THE UNIVERSITY OF ARIZONA

Lab 7

Data Acquisition, Aliasing, FFT, and the Optical Fourier Transform

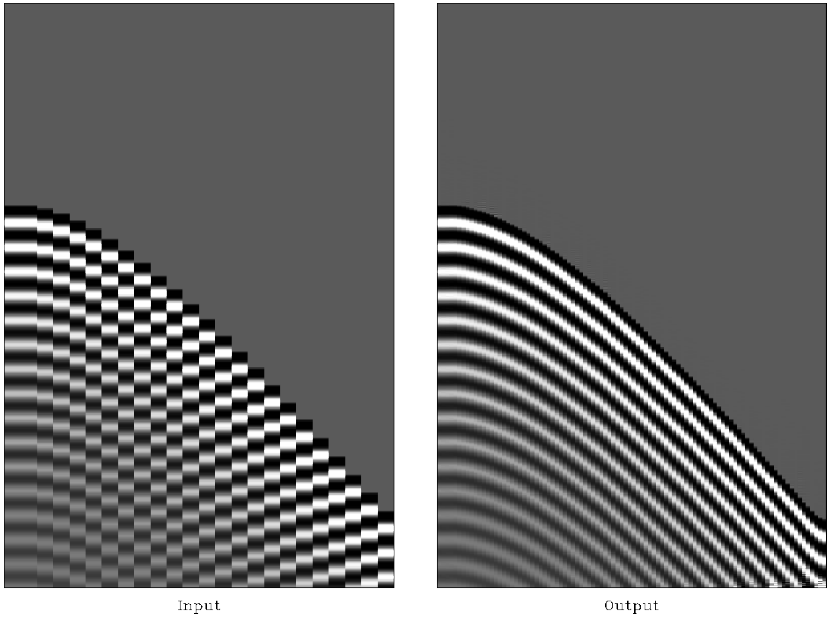
Wente Yin

February 26th, 2015

azengineering

Agenda

- DAQ
- DFTs
- DFTs in MATLAB
- Frequency Shifting
- Gibbs Phenomenon
- Aliasing
- Advice



Data Acquisition (DAQ)

- Two Sets of Inputs
  - Analogue (for…analogue)
  - Digital (for data transfer, digital sensors, etc.)
- Requires Proper Drivers for Function

Data Acquisition (DAQ, Cont'd)

Discrete Fourier Transforms

- Computers are Discrete in Nature
  - Discrete → Not continuous, requires individual input (samples)
  - Continuous functions would require infinite amount of RAM, which cannot be downloaded (non-countable infinity).

DFTs (Cont'd)

- Continuous Fourier Transform

$$F(\xi) = \int_{-\infty}^{\infty} f(x) \exp(-j2\pi x\xi)\, dx$$

$$f(x) = \int_{-\infty}^{\infty} F(\xi) \exp(j2\pi x\xi)\, dx$$

DFTs (Cont'd)

- Discrete Fourier Transform

$$F_k = \sum_{n=0}^{N-1} f_n \exp\left(-\frac{2\pi ink}{N}\right)$$

$$f_n = \frac{1}{N} \sum_{k=0}^{N-1} F_k \exp\left(\frac{2\pi ink}{N}\right)$$

DFTs (Cont'd)

- Functions Must be Discretely Sampled
  - Allows a simulated Fourier transform
  - DFT is cyclic (periodic), and multiple cycles are required for the function to appear in its appropriate dimensions
  - Several issues arise with MATLAB's implementation of the DFT

DFTs (Cont’d)

- DFT in MATLAB
  - *fft* (Single-Dimension), *fftn* (N-Dimension)
  - The *fft* function will carry out the DFT in a manner that places zero frequencies to the left (DC Bias, Single Dimension) or corners (Two-Dimension)
  - *fft* also limits DFT to a single cycle (appears more like a continuous FT), which is the source of problems like aliasing

DFTs (Cont'd)

- *fft* (Fast Fourier Transform)
  - Function makes use of symmetry of input to reduce computational complexity from N^2 to $N\log_2(N)$
  - As such, it is best to have inputs sized to powers of 2.

Rules of DFTs

- Inherent Rules of Discrete Fourier Transforms
  1. The entirety of the span of a FT is equivalent to the inverse of the sampling rate, centred if necessary.
  2. Said sampling rate also determines normalisation of the dependent axis, due to nature of the summation in DFTs
  3. Always sample by a power of 2
  4. It is recommended the final value of an even function be dropped when performing DFTs (cyclic nature)

Rules of DFTs (Cont'd)

1. The entirety of the span of a FT is equivalent to the inverse of the sampling rate, centred if necessary. Inverse is true.

$$Entirety\ of\ Frequency\ Spectrum = \frac{1}{\Delta x}$$

$$\Delta\xi = \frac{1}{ALL\ OF\ THE\ NORMAL\ SPACE}$$

Rules of DFTs (Cont'd)

2. Said sampling rate also determines normalisation of the dependent axis, due to the nature of the summation in DFTs

$$Normalisation\ Factor = \Delta x$$

Rules of DFTs (Cont'd)

3. Always sample by a power of 2
    - MATLAB adores symmetry and improves calculation speed for large arrays
    - MATLAB also has crazy, mysterious, trademarked, copyrighted, patented, and highly lucrative optimisation techniques
    - Guideline for efficiency and good form

Rules of DFTs (Cont'd)

- Effects of Optimisation Apparent with *tic/toc*

```
Average Time for Powers of 2
   1.0e-03 *

    0.0304    0.0295    0.0384    0.1844

Average Time for Powers of 2 Minus 1
   1.0e-03 *

    0.0257    0.0327    0.0451    0.5386
```

Rules of DFTs (Cont'd)

4. It is recommended the final value of an even function be dropped when performing DFTs (cyclic nature)
    - Final point is unnecessary and potentially problematic in repeating signals

