[Tyler Olivieri],

ECE 3512: Stochastic Processing in Signals and Systems

Department of Electrical and Computer Engineering, Temple University, Philadelphia, PA 1912

# Problem Statement

Given several different common signals, the autocorrelation and power spectral density will be computed and plotted. The power spectral density is simply the fourier transform of the autocorrelation function. The signals given are Gaussian white noise, an impulse function, a periodic impulse train with a period of 20 samples, a sinewave with a period of 20 samples, and a sine wave with Gaussian white noise with SNR = 10.

# Approach and Results

For white noise, we expect a flat frequency response, and an impulse as the autocorrelation. With an impulse autocorrelation function, a flat power spectral density is expected. MATLAB cannot create an impulse function, therefore the results are not perfect, but the autocorrelation function starts at 1 and then quickly drops to zero. The power spectral density can also be seen to be nearly flat.

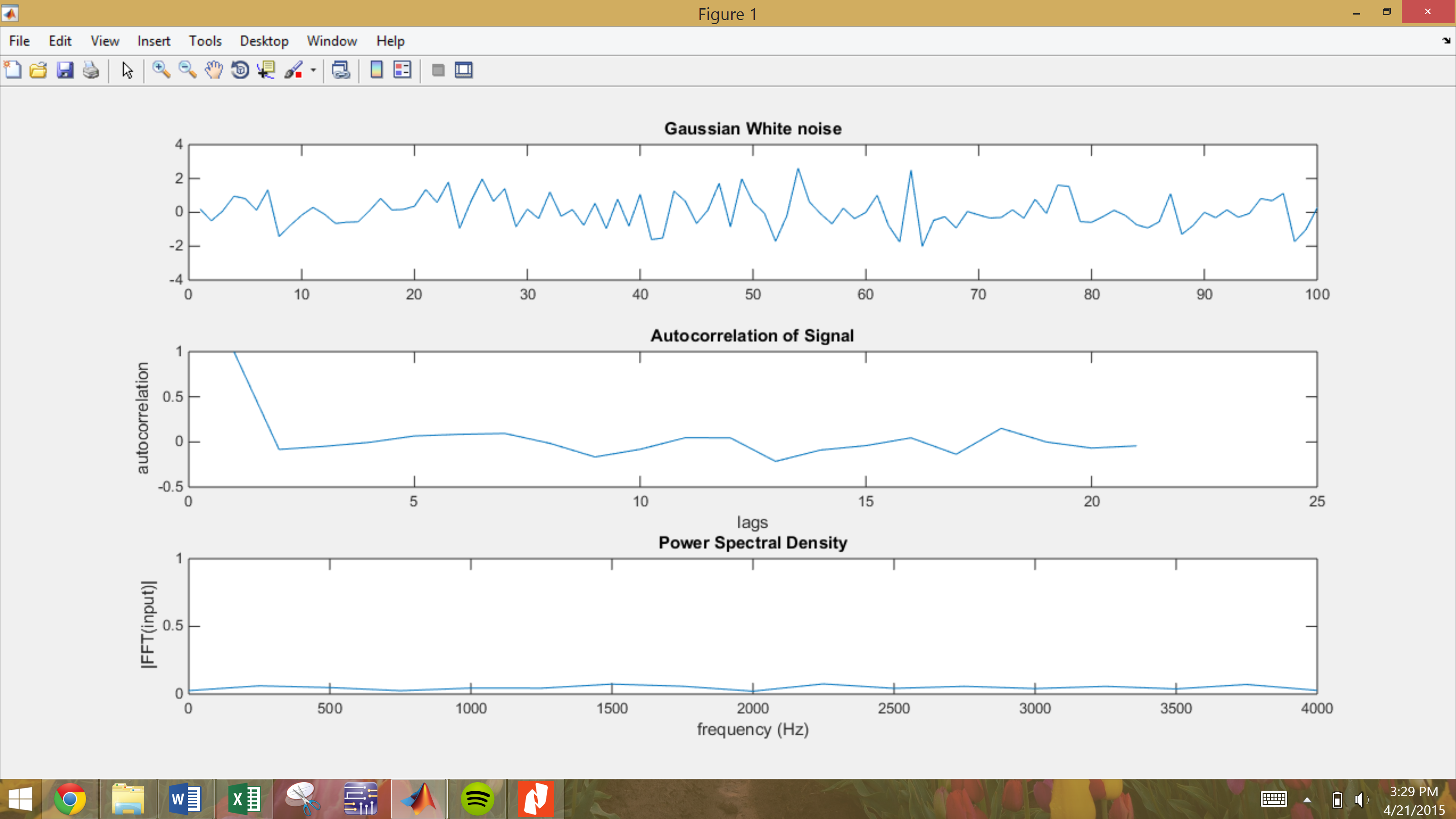


Figure : Gaussian white noise, autocorrelation, power spectral density

For an impulse function, we would expect the autocorrelation function to be an impulse and then the power spectral density would be flat. For the autocorrelation function, all lags are zero except the first lag. The fourier transform of an impulse is a constant. Again, MATLAB cannot create an impulse function, but similar results can be seen.

