James Saw

Problem 1
a)

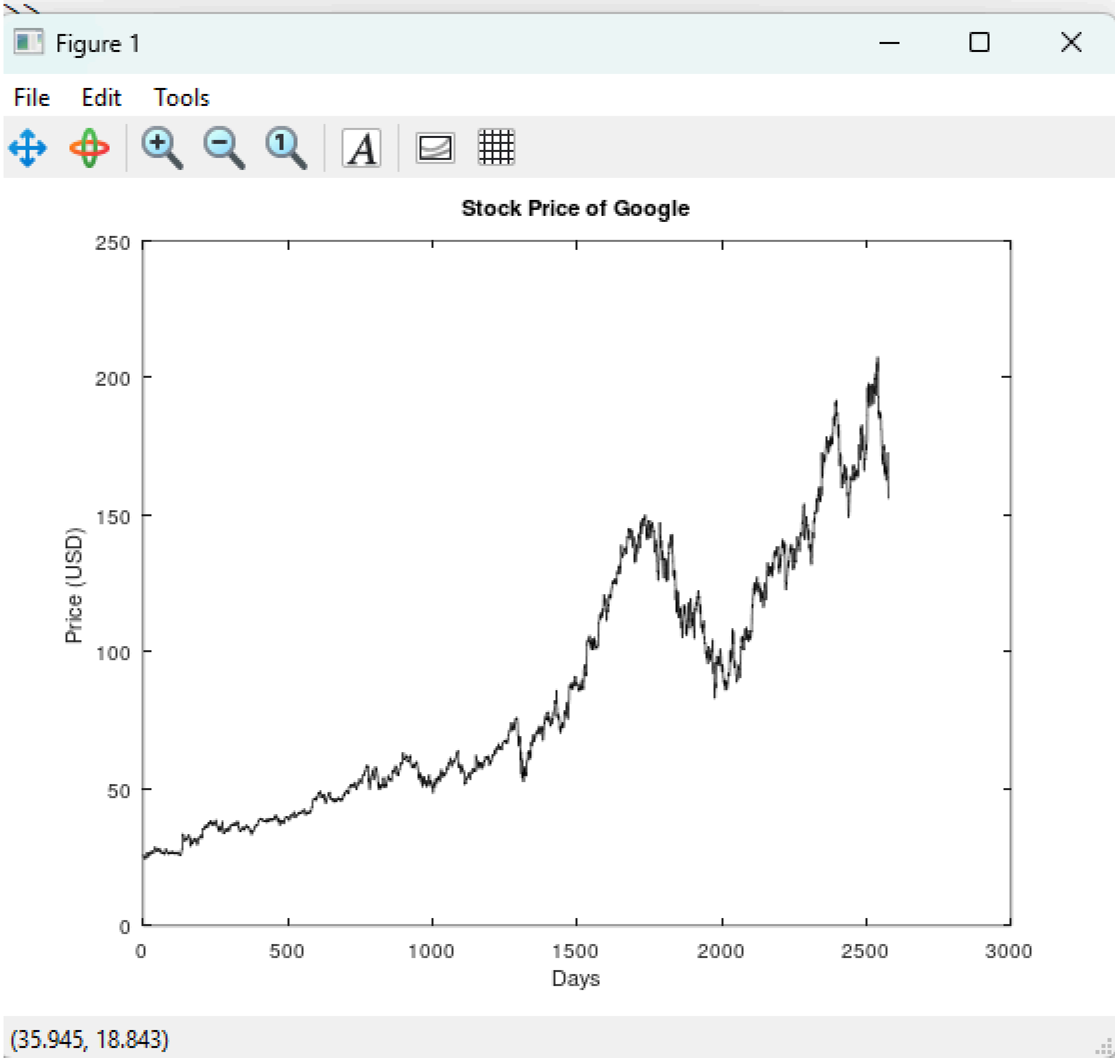
1) a) $$x(t) = \sum_{n=1,3,5}^{\infty} \frac{4}{n\pi} \sin\left(\frac{2\pi n t}{T}\right)$$

