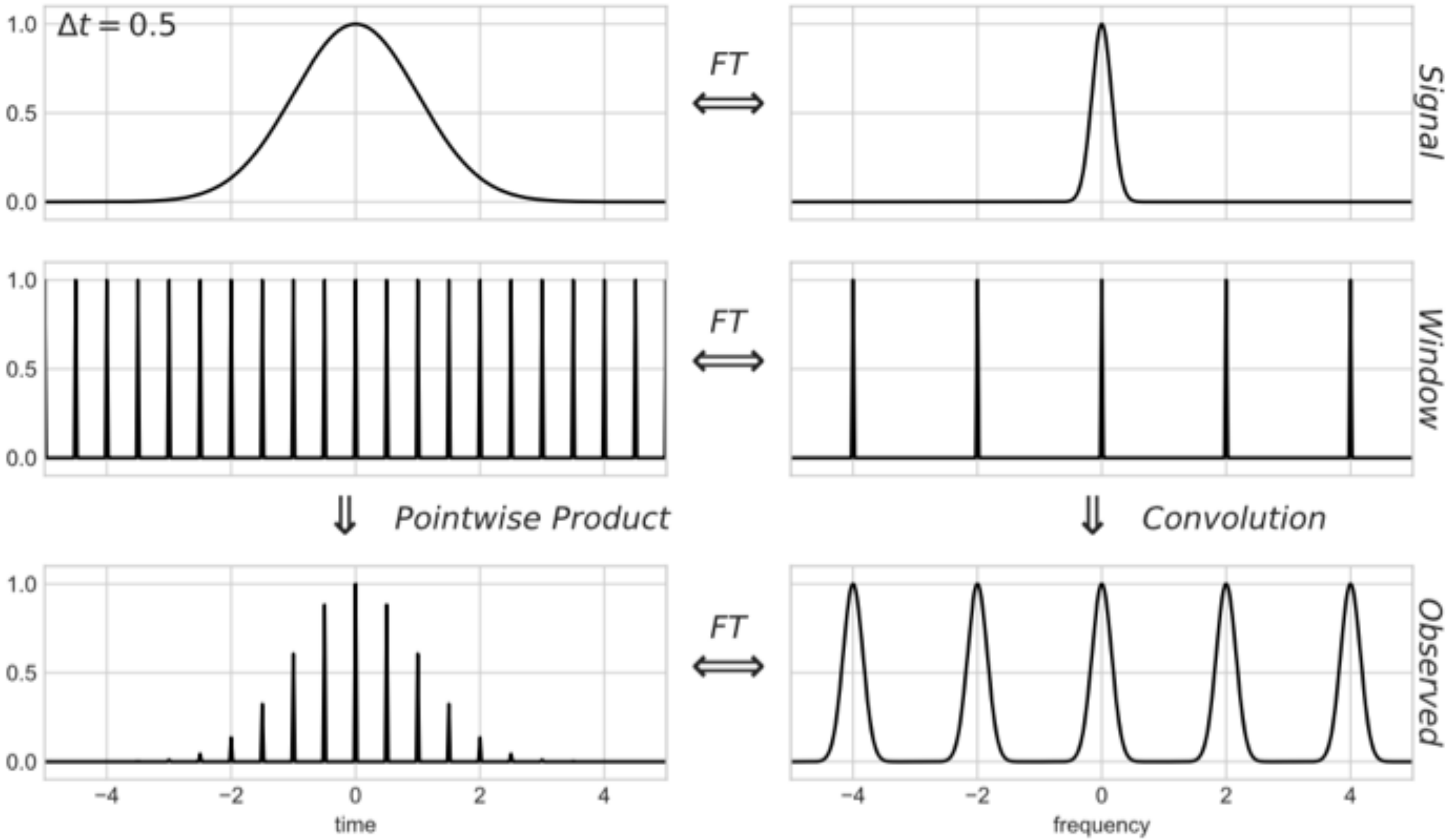
- Here one smoothly transitions from 1 to 0 on the edge only
- Leakage is higher, but much lower than square window
- Less signal to noise loss



