Fourier Transforms – Some Properties of DFT

CS370 Lecture 20 – Feb 17, 2017

Midterm details, etc.

Materials allowed: Official course notes, handwritten notes, assignments, non-programmable calculators.

Data/Time: **Tues, Feb 28, 7-9pm**.

Location: To be announced. (You'll be divided among 3 rooms.)

Covers: Floating point, interpolation and splines, ordinary differential equations. (No Fourier transforms.)

Q/A Session: Sunday, Feb 26. Time/Location: TBA (on Piazza).

Lecture on Mon, Feb 27: Assignment return, and questions

Lecture on Wed, March 1: **Canceled**.

Lecture on Fri, March 3: (Probably) discussion of midterm.

Recall: Discrete Fourier Transform

Last time, we developed the Discrete Fourier Transform pair:

$$f_n = \sum_{k=0}^{N-1} F_k W^{nk} \qquad \text{and} \qquad F_k = \frac{1}{N}\sum_{n=0}^{N-1} f_n W^{-nk}$$

where $W = e^{\frac{2\pi i}{N}}$, recalling Euler's formula $e^{i\theta} = \cos\theta + i\,\sin\theta$.

This allows a perfect transformation between N "time-domain" data points f_n and N "frequency-domain" Fourier coefficients F_k.

Emphasizes the frequencies present in the data.

Note: A Lack of Standardization

There is no single accepted standard definition of the DFT/IDFT pair.

We use the following, with 0-based indexing:

$$f_n = \sum_{k=0}^{N-1} F_k W^{nk} \quad \text{and} \quad F_k = \frac{1}{N}\sum_{n=0}^{N-1} f_n W^{-nk}.$$

Matlab uses this instead, with 1-based indexing:

$$f_n = \frac{1}{N}\sum_{k=1}^{N} F_k W^{(n-1)(k-1)} \quad \text{and} \quad F_k = \sum_{n=1}^{N} f_n W^{-(n-1)(k-1)}.$$

Note the different placement of the scaling by $\frac{1}{N}$.

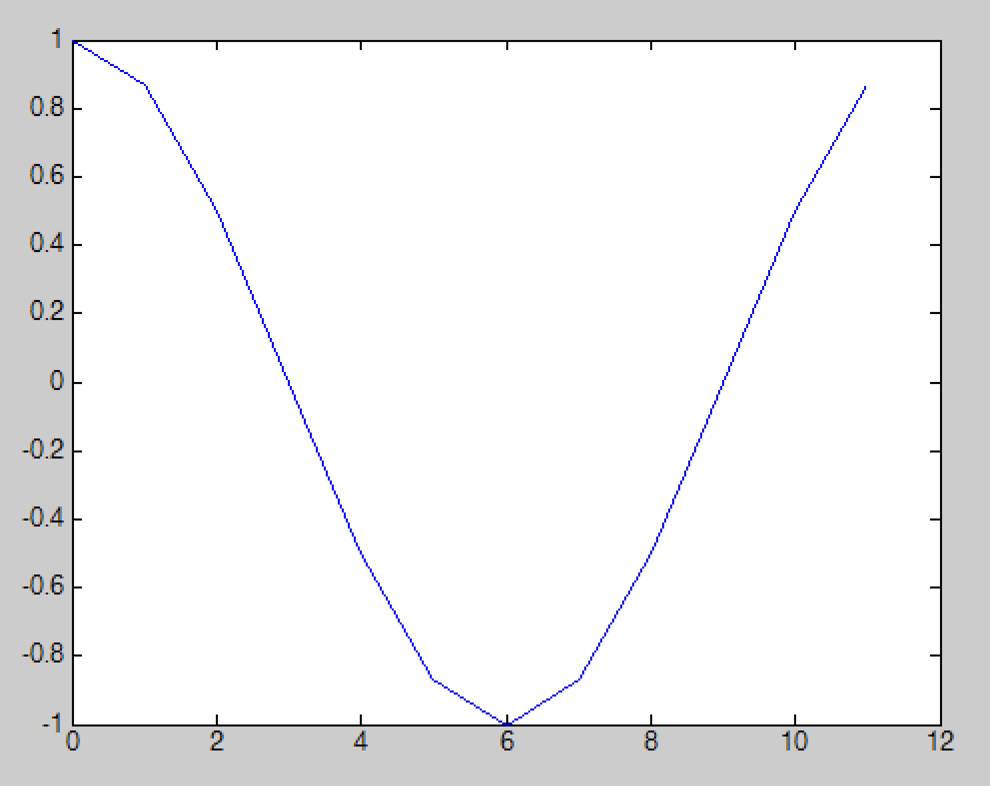
So be careful (a) when coding in Matlab, and (b) reading other sources!

