CS170A — Mathematical Modeling & Methods for Computer Science — Fall 2014
HW#3 *Fourier Transform and Friends*
Due: 11:55pm Sunday November 30, 2014

D. Stott Parker, Luis Angel Larios Cardenas
stott@cs.ucla.edu, langel@cs.ucla.edu

Using CCLE, please turn in your Matlab code and resulting output (enough to show that the program is working) as a document, such as a MSWord file or PDF document.

1. **Comet Music (35 points)**
   The Matlab commands

   ```
   audiofile = 'SingingComet.mp3';
   [comet,Fs] = audioread(audiofile);
   y = comet(:,1);  % first track of the stereo clip
   n = length(y)
   ```

   create a 3854592×2 matrix `comet` that contains a stereo sound track (audio sequence) for a 'song' induced by the magnetic field of the comet 67P/Churyumov-Gerasimenko on which the probe of the Rosetta Plasma Consortium (RPC) landed on November 14. (If you want to get a feel for it there is a an audio track playable on the web).

   After this, executing `sound(comet, Fs)` will play the sound track at the sampling frequency `Fs`. There are several questions below about `y`, which is the first of the 2 stereo audio tracks.

   (a) What is `Fs`? How does this compare with CD-quality sound?

   (b) The length of `y` is $n = 3854592$. Find the integer factors of n by using the Matlab function `factor()`. Also find the integer factors of $(n - 11)$.

   (c) Using the `profile` command in Matlab, find out how much cpu time it takes to compute `fft(y(1:n)));` and `fft(y(1:(n-11)));` — and explain the difference in timing.

   (d) For this assignment we define the *power* of a complex value z to be its complex absolute value: $\text{power}(z) = |z| = \sqrt{z\,\bar{z}}$.

   The *frequency spectrum* is a sequence of frequency values that has the same length n as the power values. The lowest frequency value is 0, and the highest frequency value is `Fs`. The frequency spectrum for `y` should be a linearly spaced between these two limits.

   Plot the power of the Fourier transform, using the frequency spectrum values for the x-coordinate and power values for the y-coordinate. Your plot must show correctly how the frequency axis is scaled.

   (e) Do the same plot, but only for the 'first half' of the power (since the second half is a kind of mirror image when `y` is real). Here by *'first half'* of a sequence $s = [s_0, \ldots, s_{n-1}]$ we mean $[s_1, \ldots, s_m]$ where $m = \lfloor n/2 \rfloor$. For example, the 'first half' of $[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]$ is $[1, 2, 3, 4, 5]$.

   The highest frequency of this half is called the *Nyquist frequency*. What is its value?

   (f) Write a Matlab function `[spike_freq,spike_power] = top_spike(frequency_values,power_values)` that finds the highest value in a music power spectrum `power_values`, which have the same length as the corresponding interval of (consecutive) `frequency_values`. Using this function, by picking five appropriate intervals of frequencies (excluding zero), find the frequencies of the 'top 4 spikes' – i.e., four frequencies at which the power spectrum has the highest power values.

   (Hint: if you are confused about how to convert subscripts to frequency values, the Matlab `sunspots` demo may help; try `playshow sunspots`.)

   (g) Write another Matlab function `notes = piano_notes(filename)` that reads a file using `audioread()`, takes a prefix of the music whose size is a large power of 2, and converts the resulting vector of power values to musical notes using the *piano scale* — which assumes that the k-th note in the scale has frequency c^k times the frequency of C, where $c = 2^{1/12} = 1.059463\ldots$ and all frequencies of C are powers of two as in the table below.

   Use your function to find the notes for the top 4 note frequencies in the clip `mystery.m4a`. What 'intervals' in the table above appear among these notes?

note	frequency (Hz)	frequency/C	interval
Middle C	256.0	$c^0 = 1.000$	
C#	271.2	$c^1 = 1.059$	minor 2nd
D	287.3	$c^2 = 1.122$	major 2nd
D#	304.4	$c^3 = 1.189$	minor 3rd
E	322.5	$c^4 = 1.260$	major 3rd
F	341.7	$c^5 = 1.334$	4th
F#	362.0	$c^6 = 1.414$	diminished 5th
G	383.6	$c^7 = 1.498$	5th
G#	406.4	$c^8 = 1.587$	minor 6th
A	430.5	$c^9 = 1.682$	major 6th
A#	456.1	$c^{10} = 1.782$	minor 7th
B	483.3	$c^{11} = 1.888$	major 7th
High C	512.0	$c^{12} = 2.000$	octave/8th

2. **Comet Distributions (15 points)**
We can plot the histogram of 'comet song' values, and also get the counts in each bin of the histogram, with commands like these:

```
nbins = 10000
data = comet(:,1);
hist( data, nbins );              %  plot histogram
bincounts = hist( data, nbins );  %  get counts for the histogram bins
```

Question: what distribution is this?

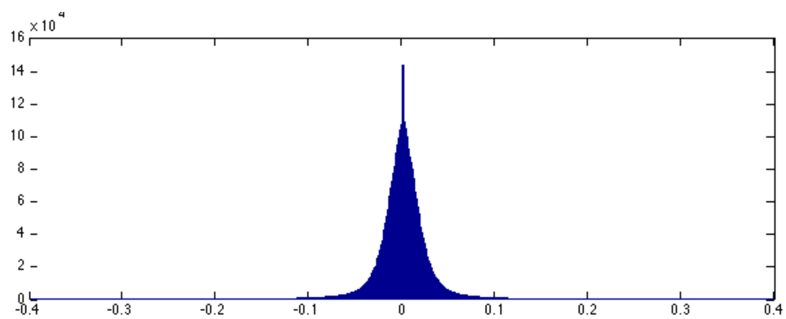

Figure 1: Distribution of the comet song values. Is this a normal distribution? Cauchy? Student's t?? It is centered at zero, that much seems clear, but — is it normal?

