Math 100 — MATLAB Project

For this project you will need an audio file such as the file `handel.mat` of the first few seconds of G. F. Handel's *Hallelujah Chorus* provided as part of MATLAB and to be found in the folder `matlabroot\toolbox\matlab\audiovideo` on any Windows PC with MATLAB installed. Here `matlabroot` is the folder in which MATLAB resides (something like `C:\Program Files\MATLAB\R2007b`).

You can load and listen to this file in MATLAB using the following commands:

```
load handel;
sound(y);    % since the sound is stored in the variable y in this .MAT file
```

You can also use any other audio file in the .wav format such as the recording of former IIT mathematics professor Karl Menger's voice to be found at `http://math.iit.edu/Menger/curve.wav` (which already contains a certain amount of noise).

You can load and listen to a .wav file in MATLAB using the following commands:

```
[y,Fs,nbits] = wavread('curve');
sound(y,Fs);
```

Here I'm assuming you downloaded `curve.wav` to your MATLAB working folder. With the command above you save the signal (which can be multi-channel) in the matrix `y`, the sample rate (in Hz) in the variable `Fs`, and the number of bits used per sample in the variable `nbits`. The default sample rate for MATLAB is 8192 Hz (which is why you need to include the correct – in this case 44 kHz – rate to play the `curve.wav` sample).

This project is about the application of the fast Fourier transform (FFT) to audio filtering. You should answer the following questions:

1. Read up on the FFT, and present a summary relevant to your project. Possible starting points for your reading are MATLAB's built-in help on the `fft`, Chapter 8 in Cleve Moler's online book (which you can get from `http://www.mathworks.com/moler/fourier.pdf`), or the Wikipedia entry `http://en.wikipedia.org/wiki/Fast_Fourier_transform`.

2. Choose a signal/audio file for your project and determine:
   - What is the size of the signal (how many samples, how many channels)?
   - What is the sample rate, and how many bits are used per sample?
   - How long is your sample (in seconds)?

3. Play your original audio signal in MATLAB and produce a plot of the first 100 samples of your signal as in Figure 8.1 of Moler's book.

4. Use (and modify) the script `dftfilter.m` to filter your audio signal with different choices of m (the number of Fourier modes kept) such as $n/2$, $n/4$ and $n/8$. Also plot the resulting filtered signals.

5. The script `wienerfilter.m` also filters an audio signal. It has been written so that your clean initial signal is artificially corrupted by noise, and then an attempt is made to filter out the noise. Change the noise level to different values (such as 10, 25, 100, or 200 percent) and compare the results.

6. To make a more accurate comparison, compare the root-mean-squared errors of the corrupted signal and the filtered signal for your different noise levels.

7. Now fix the noise level (e.g., at 50%), and then find the best value of p for the filter (see line 5 of the script `wienerfilter.m`). Does the value with the smallest root-mean-squared error also sound best?

8. Match the script `dftfilter.m` to `wienerfilter.m` so that you can compare their effects on an audio signal of your choice corrupted by a certain amount of noise.