Exercise 11 – Time series. Cross-correlation. Autocorrelation. Detection of cycles.

1. Open the file 'Tsup.mat'. This file contains the hourly air temperatures at two locations (termed 'stations') on Lake Superior.
   Station 1: Lat 47.08 Lon -90.73
   Station 4: Lat 47.18 Lon -87.22
   You can use Google Maps to visualize the locations. The record starts on midnight January 1, 2000 and continues until the fall of 2006. The NaN values that were present in the original series were removed by replacing them by the preceding temperature measurements (yes, this *will* generate some artifacts). The columns of the dataset are 1) time in hours; 2) temperature at the first station 3) temperature at the second station.

Look at the data. What does it tell you? For example, is there a relationship between the temperatures at the two locations? If you know the temperature at one location, with what accuracy can you predict the temperature at the other? What are the trends in the data? Can they be quantified? Are there any cycles and what causes them? Is it possible to determine their periods?

2. Copy the data into separate variables and plot the data

```
t=Tsup(:,1);
T1=Tsup(:,2);
T2=Tsup(:,3);
plot(t,T1,t,T2)
```

   Notice that the two series are noisy but obviously correlated because of the annual temperature cycles.

   Another good way to visualize the correlation in two series is to plot them against each other:

```
plot(T1,T2,'o')
```

   Note that this relies on the fact that both series have identical time vectors. Compute the correlation coefficient

```
corrcoef(T1,T2)
```

3. In most time series analyses, it is necessary to remove any linear trends that may be present in the data. The first thing to do is to check if any significant trends exist in our series. For example:

```
fit(t,T1,'poly1')
fit(t,T2,'poly1')
```

You may also use the ‘polytool’ command or any linear fit commands that you know. Is the linear trend (the slope) in your data significant?

Let’s remove the linear trend. For that, we will subtract the linear trend line from the data. Define the two variables for the slope and intercept that you found with the ‘fit’ tool for the first series (T1):

```
slope1 = ______    % enter the found value for slope here
intercept1 =  _____ % enter the found value for intercept here
```

Subtract the fit line from the series T1 and make it into a new series S1

```
S1=T1 – (slope1*t + intercept1);
```

Repeat for the second series (T2) and save it into a new series S2. Plot the series to make sure they look ok. The fact that we subtracted not only the slopes but also the intercepts means that we also set the means of the series to zero, which is fine for our purposes.

4. First, let’s look at the autocorrelations. This is the correlation of the series with itself. It can be calculated using the same command ‘xcorr’ but with a single argument:

   ```
   c=xcorr(S1); plot(c)
   ```

   We will discuss the output in a moment. The function ‘xcorr’ calculates correlations for both positive and negative lags (τ), so the lag=zero is plotted in the middle of the graph. To mark the position of the zero lag, we can, for example, draw a vertical line:

   ```
   hold on
   d=max(size(c));
   mc=max(c);
   plot([d/2  d/2] , [-mc  mc], 'r')
   ```

For zero lag, the correlation obviously has to be perfect, with the correlation coefficient =1. By default, however, the function ‘xcorr’ does not normalize its output, so the y-axis of your graph does have units (what are they?). Moreover, as it shifts the series to calculate the correlations at larger lags, it progressively runs out of points to base the calculation on. As a result, the correlations decrease for large lags. This is called a ‘bias’. One way to alleviate the problem is to specify additional options to xcorr:

```
hold off
c=xcorr(S1,’unbiased’); plot(c)
```

The result is better but the correlations now increase at the edges because of the increased role of statistical fluctuations for a small number of data points. Another option lets you force the correlation coefficient at zero lag to be 1:

```
c=xcorr(S1,'coeff'); plot(c)
```

So what does this autocorrelation function tell you? The fact that the autocorrelation function is periodic means that shifting the function relative to itself by a certain lag τ produces a function that is similar to the original. In other words, your function is periodic. The lag τ at which the maximum occurs is the function's period.

5. Now let's look at the cross correlations between series T1 and T2:

```
c=xcorr(S1,S2,'unbiased'); plot(c)
hold on
mc=150;              % to plot a center line
plot([d/2  d/2] , [-mc  mc], 'r')
```

There is obviously correlation between the two series, with a period corresponding to one year. What would you expect to see if the temperature at the second station followed the temperature at the first station with a delay of several days?

6. So far, we looked only at the large scale annual signals. But are there correlations in the daily temperature fluctuations? To check this, we need to remove the annual signal from the series. One way to do this is to subtract the running mean.

```
hold off
ws=24;   % size of averaging window is 1 day
rmean1=filter(ones(1,ws)/ws,1,S1);
Q1 = S1 - rmean1;
plot(t,Q1)

rmean2=filter(ones(1,ws)/ws,1,S2);
Q2 = S2 – rmean2;
plot(t,Q2)
```

The newly created series Q1 and Q2 have the annual trend removed from them and contain mostly the high-frequency temperature fluctuations. Now let's see if these fluctuations are correlated:

```
c=xcorr(Q1,Q2,'unbiased'); plot(c)
```

What can you say about the nature of these high-frequency temperature variations? Investigate the autocorrelations for Q1 and Q2 and experiment with the size of the averaging window 'ws'. (To automate the process, you may copy the above sequence of commands in a .m file and run them for a number of values of ‘ws’.) Look closely at the area around the peak in the autocorrelation function; see if there is any characteristic time scale (perhaps, hours?) on which the temperature fluctuations are still correlated and not random.