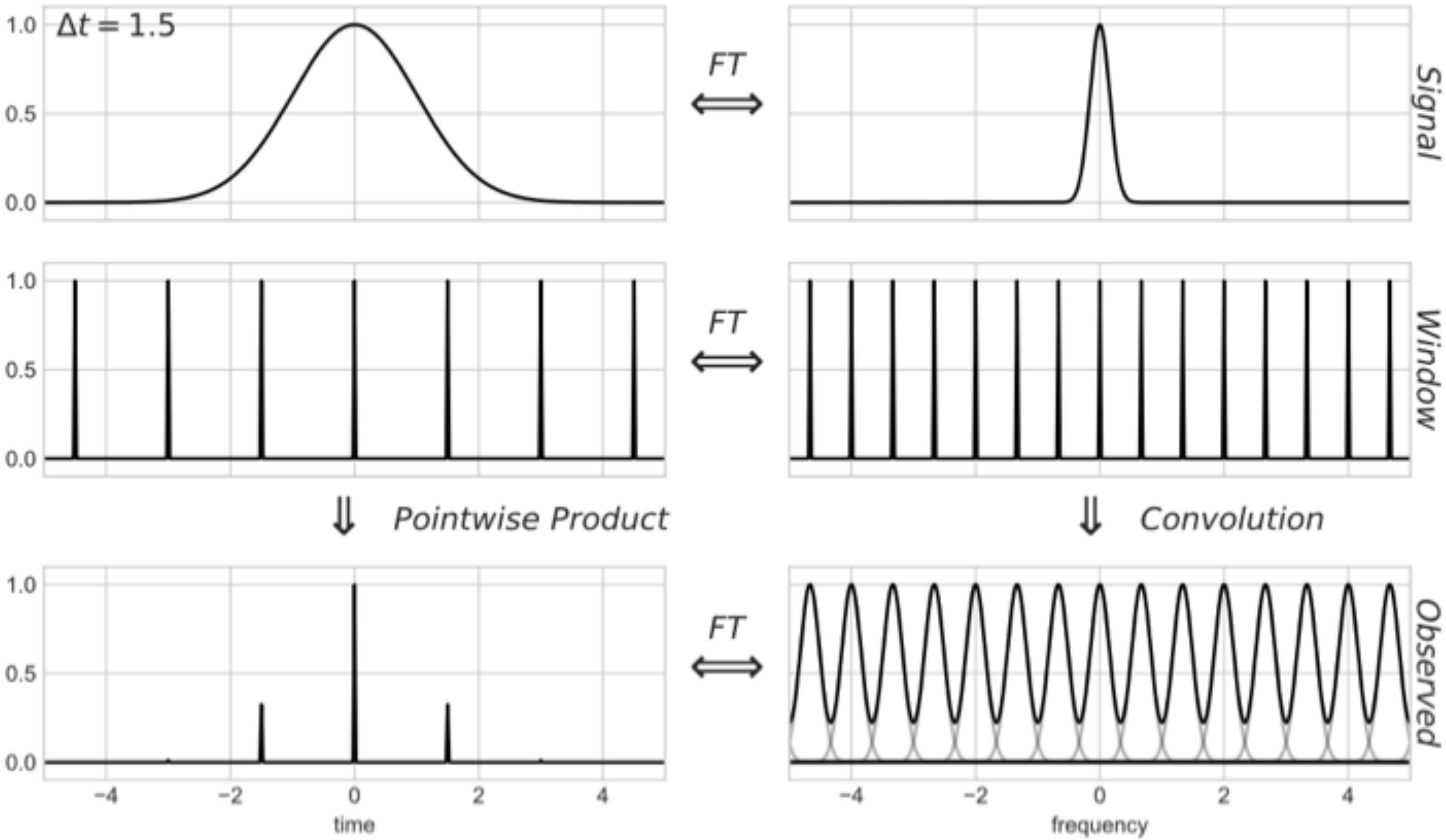
- Optimal window depends on frequency we wish to estimate: Wiener filter (see below)


FT
FT
FT
Signal
Window
Observed
⇓ Pointwise Product
⇓ Convolution
time
frequency


Data: $D(t)$
Data FT: $\mathcal{F}(D)$
FT
Window: $W(t)$
Window FT: $\mathcal{F}(W)$
FT
⇓ Convolution
⇓ Pointwise Product
$[D * W](t) = \mathcal{F}^{-1}\{\mathcal{F}[D] \cdot \mathcal{F}[W]\}$
$\mathcal{F}(D) \cdot \mathcal{F}(W)$
FT
t
f



$\Delta t = 0.5$
FT
FT
FT
Signal
Window
Observed
Pointwise Product
Convolution
1.0
0.5
0.0
-4
-2
0
2
4
time
frequency



$\Delta t = 1.5$
FT
FT
FT
Signal
Window
Observed
Pointwise Product
Convolution
time
frequency
1.0
0.5
0.0
−4
−2
0
2
4

