Using the DFT For Data Analysis

MATH 471, NUMERICAL ANALYSIS LAB
Worksheet #8

In this lab we will make use of the *discrete Fourier transform* (DFT) discussed in class. The MATLAB functions that implements the DFT, and its inverse the IDFT, are `fft` and `ifft`. They are given by the equations

$$\hat{f}(k) = [\texttt{fft}(\mathbf{f})]_k = \sum_{n=1}^{N} f(n) e^{-i2\pi(k-1)(n-1)/N}, \quad 1 \le k \le N; \tag{1}$$

$$f(k) = [\texttt{ifft}(\hat{\mathbf{f}})]_k = \frac{1}{N} \sum_{n=1}^{N} \hat{f}(n) e^{i2\pi(k-1)(n-1)/N}, \quad 1 \le k \le N; \tag{2}$$

where $\mathbf{f} = (f_1, \ldots, f_n)$ and $\hat{\mathbf{f}} = (\hat{f}_1, \ldots, \hat{f}_n)$. Note that `ifft` coincides with $\frac{1}{N}$ IDFT and `fft` coincides with N DFT from class. This will not effect our analysis.

Example 1: In the m-file `DataAnal.m` on the website, the DFT is used to find the frequency components of a signal buried in a noisy time domain signal. The true signal x is a sum of two sine waves with frequencies 50 Hz and 120 Hz, i.e.

$$x(t) = \sin 50(2\pi t) + \sin 120(2\pi t).$$

The data has been sampled at 1000 Hz, is corrupted with zero-mean random noise, and can be found in the vector $\mathbf{y}$.

To filter the noisy signal, we compute the *power spectrum*, which is defined by

$$\mathbf{P}(\mathbf{y}) = |\texttt{fft}(\mathbf{y})|^2/N,$$

where N is the number of elements in $\mathbf{y}$. Note that since $[\texttt{fft}(\mathbf{y})]_k = \overline{[\texttt{fft}(\mathbf{y})]_{N-k}}$, we only plot the first $N/2$ elements of $\mathbf{P}$. These correspond to the frequencies

$$f = \frac{1000}{N} j, \quad j = 0, 1, ..., N/2.$$

Example 2: On the website you will find data from the Bonneville Power Corporation on power usage in the Pacific Northwest over a specific time period. The filename is `freqdata.csv`. Use `Bonneville.m` to filter the signal. What value of α do you think provides the most accurate fit of the "true signal". How did you make your choice?

Problem 1: The data in `signal.mat` on the web site has been sampled at 100 Hz and contains a signal of the form

$$x(t) = A \sin(k_1 t) + B \cos(k_2 t) + C \sin(k_3 t)$$

corrupted with zero-mean Gaussian noise. Filter the signal as is done in `DataAnal.m` and plot it. Next determine first k_1, k_2, k_3 and then A, B, C.

Calculating Sunspot Periodicity

We want to analyze the variations in sunspot activity over the last 300 years. Sunspot activity is cyclical, reaching a maximum about every 11 years. To see this, we analyze measurements of the Wolfer number, which has been collected by astronomers for almost 300 years. This quantity measures both number and size of sunspots. Load and plot the sunspot data.

```
>> load sunspot.dat
>> year = sunspot(:,1);
>> wolfer = sunspot(:,2);
>> plot(year,wolfer)
>> xlabel('Year')
>> ylabel('Wolfer Number')
>> title('Sunspot Data')
```

Now take the FFT of the sunspot data.

```
>> Y = fft(wolfer);
```

We now plot the power spectrum versus frequency, which is a called a "periodogram". Before plotting, we remove `Y(1)`, which is known as the DC component. We do this because the DC component corresponds to the zero frequency in the DFT, and hence, contains no information about periodicity in the data.

```
>> N = length(Y);
>> Y(1) = [];
>> power = abs(Y(1:N/2)).*abs(Y(1:N/2))/N;
>> freq = (1:N/2)/N;
>> plot(freq,power), grid on
>> xlabel('cycles/year')
>> ylabel('Power')
>> title('Periodogram')
```

Plotting power spectrum versus frequency (= cycles/year) is somewhat inconvenient. We now plot, instead, the power spectrum versus period (= years/cycle), which will allow us to estimate what one cycle is. For convenience, we plot the power versus period (where period = 1./freq) from 0 to 40 years/cycle.

```
>> period = 1./freq;
>> plot(period,power)
>> axis([0 40 0 7e4])
>> grid on ylabel('Power')
>> xlabel('Period(Years/Cycle)')
```

Note that the maximum power occurs at approximately 11 years, verifying the claim made at the outset.