DFTs in MATLAB

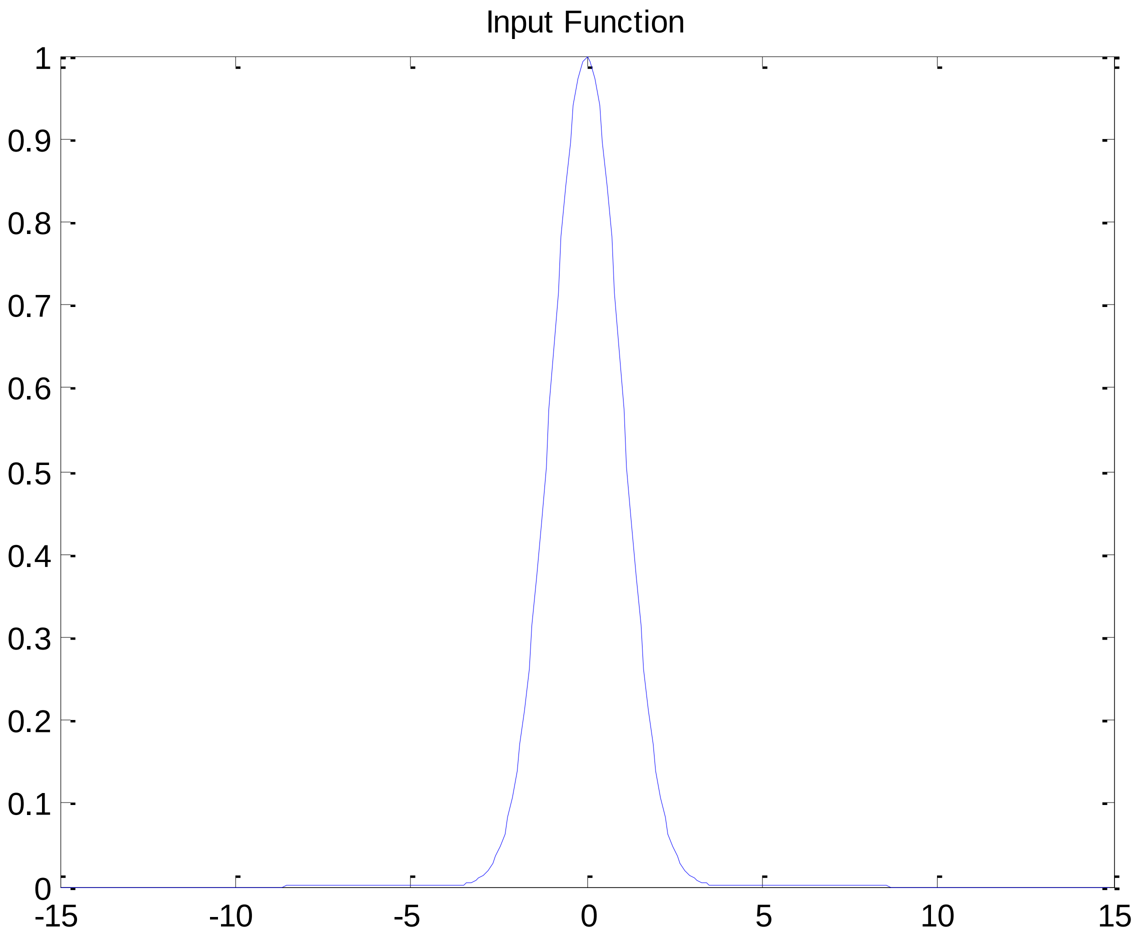
- Plot a Gaussian Function

$$x = linspace\,(-15{,}15{,}257);$$

$$x = x(1{:}\,256);$$

$$y = \exp\left(-\frac{x^2}{2}\right);$$

DFTs in MATLAB (Cont'd)

