ECE 595, Section 10
Numerical Simulations
Lecture 12: Applications of FFT

Prof. Peter Bermel

February 6, 2013

Outline

- Recap from Friday
- Real FFTs
- Multidimensional FFTs
- Applications:
  - Correlation measurements
  - Filter diagonalization method

Recap from Friday

- Recap from Wednesday
- Fourier Analysis
  - Scalings and Symmetries
  - Sampling Theorem
- Discrete Fourier Transforms
  - Naïve approach
  - Danielson-Lanczos lemma
  - Cooley-Tukey algorithm

Real FFTs

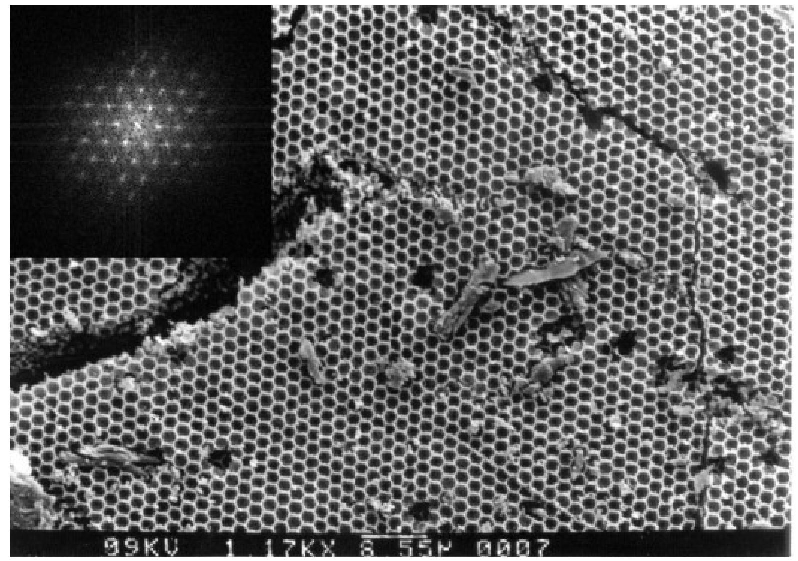
- For real functions, the general complex FFT procedure is wasteful
- Solutions:
  - Pack twice as many FFTs into each calculation
  - Reduce length by half, sort out result
  - Use sine and cosine transforms
- Application: signal processing of experimental measurement data

Multidimensional FFTs

- Applications: image processing, band structures
- Definition:

$$F\left(n_x, n_y\right) = \sum_{k_x=1}^{N} \sum_{k_y=1}^{N} f_{k_x k_y} e^{2\pi j\left[k_x n_x + k_y n_y\right]/N}$$

- For FFT data in 2D or 3D, can efficiently perform FT in each dimension successively



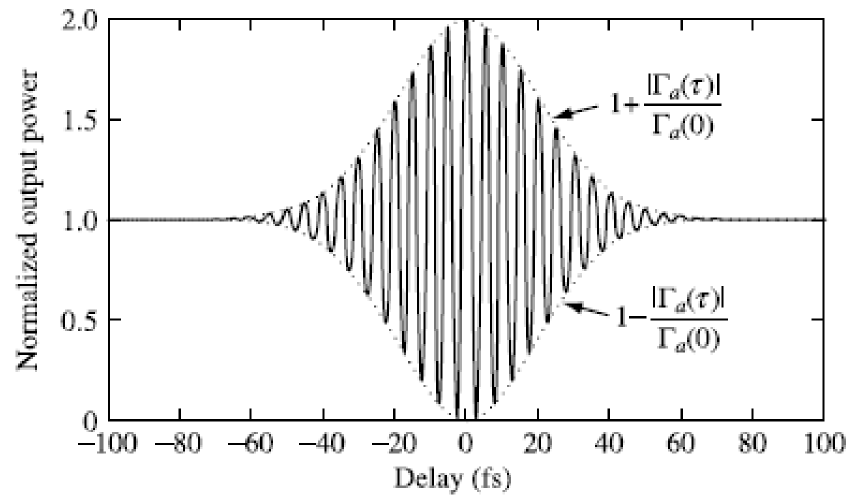
M. Leistikow et al., Phys. Rev. Lett. **107**, 193903 (2011).