Example 5.1: $\mathrm{f_n} = \cos\left(\frac{2\pi n}{N}\right)$

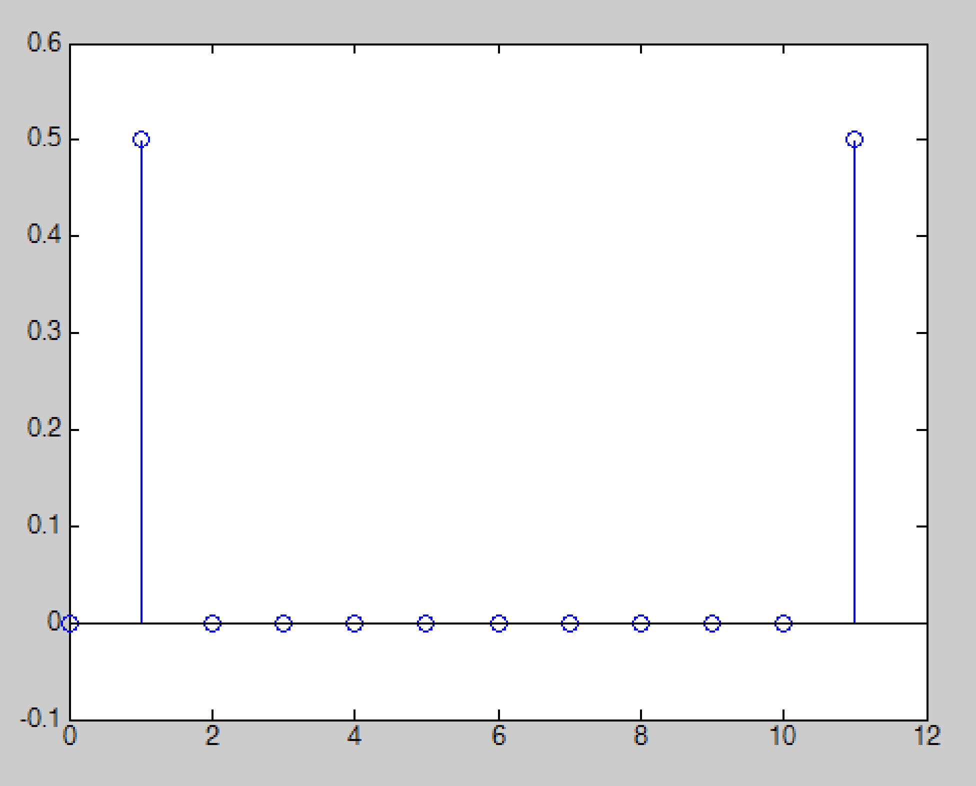
Consider a set of N data points such that $\mathrm{f_n} = \cos\left(\frac{2\pi n}{N}\right)$.

Show that $F_1 = F_{N-1} = \frac{1}{2}$, and for all other coefficients, $F_k = 0$.

Example 5.1: DFT of N=12 points for $\cos\left(\frac{2\pi n}{N}\right)$



Original Data f_n



Resulting Fourier coefficients F_k

Example 5.1: $\mathrm{f_n} = \cos\left(\frac{2\pi n}{N}\right)$

Consider a set of N data points such that $\mathrm{f_n} = \cos\left(\frac{2\pi n}{N}\right)$.

Show that $F_1 = F_{N-1} = \frac{1}{2}$; for all other coefficients, $F_k = 0$.

Therefore we can express f_n in our Fourier representation as

$$f_n = \cos\left(\frac{2\pi n}{N}\right) = \sum_{k=0}^{N-1} F_k W^{nk} = \frac{1}{2} W^{n(1)} + \frac{1}{2} W^{n(N-1)}.$$

Two Properties of the DFT

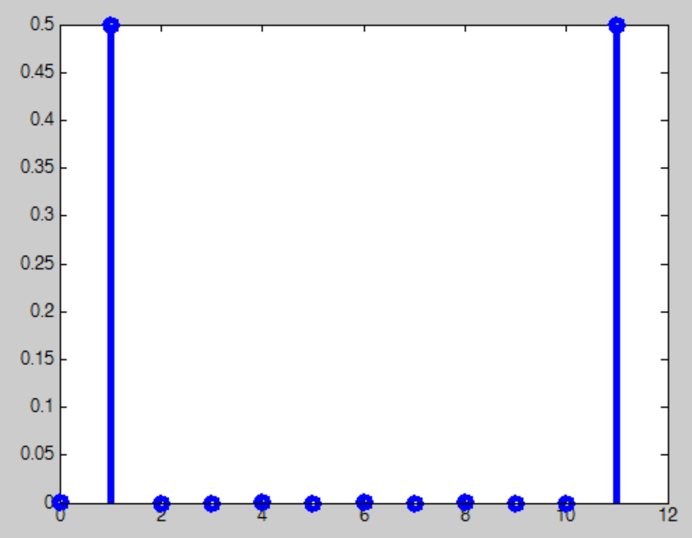
As consequences of the properties of Nth roots of unity…

1. The sequence $\{F_k\}$ is doubly infinite and periodic.

i.e., If we allow $k < 0$ or $k > N - 1$, the F_k coefficients repeat.

