Name: Thong Thach

Lab session: 2-4:50 pm

EE110B LAB 7

1)

1) First, determine the analytical form of $h[n]$ by performing the inverse DTFT of $H(f)$. In fact, you can verify that $h[n] = F^{-1}\{H(f)\} = \int_{-0.2}^{0.2} e^{j2\pi fn} df = \frac{\sin(\pi 0.4n)}{\pi n} = 0.4sinc(0.4n)$. Here $h[0] = 0.4$.

For computing h[n]:

```
l=20;
n=-l-1:l+1;
h=0.4.*sin(pi*0.4*n)./(pi*0.4*n);
h(l)=0.4;

%h=0.4.*sinc(0.4*n);
```

2)

2) Second, compute $g[n] = h[n - n_0]w[n]$ where

$$w[n] = \begin{cases} n, & 0 \le n \le n_0 \\ 2n_0 - n, & n_0 \le n \le 2n_0 \end{cases}$$

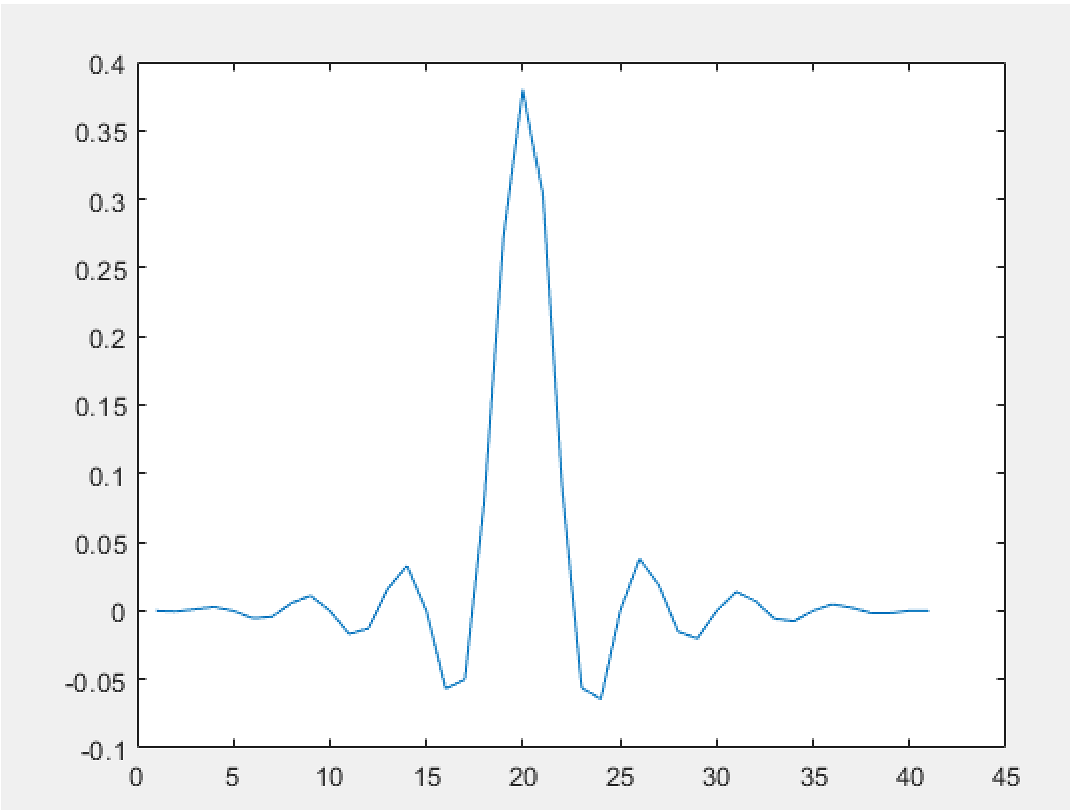
Unlike $h[n]$, $g[n]$ is a finite casual impulse response.

When l=20

For the computing part of g[n]:

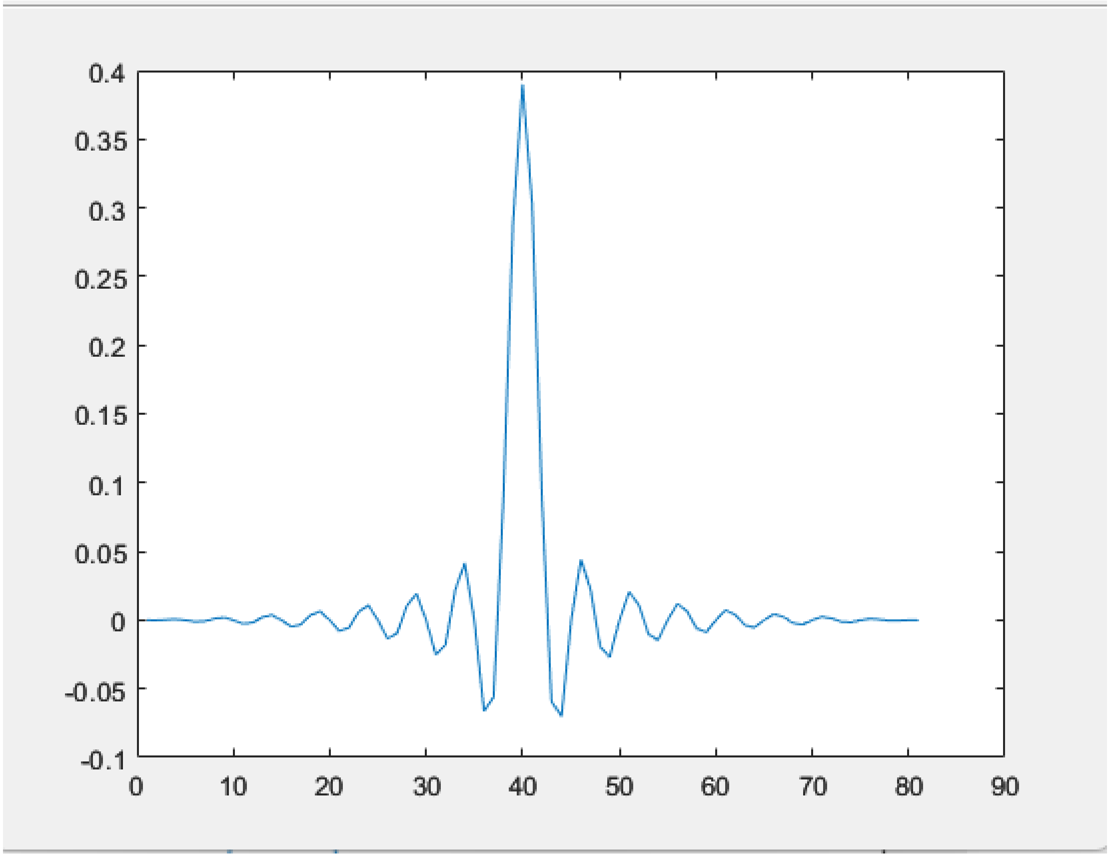
```
l=20;
n=-l+1:l+1;
h=0.4.*sin(pi*0.4*n)./(pi*0.4*n);
h(l)=0.4;

w=[0:l,(l-1):-1:0];
g=(h.*w)/l;
plot(g);
```

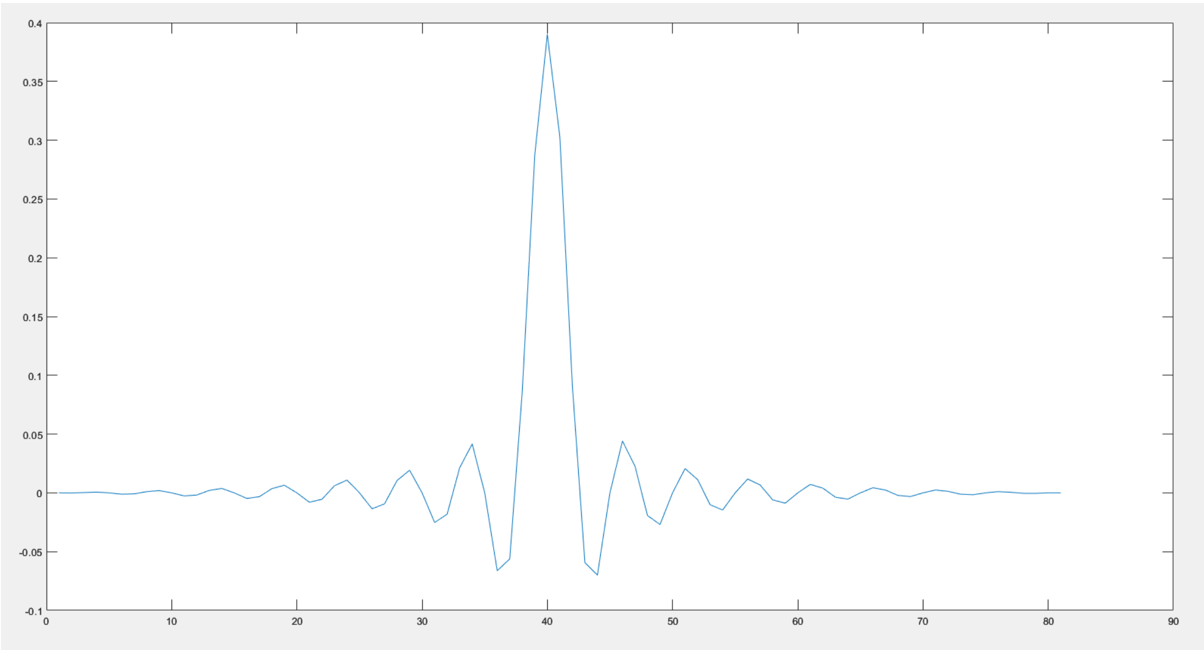
For the graphing part of g[n]



