1 FFT and Spectrogram

1.1 Fourier Transform for Finite Duration Signals

In order to analyze the frequency content of a finite duration discrete time signal x with N samples, we use the Discrete Fourier Transform (DFT):

$$\hat{x}(k) = \sum_{n=0}^{N-1} x(n) e^{-i\frac{2\pi k}{N}n}, \quad k = 0, \ldots, N-1.$$

This can be interpreted as the Fourier Transform of the finite duration signal evaluated at the frequencies $f = k/N$. Another interpretation is that the DFT is the Fourier Series of the periodic extension of x but is missing the $1/N$ scaling factor. This second interpretation gives rise to the Inverse DFT formula.

In order to go from $\hat{x}$ back to x, we use the Inverse DFT:

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} \hat{x}(k) e^{-i\frac{2\pi k}{N}n}, \quad n = 0, \ldots, N-1$$

These equations can be written in matrix form as

$$\begin{aligned} x &= \frac{1}{N} F \hat{x}, \\ \hat{x} &= \overline{F} x, \end{aligned}$$

where $\bar{F}$ is the complex conjugate of F, and F is the $n \times n$ Fourier matrix:

$$F = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 \\ 1 & e^{i\frac{2\pi}{N}} & e^{i\frac{4\pi}{N}} & \ldots & e^{i2\pi\frac{N-1}{N}} \\ 1 & e^{i\frac{4\pi}{N}} & e^{i\frac{8\pi}{N}} & \ldots & e^{i2\pi\frac{2(N-1)}{N}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e^{i2\pi\frac{N-1}{N}} & e^{i2\pi\frac{2(N-1)}{N}} & \ldots & e^{i2\pi\frac{(N-1)^2}{N}} \end{bmatrix}.$$

We will often want to graphically depict the frequency content found in the DFT $\hat{x}$. Remember that $\hat{x}$ is in general a complex valued vector even if x is real valued, so it is usual to plot both the magnitude $|\hat{x}(k)|$ and phase $\arg(\hat{x}(k))$ on separate graphs.

The elements in the $\hat{x}$ vector are coefficients for the frequencies $f = 0, 1/N, 2/N, \ldots, (N-1)/N$. However, notice that the high frequencies are equivalent to low negative frequencies due to aliasing. For example, $f = (N-1)/N$ is equivalent to $f = -1/N$. It is common to rearrange the vector $\hat{x}$ to represent the frequency components in the range $[-1/2, 1/2]$ rather than $[0, 1]$. Please read about the Matlab function `fftshift` which is used for this purpose.

1.2 Matlab: `fft`, `ifft` and `fftshift`

To calculate the DFT of a function in Matlab, use the function `fft`. FFT stands for Fast Fourier Transform, which is a family of algorithms for computing the DFT. A straight computation of the DFT from the formulas above would take n^2 complex multiplications and $n(n-1)$ complex additions. Algorithms have been developed (for example, the Cooley-Tukey FFT) which exploit symmetries of the roots of unity to reduce the number of calculations to $O(n \log_2 n)$. It is fastest when n is a power of 2, followed by composite numbers of small primes. Read the Matlab help page for more information.

To take the DFT of a vector x of length n, use the command:

```
hx=fft(x);
```

If you want to explicitly specify the length m, then use:

```
hx=fft(x,m);
```

This will append the vector with zeros if m is greater than the length of x and it will truncate x if m is less than the length of x. To compute the IDFT of a vector hx use:

```
x=ifft(hx);
```

The command `fftshift` swaps the first and the second half of a vector. This is used before plotting `hx` to ensure the frequency range is in $[-1/2, 1/2]$. To plot the magnitude versus frequency with zero frequency centered on the axis, you could use the commands (assuming N is even):

```
hx=fft(x);
shx=fftshift(hx);
f=[-N/2:N/2-1]/N;
figure(1)
stem(f,abs(shx),'r')
xlabel('Frequency in [-1/2,1/2]')
ylabel('Magnitude of DFT(x)')
axis([-1/2 1/2 0 inf]);
grid
```

1.3 Sampled Signals

Suppose you begin with a continuous time signal $x(t)$ defined over a finite time period $[0, T]$ and then you construct a discrete-time signal $x[n]$ by taking uniformly spaced samples of $x(t)$. If N uniformly spaced samples of x are taken, then the sampling frequency is $F_s = N/T$ (the sampling frequency would technically be $(N-1)/T$ if the first sample is $x(0)$ and the last sample is $x(T)$).

We can take the DFT of the discrete-time signal $x[n]$ and use this to represent the frequency content of the continuous-time signal $x(t)$. This may not be valid, but let's go with it anyway. You'll learn more about this in the lecture on sampling. If we represent the discrete-time frequency spectrum on the range $[-1/2, 1/2]$, then we translate this to continuous-time frequencies by multiplying by F_s. Thus, given a sampled signal with sampling frequency F_s, we plot its DFT magnitude and phase versus frequencies in Hz in the range $[-F_s/2, F_s/2]$.

1.4 The Spectrogram

Let x be signal of length N. Consider consecutive segments (or "clips") of x of length m where $m \ll n$ and let $X \in \mathbb{R}^{m \times (N-m+1)}$ be the matrix with the consecutive segments as consecutive columns. In other words, $[x[0], x[1], \ldots, x[m-1]]^T$ is the first column, $[x[1], x[2], \ldots, x[m]]^T$ is the second column, and so forth. Both the rows and columns of X are indexed by time. We see that X is a highly redundant representation of x.

The **spectrogram** of x with window size m is the matrix $\hat{X}$ whose columns are the DFT of the columns of X. So

$$\begin{aligned} \hat{X} &= \overline{F} X \\ X &= \frac{1}{m} F \hat{X} \end{aligned}$$

Note that the rows of $\hat{X}$ are indexed by frequency and the columns are indexed by time. Each location on $\hat{X}$ corresponds to a point in frequency and time. So $\hat{X}$ is a mixed time-frequency representation of x. Because the transformation between X and $\hat{X}$ is invertible, $\hat{X}$ is also highly redundant.

The spectrogram is a matrix. To visualize it we can view the matrix as an image with the i, j-th entry in the matrix corresponding to the intensity or color of the i, j-th pixel in the image. In Matlab this is done

by calling the function `imagesc`. This first scales the image to the full range of the color map and then displays it as an image. The x-axis is the time axis and the y-axis is the frequency axis. You can change the color map with the command `colormap`. Since the signal is real, the first half and the second half of each column are redundant by a symmetry property of the DFT. Therefore, when you display $\hat{X}$, you only need to show the first half of each column. Also, by default `imagesc` plots from the top left corner, and we will want to see it upside down, so you use the following commands (assuming `HX` is your spectrogram):

```
ax=imagesc(t,f,HX); % t is the time vector for the x-axis
set(ax,'YDir','normal'); % and f is the frequency vector for the y-axis
```

Given that all of the information in $\hat{X}$ is already in x what is its value? We are interested in analyzing signals whose frequency content changes in time (e.g. music) and we want to analyze how this occurs. Taking the DFT of the entire signal will show us the frequency content over the entire time period. We would like to take the DFT over a short period of time because this will give us a local snapshot in time of the frequency content of the signal during that short time period. We can do this by first taking the DFT of $x(1:m)$, then we can take the DFT of $x(2:m+1)$, and so on until we hit the end of the signal. This is exactly the spectrogram.