2. *Conjugate symmetry*: If data f_n is *real,* $F_k = \overline{F_{N-k}}$.

Hence the $|F_k|$ are symmetric about $k = \frac{N}{2}$, as we saw in the cos example's power spectra.



F_k plot for $\cos\left(\frac{2\pi n}{N}\right)$ exhibits conjugate symmetry.

Yet to come in Fourier Transforms…

- We’ll try to develop more intuition for what the DFT *means*.
- We’ll look at how to do the DFT *quickly* (Fast Fourier Transform, FFT).
- We’ll discuss some *applications*.

Discrete Fourier Transform

Typically, we want to learn/achieve something by processing the data.

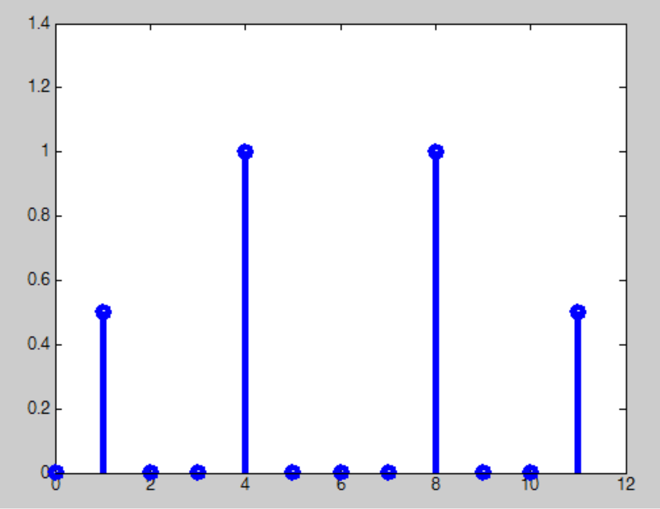
(Image data, audio samples, prices, intensities, etc.)

In theory, the time-domain data tells us everything!

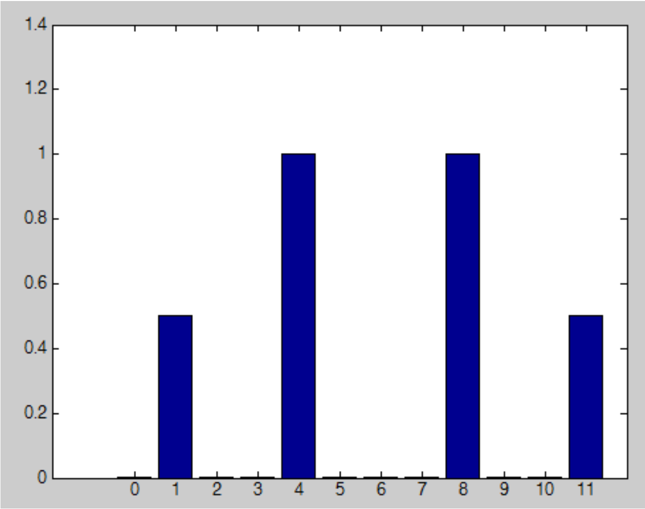
In practice, Fourier coefficients provide easier access to useful insights/information for certain problems.

Power/Fourier Spectrum

The "power spectrum" visualizes the Fourier coefficients by plotting their moduli/magnitudes $|F_k|$, e.g., using Matlab.



stem(abs(F))