Optimal (Wiener) Filtering

- Suppose we have noisy and convolved time stream of gaussian data and we wish to determine its best reconstruction in absence of noise and smearing
- In absence of noise we would deconvolve, but in presence of noise this would amplify noise
- This corresponds to a gaussian process: how do we choose the kernel K_N? K_N = Corr.

$$\mu_* = \mathbf{K}_{*N}(\mathbf{K}_N + \sigma^2\mathbf{I})^{-1}\mathbf{y}$$

- We have kernel-basis function duality: a gaussian process corresponds to an infinite basis. Fourier basis is complete, so we can rewrite the kernel in terms of sum over Fourier basis: in this basis Corr becomes power spectrum.

Wiener Filter Derivation

- Response convolution $s(t) = \int_{-\infty}^{\infty} r(t-\tau)u(\tau)\,d\tau$ or $S(f) = R(f)U(f)$
- Noise $c(t) = s(t) + n(t)$
- Linear ansatz: determine Φ $\tilde{U}(f) = \dfrac{C(f)\Phi(f)}{R(f)}$
- Least squares: minimize χ^2 $\int_{-\infty}^{\infty} |\tilde{u}(t) - u(t)|^2\,dt = \int_{-\infty}^{\infty} \left|\tilde{U}(f) - U(f)\right|^2 df$ is minimized.
- No cross-term between N and S

$$\int_{-\infty}^{\infty} \left| \frac{[S(f)+N(f)]\Phi(f)}{R(f)} - \frac{S(f)}{R(f)} \right|^2 df$$

$$= \int_{-\infty}^{\infty} |R(f)|^{-2} \left\{ |S(f)|^2 \, |1-\Phi(f)|^2 + |N(f)|^2 \, |\Phi(f)|^2 \right\} df$$

- Take derivative wrt Φ: $\Phi(f) = \dfrac{|S(f)|^2}{|S(f)|^2 + |N(f)|^2}$
- Typically: $\Phi(f)=1$ for low f, $\Phi(f)=0$ for high f.

WF relation to Gaussian Process

$$\mathrm{DATA:}\quad d_i = \sum_j R_{ij} s_j + n_j, \qquad \langle n_i^2 \rangle = N_i$$

$$\mathrm{PRIOR:}\quad p_{pr} \propto e^{-\frac{\sum s_j^2}{2s_j}}, \qquad {}_j = \langle s_j^2 \rangle$$

$$\mathrm{LIKELIHOOD:}\quad \chi^2 = \sum_i \Big(\sum_j R_{ij} s_j - d_i \Big)^2 / N_i$$

$$-\ln p_{post} = \sum_i \frac{\left(\sum_j R_{ij} s_j - d_i\right)^2}{2N_i} + \frac{\sum_j s_j^2}{2s_j}$$

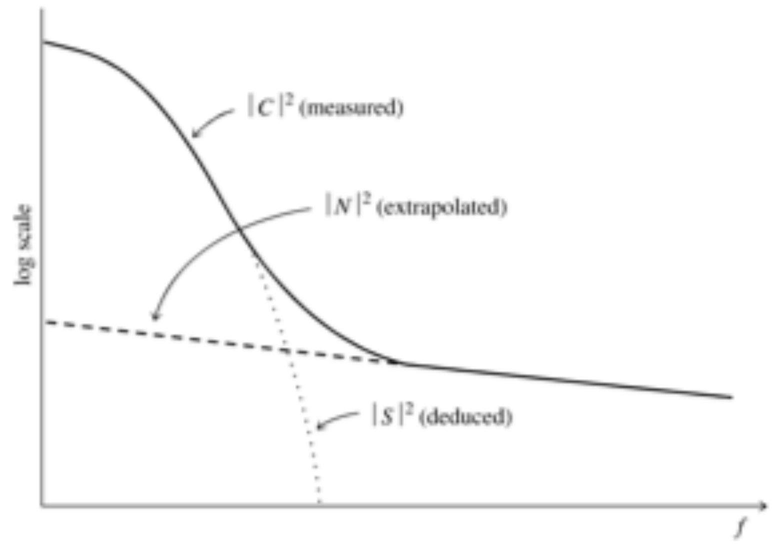
$$-\frac{\partial \ln L}{\partial s_j}\Big|_{\hat{s}} = 0 \rightarrow \sum_i \frac{R_{ij}\left(\sum_j R_{ij}\hat{s}_j - d_i\right)}{N_i} + \frac{\hat{s}_j}{s_j} = 0$$