Correlation Measurements

- Application: ultrafast optics, quantum optics
- Correlation for discrete data defined by:

$$g_2(m) = \sum_{k=1}^{N} f_k h_{k+m}$$

- Autocorrelation: special case where $f = h$



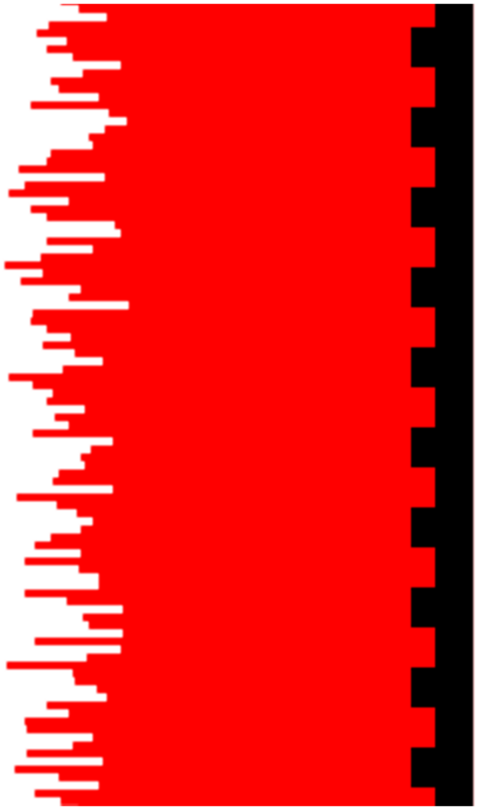
From A.M. Weiner, *Ultrafast Optics* (2009).

Correlation Measurements

- Autocorrelation powerful signature of the nature of one's data set
- Largest value for m=0
- Pure noise: δ-function correlated
- Pure periodic signal: cross-correlation also has same period
- Most signals decay with characteristic correlation time τ_c

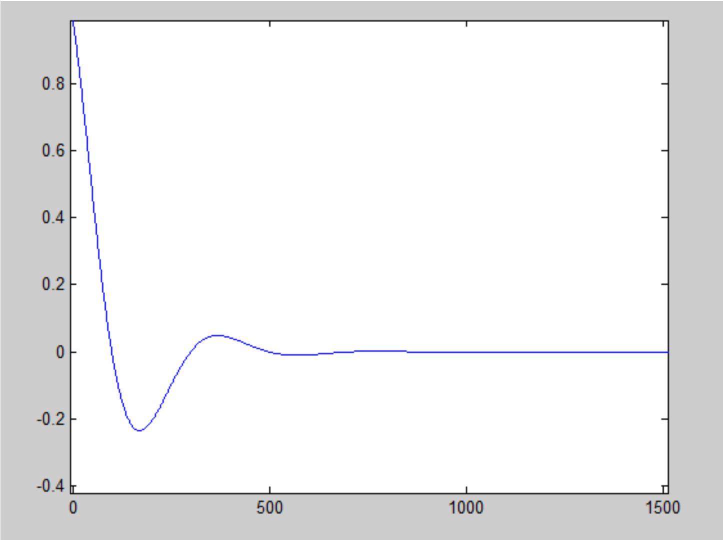
Time-domain data analysis

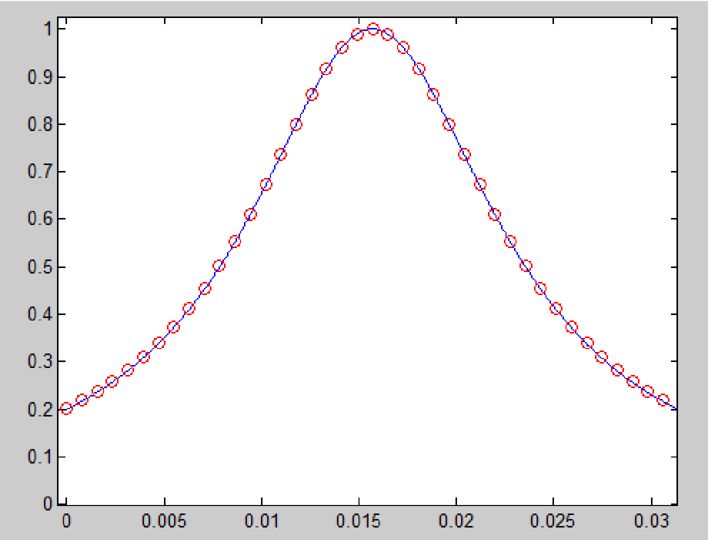
- Many PDE solvers produce a time series of data warranting spectral analysis
- Examples: finite-difference time domain, drift-diffusion models