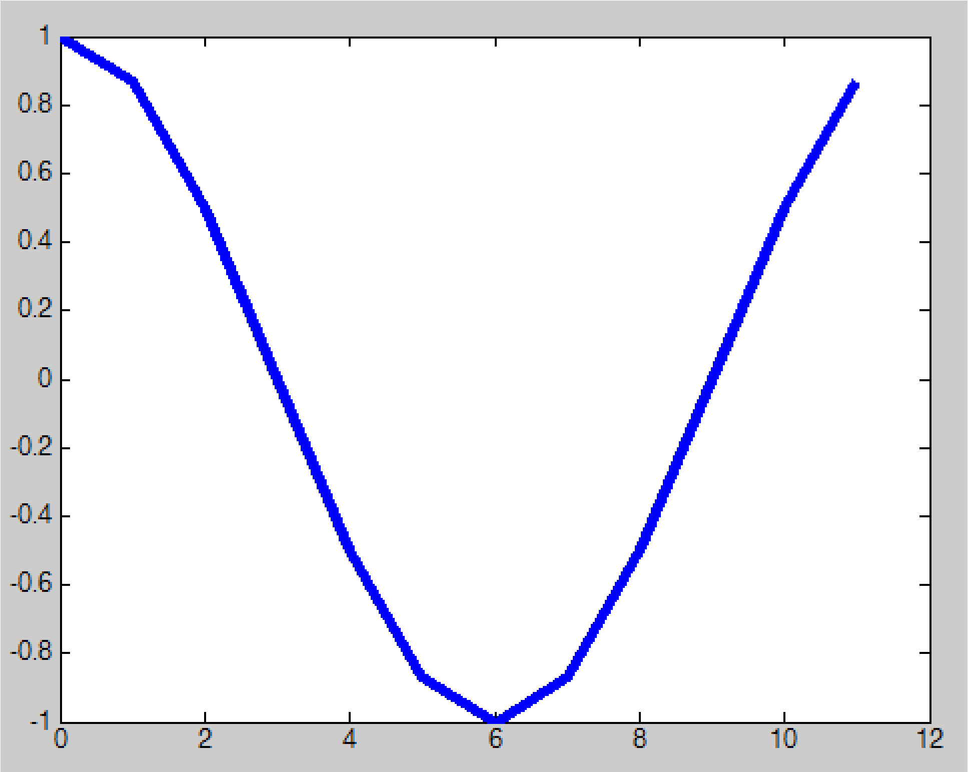
bar(abs(F))

This expresses the magnitude of the frequencies, but ignores phase.

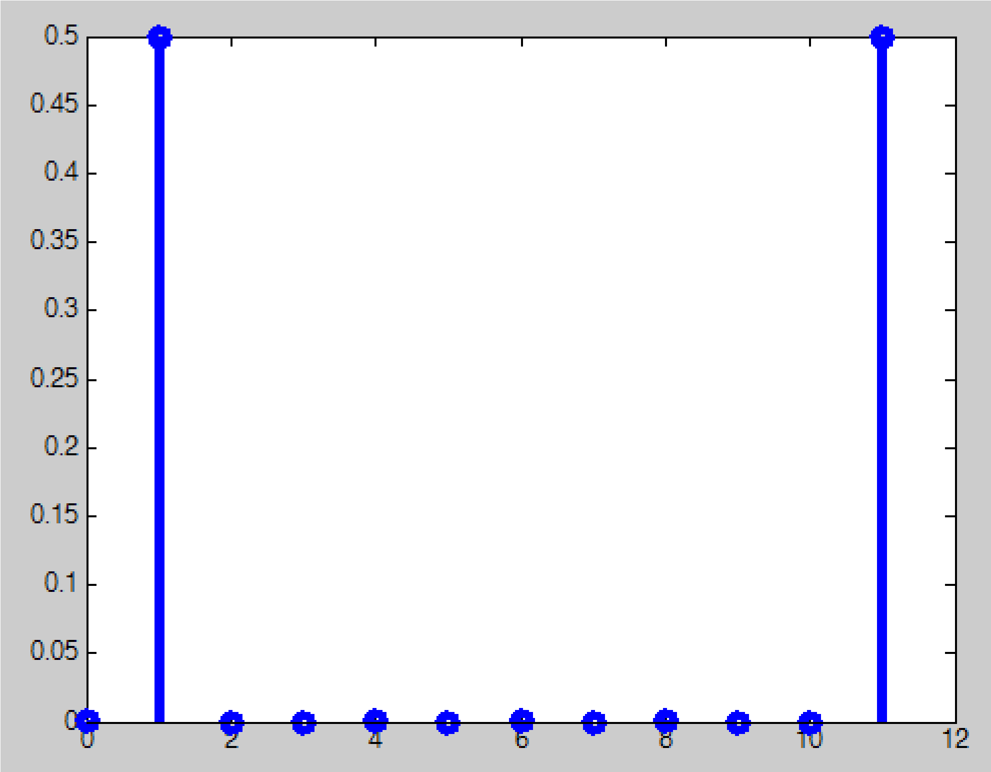
(Note: other sources use $|F_k|^2$ as power spectrum.)

Example Cosine (Notes, Example 5.1)

Last time, we saw a (rather artificial) example, $\cos\left(\frac{2\pi n}{N}\right)$.