Try to fit the normal (gaussian) distribution and the Cauchy distribution to the data, assuming it is centered at zero (i.e., the mean value is zero). Which fits better?

```
%% First:  superimpose a cauchy fit on the histogram
normal = @(x,sigma) 1 / (sqrt(2 * pi) * sigma)  *  exp( -1/2 * (x/sigma) .^ 2 )
normal = @(x,sigma)  pdf( 'Normal', x, 0, sigma )         %%  we can use the general builtin function pdf()
sigma = std(data);
xvalues = linspace(-0.8, 0.8, 10000);
normal_yvalues = normal(xvalues, sigma);
normal_area = sum( normal_yvalues );
histogram_area = sum( bincounts );
hold on;   axis( [-0.4, 0.4, 0.0, 3e4] )
plot( xvalues, normal_yvalues / normal_area * histogram_area, 'r-' )  %% superimpose a normal fit on the histogram

%% Second:  superimpose a cauchy fit on the histogram
cauchy = @(x,gamma) gamma ./ ( pi * (gamma^2 + x.^2) )
```

3. **Stock Analysis (30 points)** From the site http://finance.yahoo.com/q you can download historical stock quotes on most U.S. securities. For example, from http://finance.yahoo.com/q/hp?s=TSLA you

can download the quotes for Tesla, and save it to a file with a name like `TSLA.csv`. Similarly you can also get quotes for Tableau software, a data visualization company, with http://finance.yahoo.com/q/hp?s=DATA — and save them in `DATA.csv`.

Download the complete history of prices for a symbol of your choice that has a fairly long history. Your history should be at least several years long; longer is better. You can read in your stock history using the Matlab function `read_stock.m` provided with the other files for this assignment; try e.g.:

```
[time quotes] = read_stock('DATA.csv');
n = length(quotes)
power_spectrum = abs(fft(quotes)).^2
frequencies = linspace(0, 1.0, n)
  %% 1.0 <=> stock sampling frequency is once per day       <=> units are in days
plot(frequencies(2:floor(n/2)), power_spectrum(2:floor(n/2))) % ignore 1st coeff
  %% zoom in and inspect the power spectrum at the especified frequencies
frequencies = linspace(0,  252, length(power));
  %% 252 <=> stock sampling frequency is 252 times per year <=> units are in years
figure; plot( frequencies, power )
```

Compute the Fourier transform of this history. Remember that stocks are sampled at a frequency of 1/day (once per day), but there are only five business days per week, and with holidays and other exceptional situations altogether prices are sampled only on about 252 days per year. In other words, assume a sampling frequency of 1/day (once per day), and that 1/(252 days) = 1/year. (Remember that the far right value in the frequency range should be the sampling frequency, and for stocks this is once per day = 1/day.)

In other words, please truncate the stock's history to be a multiple years in length; and if possible, make it a multiple of 4 years with 1009 days per 4 years. This can matter since the stock market is extremely sensitive to quarterly and annual reports.

(a) First, notice that the the power spectrum of many stocks has spikes at frequencies near `0.015873`/day or `0.007937`/day. These frequencies are in units of 1/day. What annual frequencies (with 252 days/year) do these correspond to?

(b) The Matlab command `s = urlread('http://ichart.finance.yahoo.com/table.csv?s=TSLA');` gets the stock history for TSLA (Tesla) as a string. If you store it as a .csv file (e.g., by using basic Matlab I/O functions: fopen(), fwrite(), fclose()), and use `read_stock()` to read it, you can find the largest power spectrum spike at a low frequency. What annual frequency does it correspond to?

(c) Write a function `find_freq` that takes the symbol for any stock (like 'DATA' or 'TSLA'), then uses `urlread()` to download its price history, saves the history to a file and uses `read_stock()` to read it in, then uses something like the `top_spike()` function from the first part of this homework to find significant annual frequencies. Show the result of your function for DATA and for TSLA.

4. **A Simple 'Fake Photo' Detector (20 points)**
A simple way to check if a $m \times n$ RGB image has been faked (with Photoshop, say) is to use the `reshape` function to convert it into a $(m\,n) \times 3$ matrix, compute the 3 principal components of its 3×3 covariance matrix, project the reshaped image on the second principal component, and then reshape the result back to a $m \times n$ grayscale image. If this grayscale image has bright or dark spots, it is possible that the color distribution in that area (and perhaps also that part of the image) has been altered.

For example, Figure 2 shows the photo http://i.huffpost.com/gen/2205440/thumbs/o-NINA-PHAM-facebook.jpg of the now virus-free Ebola nurse Nina Pham — does it look photoshopped? Below it is the output from a sample detector (included as detector_output_for_o-NINA-PHAM-facebook.pdf in the homework zip file). Do any areas of the detector image seem particularly bright or dark?

As another example, in the publicity photo of Rosetta's Philae lander approaching the comet that is accessible from http://www.esa.int/spaceinimages/Images/2014/11/Welcome_to_a_comet, the landing gear-like assembly in the left of the image appear quite white, so it's conceivable that it was added in.

(a) Write a program to analyze any image file. Show the results on the (apparently controversial) image at s1.ibtimes.com/sites/www.ibtimes.com/files/styles/v2_article_large/public/2014/07/15/kim-kardashian.jpg or any photo whose veracity has been challenged.

(b) Also find a pair of 'before and after Photoshopping' images (maybe with Google image search) for which the fake photo detector can spot Photoshopping.

Figure 2: Publicity image of Ebola nurse Nina Pham, and output of a sample fake photo detector. Image intensities have been inverted, so the lightest areas have become dark. Do any outlines of clothes look a bit odd? Or does anything else look strange?