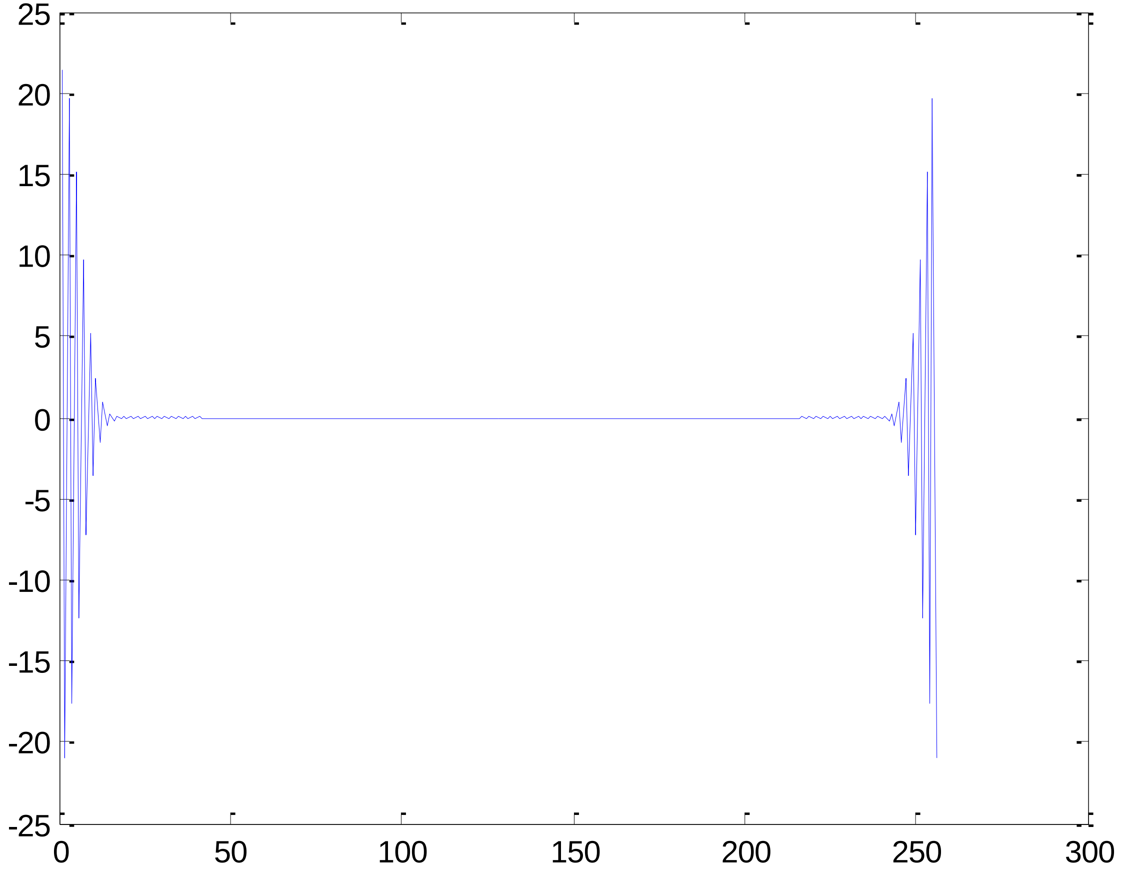


DFTs in MATLAB (Cont'd)

- Use the *fft* Function, Plot Reals

$$Y = fft(y);$$
$$plot\left(real(Y)\right);$$

DFTs in MATLAB (Cont'd)



DFTs in MATLAB (Cont'd)

- Previous Figure Obviously Incorrect
  - Fourier transform of a Gaussian function is always another Gaussian…which it was obviously not.
  - Strange figure is because of the DC bias of the *fft* function
  - We must place zero frequency in middle

Frequency Shifting

- Shifting Frequencies
  - *fftshift* (A)
  - Places zero frequency between negative and positive frequencies, essentially “flipping” the result

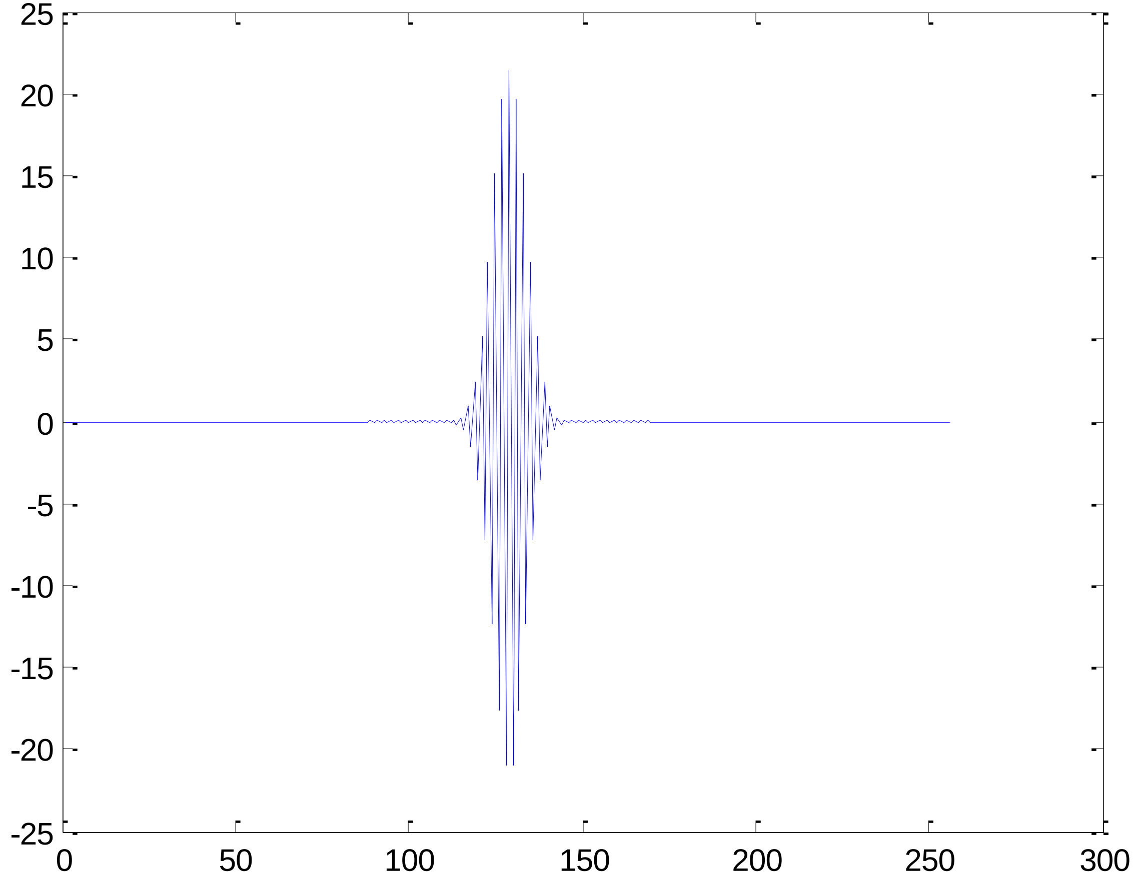
Frequency Shifting (Cont'd)

- Use Function on Previous Result, Plot Reals

$$Y1 = fftshift(Y);$$

$$plot\big(real(Y1)\big);$$

Frequency Shifting (Cont'd)



Frequency Shifting (Cont'd)

- Still Horribly Wrong
  - Oscillating amplitudes indicative of phase issues risen from input Gaussian before *fft*
  - Can be fixed by applying *fftshift* to the input as well
  - Allows for function to be represented properly in time, which is the preferred input of the *fft* function, fixing phase issues

Frequency Shifting (Cont'd)