When l=40:

For the graphing part of g[n]:





3)

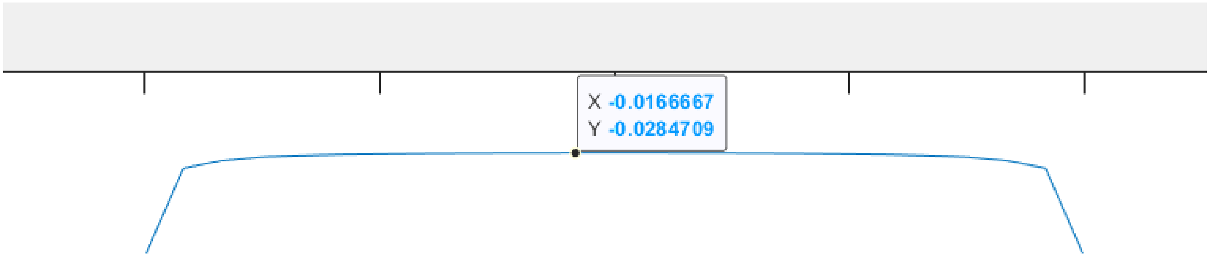
3) Choose different values of n_0 (such as 20, 30, etc) and compute and plot the amplitude spectrum $|G(f)|$ of $g[n]$ for $-0.5 < f < 0.5$.

When l=20:

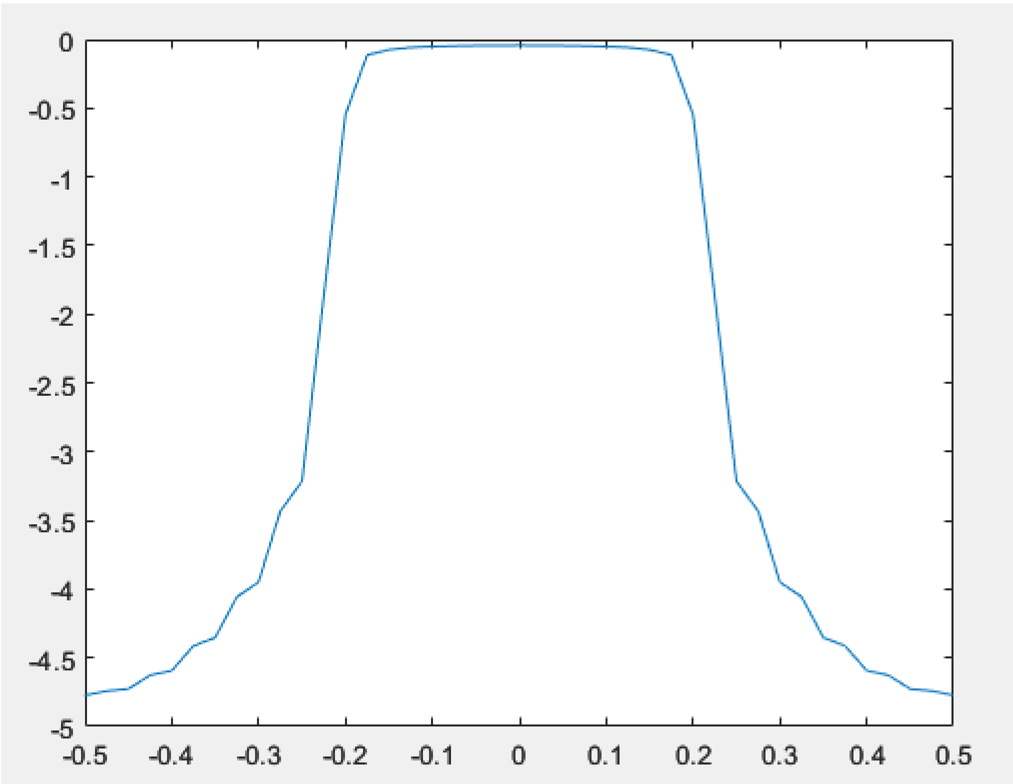
For the coding part:

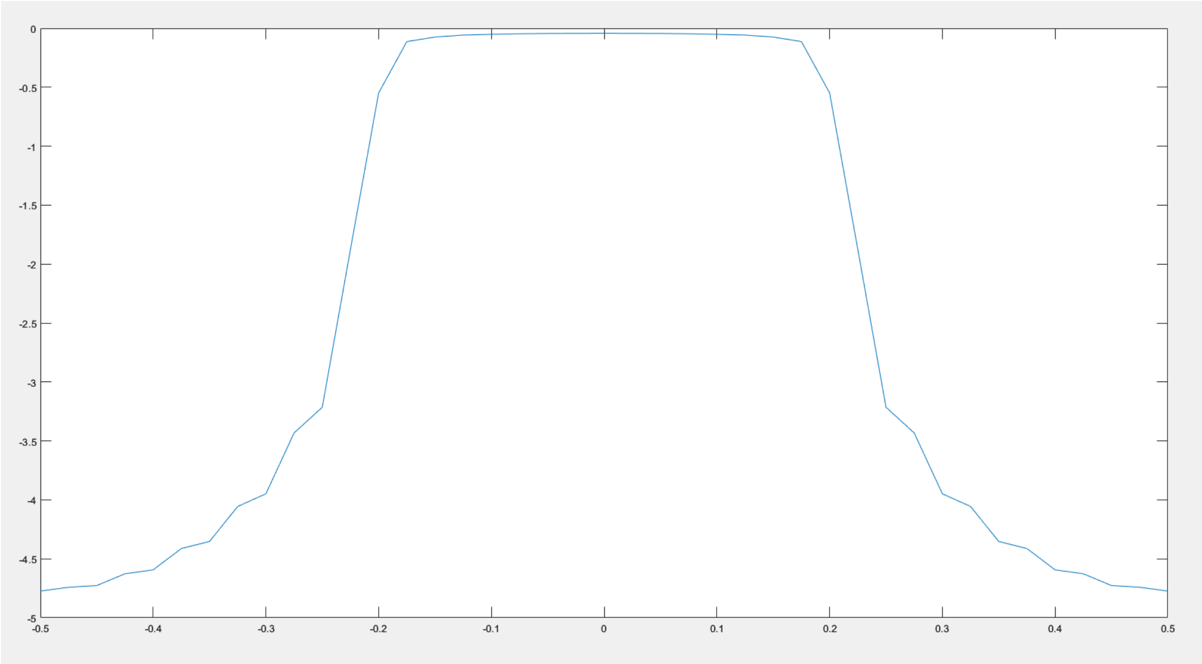
```
clc;clear;close all;
l=20;
n=-l+1:l+1;
h=0.4.*sin(pi*0.4*n)./(pi*0.4*n);
h(l)=0.4;

w=[0:l,(l-1):-1:0];
L=length(w);
g=(h.*w)/l;
f=-0.5:(1/(L-1)):0.5;
G=fftshift(fft(g));

plot(f,log(abs(G)));
plot(f,angle(G));
```

For computing the amplitude spectrum of |G(f)|



For the graphing part:



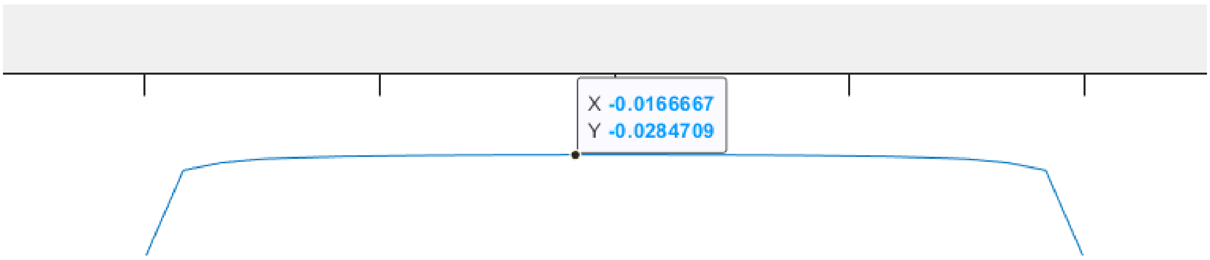


When l=30;