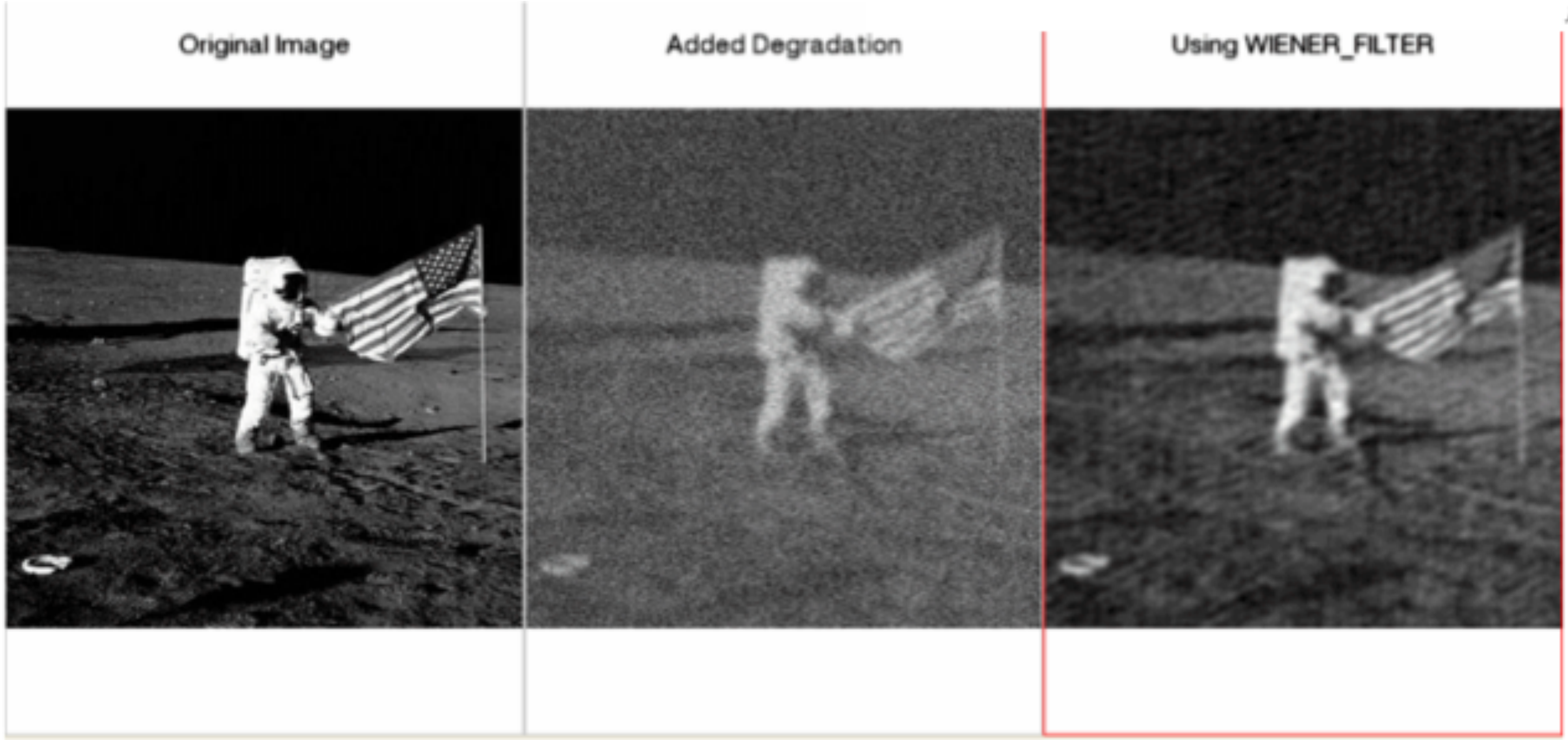
$$\hat{s}_j = \left(R^t N^{-1} R + S^{-1}\right)^{-1} R^t N^{-1} d \qquad \boldsymbol{\mu}_* = \mathbf{K}_{*N}(\mathbf{K}_N + \sigma^2 \mathbf{I})^{-1}\mathbf{y}$$

- This is the same as doing Gaussian process
- The basis is Fourier basis (complete basis) where $K_N = RSR^T$ and $N = \sigma^2 I$, R is FT times response function in Fourier space
- We are guaranteed that K is covariance matrix

