Scientific Computing

Assignment 9: Spectral Analysis and Fourier Series

With a network of underwater microphones, U.S. government scientists are able to monitor ocean sounds from around the world. In addition to obvious military uses, like detecting nuclear submarines and weapons tests, these hydrophones pick up a wide range of natural phenomena, including whale songs, earthquakes, and iceberg calving.

In the summer of 1997, NOAA hydrophones in the South Pacific detected a truly strange signal, which came to be known as “The Bloop”. It lasted about ten seconds, and was too low-pitched for the human ear to hear. It can be made audible by speeding up the recorded signal by 16 times: listen to it here (http://www.pmel.noaa.gov/acoustics/sounds/bloop.wav) The oddest thing about the Bloop was how *loud* it was: far louder than any known animal, it was heard by listening stations 5000 km apart!

What created the Bloop? Was it an earthquake? Some military super-weapon? A very large sea creature we have yet to discover? The snores of the sleeping god Cthulhu in his underwater city of R’lyeh? Conspiracy theories abound, but NOAA believes it was an “icequake” -- a large iceberg or ice sheet breaking up. (Or maybe that’s what they *want* you to think…)

An important part of figuring out what the Bloop was is to do Fourier analysis to figure out what its dominant frequencies are.

Task 1: Collect and isolate the data.

Download the audio data file at the link above, or get it from OnCourse. Use the **`audioread()`** function to read the data and the "sample rate" *Fs,* which is the number of data points per second in the audio file.

The audio file contains a bunch of quiet noise that's not part of the actual "bloop". Using the command line, plot the data, listen to it using the **`sound()`** function, and extract a new list of data called y containing the "bloop" itself. Your y variable should have a length of $N = 4000$.

Question 1: Given the sampling rate Fs and the data length N, what is the duration of y in seconds?

Task 2: Create a Fourier decomposition function:

```
function [A, B, A0] = fourier(y)
```

where A(k) and B(k) are vectors containing the cosine and sine amplitudes of each frequency k, and the A_0 is stored separately. These should be calculated as follows:

$$A_k = \frac{2}{N} \sum_{i=1}^{N-1} y_i \cos(2\pi k i / N)$$

$$B_k = \frac{2}{N} \sum_{i=1}^{N-1} y_i \sin(2\pi k i / N)$$

with k = 1 to N/2, and

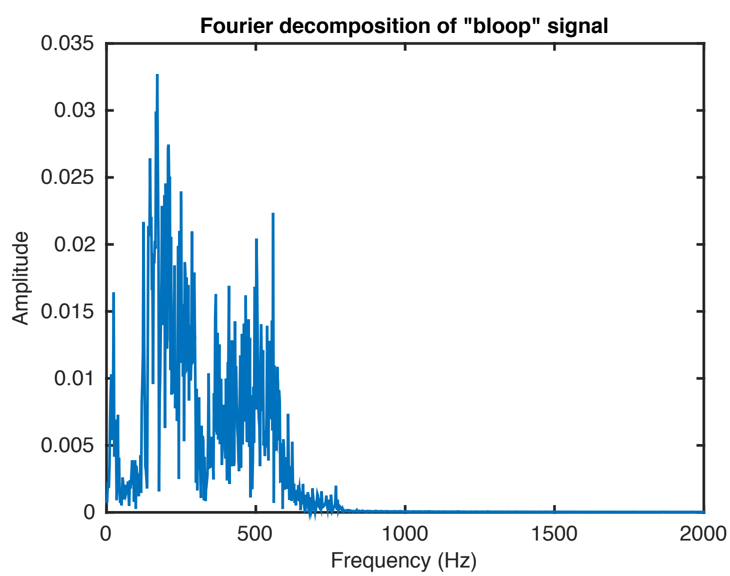
$$A_0 = \frac{1}{N} \sum_{i=1}^{N-1} y_i$$

Also write code to calculate the periodogram (the overall strength of each frequency):

$$p_k = \sqrt{A_k^2 + B_k^2}$$

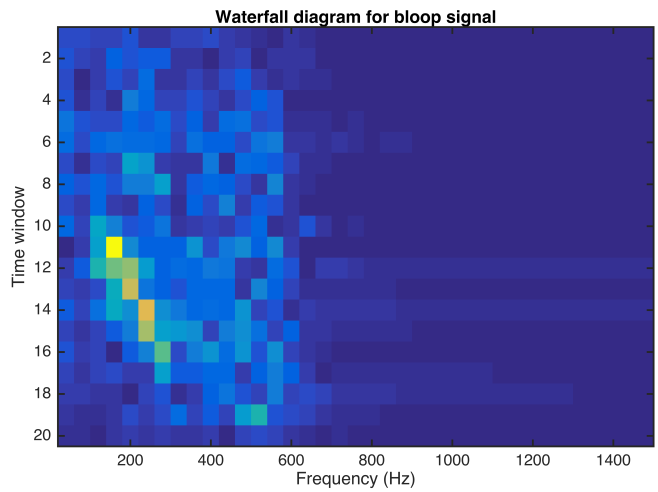
and the frequency f_k of each spectral component, in Hertz. (k is the number of cycles in the entire dataset y; you'll need to convert to get the number of cycles per <u>second</u>.)

Task 3: Calculate the periodogram of the bloop, and plot it as a function of frequency. You should get something like the figure at right.



The signal has a pretty broad range of frequencies, mostly because, as you can probably hear with **`sound()`**, it changes in pitch over time. Let's try to describe how the frequency shifts over time.

Task 4: Divide the data set *y* into 10 equally-sized pieces, and calculate the Fourier decomposition of *each*. Store the amplitudes in a 2-d table, so that A(n,k) is the cosine amplitude of frequency *k* in the nth time window.



Calculate a periodogram, and display it as a heat-map using the **`imagesc()`** function. This frequency-vs-time graph is sometimes called a "waterfall plot". It should look like the figure at right. Warm colors indicate a large amplitude. Note the diagonal line of pixels indicating an increase in dominant frequency over time.

Question 2: What is the dominant frequency of the "bloop" audio file, and how does it change over time?

Question 3: Remember that this audio file is sped up from the original hydrophone recording by a factor of 16 times. What was the dominant frequency of the original hydrophone signal, and how did it change over time?