non-zero coefficients are

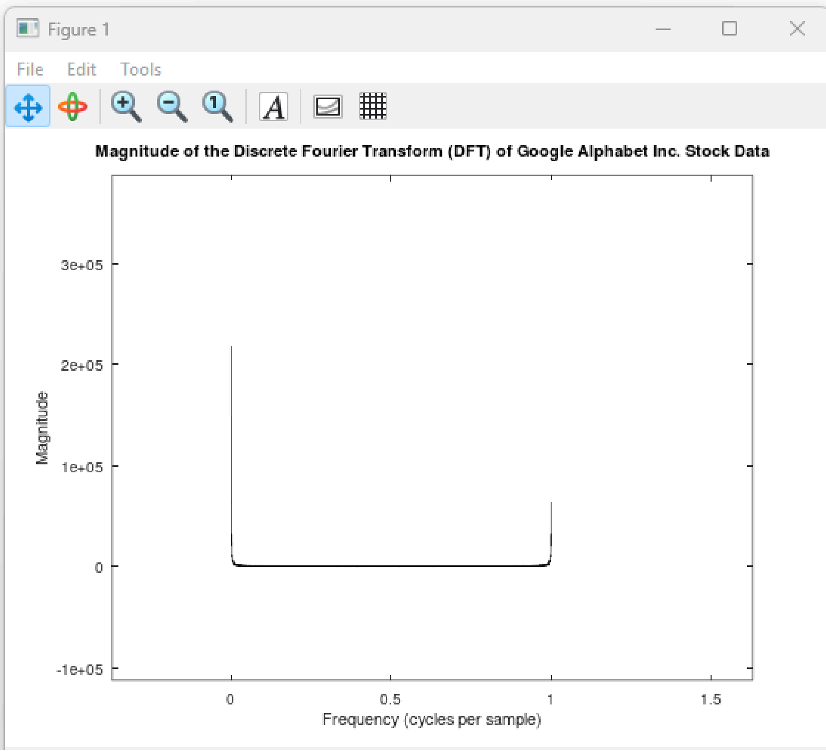
$$b_n = \frac{4}{n\pi} \text{ for odd integers of } n$$

1b, c, d)

```
>> file_path = "StockData_2015_2025.csv"
file_path = StockData_2015_2025.csv
>>
>> data = csvread(file_path);
>>
>> plot(data(:,1), data(:,2), '-', 'color', 'black');
>>
>> xlabel('Days');
>> ylabel('Price (USD)');
>> title('Stock Price of Google')
>>
```



```
>> stock_prices = data(:, 2);
>> N = length(stock_prices);
>> X = zeros(N, 1);
>>
>> % DFT
>> for k = 1:N
    sum_val = 0;
    for n = 1:N
        sum_val = sum_val + stock_prices(n) * exp(-2 * pi * i * (k - 1) * (n - 1) / N);
    end
    X(k) = sum_val;
end
>>
>> magnitude = abs(X);
>> frequencies = (0:N-1) / N;
>> plot(frequencies, magnitude, '-k');
>> xlabel('Frequency (cycles per sample)');
>> ylabel('Magnitude');
>> title('Magnitude of the Discrete Fourier Transform (DFT) of Google Alphabet Inc. Stock Dat
a');
>> |
```