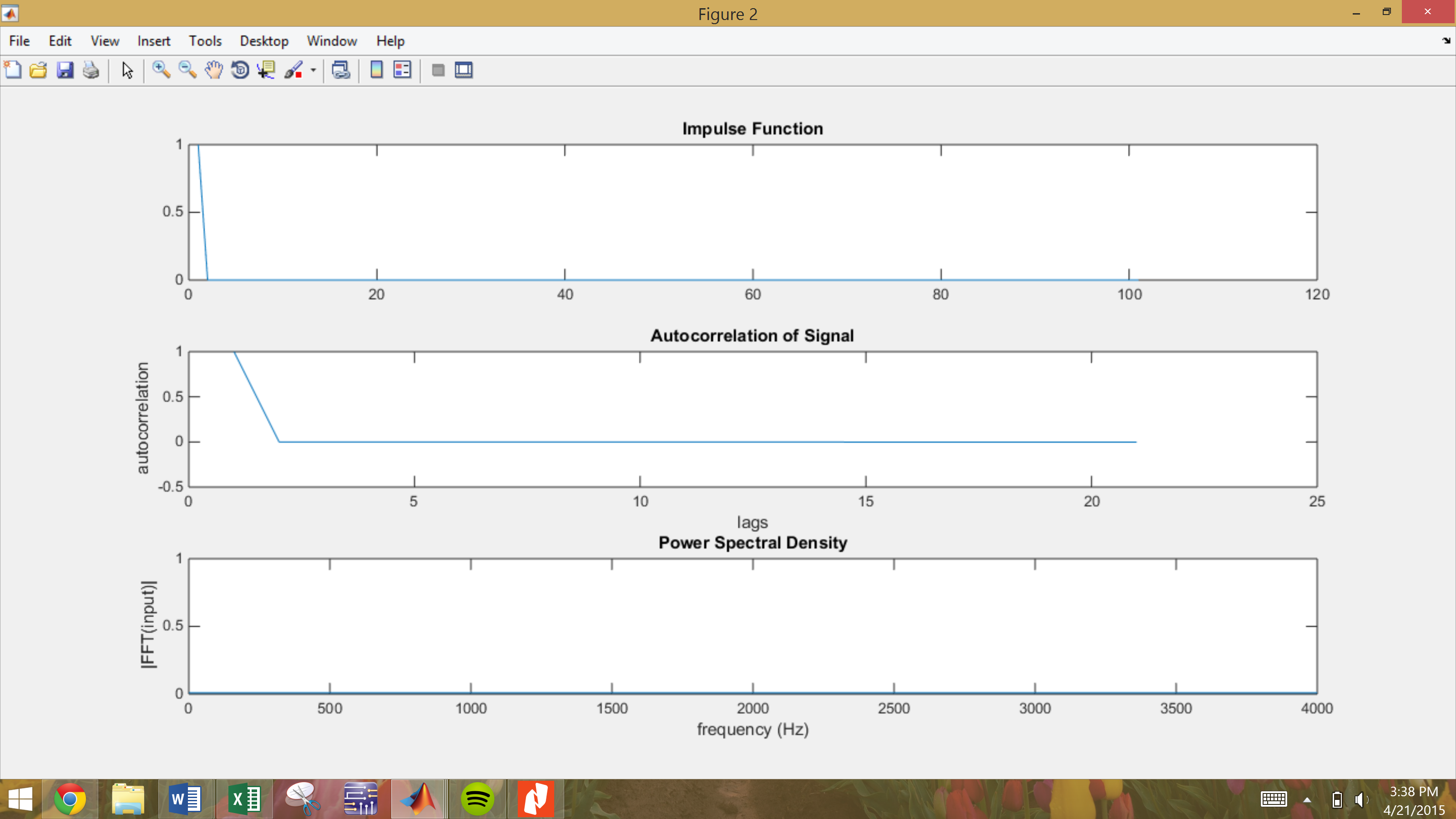


Figure : Impulse function, autocorrelation, power spectral density

For an impulse train, the autocorrelation of an impulse should be replicated periodically with the same period of the impulse train. In this case, the period is 20 lags. The autocorrelation function should have perfect impulses and the power spectral density should be completely flat, however it is not. MATLAB cannot create a perfect impulse function.

