ECE 595, Section 10
Numerical Simulations
Lecture 11: Fast Fourier Transforms

Prof. Peter Bermel

February 1, 2013

Outline

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

Recap from Wednesday

- Schrodinger's equation
- Infinite & Finite Quantum Wells
- Kronig-Penney model
- Numerical solutions:
  - Real space
  - Fourier space

Fourier Analysis

- Fourier transformation is a linear operation that maps time-series data into the Fourier-domain:

$$\tilde{f}(\omega) = \int_{-\infty}^{\infty} dt\, f(t) e^{i\omega t}$$

- Independent variable interpreted as frequency
- Inverse Fourier transform:

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} d\omega\, \tilde{f}(\omega) e^{-i\omega t}$$

Fourier Symmetries

Time-domain property of $f(t)$	Frequency-domain property of $\tilde{f}(\omega)$
$f(t) = \mathrm{Re}\{f(t)\}$	$\tilde{f}(-\omega) = \tilde{f}^*(\omega)$
$f(t) = \mathrm{Im}\{f(t)\}$	$\tilde{f}(-\omega) = -\tilde{f}^*(\omega)$
$f(t) = f(-t)$	$\tilde{f}(-\omega) = \tilde{f}(\omega)$
$f(t) = -f(-t)$	$-\tilde{f}(-\omega) = \tilde{f}(\omega)$
$f(t) = f(-t) = \mathrm{Re}\{f(t)\}$	$\tilde{f}(-\omega) = \tilde{f}(\omega) = \mathrm{Re}\{\tilde{f}(\omega)\}$
$f(t) = -f(-t) = \mathrm{Re}\{f(t)\}$	$-\tilde{f}(-\omega) = \tilde{f}(\omega) = \mathrm{Im}\{\tilde{f}(\omega)\}$
$f(t) = f(-t) = \mathrm{Im}\{f(t)\}$	$\tilde{f}(-\omega) = \tilde{f}(\omega) = \mathrm{Im}\{\tilde{f}(\omega)\}$
$f(t) = -f(-t) = \mathrm{Im}\{f(t)\}$	$-\tilde{f}(-\omega) = \tilde{f}(\omega) = \mathrm{Re}\{\tilde{f}(\omega)\}$

Fourier Scalings

Time Domain Expression	Frequency-Domain Expression
$f(t)$	$\tilde{f}(\omega)$
$\alpha f(t) + \beta g(t)$	$\alpha\tilde{f}(\omega) + \beta\tilde{g}(\omega)$
$f(at)$	$\frac{1}{\|a\|}\tilde{f}\left(\frac{\omega}{a}\right)$
$f(t - t_o)$	$\tilde{f}(\omega)e^{i\omega t_o}$
$f(t)e^{-i\omega t_o}$	$\tilde{f}(\omega - \omega_o)$
$t^n f(t)$	$i^n \frac{d^n\tilde{f}(\omega)}{d\omega^n}$
$\frac{d^n f(t)}{dt^n}$	$(i\omega)^n\tilde{f}(\omega)$

Sampling Theorem

- At finite sampling rates, need at least two data points
- For a sample time interval Δ, max measurable frequency is the Nyquist frequency $f_c=1/2\Delta$
- Sampling theorem says that f_c bandwidth-limited spectrum **completely determined** by data sampled at intervals of Δ

Discrete Fourier Transforms

- Accounts for finite time spacing of data points
- Gives rise to finite frequency spacing and maximum frequency (from sampling theorem)
- DFT defined by:
  $F(n) = \sum_{k=1}^{N} f(x_k) e^{-2\pi j (x_k n / x_N)}$
- For uniform time spacing Δ:

$$F(n) = \sum_{k=1}^{N} f_k e^{-2\pi jkn/N}$$

Discrete Fourier Transforms: Naïve Algorithm

- Rewrite DFT as:

$$F(n) = \sum_{k=1}^{N} f_k \, W^{nk}$$

- Where $W = e^{-2\pi j/N}$
- Just perform sum (*N* operations) for each frequency (*N* times)
- Overall time scales as N^2