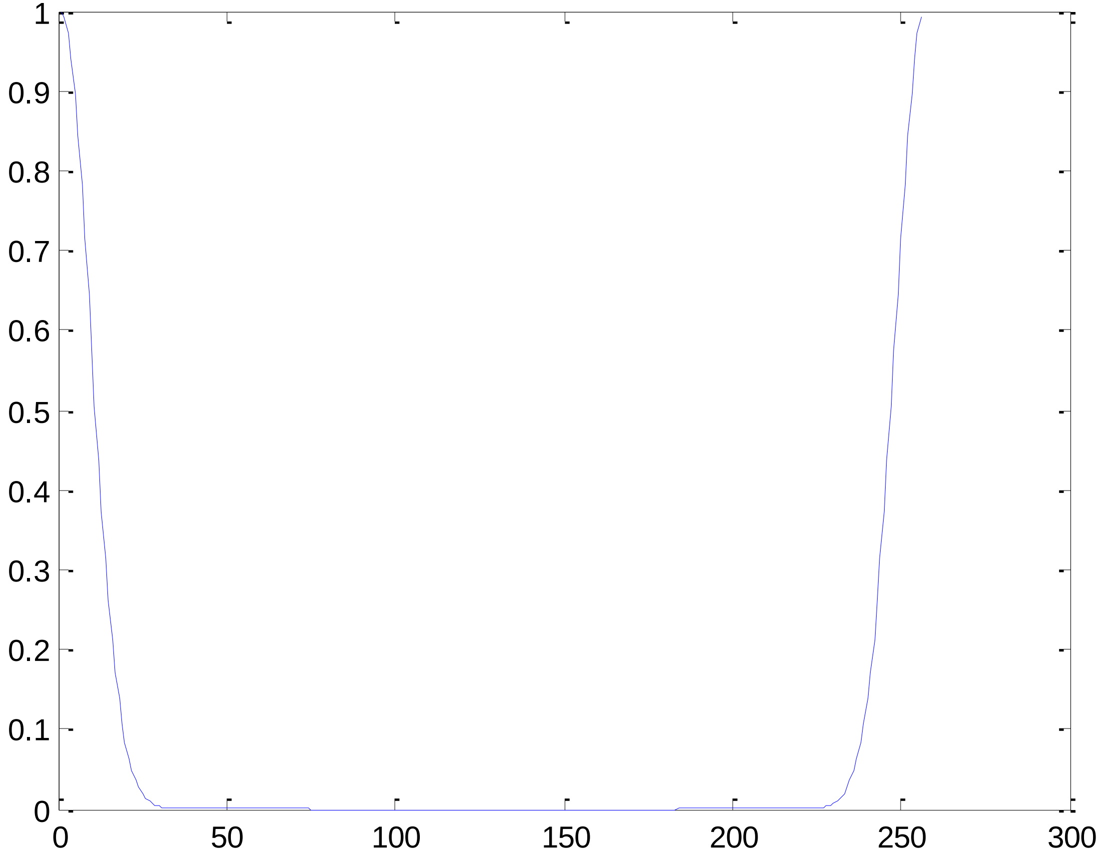
- Apply *fftshift* to Original Gaussian

$$y1 = fftshift\,(y);$$
$$plot\,(y1);$$

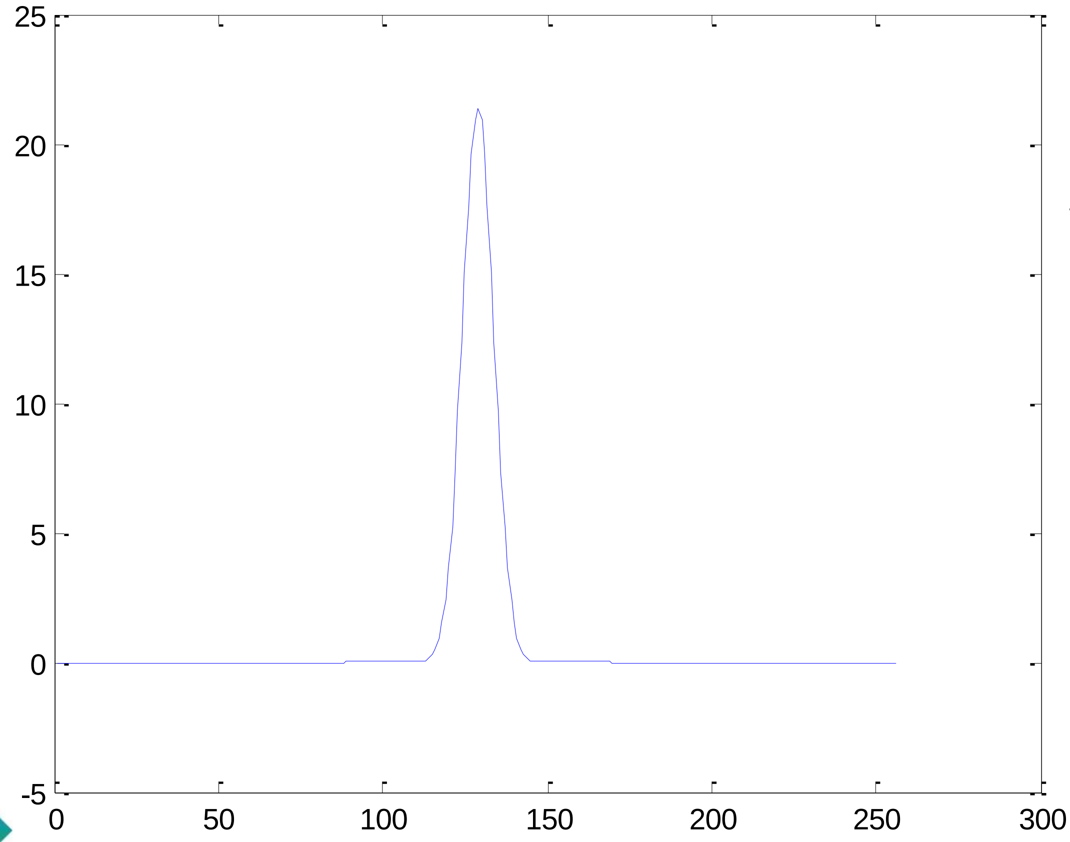
- Now apply *fft*, Shift to Centre

$$Y2 = fftshift\,(fft(y1);$$
$$plot\,(real(Y2));$$

Frequency Shifting (Cont'd)



Frequency Shifting (Cont'd)



We know that the FT of a Gaussian must be a scaled Gaussian, but what does this plot mean?