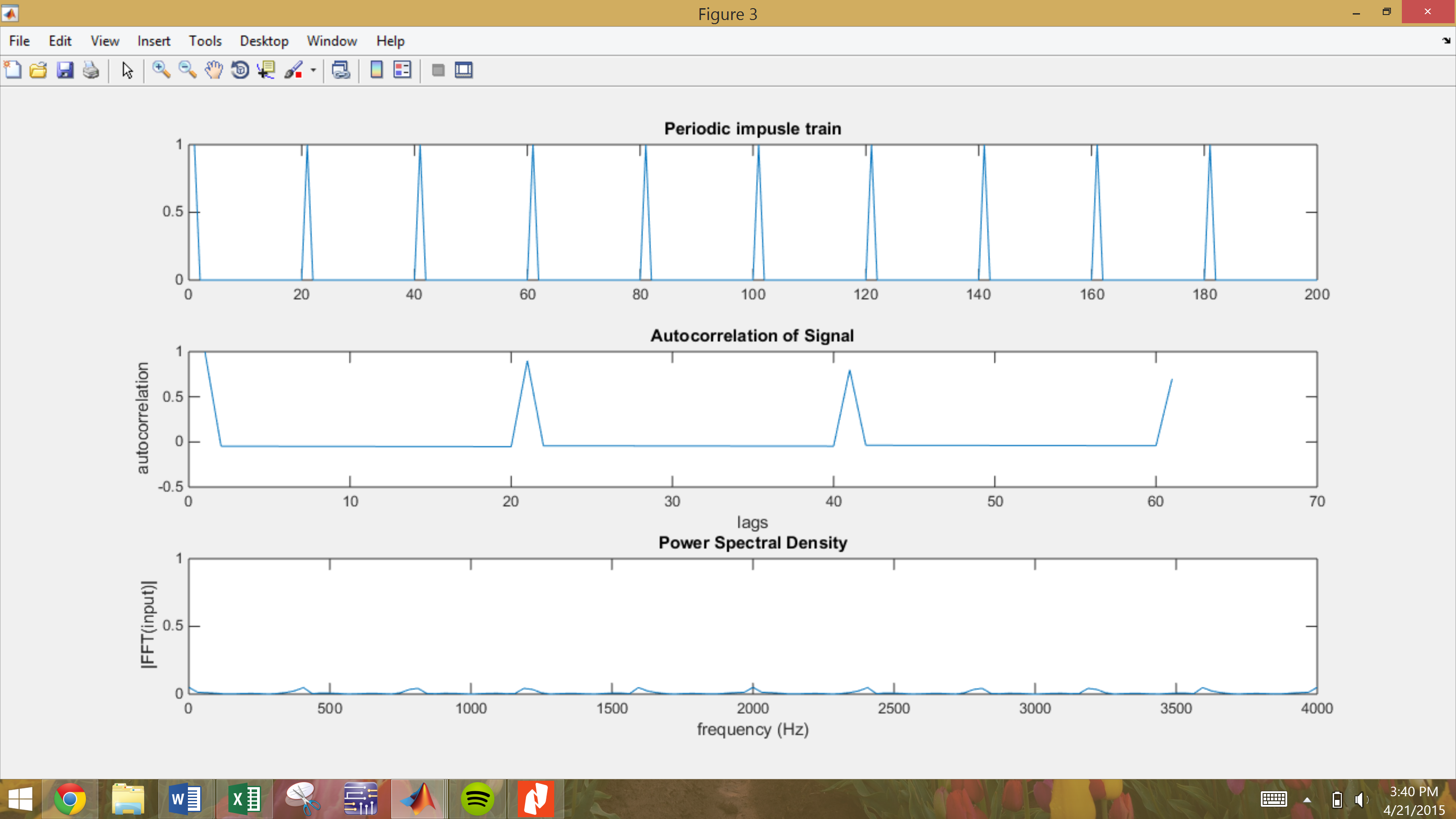


Figure : Periodic impulse train, autocorrelation, power spectral density

The autocorrelation function of a sine wave is a cosine wave. This can be observed in the figure below. The power spectral density of the cosine wave is just the fourier transform. Since the cosine only has energy at one frequency, the Fourier transform of the cosine should only have energy at one frequency, which can also be observed.

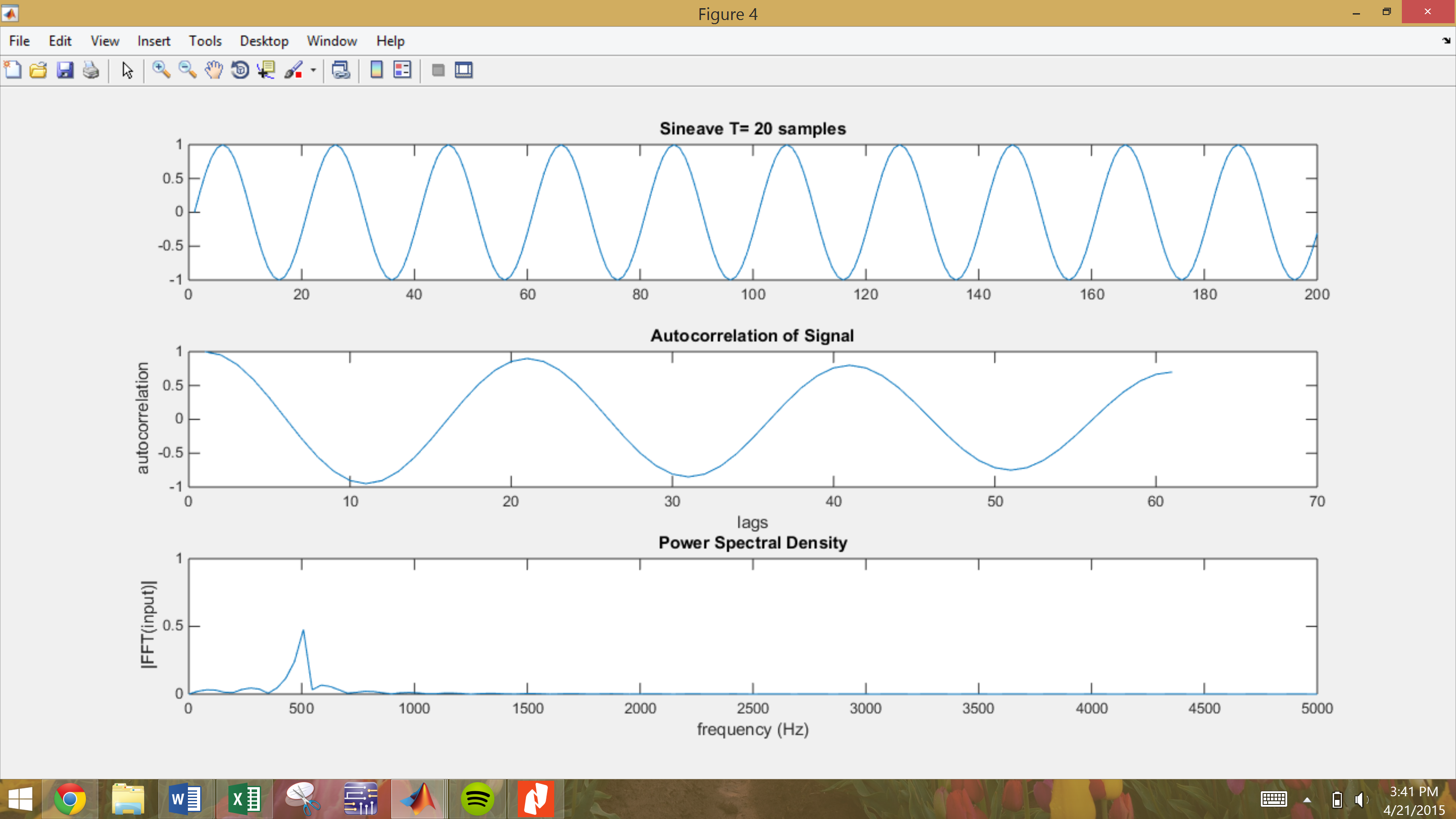


Figure : Sinewave period of 20 samples, autocorrelation, power spectral density

Now, the sine wave is generated with the same period but N changes. This was done by repeating the sinewave of N samples until the sine wave is greater than 60 samples, which is the required amount of lags.