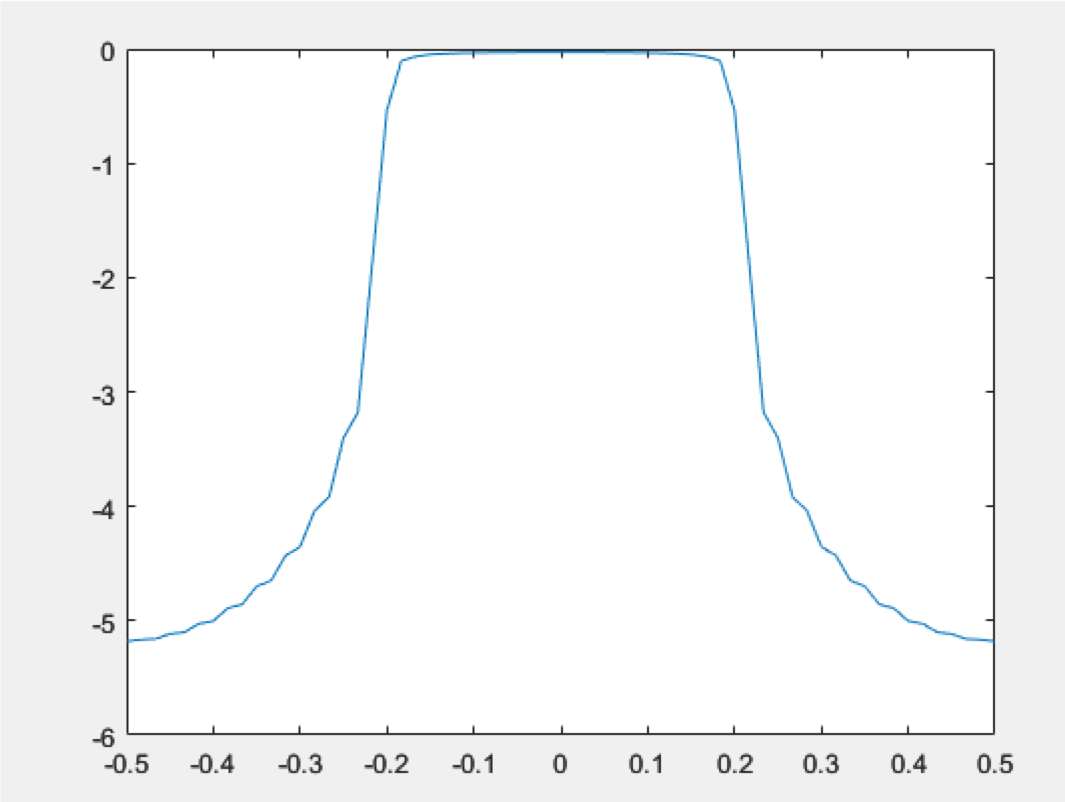
For the coding part:

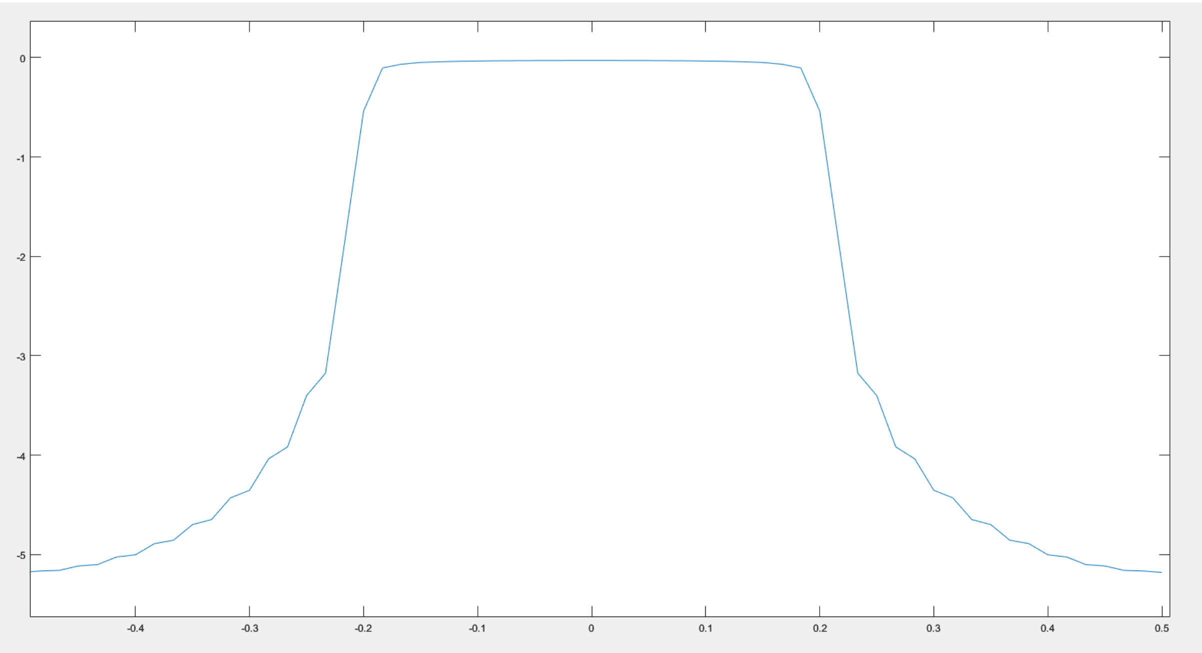
it is exactly the same as the previous code, the only difference is changing from l=20 to l=30:

For the computing part:



For the graphing part:



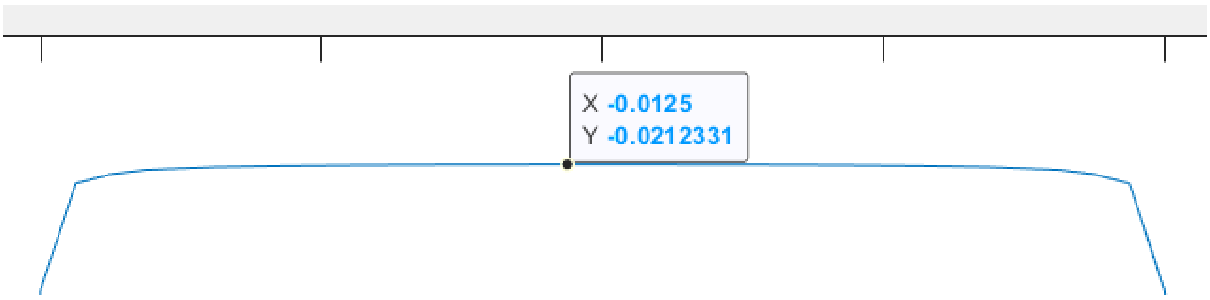


When l=40:

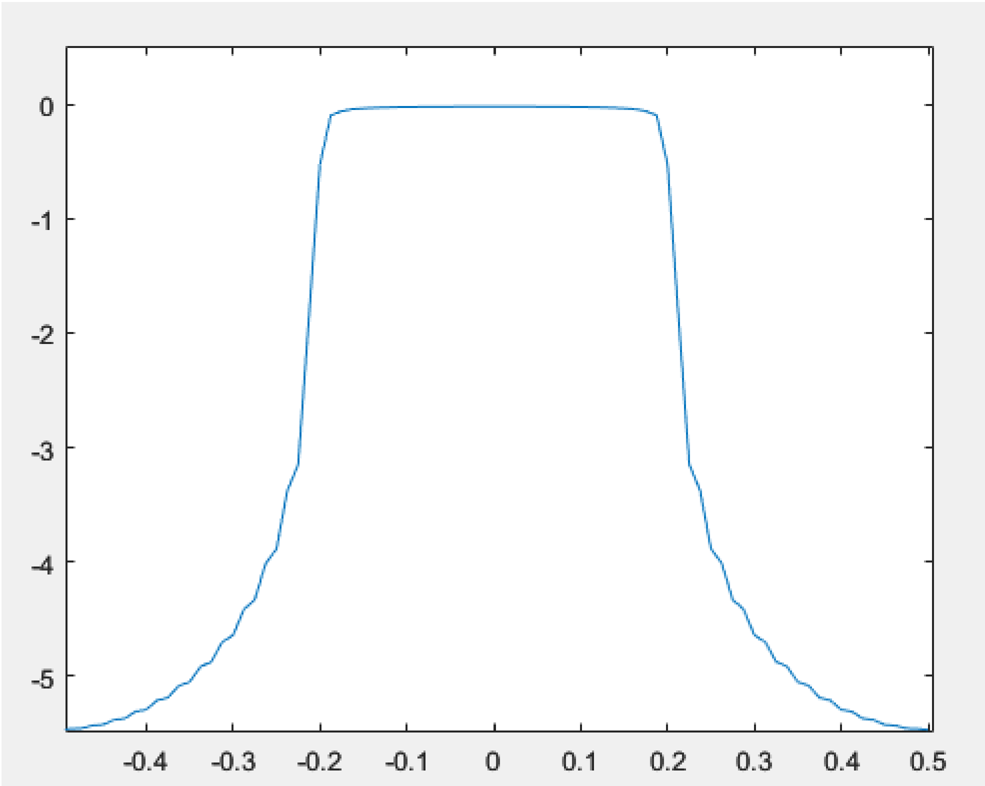
For the coding part:

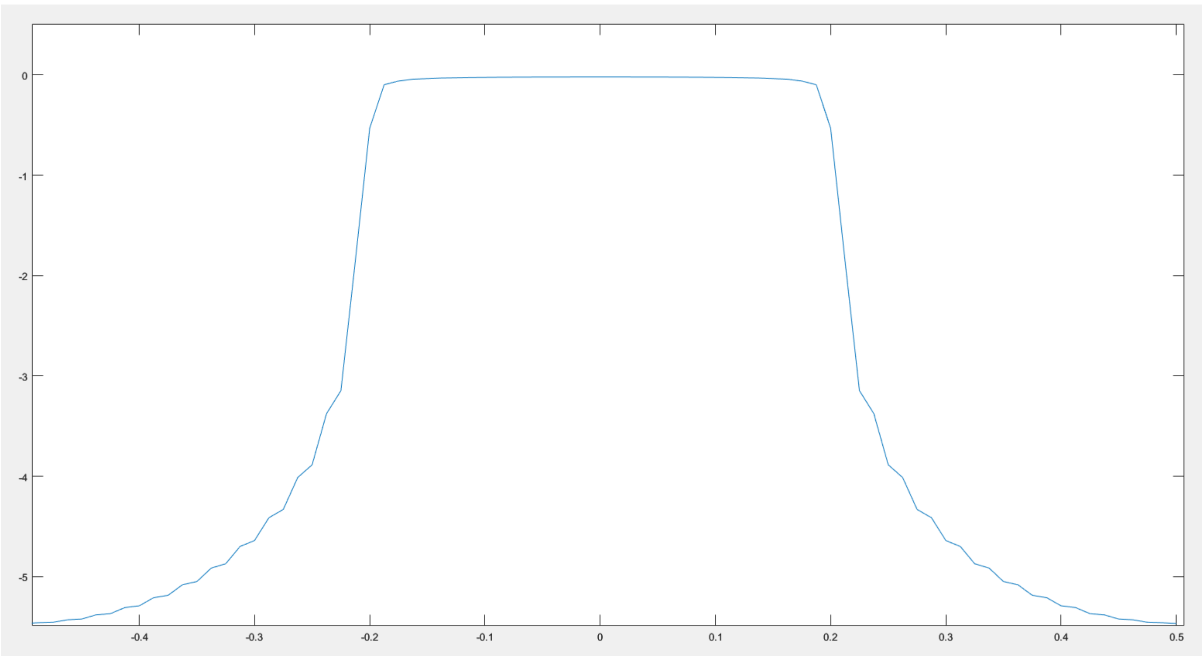
it should be similar to the previous code, the only difference is changing l=20 to l=40;

For the computing part:



For the graphing part

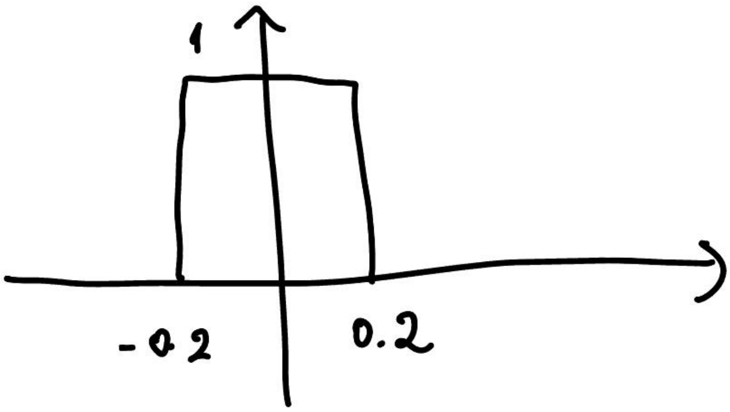

0
-1
-2
-3
-4
-5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5


0
-1
-2
-3
-4
-5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5

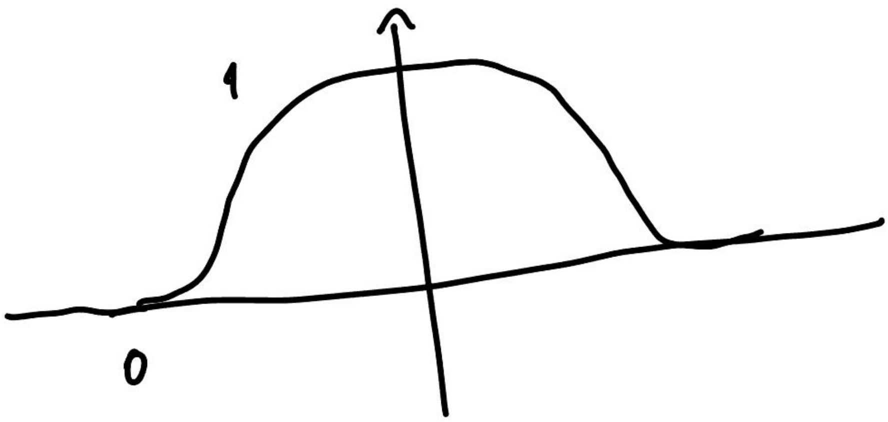
4)

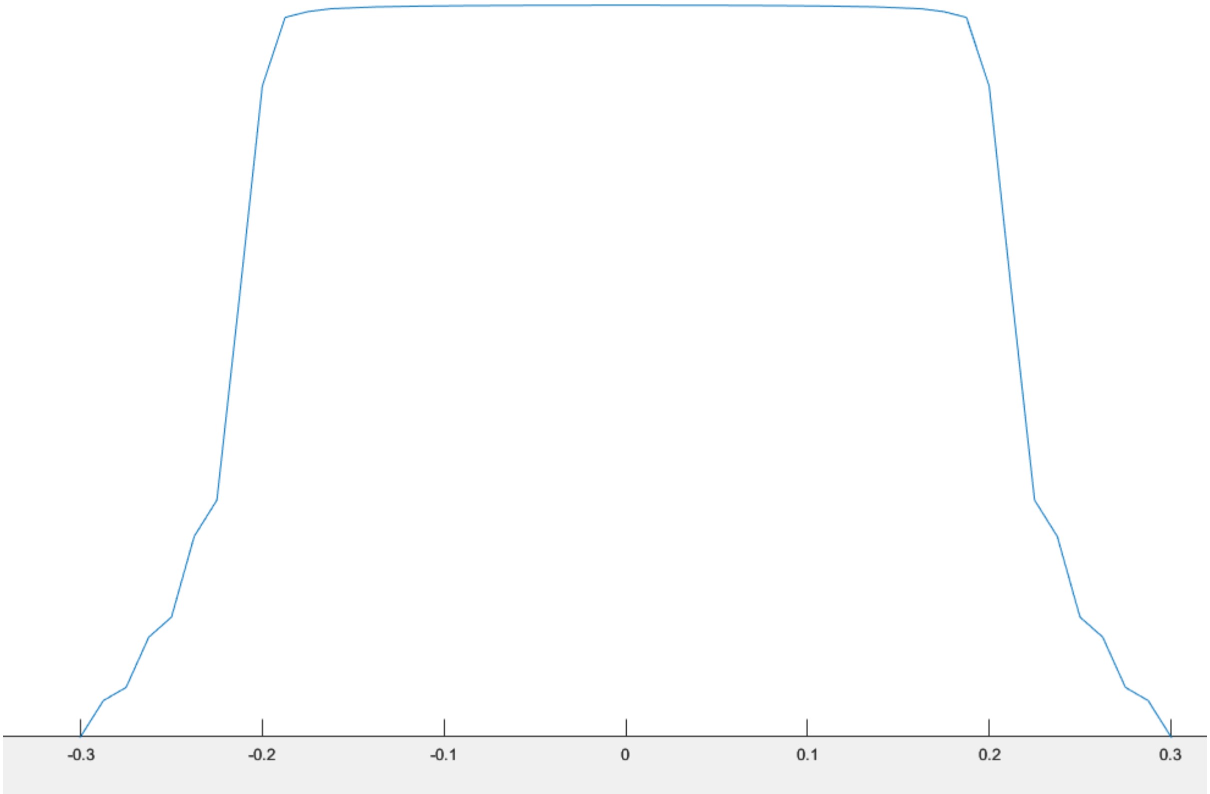
4) Compare $|G(f)|$ with $|H(f)|$ and discuss the effect of n_0 on $|G(f)|$. Hint: you can plot $|G(f)|$ on log scale vertically to see details of the small region in $|G(f)|$.

For the $H(f)$.



For the $G(f)$.