Discrete Fourier Transforms: Naïve Algorithm

- Obtain same number of points in DFT as original series
- Symmetry properties same as for continuous FT
- For inverse Fourier transform – just use the inverse of W:

$$f_k = \frac{1}{N}\sum_{k=1}^{N} F(n)\left(\frac{1}{W}\right)^{nk}$$

Fast Fourier Transforms: Danielson-Lanczos Lemma

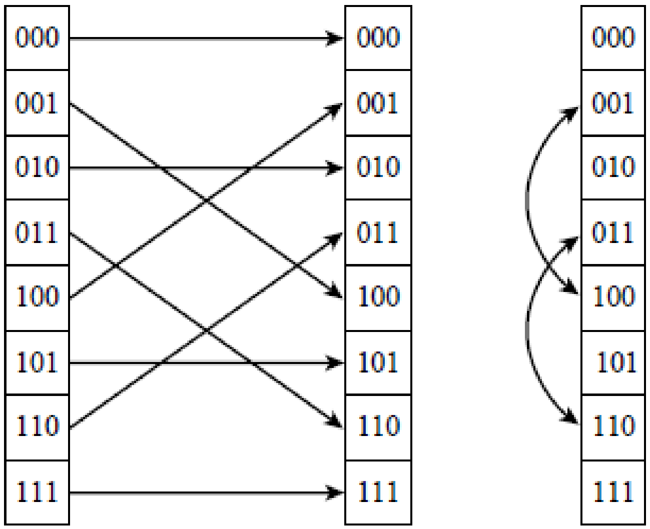
- Based on Danielson-Lanczos lemma:

$$F(n) = \sum_{k=1}^{N} f_k \, W^{nk}$$
$$= \sum_{k=1}^{N/2} f_{2k} \, W^{n2k} + \sum_{k=1}^{N/2} f_{2k+1} \, W^{n(2k+1)}$$
$$= F^e(n) + W^n F^o(n)$$

- Can be applied recursively

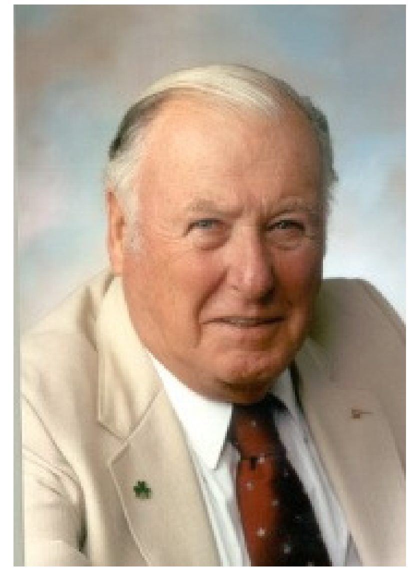
Discrete Fourier Transforms: Cooley-Tukey Algorithm

- If one applies D-L lemma recursively to a data set with $N = 2^m$, reduce to FT of single point!
- Key is to order everything to keep track of where everything should go – then work backwards



Discrete Fourier Transforms: Cooley-Tukey Algorithm

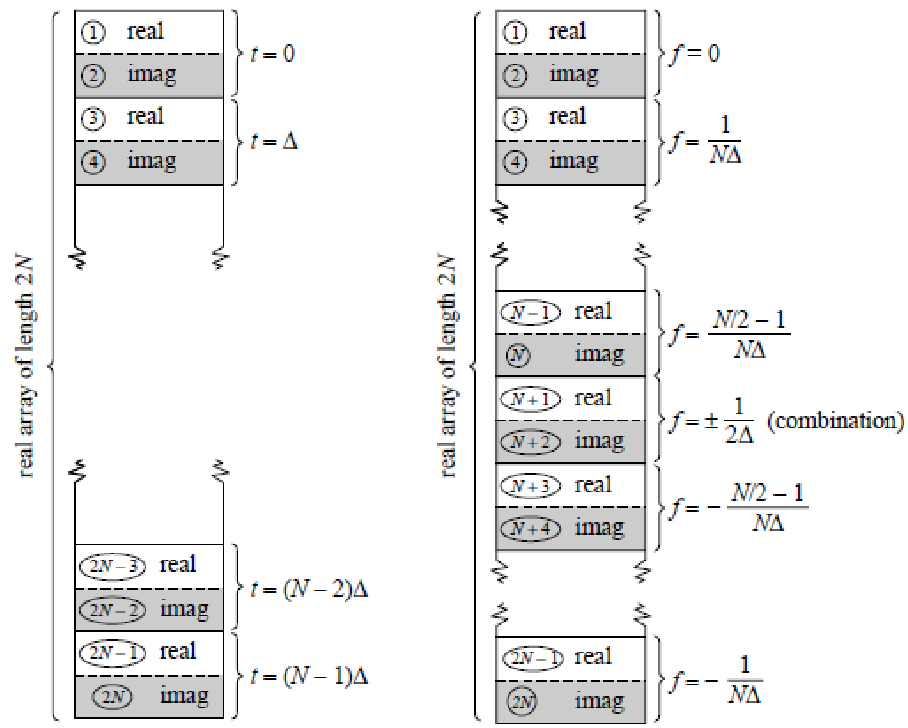
- Algorithm devised by Cooley and Tukey:
  - Sort data into bit reversed order
  - Perform FTs on lengths 1, 2, 4, 8, etc. with D-L lemma
- Operations for first step go as *N*
- Operations for second step go as *N* per cycle, with $\log_2 N$ cycles
- Overall time is $N \log_2 N$– considerably better than naïve approach