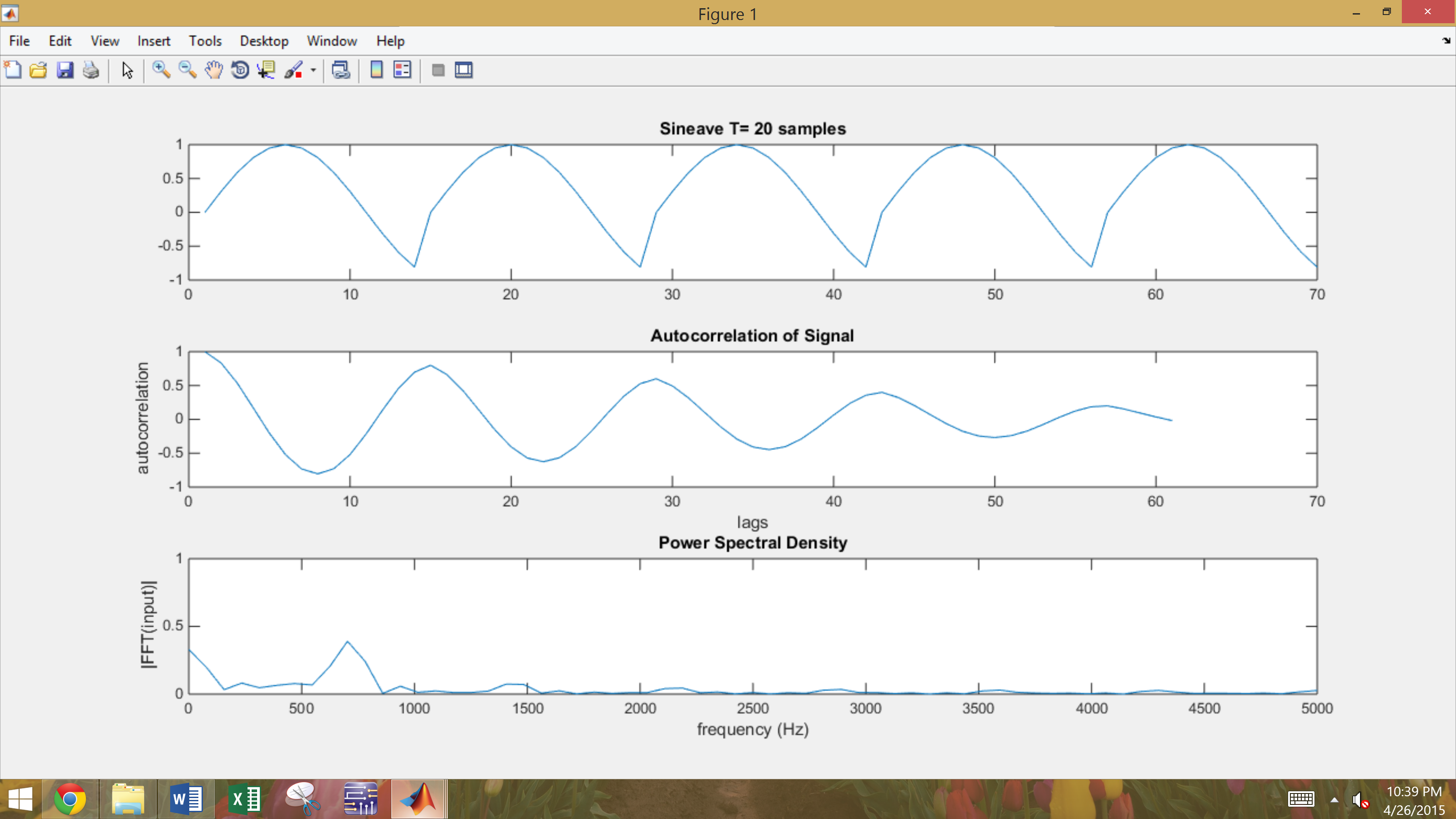


Figure : Sinewave period of 20 samples, N = 14

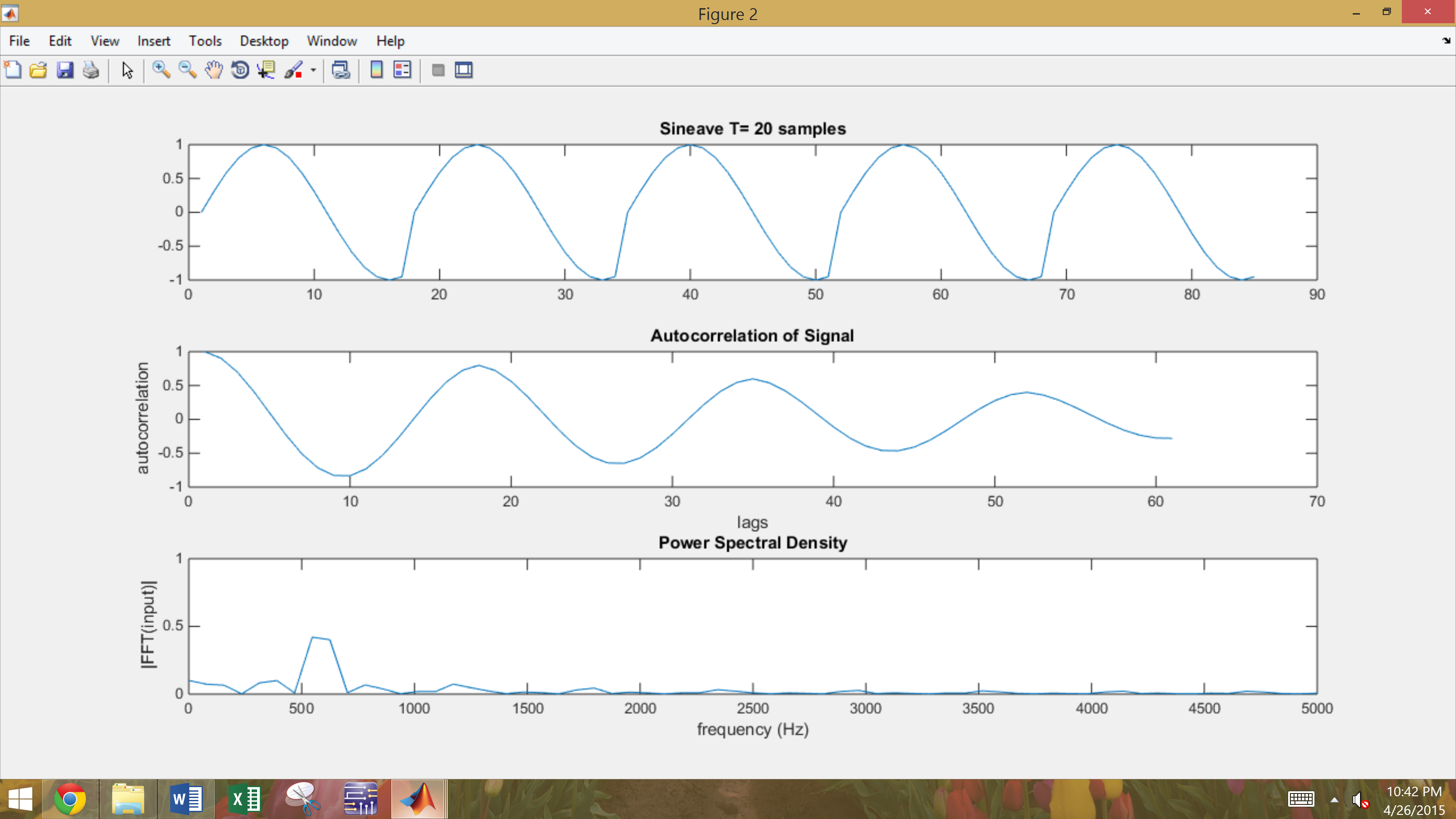


Figure : Sinewave period of 20 samples N = 17

With N = 20, the sine wave looks identical to N = 200 and so do the autocorrelation and power spectral density.