Scaling

- Axes Currently Convey no Useful Information
  - x-axis indicates relative position of value plotted
  - y-axis requires normalisation
  - We must understand the nature of normal and frequency space

Scaling (Cont'd)

- Applying Scaling to Result Results in Proper Results
  - The correct way to do an *fft*

$$x = linspace(-15,15,257);$$

$$x = x(1:256);$$

$$delx = x(2) - x(1);$$

$$freq = linspace\,(-1/(2*delx), 1/(2*delx),257);$$

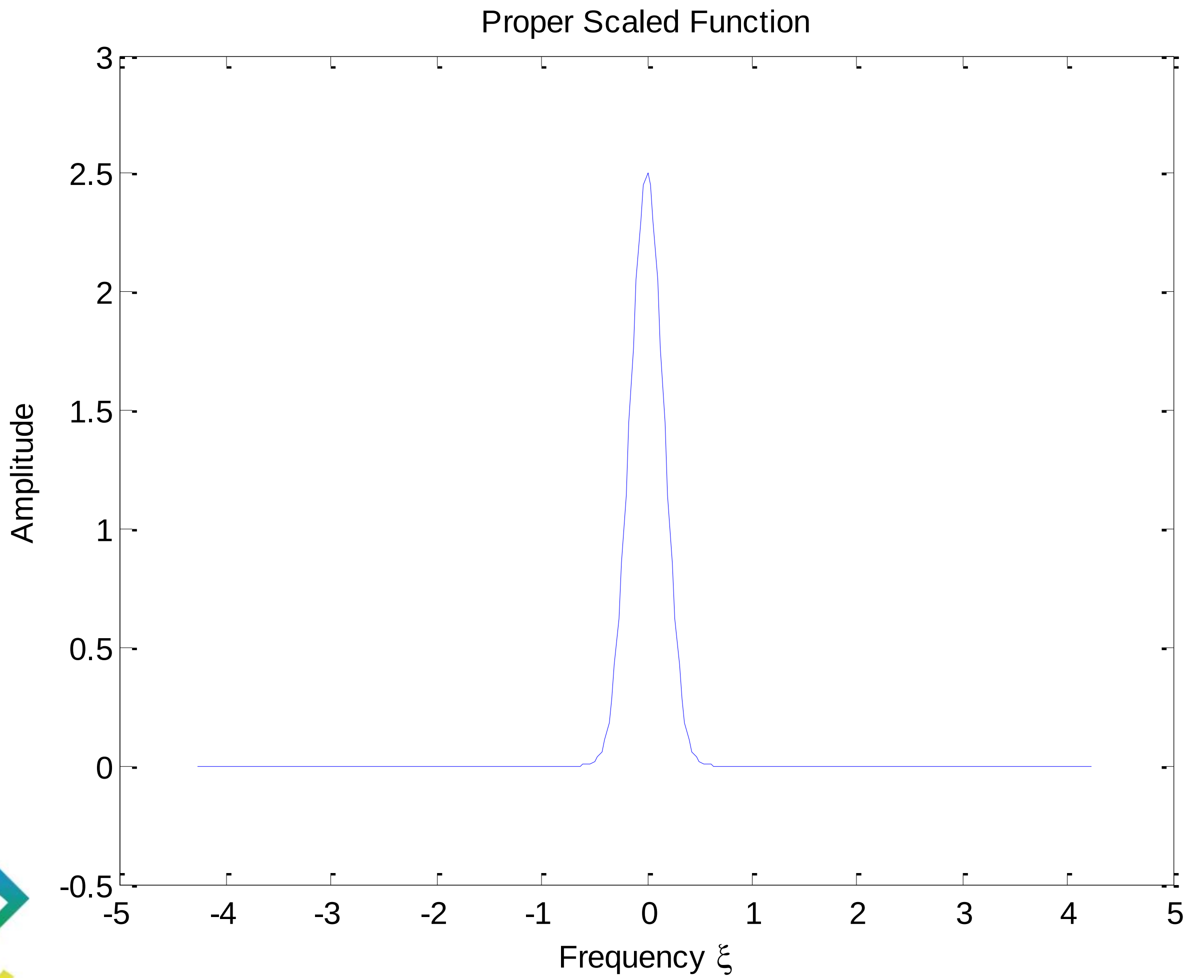
$$freq = freq\,(1:256);$$

$$y1 = fftshift\,(y);$$

$$Y2 = fftshift\,(fft(y1)\ * delx;$$