Wiener Filter

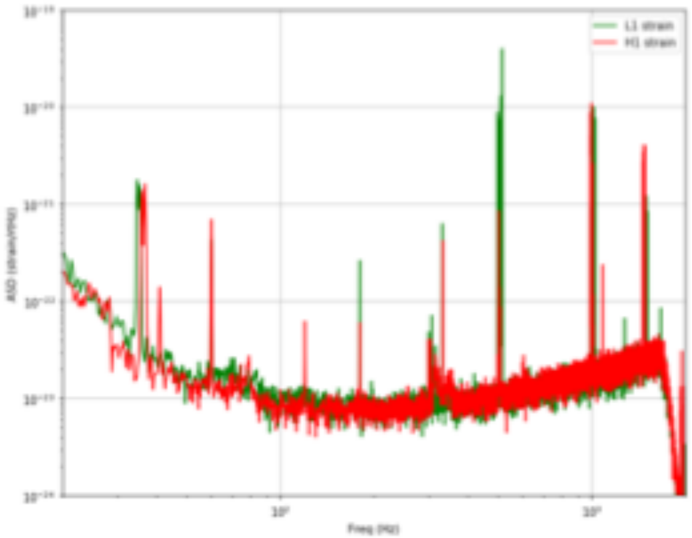
- Leads to smoothed images
  only optimal for Gaussian processes

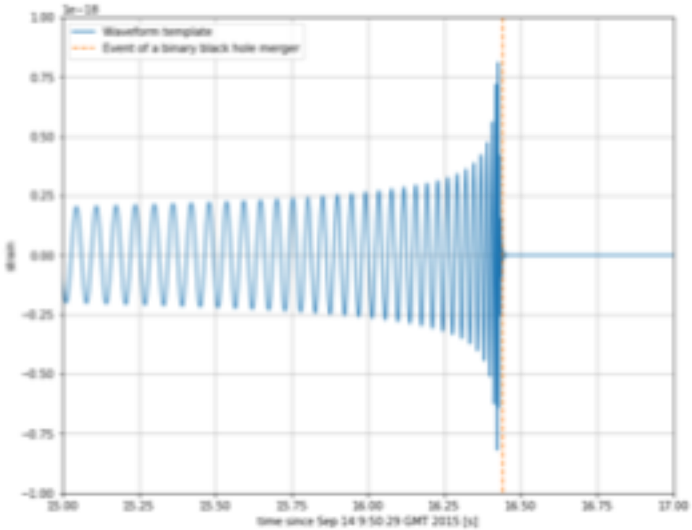


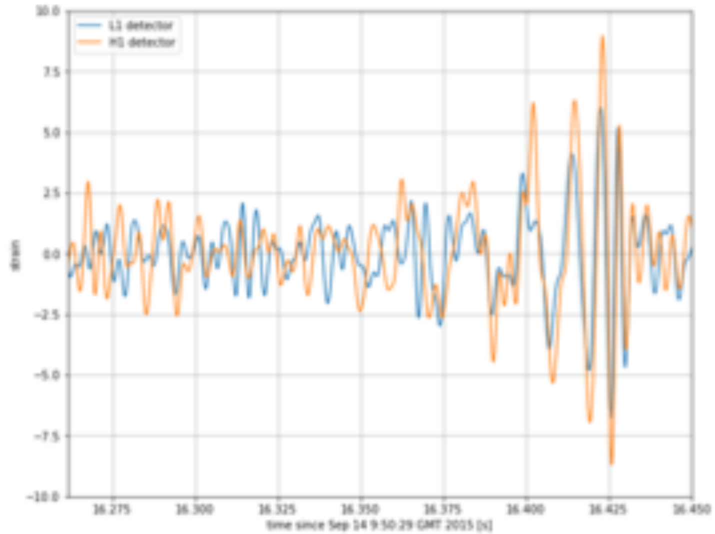


Matched Filtering

- We have stationary but not diagonal noise and a template for signal, but we do not know the signal origin.
- Example from Project 2: LIGO chirp template on top of a correlated noise







Derivation of Matched Filter: 2-d Image

- Stationary noise
$$\begin{aligned}\langle N(\mathbf{x})\rangle &= 0,\\ \langle \hat{N}(\mathbf{k})^* \hat{N}(\mathbf{k})\rangle &= (2\pi)^2\delta(\mathbf{k}' - \mathbf{k})P(\mathbf{k})\end{aligned}$$

- Template $\tau(x)$ assumed at $x' = 0$: $D(\mathbf{x}) = A\,\tau(\mathbf{x}) + N(\mathbf{x})$