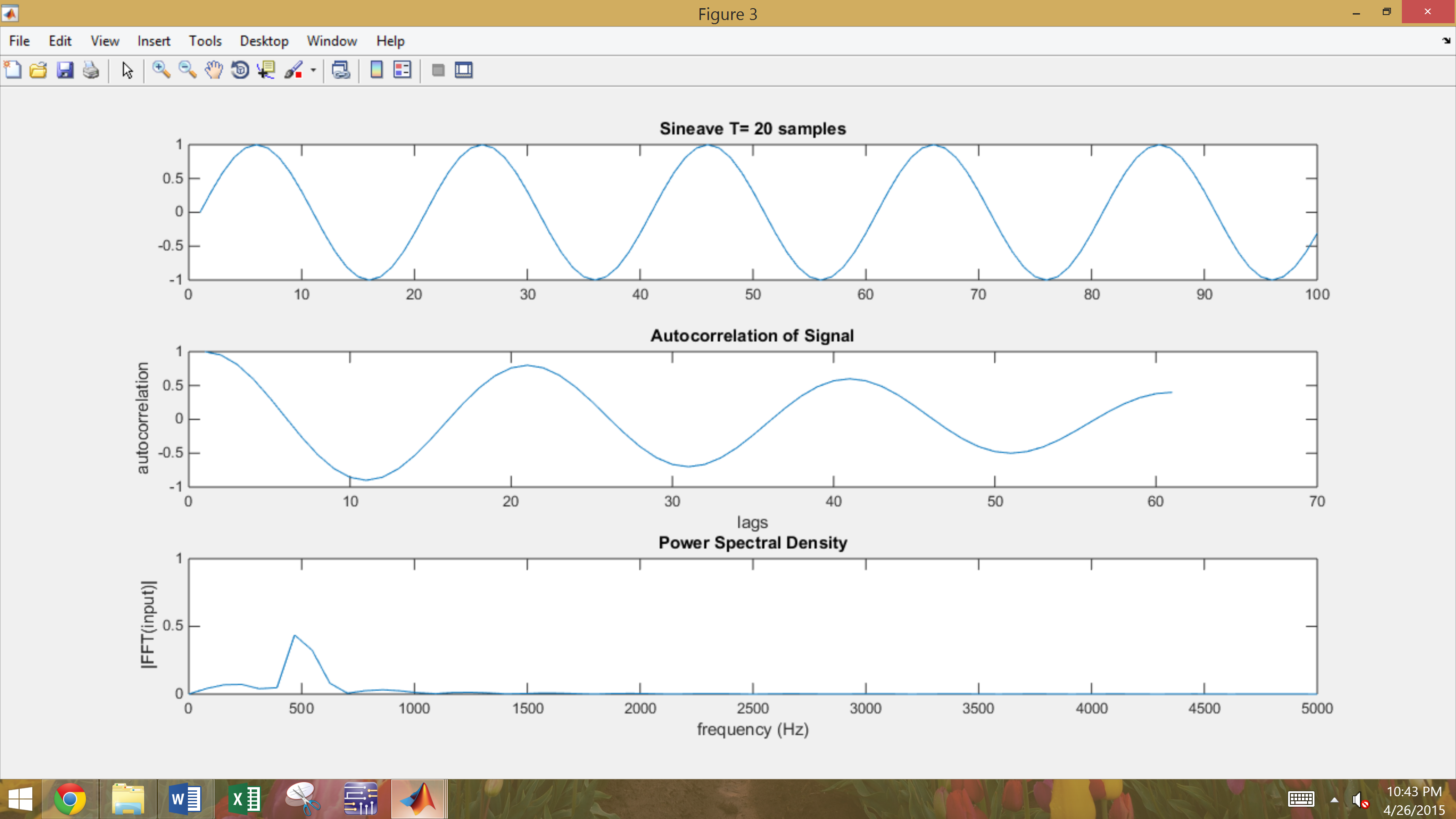


Figure : sine wave period of 20 samples, N = 20

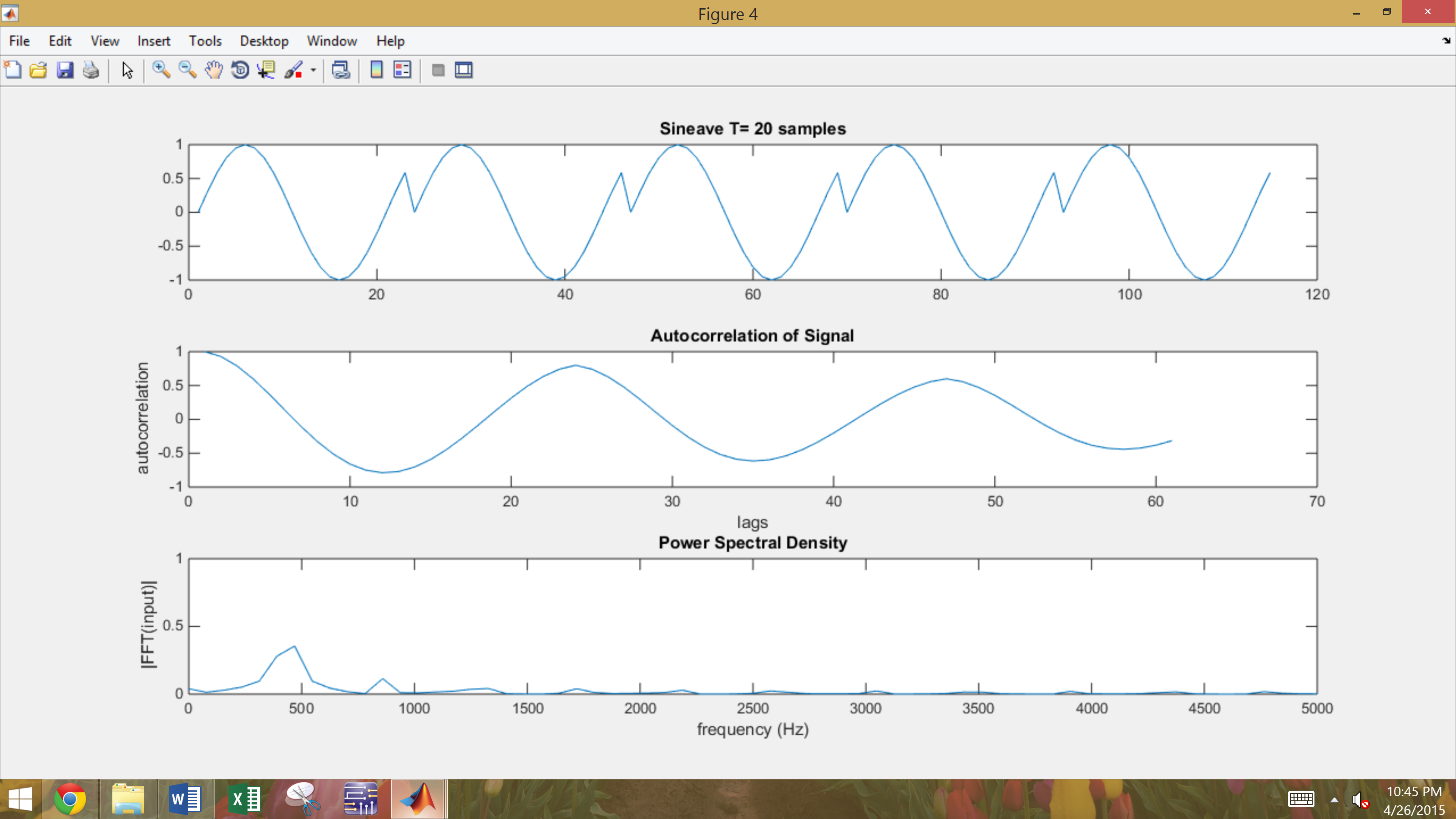


Figure : sine wave period of 20 samples. N = 23

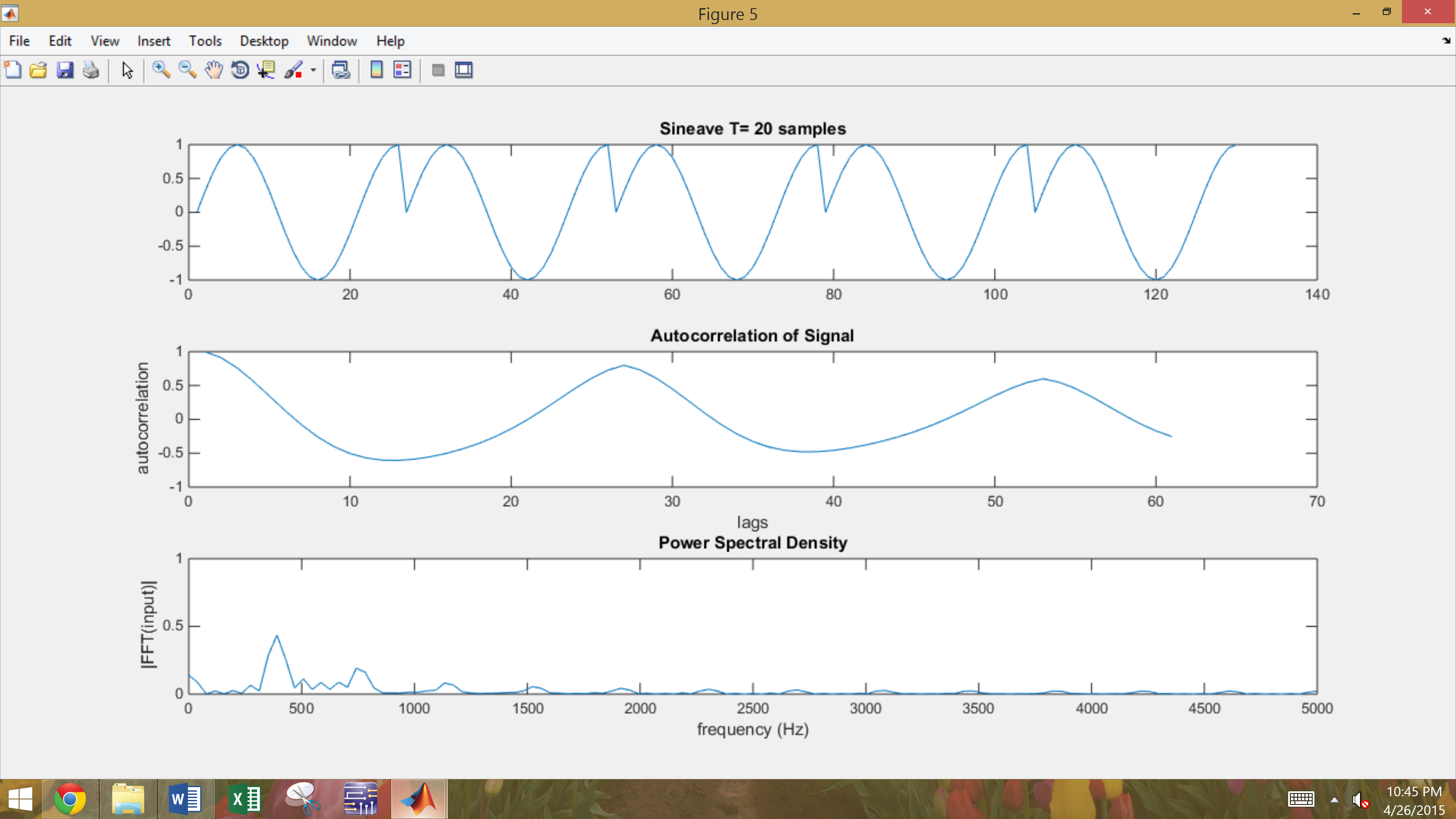


Figure : sine wave period of 20 samples N =26

Similarly to the sine wave, the autocorrelation is expected to become a cosine wave, but not a perfect cosine wave. The signal dominants the noise so the