Note that I might use the terms autocorrelation and covariance interchangeably. For zero mean data, these terms are equal.

This is a brief description of the program "directionEstimates.m"

- The program assumes that the total number of sensors is 64. It also assumes that there are two sources at directions cos(theta1)=0 and cos(theta2)=0.06. The program generates data for all 64 sensors, for a given SNR and number of snapshots.
- The first algorithm is direct MUSIC which is implemented as explained below:
  1. Find the longest possible range of lags. At each lag, estimate the covariance/autocorrelation making use of every possible sensor pair. If there are multiple sensor pairs for any lag, we take average of the estimates given by different sensor pairs. To do this in MATLAB, the easiest method is the one that utilizes convolution operation. Convolution is related to autocorrelation (*I'll explain this in the next meeting if you want*) as $autocorrelation(x_a[n], x_b[n]) = convolution(x_a[n], x_b[-n])$.
  2. Once you have a series of autocorrelation estimates, make a toeplitz matrix.
  3. Use the toeplitz matrix as the covariance matrix in MUSIC.
- The second algorithm in the code is product processing followed by MUSIC which is implemented as explained below:
  1. Apply product processing to data which gives us an estimate of power spectral density.
  2. If we take the inverse Fourier transform of the power spectral density estimate, we get a range of autocorrelation estimates.
  3. Once we have a range of autocorrelation estimates, we make a toeplitz matrix.
  4. Use the toeplitz matrix as the covariance matrix in MUSIC.
- The third algorithm in the code is min processing followed by MUSIC which is implemented as explained below:
  1. Apply min processing to data which gives us an estimate of power spectral density.
  2. If we take the inverse Fourier transform of the power spectral density estimate, we get a range of autocorrelation estimates.
  3. Once we have a range of autocorrelation estimates, we make a toeplitz matrix.
  4. Use the toeplitz matrix as the covariance matrix in MUSIC.
- The program also applies MUSIC to the whole full ULA. This obviously will have the best outcome. We can use this as a benchmark.
- **Nested Team**: An example of input parameters for Nested Team: directionEstimates(4,16,1,4,10,200,1). See how good or bad the estimates are when you change SNR, number of snapshots, or the number of sensors.
- **Coprime Team**: An example of input parameters for Coprime Team: directionEstimates(12,12,3,2,10,100,1). See how good or bad the estimates are when you change SNR, number of snapshots, or the number of sensors.