- Linear ansatz: find optimal ψ $A_{\rm est} \equiv \int \psi(\mathbf{x})D(\mathbf{x})d^2x$

- Bias $b \equiv \langle A_{\rm est} - A\rangle = \int \psi(\mathbf{x})\tau(\mathbf{x})d^2x - 1,$
- Variance $\sigma^2 \equiv \langle (A_{\rm est} - A)^2\rangle = b^2 + \frac{1}{(2\pi)^2}\int \left|\hat{\psi}(\mathbf{k})\right|^2 P(k)d^2k.$

- We minimize variance subject to zero bias: Lagrange multiplier
- Use Parseval's th.
$$L = \frac{1}{(2\pi)^2}\int \hat{\psi}(\mathbf{k})^* \left[\hat{\psi}(\mathbf{k})P(\mathbf{k}) + \lambda\hat{\tau}(\mathbf{k})\right] d^2k - \lambda$$

Matched Filter

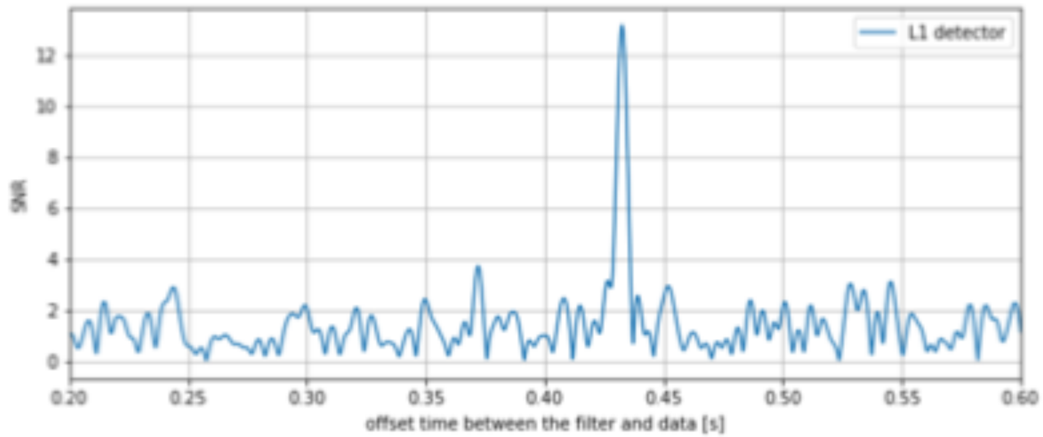
- Minimizing L w.r.t. ψ $$L = \frac{1}{(2\pi)^2}\int \hat{\psi}(\mathbf{k})^* \left[\hat{\psi}(\mathbf{k})P(\mathbf{k}) + \lambda\hat{\tau}(\mathbf{k})\right] d^2k \; - \lambda$$

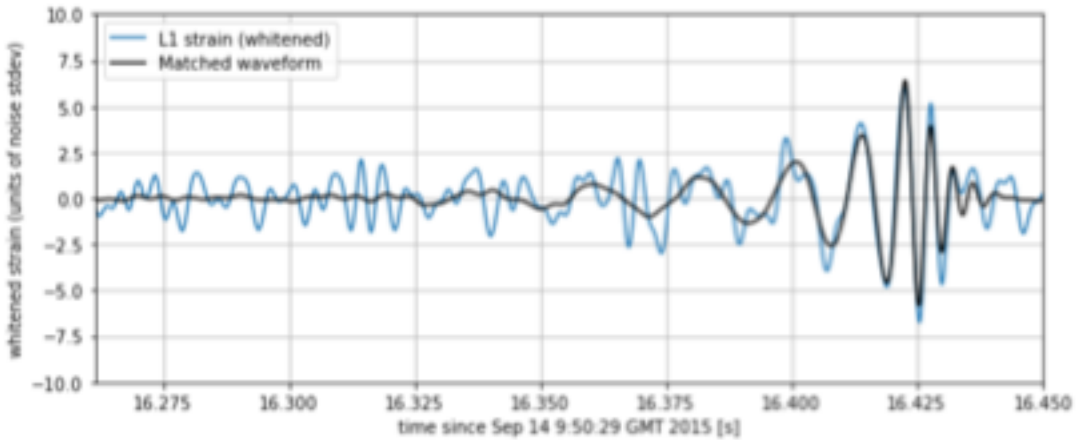
$$\hat{\psi}(\mathbf{k}) \propto \hat{\tau}(\mathbf{k})/\breve{P}(\mathbf{k}) \qquad \hat{\psi}(\mathbf{k}) = \left[\frac{1}{(2\pi)^2}\int \frac{|\hat{\tau}(\mathbf{k}')|^2}{P(\mathbf{k}')} d^2k'\right]^{-1} \frac{\hat{\tau}(\mathbf{k})}{P(\mathbf{k})}$$