cosine should be dominant in the autocorrelation. The power spectral density should be focused around the frequency of the cosine wave but is again a little less defined then a pure sine wave.

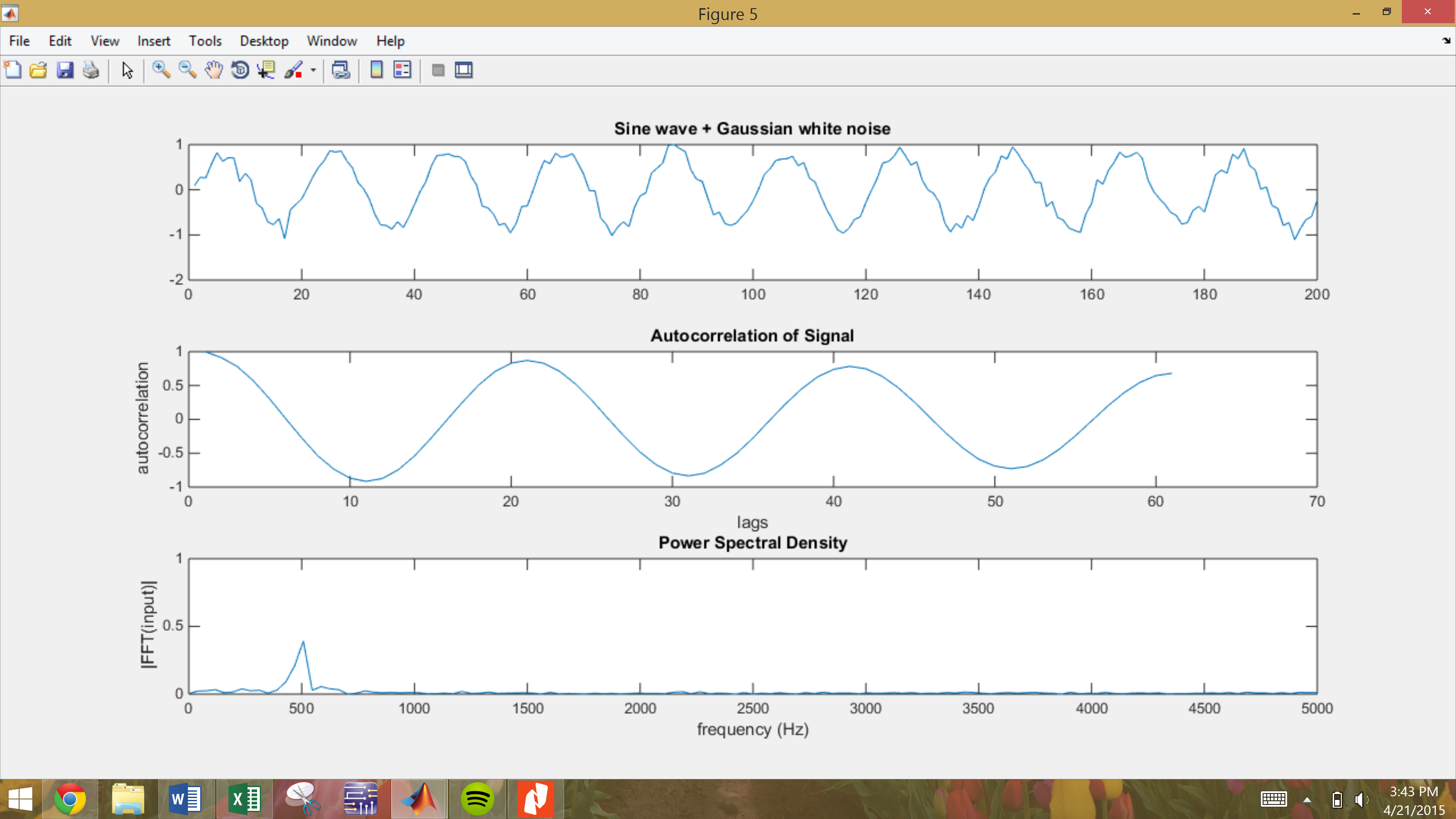


Figure : sine wave w/ Gaussian white noise SNR = 10, autocorrelation, power spectral density

# MATLAB Code

Below is sectioned MATLAB code that generated the signal, plotted the signal, used a function to calculate and plot the autocorrelation function which can be seen at the bottom and used a function to calculate and plot the fourier transform, which was used on the output of the autocorrelation function. Subplot was used to create a nice graphic of all plots. For the sine wave with less points than lags, the “sine wave” was made periodic until the number of samples was greater than the number of lags.