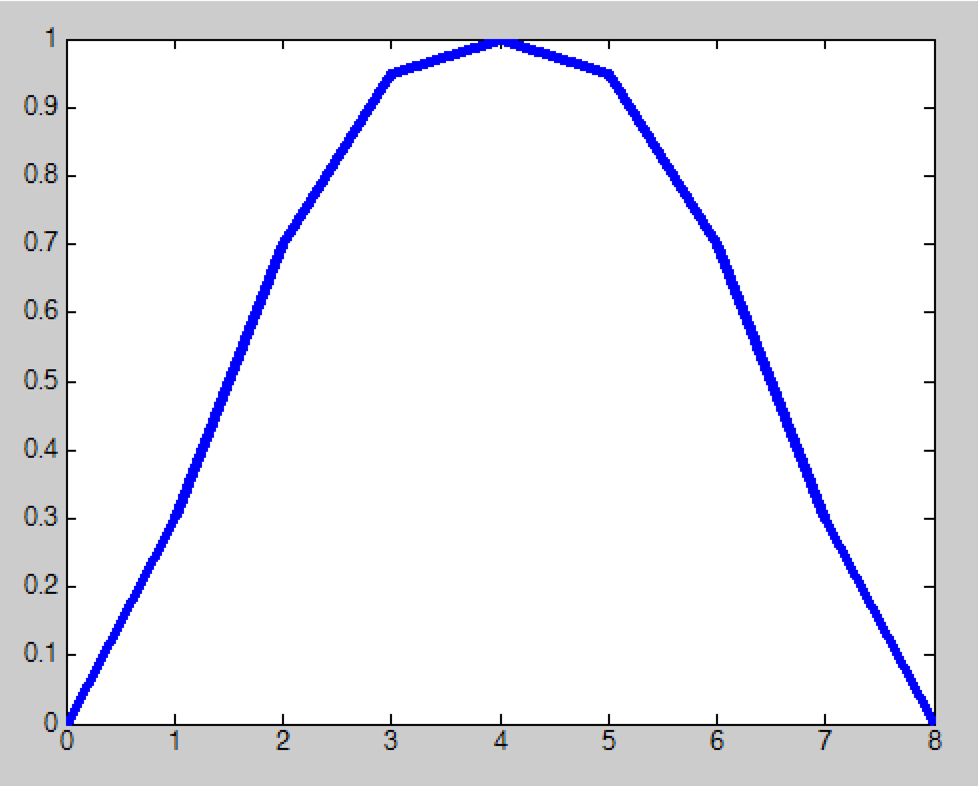
f_n



F_k

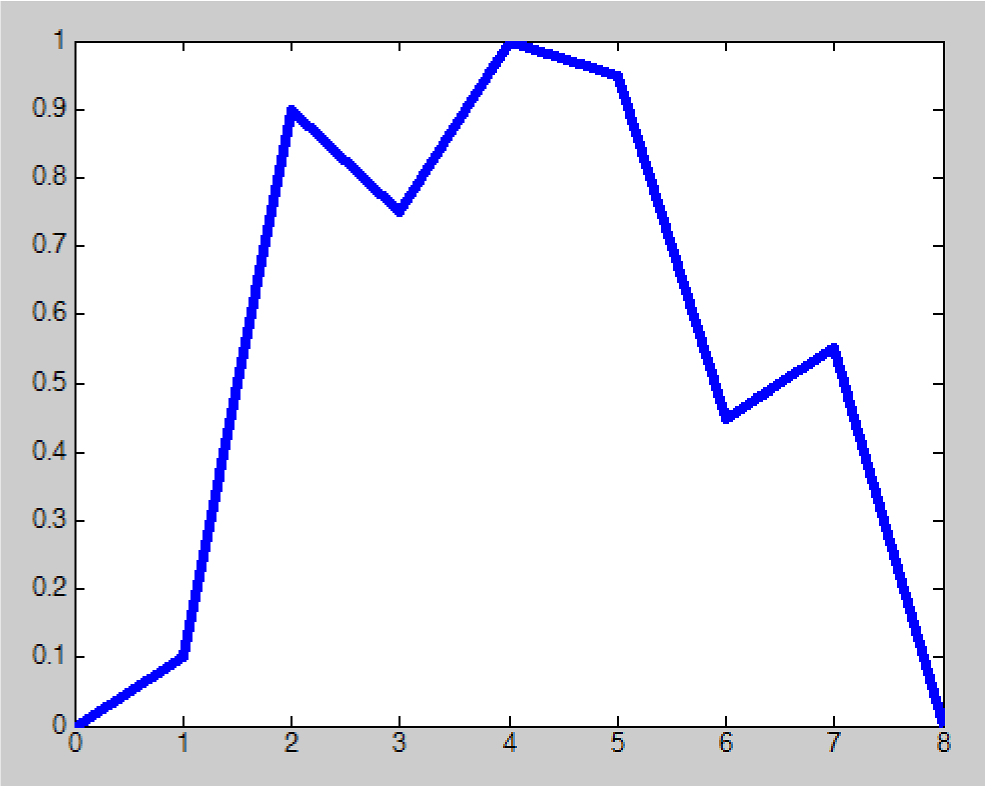
Discrete Data Examples

What if our data is less artificial and more irregular? (N=9 here.)