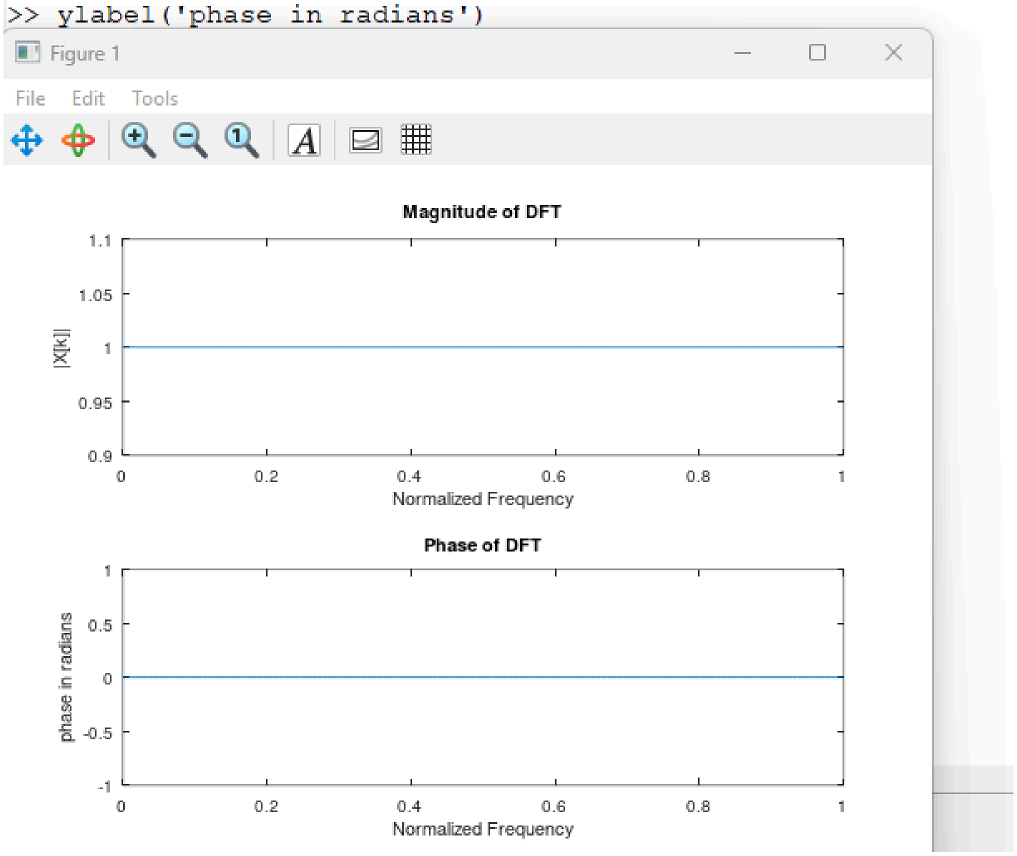


Problem 2

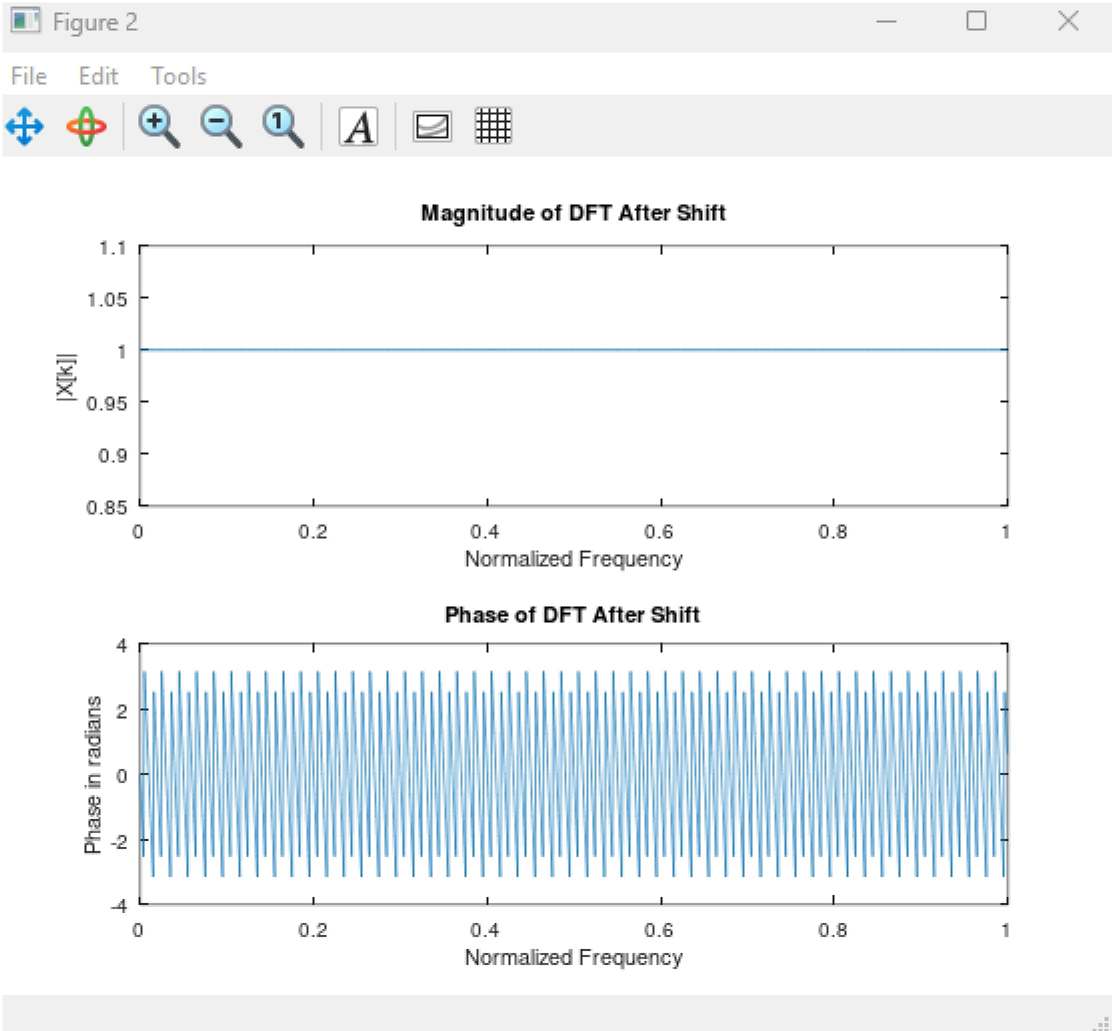
The Gibbs effect is observable as there are squarewaves with discontinuities

Problem 3ab)

```
>> N = 1000;
>> delta = zeros(1, N);
>> delta(1) = 1;
>>
>> X = fft(delta);
>> magX = abs(X);
>> phaseX = angle(X);
>> f = (0:N-1) * (1/N);
>> figure;
>>
>> subplot(2,1,1);
>> plot(f, magX);
>> title('Magnitude of DFT');
>> xlabel('Normalized Frequency')
>> ylabel('|X[k]|');
>>
>> subplots(2,1,2);
error: 'subplots' undefined near line 1, column 1
>> subplot(2,1,2);
>> plot(f, phaseX);
>> title('Phase of DFT')
>> xlabel('Normalized Frequency')
>> ylabel('|X[k]|');
>> ylabel('phase in radians')
```