%%Tyler Olivieri  
%CA 11  
clc;clear;  
fs = 8000  
NFFT = nextpow2(fs);  
lags = 20;  
  
%generate gaussian white noise  
u1 = rand(100,1);  
u2 = rand(100,1);  
gwn = sqrt(-2\*log(u1)).\*cos(2\*pi\*u2);  
  
figure(1);  
subplot(3,1,1)  
plot(gwn)  
title('Gaussian White noise')  
subplot(3,1,2)  
autocorr1 = Tyautocorr(gwn,lags);  
subplot(3,1,3)  
TyFFT2(autocorr1,fs);  
title('Power Spectral Density')  
ylim([0 1])

clc;clear;  
impulse\_train = [1 zeros([1 19])];  
lags = 60;  
fs = 8000;  
%periodic impulse train with a period of 20 samples  
  
for n = 1:10  
 temp = [1 zeros([1 19])];  
 if (n ~= 10)  
 impulse\_train = [impulse\_train temp];  
 else  
 impulse\_train = impulse\_train;  
 end  
end  
  
figure(3);  
subplot(3,1,1)  
plot(impulse\_train)  
title('Periodic impusle train')  
subplot(3,1,2)  
autocorr1 = Tyautocorr(impulse\_train,lags);  
subplot(3,1,3)  
TyFFT2(impulse\_train,fs);  
title('Power Spectral Density')  
ylim([0 1])

clc;clear;  
lags = 20;  
fs = 8000;  
impulse = [1 zeros([1 100])];  
  
figure(2);  
subplot(3,1,1)  
plot(impulse)  
title('Impulse Function')  
subplot(3,1,2)  
autocorr1 = Tyautocorr(impulse,lags);  
subplot(3,1,3)  
TyFFT2(impulse,fs);  
title('Power Spectral Density')  
ylim([0 1])

clc;clear;  
f = 500;  
fs = f\*20;  
N = 200;  
t= 0:1/fs:(N-1)/fs;  
lags = 60;  
sine = sin(2\*pi\*f\*t);  
  
figure(4);  
subplot(3,1,1)  
plot(sine)  
title('Sineave T= 20 samples')  
subplot(3,1,2)  
autocorr1 = Tyautocorr(sine,lags);  
subplot(3,1,3)  
TyFFT2(sine,fs);  
title('Power Spectral Density')  
ylim([0 1])

f=500;  
fs = f\*20;  
N = 200;  
lags = 60;  
t= 0:1/fs:(N-1)/fs;  
snr\_db = 10;  
sine\_noise = generate\_sine(f,max(t),fs,snr\_db);  
  
figure(5);  
subplot(3,1,1)  
plot(sine\_noise)  
title('Sine wave + Gaussian white noise')  
subplot(3,1,2)  
autocorr1 = Tyautocorr(sine\_noise,lags);  
subplot(3,1,3)  
TyFFT2(sine\_noise,fs);  
title('Power Spectral Density')  
ylim([0 1])

%x- input  
% - lags  
function acf = Tyautocorr(x,lags)  
  
acf = autocorr(x,lags);  
plot(acf)  
title('Autocorrelation of Signal')  
xlabel('lags')  
ylabel('autocorrelation')

function TyFFT2(Signal, fs)  
  
 samL = length(Signal);  
  
 NFFT = 2^nextpow2(samL);  
  
 Signalf = fft(Signal, NFFT)/samL;  
 f = fs/2\*linspace(0,1,NFFT/2+1);  
  
 plot(f, abs(Signalf(1:NFFT/2+1)));  
 xlabel('frequency (Hz)')  
 ylabel('|FFT(input)|')  
 title('Magnitude Spectrum')  
  
end

%%

clc;clear;

f = 500;

fs = f\*20;

for N = [14 17 20 23 26]

t= 0:1/fs:(N-1)/fs;

lags = 60;

sine = sin(2\*pi\*f\*t);

sine\_periodic = sine([1:N]);

for n = 1:5

temp = sine([1:N]);

if(n ~= 4)

sine\_periodic = [sine\_periodic temp];

else

sine\_periodic =sine\_periodic;

end

end

figure;

subplot(3,1,1)

plot(sine\_periodic)

title('Sineave T= 20 samples')

subplot(3,1,2)

autocorr1 = Tyautocorr(sine\_periodic,lags);

subplot(3,1,3)

TyFFT2(sine\_periodic,fs);

title('Power Spectral Density')

ylim([0 1])

end

# Conclusions

The autocorrelation and power spectral density of several common waveforms were explored. The Gaussian white noise showed an almost flat power-spectral density and an impulse in the autocorrelation function (Figure 1). Matlab cannot create a true impulse, so a value was set to 1 with all other values zero. Matlab has a slope to reach back to 0 as it does not do it instantaneously, this caused some minor issues with the results. As seen when the autocorrelation of the impulse was taken (Figure 2). The autocorrelation function should be a 1 at zero lags and zero elsewhere. Then the power spectral density should be flat and it basically is. When creating an impulse train, the autocorrelation function shows a periodic impulse train as well at the sample period as the impulse train. This makes sense as the autocorrelation function will have a value at every 20 lags, were every impulse is. The power spectral density, should still be flat for the impulse train and is fairly flat (Figure 3). Then a sine wave is created with a period of 20 samples. The autocorrelation should be a cosine wave, and that is confirmed (Figure 4). The power spectral density should show energy at the frequency of the sine wave. I set the frequency of the sine wave to be 500 Hz and that is where the energy is seen. In Figure 5, Figure 6, Figure 7, Figure 8, and Figure 9 the total amount of points in the sine wave is changed. Since the autocorrelation function cannot compute without the total number of points in the wave to be greater than or equal to the number of desired lags, the sinewaves had to be repeated to get them to be over 60 points. This showed a sine wave like autocorrelation and the power spectral density was always around 500 Hz, but since the sine wave was truncated, the frequency was changed. Finally, the sine wave plus Gaussian white noise showed almost a perfect cosine autocorrelation function and energy at 500 Hz (Figure 10). This is an amazing result that seemed to remove the noise of the sine wave and show the correct frequency domain.