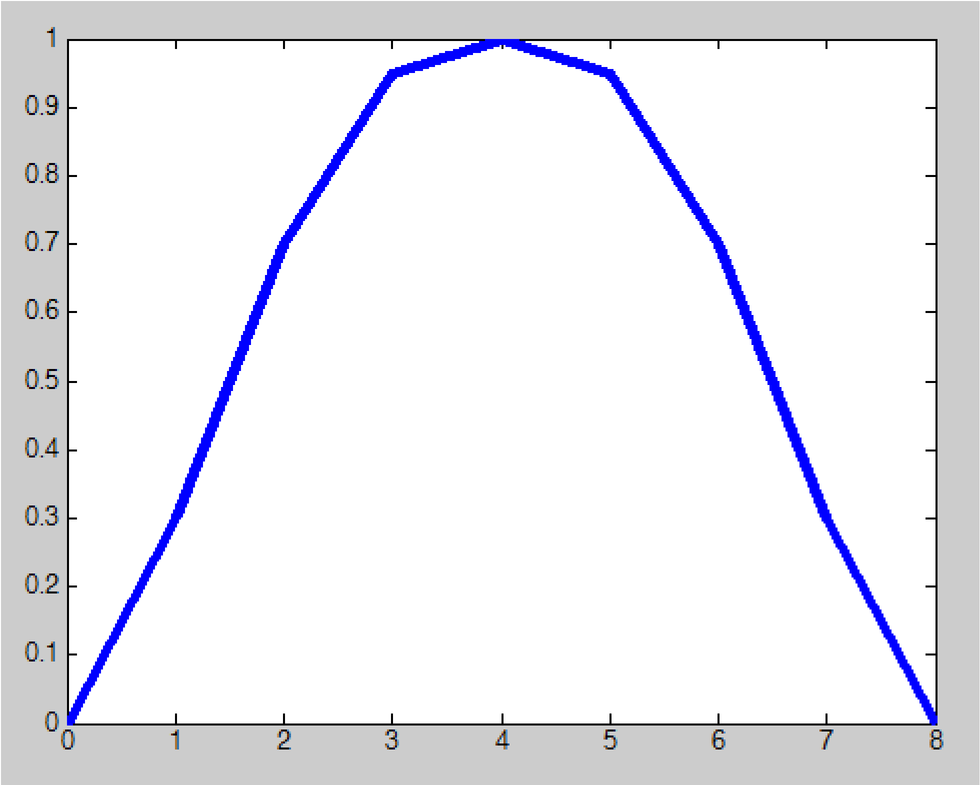
A smooth hump - data.

or

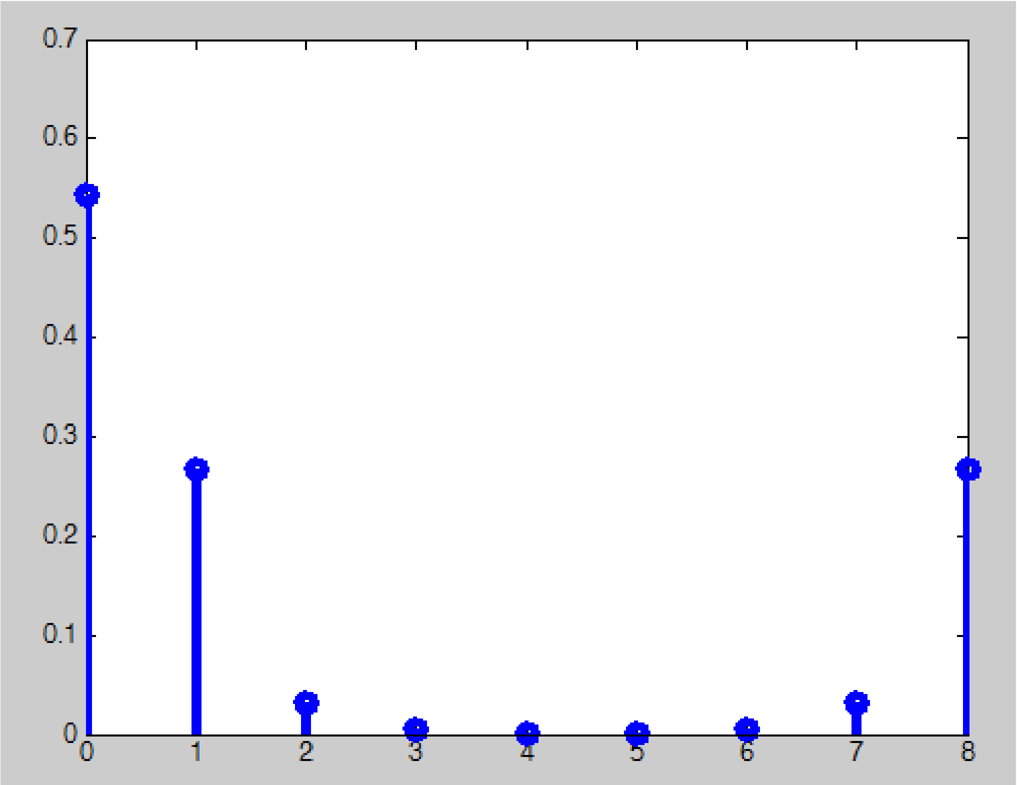


A rough hump - data.