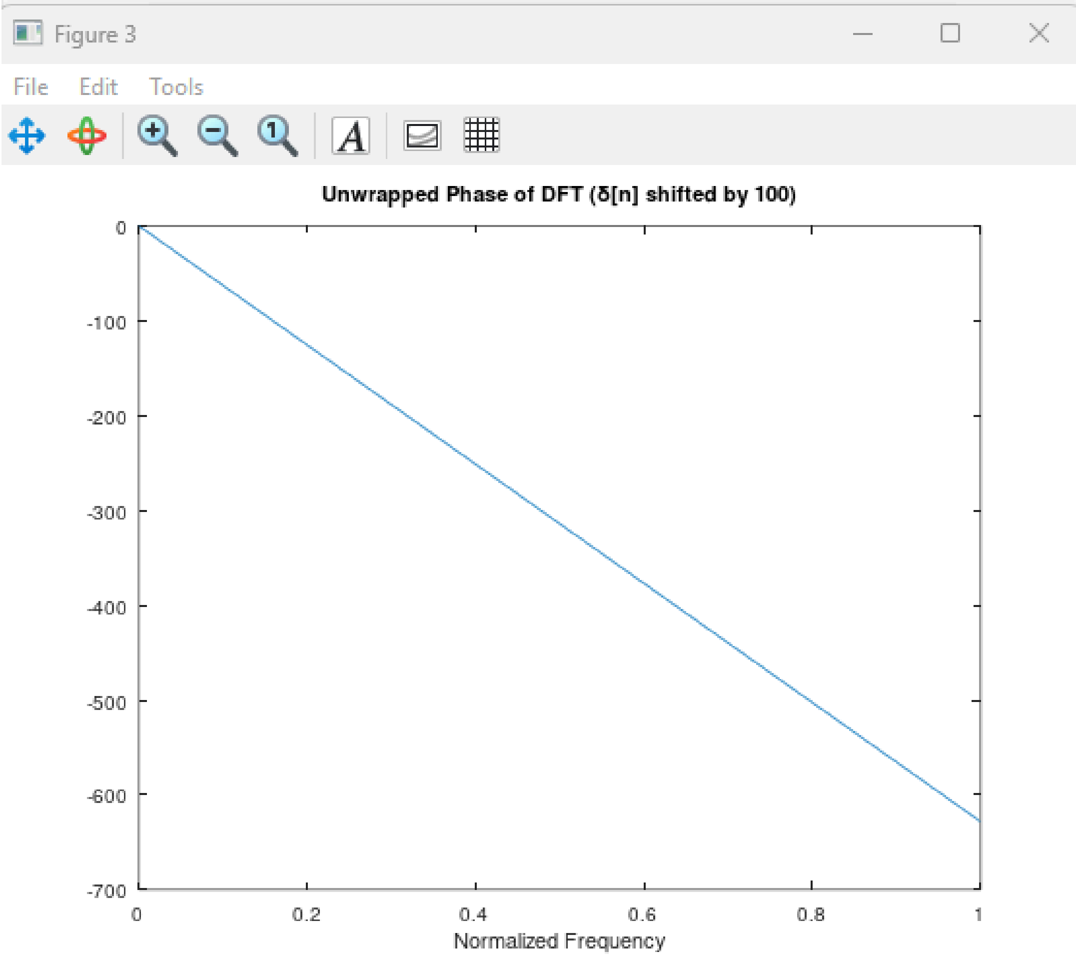
3c




```
>> figure;
>> subplot(2,1,1);
>> plot(f, magX2);
>> title('Magnitude of DFT After Shift')
>> xlabel('Normalized Frequency');
>> ylabel('|X[k]|');
>>
>> subplot(2,1,2);
>> plot(f, phaseX2);
>> title('Phase of DFT After Shift');
>> xlabel('Normalized Frequency');
>> ylabel('Phase in radians');
>>
```

3d)

```
>> unwrapped_phase = unwrap(phaseX2);
>> figure;
>> plot(f, unwrapped_phase);
>> title('Unwrapped Phase of DFT (δ[n] shifted by 100)');
>> xlabel('Normalized Frequency');
>> ylabel('Phase (radians)');
```