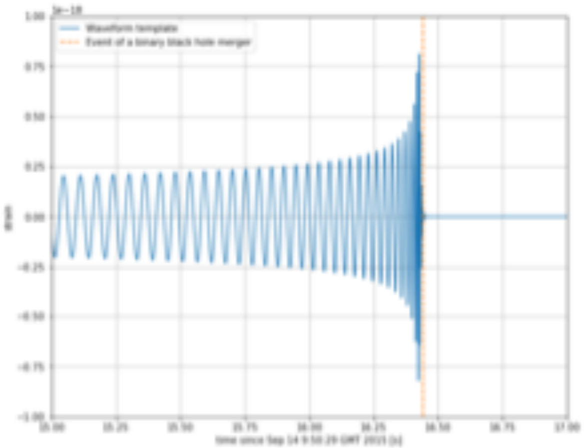
- MF solution filters out frequencies where signal to noise τ/P is low
- Signal to noise estimate $$\frac{A}{\sigma} = \left[\frac{1}{(2\pi)^2}\int \frac{|\hat{\tau}(\mathbf{k})|^2}{P(\mathbf{k})} d^2k\right]^{1/2} A.$$

- Procedure: we want to evaluate S/N for every point, not just $x' = 0$, so we want to convolve data $D(x)$ with matched filter $\tau(x+x')$: we FT $D(x)$ and template $\tau(x)$, evaluate $\psi(k)$, multiply with $D(k)$, FT back, divide by σ to get S/N, look for S/N peaks.