For the spectrum |G(f)|, the graph is increasing gradually from 0 to 1. However, for the spectrum |H(f)|, the graph is increasing instantly from 0 to 1

The effect of n(o) on |G(f)| is as n is increasing from -0.3 to -0.2, the spectrum of |G(f)| increases, and then it stays constantly from -0.2 to 0.2. When n>0.2, the spectrum of |G(f)| decreases.

5)

5) Also compute and plot the phase spectrum $\angle G(f)$ of $g[n]$. Is the phase spectrum always a linear function of f within $-0.5 < f < 0.5$? Why?

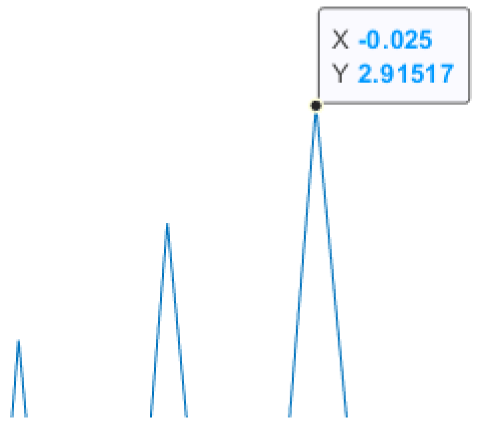
When l=20:

For the coding part:

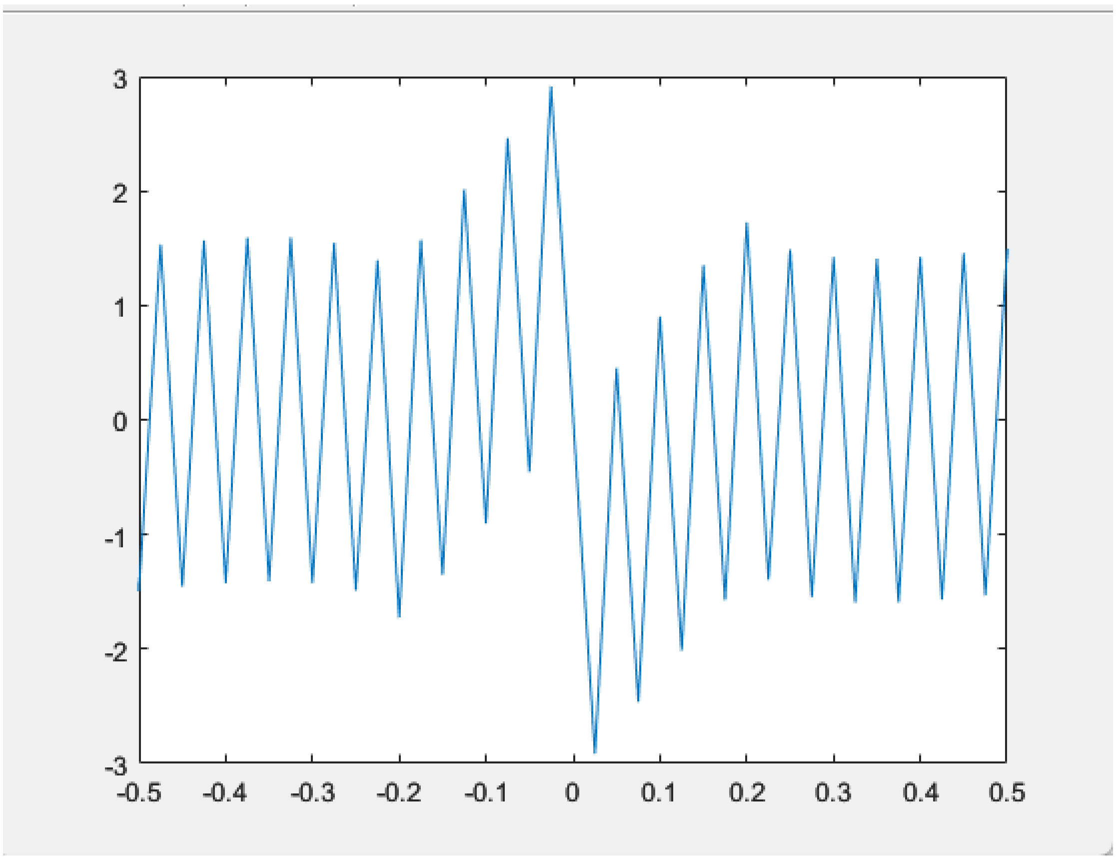
```
clc;clear;close all;
l=20;
n=-l+1:l+1;
h=0.4.*sin(pi*0.4*n)./(pi*0.4*n);
h(l)=0.4;

w=[0:l,(l-1):-1:0];
L=length(w);
g=(h.*w)/l;
f=-0.5:(1/(L-1)):0.5;
G=fftshift(fft(g));

plot(f,log(abs(G)));
plot(f,angle(G));
```

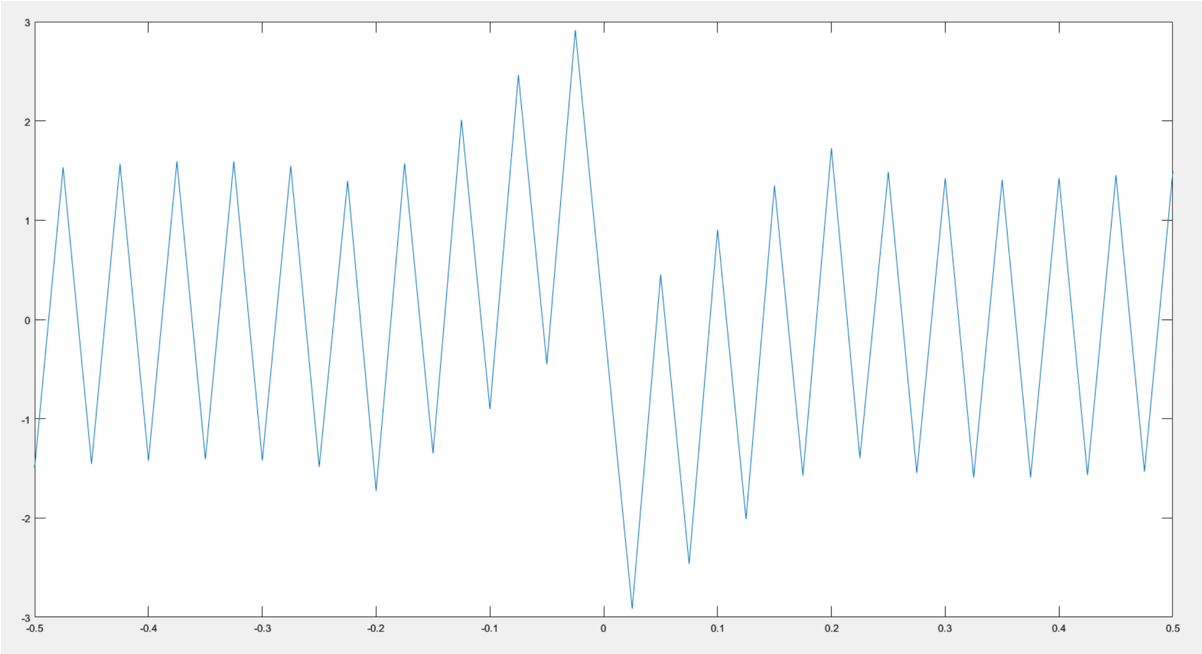
Computing the phase spectrum |G(f)|

The maximum phase of this spectrum is **2.91517**



For the graphing part:



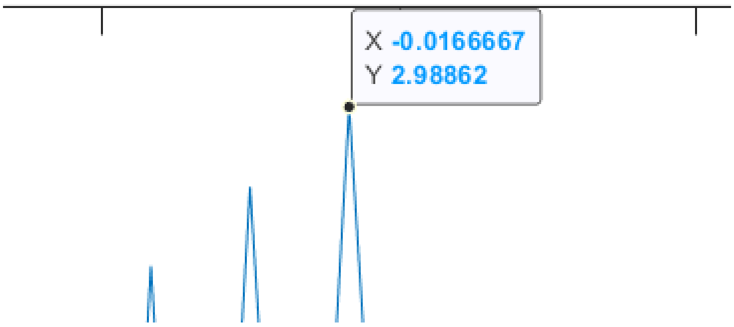


When l=30;