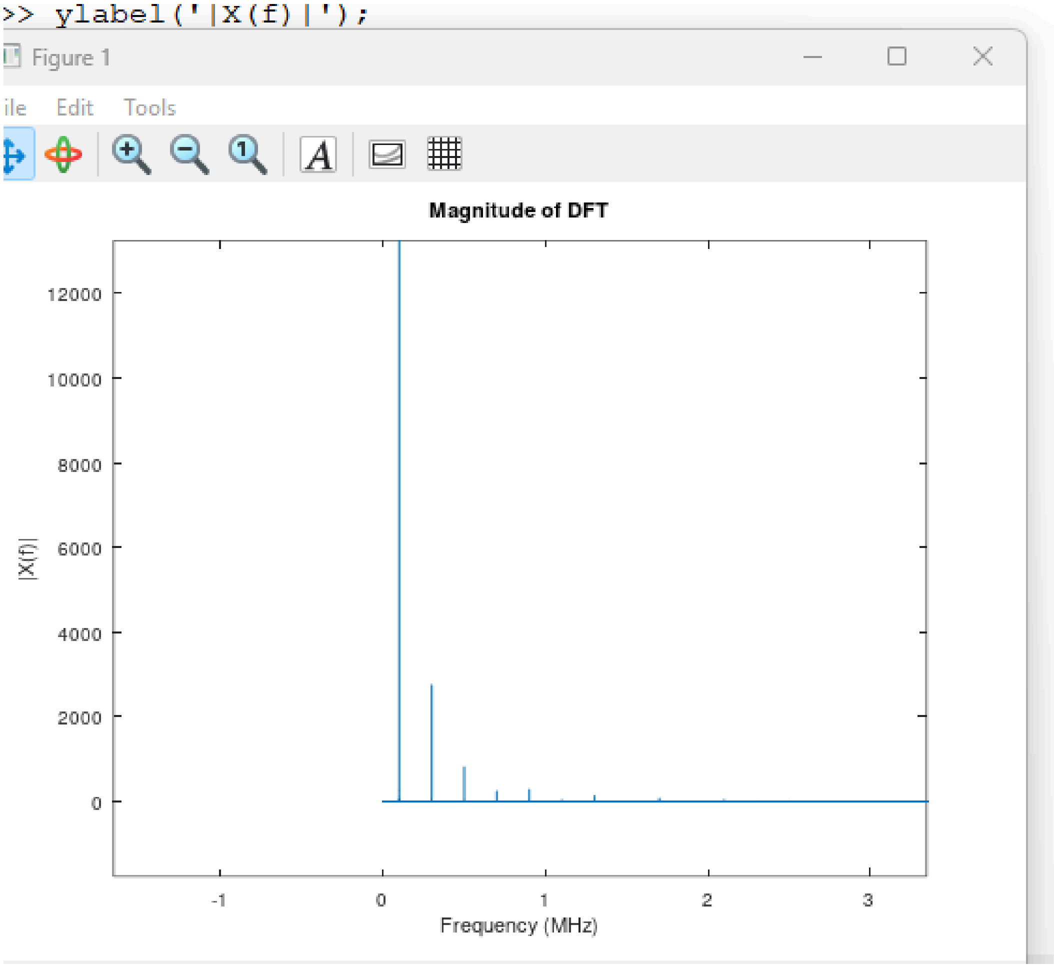
3e

```
>> dphi = diff(unwrapped_phase);
>> df = diff(f);
>> group_delay = -mean(dphi ./ df);
>> disp(['Estimated group delay: ', num2str(group_delay)]);
Estimated group delay: 628.3185
>> 
```

I believe in the result that I obtained that this is roughly correct as 628 was concluded without being divided by 2pi which should come to the expected number of around ≈ 100

Problem 4

```
>> Fs = 10e6;
>> f = 100e3;
>> Tmax = 6e-3;
>> A = 1.0;
>> t = 0:1/Fs:Tmax;
>> x = A * sin(2 * pi * f * t);
>> length(x)
ans = 60001
>>
>> x(find(x > 0.75)) = 0.75;
>> x(find(x < -0.75)) = -0.75;
>>
>> N = length(x);
>> X = fft(x);
>> f_axis = (0:N-1) * Fs / N;
>>
>> figure;
>> plot(f_axis/1e6, abs(X));
>> title('Magnitude of DFT');
>> xlabel('Frequency (MHz)');
>> ylabel('|X(f)|');
```