Signal Processing

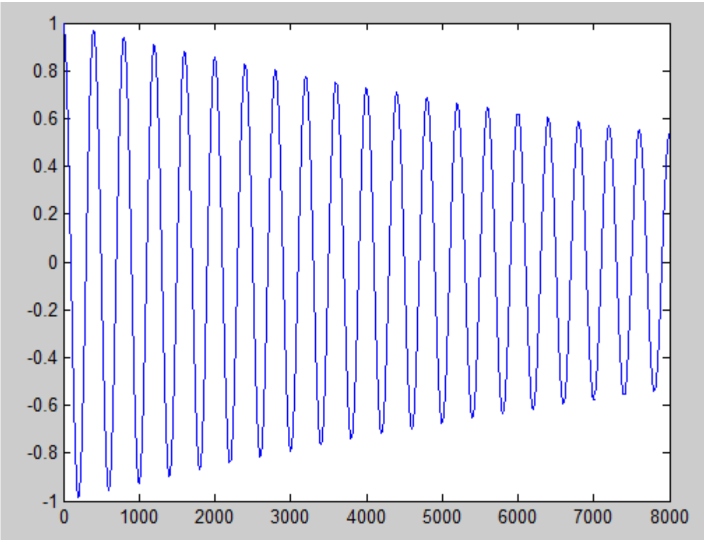
- Most obvious approach: least-squares fit to FFT of time-series data
- Given a set of narrow Lorentzian peaks, should fit well, right? Problem solved!

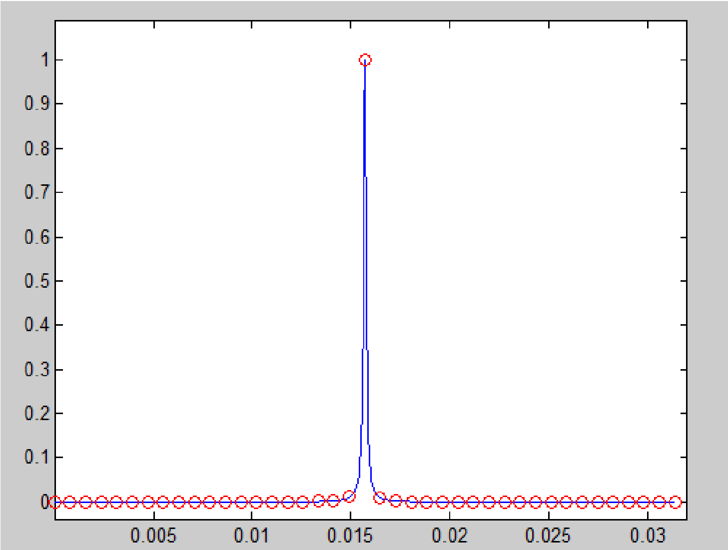




Signal Processing

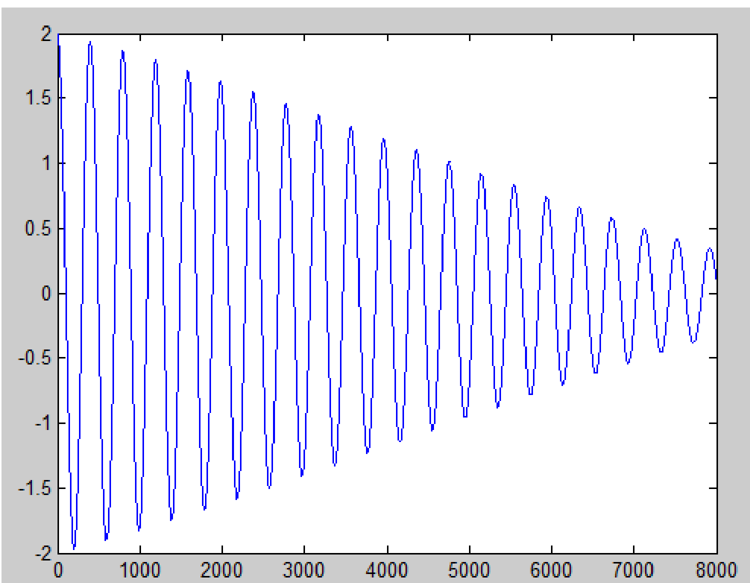
- But what if the decay is slow, and unfinished?
- The FFT of the time-series will look <u>significantly different</u> from goal