$$plot\,(freq, real(Y2));$$

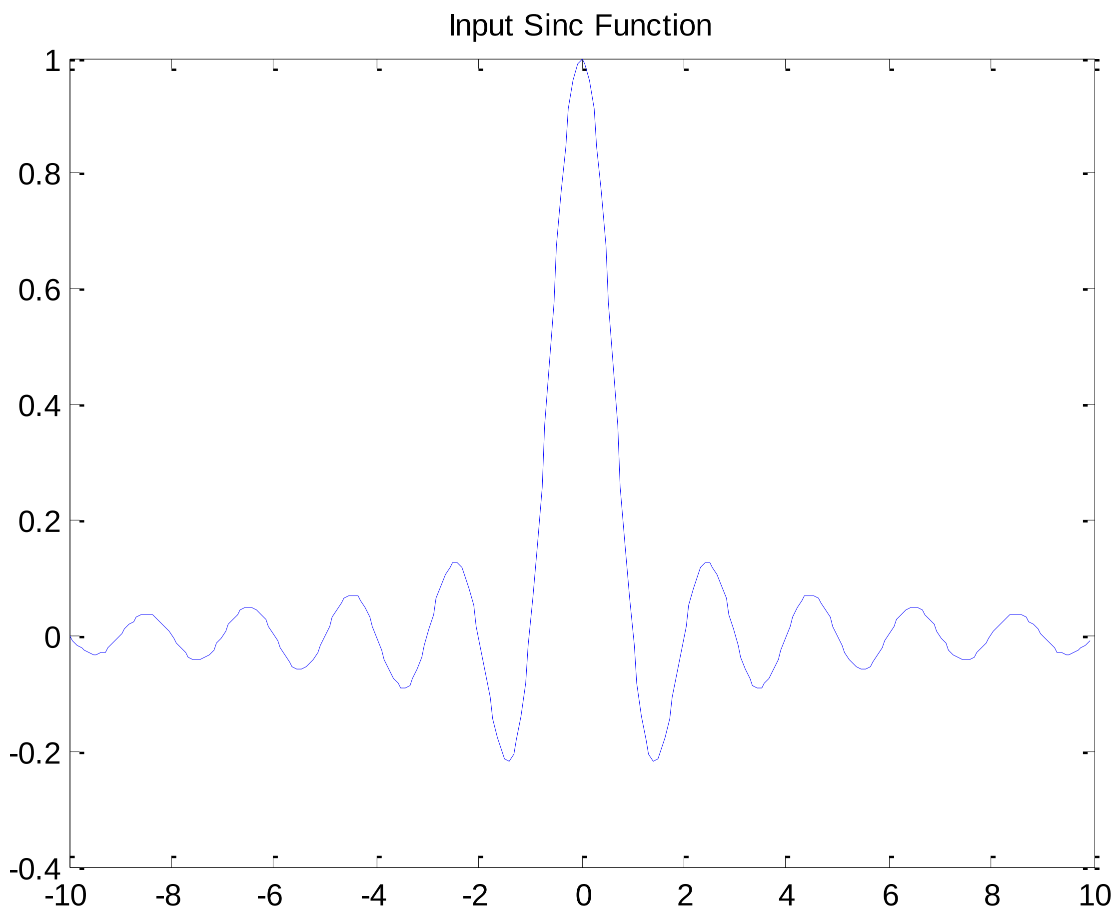
Rules of DFTs (Cont'd)



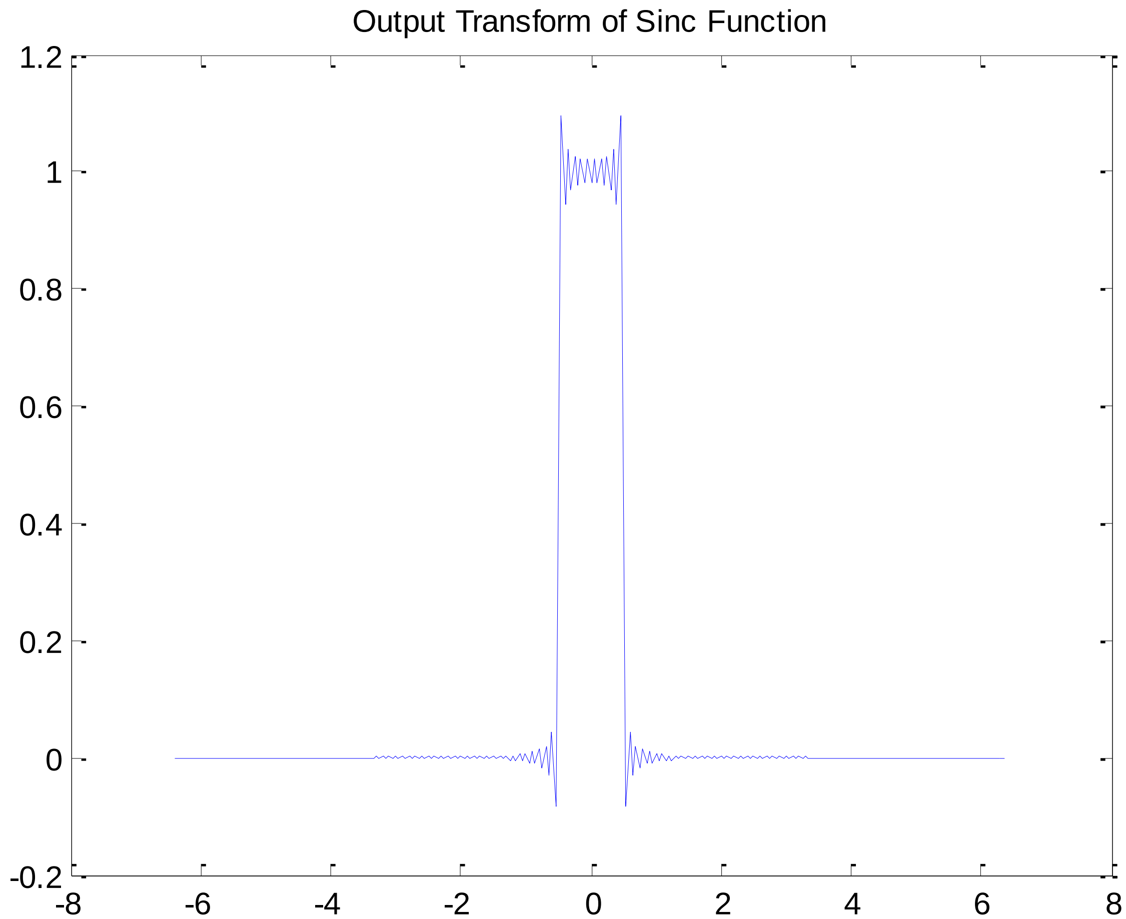
Gibbs Phenomenon

- Occurs Near Sharp Edge in Fourier Space
  - “Jump” apparent in Fourier transform
  - As number of samples increase, “jump” does not go away, merely approaches infinitely closer to the discontinuity
  - Always hovers at 1.09 magnitude of max
  - Occurs as the DFT is carried out with a limited number of frequencies of sines and cosines, which are incapable of representing the discontinuity