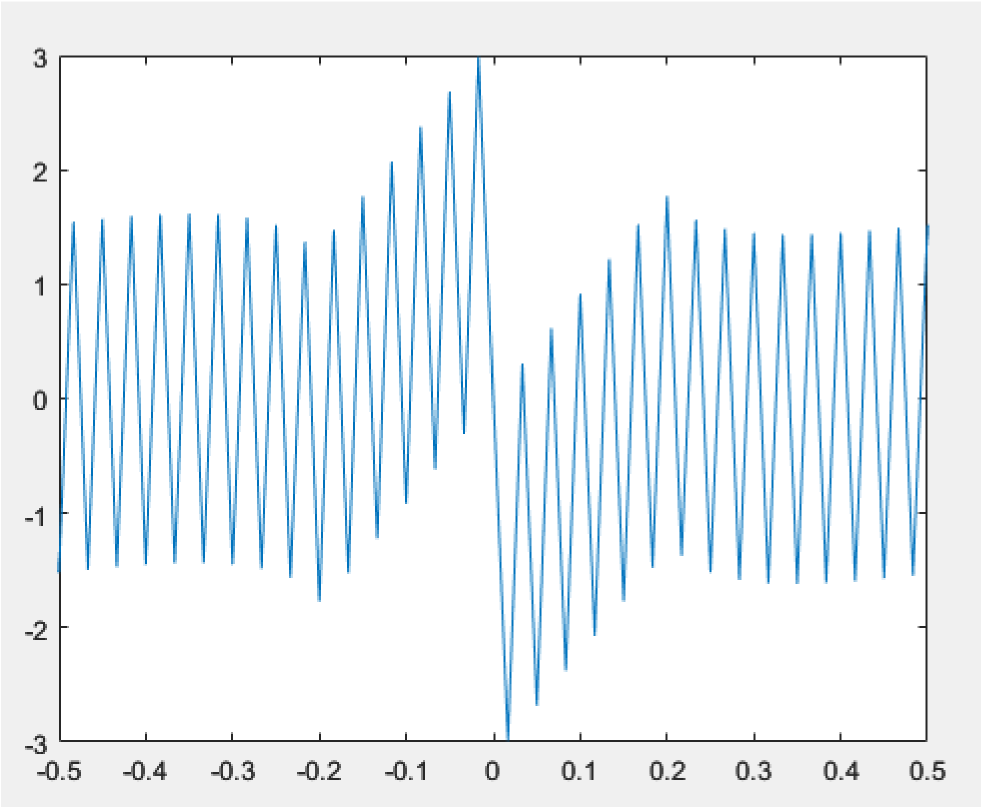
For the coding part: it is exactly same as the previous code, the only difference is changing from l=20 to l=30:

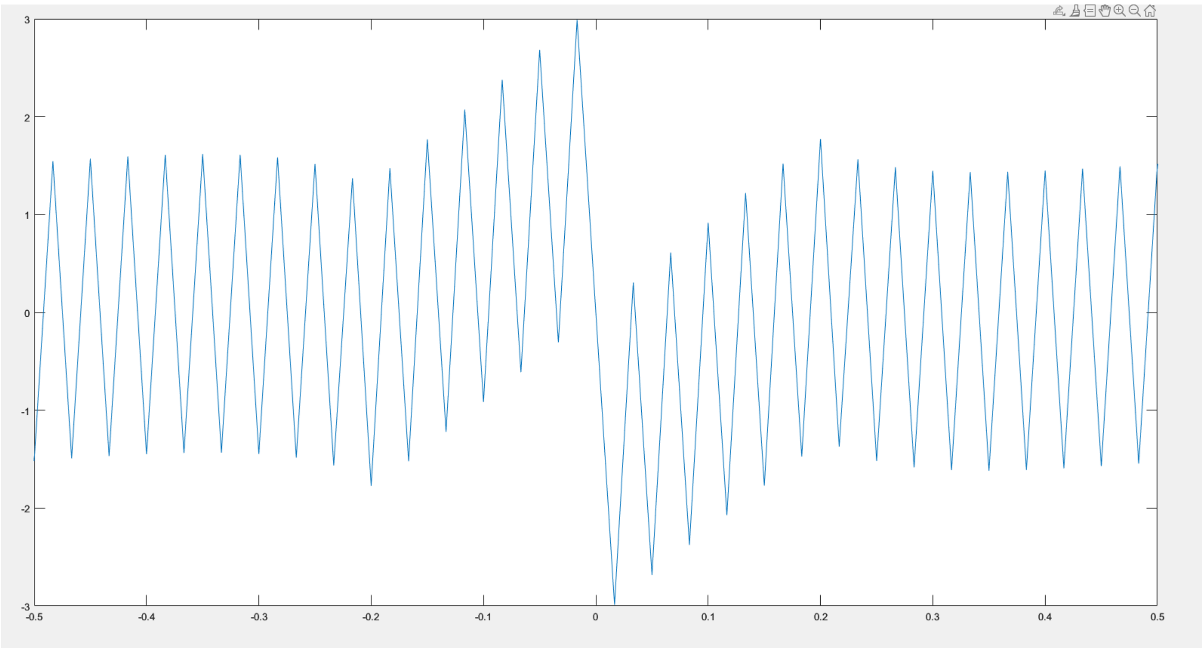
Computing the spectrum |G(f)|

The maximum phase of this spectrum is **2.98862**



For the graphing part:



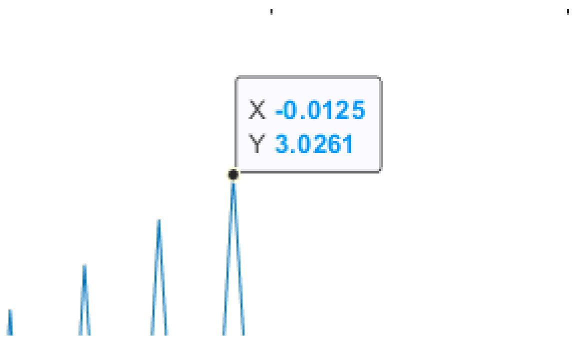


When l=40;

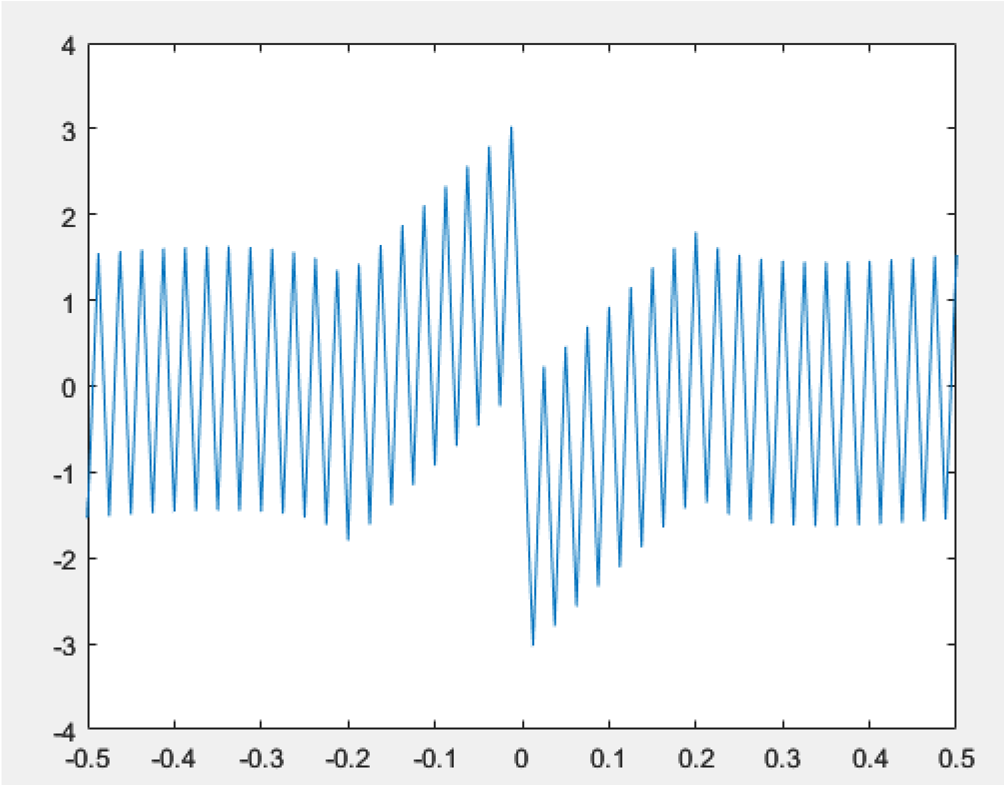
For the coding part: it should be similar to the previous code, the only difference is changing l=20 to l=40;