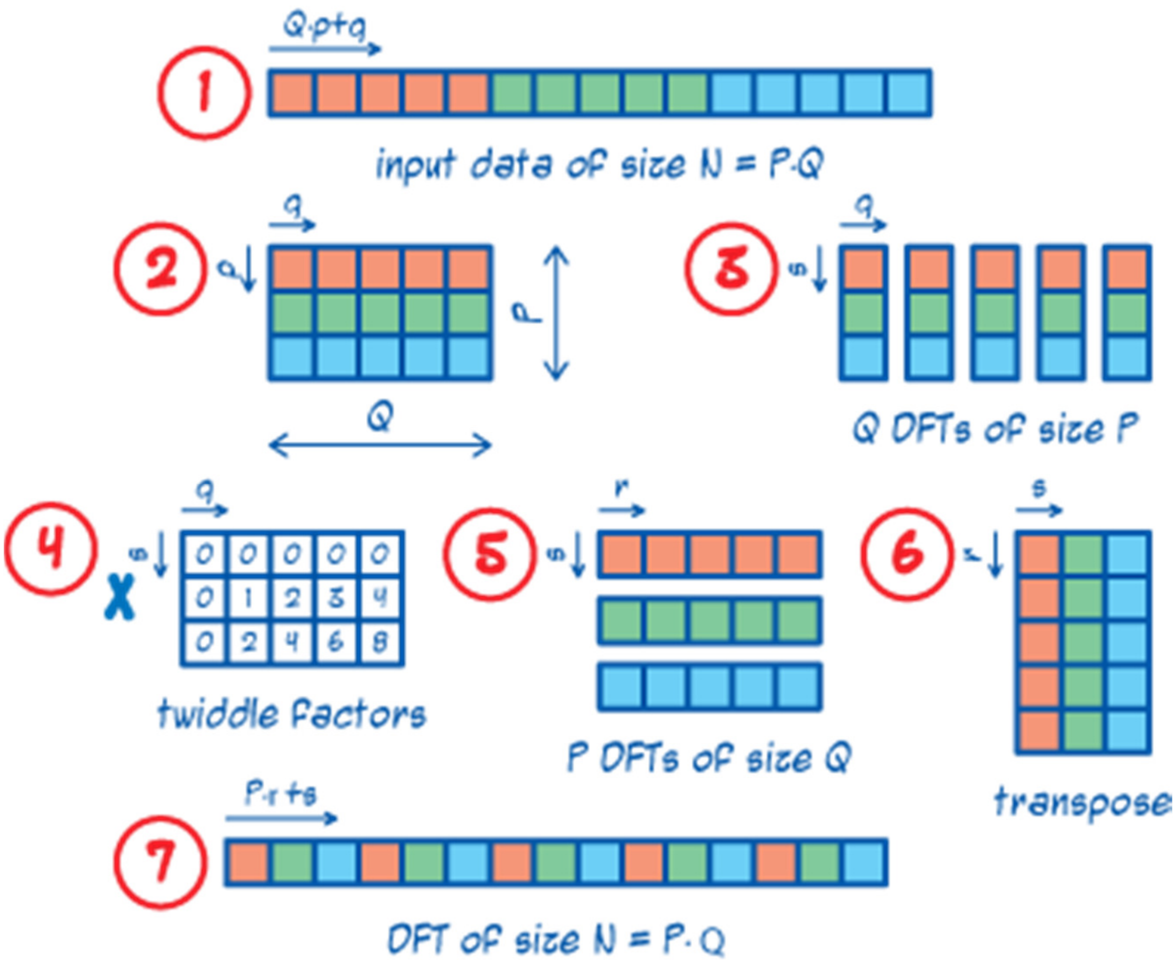
J.W. Cooley (IEEE Global History Network)

Fast Fourier Transform Data

- Input data is uniformly spaced
- Output consists of rising positive frequencies, followed by negative frequencies decreasing in magnitude



Cooley-Tukey Algorithm



Next Class

- Is on Wednesday, Feb. 6
- Will discuss numerical tools for Fast Fourier Transforms
- Recommended reading: Numerical Recipes, Chapters 12-13
- Please email me your HW #2 today (by 4:30 pm)