Gibbs Phenomenon (Cont'd)



Gibbs Phenomenon (Cont'd)



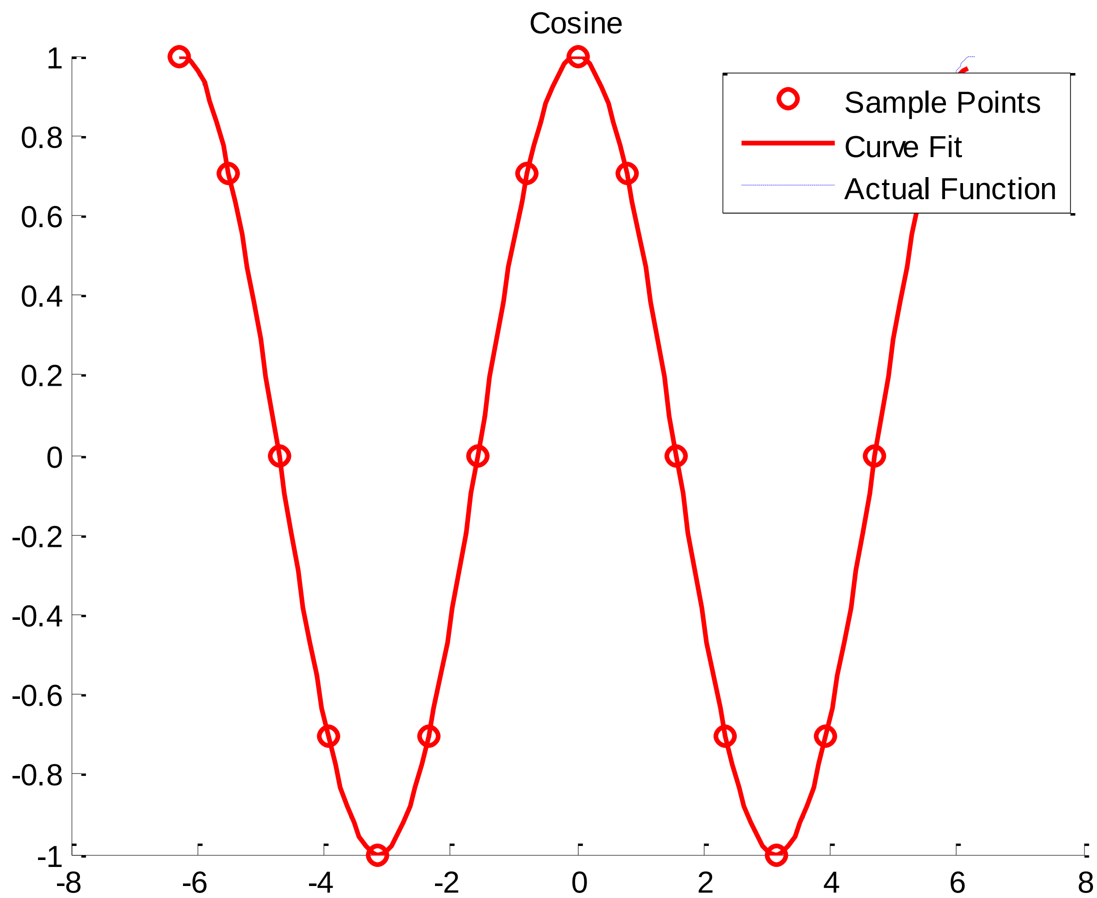
Aliasing

- Sampling Frequency of DFT > Nyquist Frequency
  - Nyquist frequency = 2x Sampling Frequency of Function
  - Nyquist frequency is regarded as minimum sampling frequency required to reconstruct a function
  - If sampling frequency of DFT is less than ideal, an identical function of lower frequency may be observed

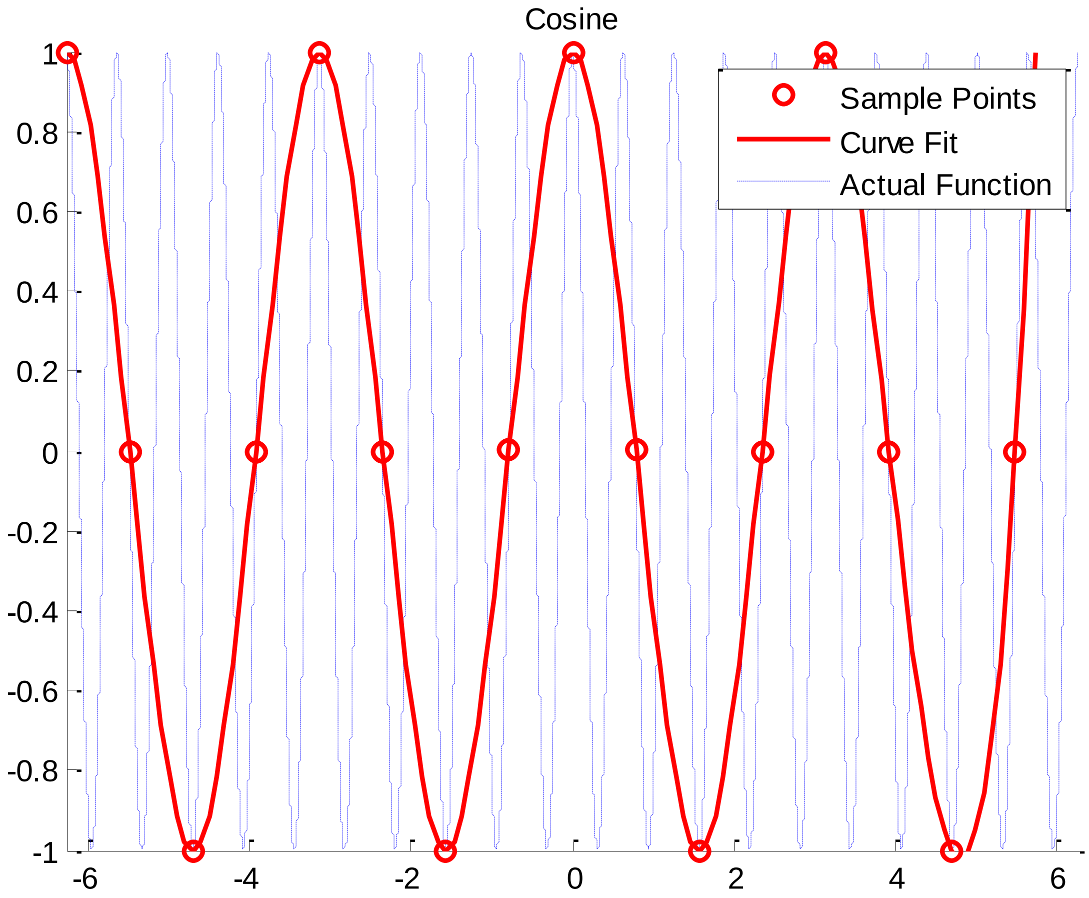
Aliasing (Cont'd)