Discrete Data – A Smooth Hump

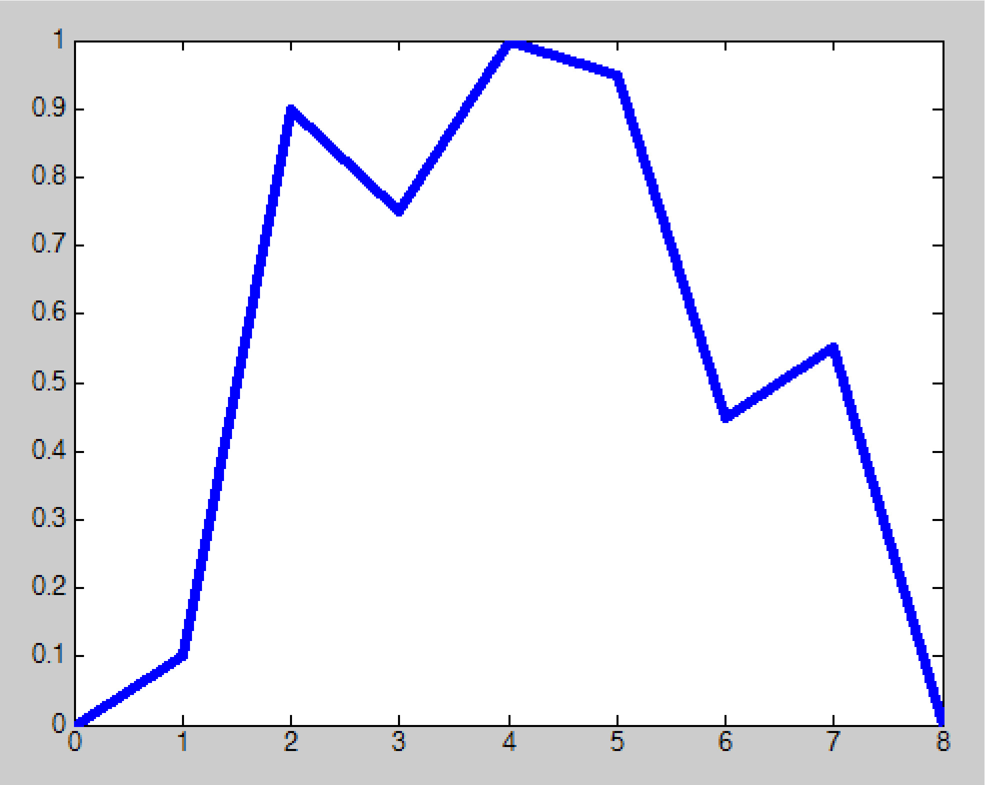


Data

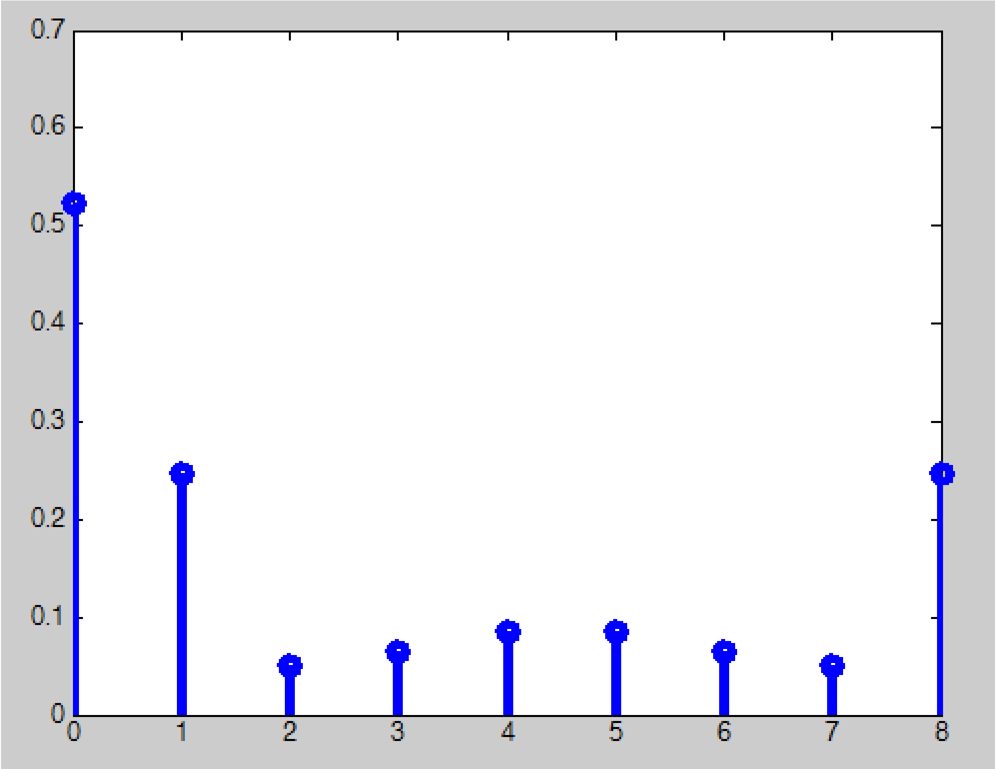


Fourier coefficients.

Discrete Data – A Rough Hump



Data



Fourier coefficients.

Discrete Data - Observations

- Coefficient F_0 is always the average of data values, $F_0 = \frac{1}{N}\sum_{n=0}^{N-1} f_n$. (Sometimes called the direct current or DC.)

- The smooth hump had one dominant frequency/wave; therefore one coefficient pair, F_1 and F_8, had large magnitude.

- The rough hump had more irregularity, so more active (higher) frequencies. Main hump still dominates, so low freqs F_1 and F_8 remain largest.

- The power spectra (F_k plots) are symmetric, since the data was real.

A Matrix View of DFT

Our usual DFT is $F_k = \frac{1}{N}\sum_{n=0}^{N-1} f_n W^{-nk}$.

Letting F and f be vectors of Fourier coefficients and data, respectively, we can write DFT as a matrix transformation, $F = Mf$, where M is a matrix where whose k^{th} column is: $\frac{1}{N}\begin{bmatrix} 1 \\ W^{-k} \\ W^{-2k} \\ \dots \\ W^{-(N-1)k} \end{bmatrix}$.

A Matrix View of DFT

e.g., For $N = 4$, we have $F_k = \frac{1}{4}\sum_{n=0}^{4-1} f_n W^{-nk}$ which in matrix form is

$$\underbrace{\frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ W^0 & W^{-1} & W^{-2} & W^{-3} \\ W^0 & W^{-2} & W^{-4} & W^{-6} \\ W^0 & W^{-3} & W^{-6} & W^{-9} \end{bmatrix}}_{M} \underbrace{\begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{bmatrix}}_{f} = \underbrace{\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix}}_{F}$$