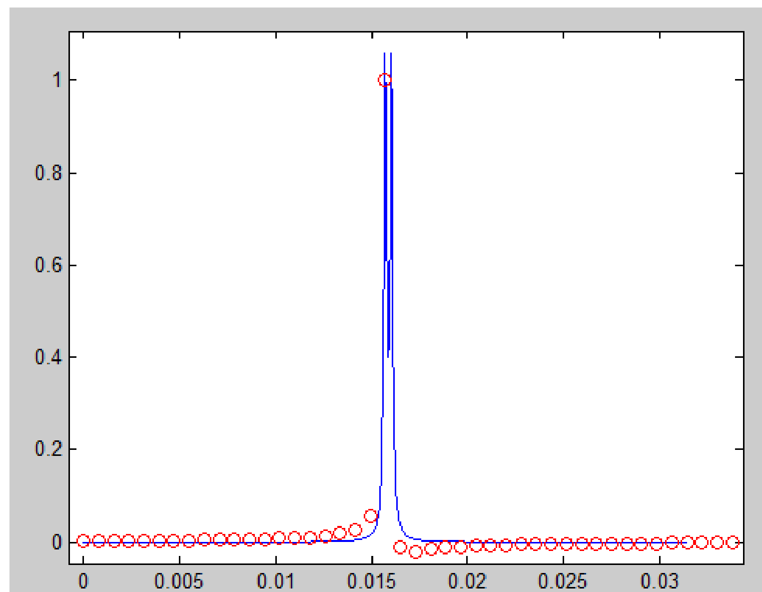




Signal Processing

- An even greater challenge – what if you have two time decays with relatively close frequencies (*this case is fairly common*)?
- Can't even detect the <u>number</u> of modes!





Signal Processing

- Need to find an alternative strategy to straightforward FFTs
- Might want to add damping explicitly
- Most obvious approach known as decimated signal diagonalization
- One particularly useful approach devised by Mandelshtam is known as filter diagonalization

Filter Diagonalization Method

[Mandelshtam, *J. Chem. Phys.* **107**, 6756 (1997)]

Given time series y_n, write: $$y_n = y(n\Delta t) = \sum_k a_k e^{-i\omega_k n\Delta t}$$

…find *complex* amplitudes a_k & frequencies ω_k
by a simple linear-algebra problem!

Idea: pretend $y(t)$ is autocorrelation of a quantum system:

$$\hat{H}|\psi\rangle = i\hbar\frac{\partial}{\partial t}|\psi\rangle$$

time-Δt evolution-operator: $$\hat{U} = e^{-i\hat{H}\Delta t/\hbar}$$

say: $$y_n = \langle\psi(0)|\psi(n\Delta t)\rangle = \langle\psi(0)|\hat{U}^n|\psi(0)\rangle$$

Filter-Diagonalization Method

[Mandelshtam, *J. Chem. Phys.* **107**, 6756 (1997)]

$$y_n = \langle \psi(0) | \psi(n\Delta t) \rangle = \langle \psi(0) | \hat{U}^n | \psi(0) \rangle \qquad \hat{U} = e^{-i\hat{H}\Delta t/\hbar}$$

We want to diagonalize *U*: eigenvalues of *U* are $e^{i\omega\Delta t}$

…expand *U* in basis of $|\psi(n\Delta t)>$:

$$U_{m,n} = \langle \psi(m\Delta t) | \hat{U} | \psi(n\Delta t) \rangle = \langle \psi(0) | \hat{U}^m \hat{U} \hat{U}^n | \psi(0) \rangle = y_{m+n+1}$$

U_{mn} given by y_n's — just diagonalize known matrix!

Filter-Diagonalization Summary

[Mandelshtam, *J. Chem. Phys.* **107**, 6756 (1997)]

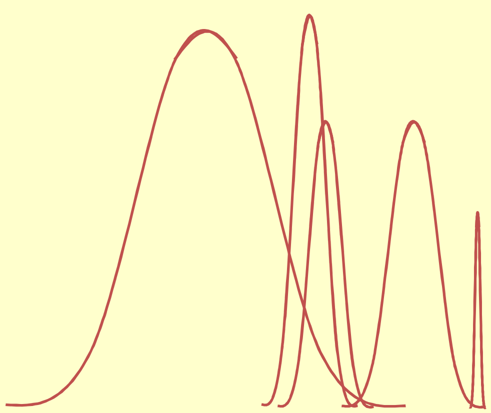
U_{mn} given by y_n's — just diagonalize known matrix!

A few omitted steps:

—Generalized eigenvalue problem (basis not orthogonal)

—Filter y_n's (Fourier transform):

small bandwidth = smaller matrix (less singular)



- resolves many peaks at once
- # peaks not known *a priori*
- resolve overlapping peaks
- resolution >> Fourier uncertainty

Next Class

- Is on Friday, Feb. 8
- Will discuss FFTW
- Recommended reading: FFTW User Guide: http://www.fftw.org/fftw3_doc/