Project 2: LIGO Matched Filter Analysis



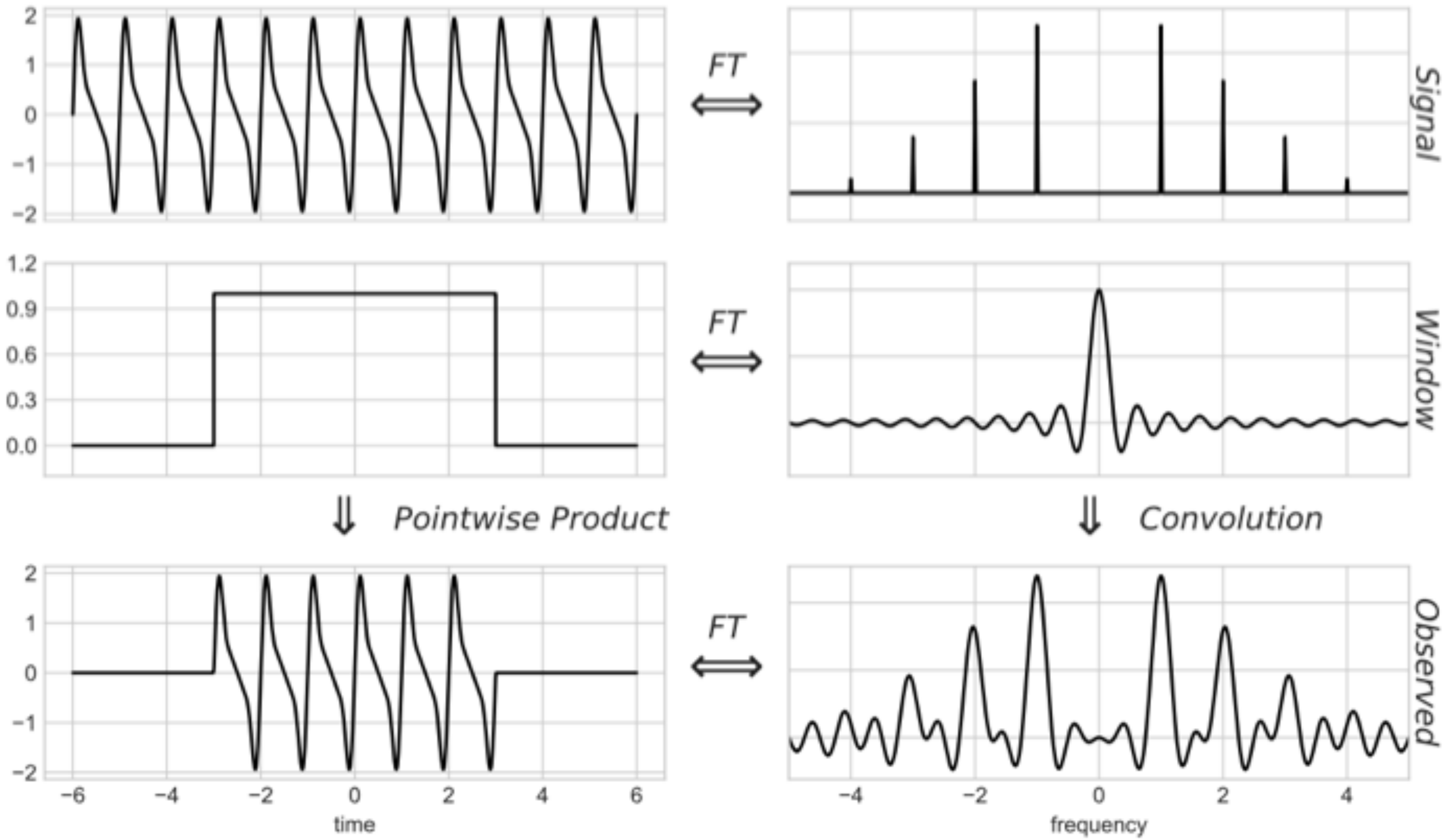




Note how low f oscillations disappeared: filtered out by matched filter because noise power is too high at low f

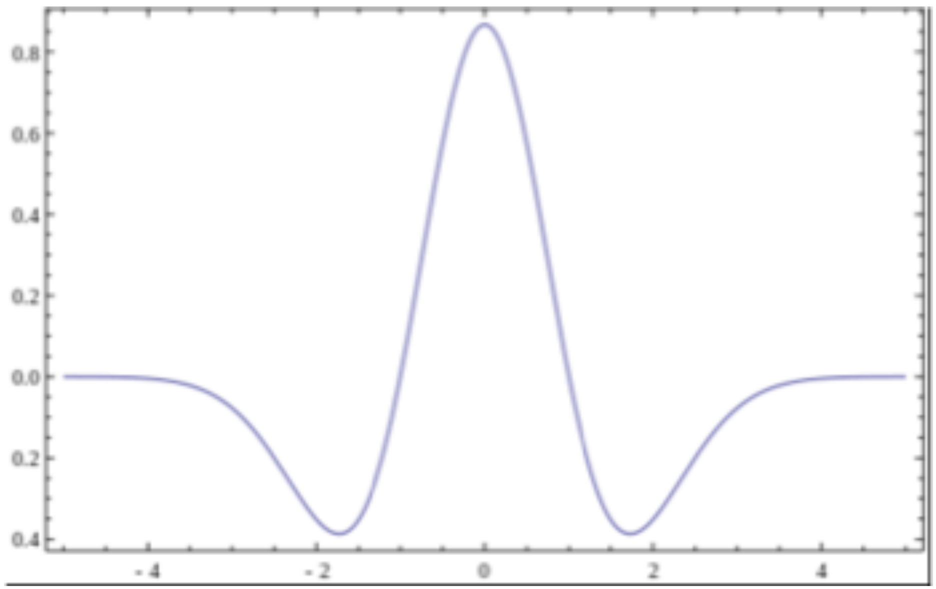
Wavelets: a brief mention

- We can view real space position basis as fully localized spatially and not at all in frequency. We can view Fourier space as fully localized in frequency and not at all spatially (a plane wave has equal support everywhere).
- Heisenberg uncertainty principle: we can localize in space or in frequency, but not both
- Basis expansions that partially localize in space and partially in frequency are called wavelets
- Computationally attractive if the data can be made sparse by doing the wavelet expansion
- One can achieve similar results with spatial FFT


FT
FT
FT
Signal
Window
Observed
Pointwise Product
Convolution
time
frequency

Wavelets: a brief mention

- There are many classes of discrete wavelets (Haar, Daubechies…)
- Some are continuous, eg mexican hat wavelet
- Typically they have the property that they integrate to 0 (hence its FT is 0 for $f = 0$)
- Sparseness and localization lead to good image compression properties



(a) 100%

(b) 23%

(c) 5.5%

(d) 5.5% Fourier

Summary

- Fourier methods revolutionized fields like image processing, Poisson solvers, PDEs, convolutions…
- At the core is FFT algorithm scaling as $N\log_2 N$
- Many physical sciences use these techniques: see Project 2, LIGO analysis of black hole black hole merger event

Literature

- *Numerical Recipes*, Press et al., Chapter 12-13