$$M \qquad f = F$$

A Matrix View of IDFT

Our orthogonality identity leads to $\overline{M^T}M = \frac{1}{N}I$, where I is the identity matrix. For $N = 4$:

$$\underbrace{\frac{1}{4}\begin{bmatrix} 1 & W^0 & W^0 & W^0 \\ 1 & W^1 & W^2 & W^3 \\ 1 & W^2 & W^4 & W^6 \\ 1 & W^3 & W^6 & W^9 \end{bmatrix}}_{\overline{M^T}} \underbrace{\frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ W^0 & W^{-1} & W^{-2} & W^{-3} \\ W^0 & W^{-2} & W^{-4} & W^{-6} \\ W^0 & W^{-3} & W^{-6} & W^{-9} \end{bmatrix}}_{M} = \underbrace{\frac{1}{4}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\frac{1}{N}I}$$

Therefore $M^{-1} = N\overline{M^T}$.

The IDFT becomes $f = M^{-1}F = N\overline{M^T}F$.

A Matrix View of IDFT

e.g., For $N = 4$, we have IDFT $f_n = \sum_{k=0}^{3} F_k W^{nk}$ in matrix form:

$$\underbrace{\begin{bmatrix} 1 & W^0 & W^0 & W^0 \\ 1 & W^1 & W^2 & W^3 \\ 1 & W^2 & W^4 & W^6 \\ 1 & W^3 & W^6 & W^9 \end{bmatrix}}_{N\overline{M^T}} \underbrace{\begin{bmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \end{bmatrix}}_{F} = \underbrace{\begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \end{bmatrix}}_{f}$$

Summary: Matrix Form of DFT pair

DFT: $F = Mf$

Inverse DFT: $f = M^{-1}F = N\overline{M^T}F$

DFT So Far

- We derived the Discrete Fourier Transform (DFT) and its inverse, IDFT.
- We examined the DFT for a few data sets, to gain intuition.
- We identified some key properties of the DFT, and saw a matrix formulation of it.
- We'll next look at DFT's computational cost, and how to accelerate it!