- Demonstration
  - Cosine was sampled and fitted with a curve fit to give better appearance (do this in radiometry or any class with Barrett or Dubin and burn like the unwashed heathen you are)
  - Frequency of the cosine function was then increased, but sampling for the DFT was not (this is dangerous in optics)
  - Perceived frequency in frequency space was then plotted

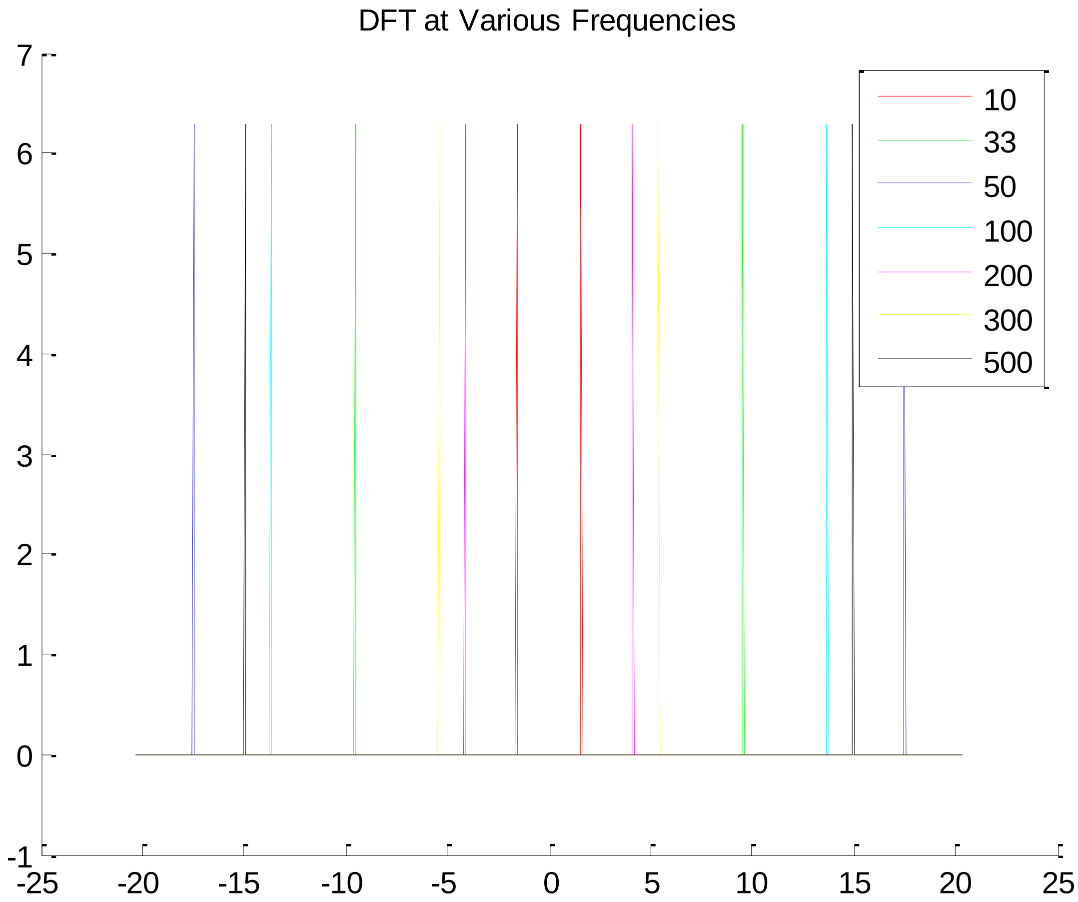
Aliasing (Cont'd)



Aliasing (Cont'd)



Aliasing (Cont'd)



Advice

- Low-Pass Filter → Zero Higher Frequencies
- High-Pass Filter → Zero Lower Frequencies
- Reconstruct to See Results: Inverse DFT
- Use "*help*" Command in MATLAB to Understand "*interp*" and "*interp2*" or learn to zero-pad for final section of lab (*padarray*)

Advice (Cont'd)

- Method to Import Image into MATLAB
- Think About Effects of Filtering
- Use a Black-White Image, or Greyscale a Coloured One
- Consider the Effects of Multiple Dimensions on Scaling

Quiz

- Using what you've learned of DFTs, explain the effects of adding a series of zeros in front of and after your function before attempting the DFT in MATLAB