Supplement to Chapter 4 of *The Science of Digital Media* – Digital Audio Representation

Programming Exercise – Digital Audio Representation > Fourier Transform[1]

Objectives:

Introduction:

The Fourier transform is used in digital audio in order to convert data from the time domain to the frequency domain. The formula for the discrete Fourier transform (DFT) is given below.

$$\boldsymbol{F}_n = \frac{1}{N}\sum_{k=0}^{N-1} \boldsymbol{f_k} \cos\left(\frac{2\pi nk}{N}\right) - i\, \boldsymbol{f_k} \sin\left(\frac{2\pi nk}{N}\right) =$$
$$\frac{1}{N}\sum_{k=0}^{N-1} \boldsymbol{f_k} e^{\frac{-i2\pi nk}{N}}$$

The algorithm operates on two arrays: ***f****(k)* and ***F****(n)*. The f array is the array of signal sample values (in the time domain) that is used as the input to the transform. The ***F*** array is the result of the transform. It is an array of complex numbers of the form x+y*i*. For our purposes, we will assume that both arrays have length *N*.

The Fourier transform is used in digital audio in order to produce the frequency components that compose a complex signal. This property can be seen in the inverse discrete Fourier transform, which is given below.

$$\boldsymbol{f}_k = \sum_{n=0}^{N-1}\left[\boldsymbol{a}_n \cos\left(\frac{2\pi nk}{N}\right) + \boldsymbol{b}_n \sin\left(\frac{2\pi nk}{N}\right)\right] =$$
$$\sum_{n=0}^{N-1} \boldsymbol{F}_n e^{\frac{i2\pi nk}{N}}$$

From this formula, you can see that the time-domain signal, represented by ***f***, can be formed by summing a series of sine and cosine waves multiplied by constants $\boldsymbol{a}_n$ and $\boldsymbol{b}_n$. The value of $\boldsymbol{F}_n$ is related to $\boldsymbol{a}_n$ and $\boldsymbol{b}_n$ by the equation

$$\boldsymbol{F_n} = \boldsymbol{a}_n + i\,\boldsymbol{b}_n$$

The Fourier transform is also used to calculate the frequency spectrum of a signal. The frequency spectrum shows the “amount” of each component frequency exists in the complex wave.

[1]This material is based on work supported by the National Science Foundation under Grant No. DUE-0340969. This worksheet was written by Todd Martin.

You will use this information to implement the discrete Fourier transform in the programming language of your choice. Your program will both calculate the values in the *F* array and convert these values into a frequency spectrum that can be used to view the component frequencies.

Instructions:

The Assignment
Implement the discrete Fourier transform in the programming language of your choice. Your program should calculate the values in the *F* array and thenconvert these values into a frequency spectrum showing the magnitude of the valid frequency components.

The program should

1. Read audio data from file
2. Compute the DFT on audio data and write the results to a file
3. Compute the frequency spectrum of audio data and write the results to a file
4. Exit the program

The user should be prompted for an input file name from which the audio data is read. The user should also be prompted for an output file name.

Remember that the result of the DFT is a complex number with real and imaginary components. You can use a two-dimensional array with dimensions of *n* x 2. Use the first set of *n* doubles to store the real data for each ***F****(n)* and the second to store the imaginary data.

Format the output data for the DFT to show the real and imaginary components in the form a + bi.

The magnitude of the frequency components can be calculated with this equation giving the magnitude of a complex number a + bi:

$$magnitude(a+bi) = \sqrt{a^2 + b^2}$$

The values for the frequency spectrum can be output as real numbers. Remember that for an N-element transform, only N/2 frequency components are valid, by the Nyquist theorem.

Before Writing the Program
You'll need to create a text file that contains an array of audio samples. You can do this by hand by opening a text editor, you can export raw audio data from an audio processing program, or you can generate the data in MATLAB.