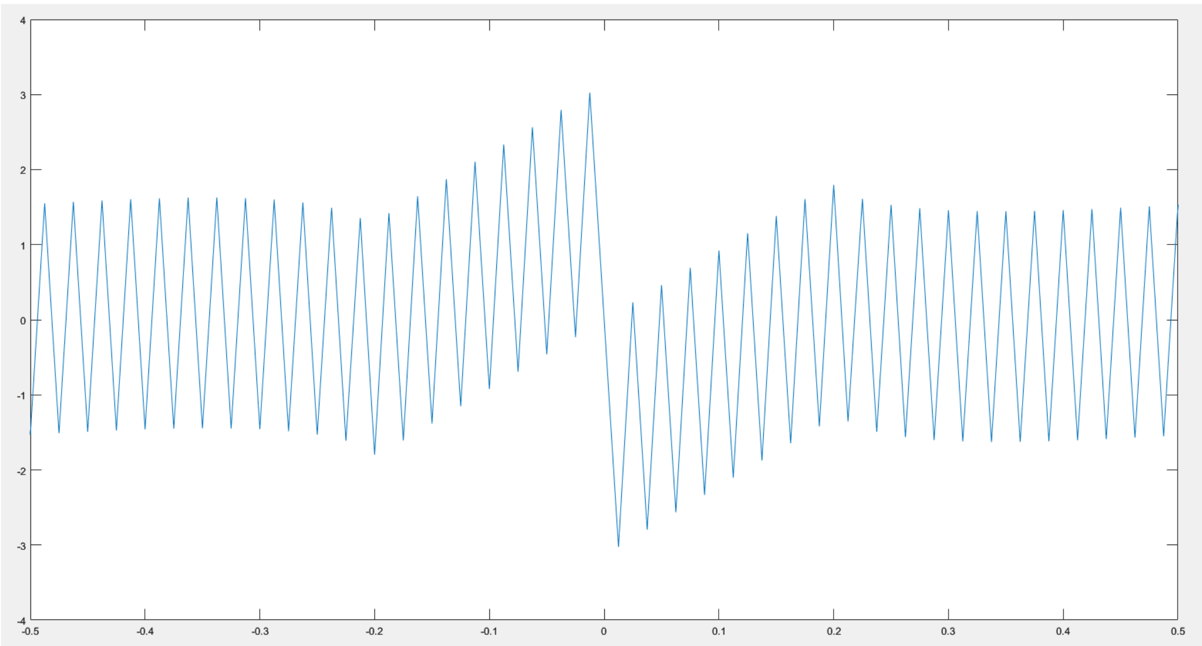
Computing the spectrum |G(f)|

The maximum phase of this spectrum is **3.0261**

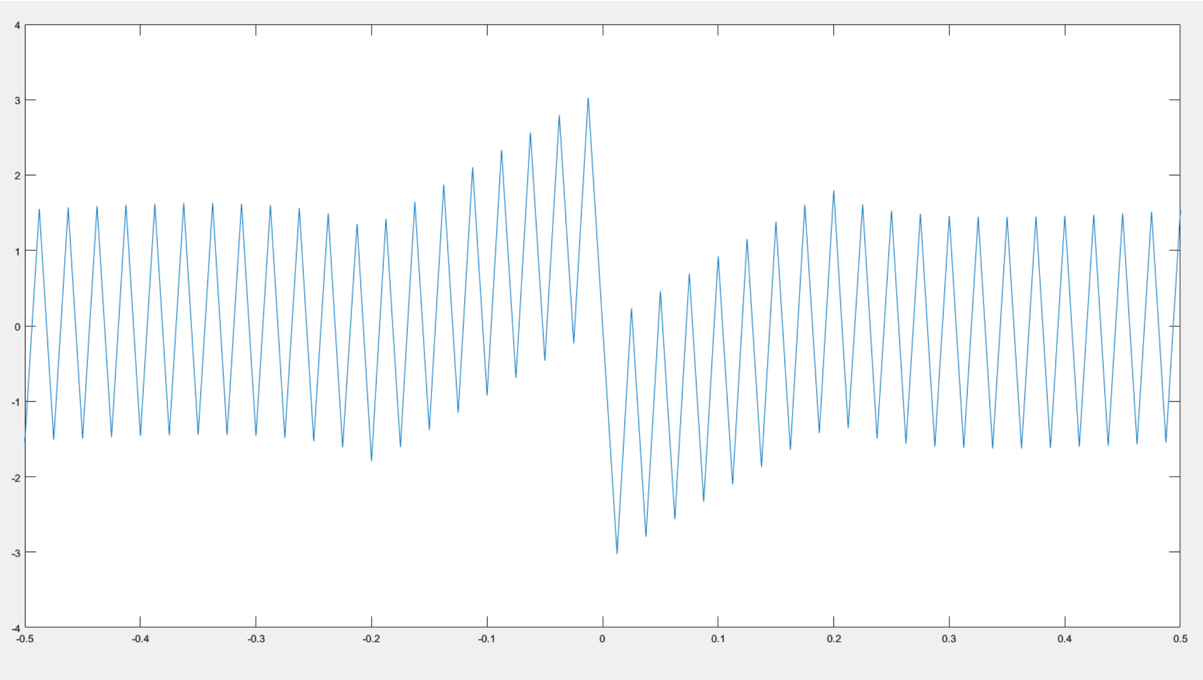


For the graphing part:


4
3
2
1
0
-1
-2
-3
-4
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5


4
3
2
1
0
-1
-2
-3
-4
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5

No, the phase isn't staying linear all the time as the phase spectrum oscillates with different amounts of phases when the frequency is changing from -0.5 to 0.5 because the phase spectrum is not constant, they keep changing when f is increasing so that's a reason why the phase spectrum is nonlinear.



6)

6) If the input applied to your designed filter is $x[n] = \cos(2\pi 0.1n) + 2\sin(2\pi 0.3n)$, what do you expect at the output of the filter? Can you verify it by computing and plotting $y[n] = x'[n] * g[n] = \sum_{m=0}^{2n_0} g[m]x'[n-m]$ where $x'[n] = \cos(2\pi 0.1n)u[n] + 2\sin(2\pi 0.3n)u[n]$? Here we make the nonzero value of $x'[n]$ to start from $n = 0$ because an ideal sinusoidal signal cannot be implemented exactly as it starts at $n = -\infty$. As n becomes large, the effect of the initial condition of $x'[n]$ on $y[n]$ will disappear.

The output of the filter:

For the x[n], the initial output of the filter will have a lot of 0 because the range for x(n) is from 1 to 50 compares to the output of y[n] which the range is from 1 to 1001.

For the output of x[n], the maximum peak is around 2.7113. The output of y[n] would be around 1.03102

For the output of x[n], as n gets larger and larger, the effect of the initial condition will disappear. As the observation from the graph, when l=20 (L is the boundary for n from l-1 to l+1), the ideal sinusoidal of the signal can be implemented. However, when the l>20, the ideal of the sinusoidal cannot be implemented because I got an error output that cannot compute the output of x[n] because the array must have an output result of positive integers or logical values.

When l=20:

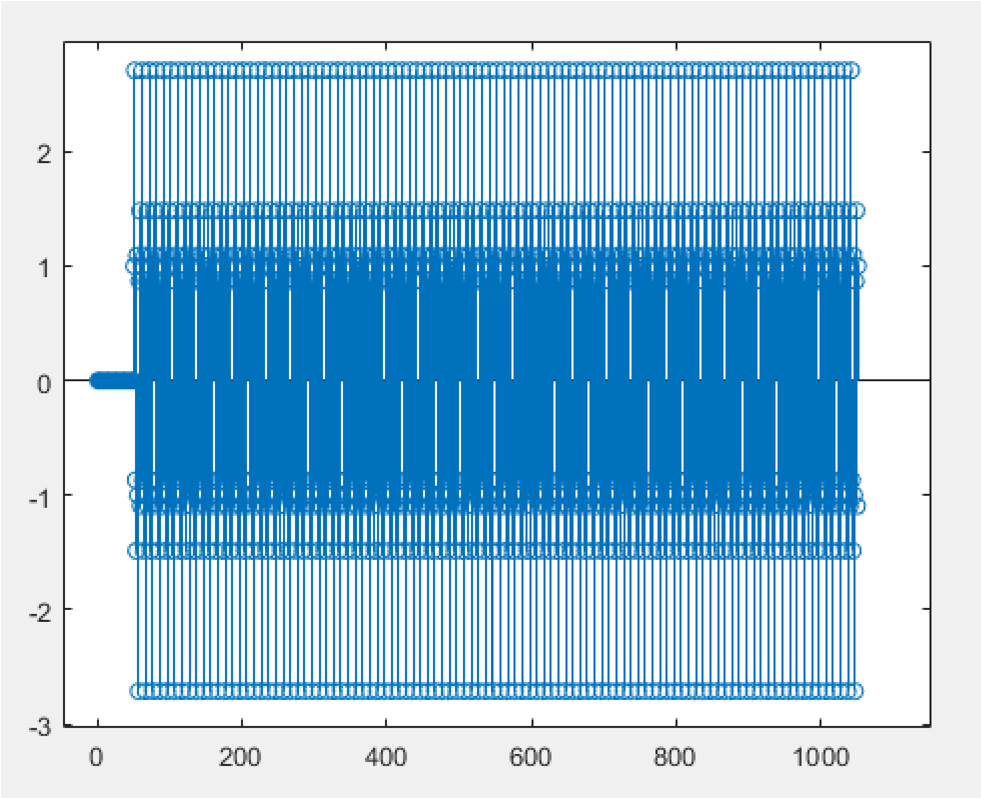
For the coding part:

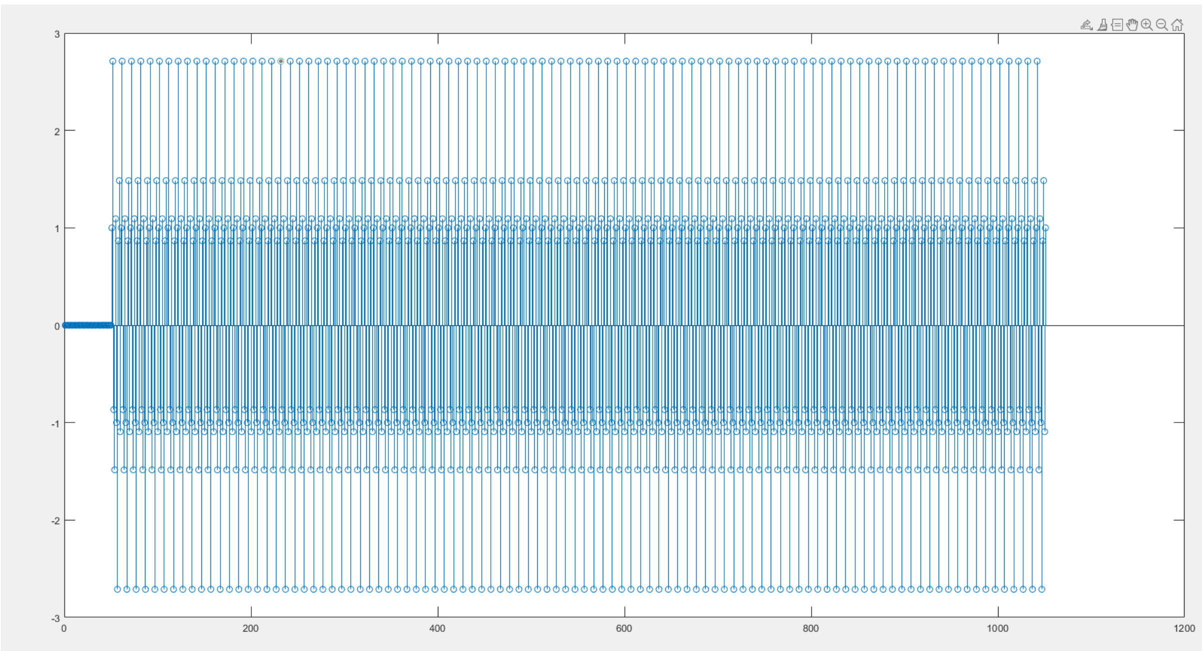
```
l=20;
n=-l-1:l+1;
n=0:1000;
xn1=cos(0.1*2*pi*n);
xn2=2*sin(0.3*2*pi*n);
xn=xn1+xn2;
xn=[zeros(1,50),xn];
yn=zeros(1,1001);
for k=0:1000
    for m=0:2*l
        yn(k+1)= yn(k+1)+g(m+1)*xn(k-m+51);
    end
end
stem(yn);
stem(xn);
```

For computing x(n)

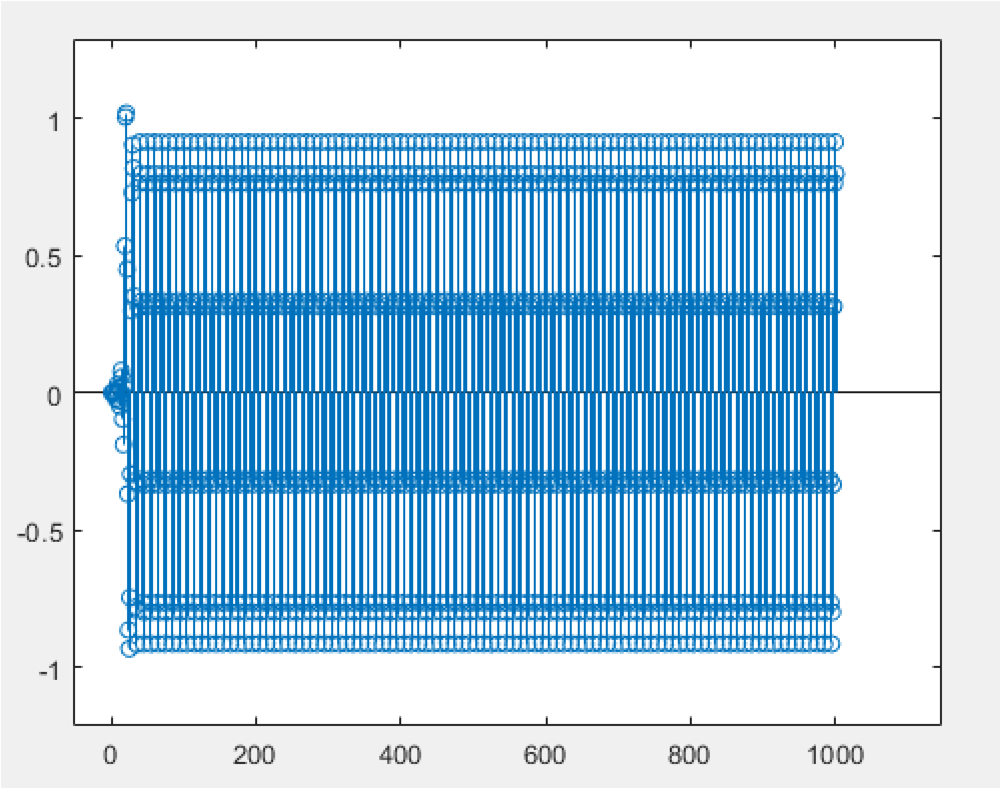
For computing y(n):

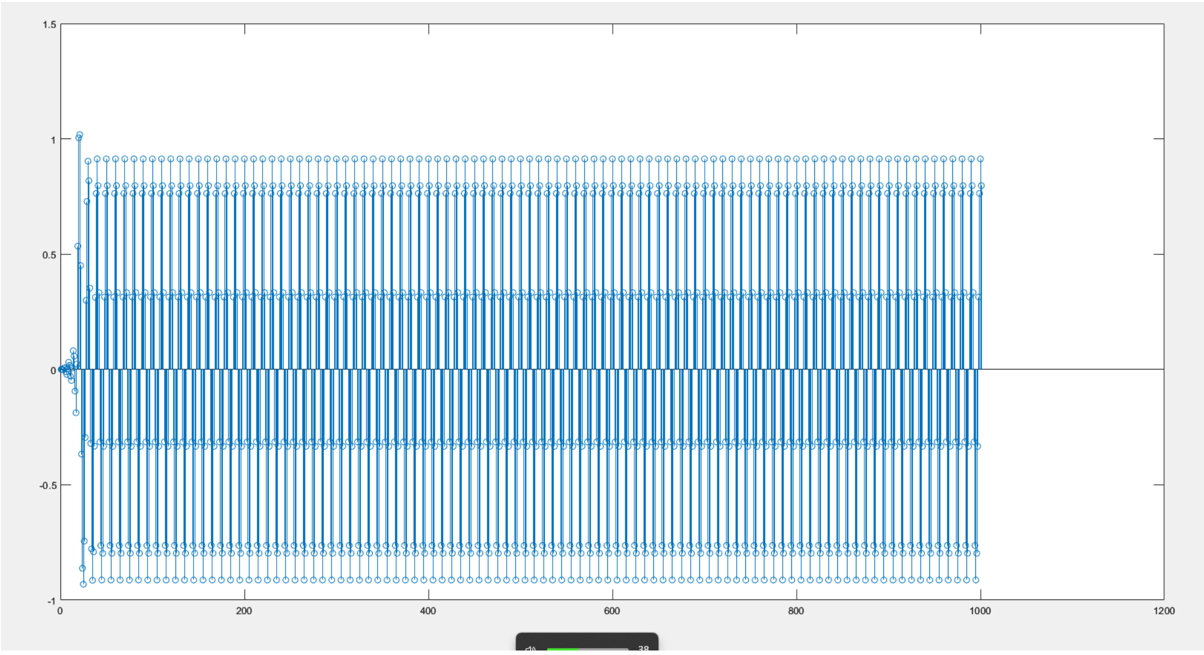
For the x(n) graph:





For the y(n) graph:


1
0.5
0
-0.5
-1
0
200
400
600
800
1000


1.5
1
0.5
0
-0.5
-1
0
200
400
600
800
1000
1200

When l=30:

For the coding part:

```
l=30;
n=-l-1:l+1;
n=0:1000;
xn1=cos(0.1*2*pi*n);
xn2=2*sin(0.3*2*pi*n);
xn=xn1+xn2;
xn=[zeros(1,50),xn];
yn=zeros(1,1001);
for k=0:1000
    for m=0:2*l
        yn(k+1)= yn(k+1)+g(m+1)*xn(k-m+51);
    end
end
```

Cannot compute the output of x[n] because the array must have an output result of positive integers or logical values.

When l=40:

For the coding part:

It's exactly the same as the coding part for l=30, the only difference is l=40

For the computing part:

Cannot compute the output of x[n] because the array must have an output result of positive integers or logical values.

e-> any l values are greater than 20 cannot able to compute or graph