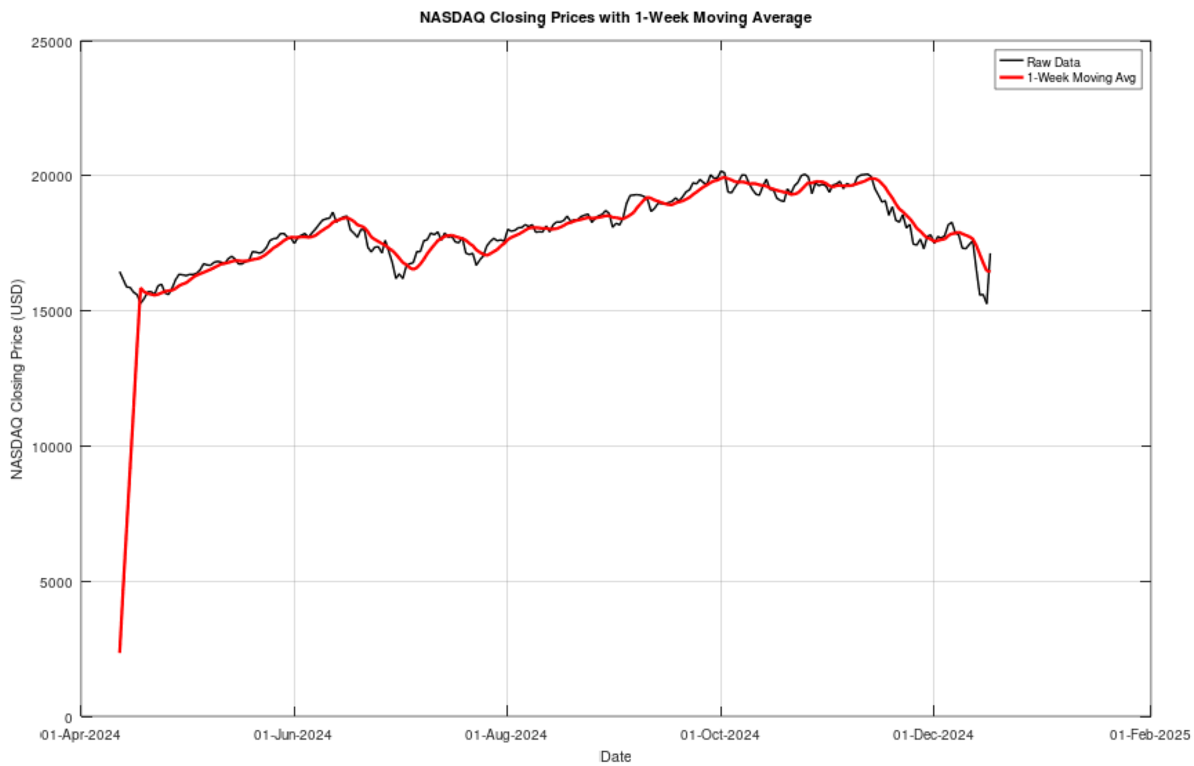


4d)

I do observe harmonics based on the plot that was made. The fundamental frequency is shown to be around 0.1 MHz which corresponds to the 100 kHz frequency. Frequencies at 300 kHz are noticeably smaller and 500 kHz are shorter than 300 kHz.

Problem 5

```
clear;
close all;
pkg load signal;

filename = 'NASDAQ.txt';
fid = fopen(filename, 'r');
fgetl(fid);
data = fscanf(fid, '%f %f', [2, Inf])';  % Read two columns
fclose(fid);

days = data(:,1);
prices = data(:,2);

% 1-week moving average filter (7-day window)
window_size = 7;
b = ones(1, window_size)/window_size;  % Simple averaging filter
a = 1;

smoothed = filter(b, a, prices);

% Convert day numbers to dates starting from April 10, 2024
start_date = datenum('10-Apr-2024');
dates = start_date + days;

% Plot raw and smoothed data
figure('position', [100, 100, 1000, 600]);
plot(dates, prices, 'k', 'LineWidth', 1.2); hold on;
plot(dates, smoothed, 'r', 'LineWidth', 2);
datetick('x', 'dd-mmm-yyyy');
xlabel('Date');
ylabel('NASDAQ Closing Price (USD)');
title('NASDAQ Closing Prices with 1-Week Moving Average');
legend('Raw Data', '1-Week Moving Avg');
grid on;
```

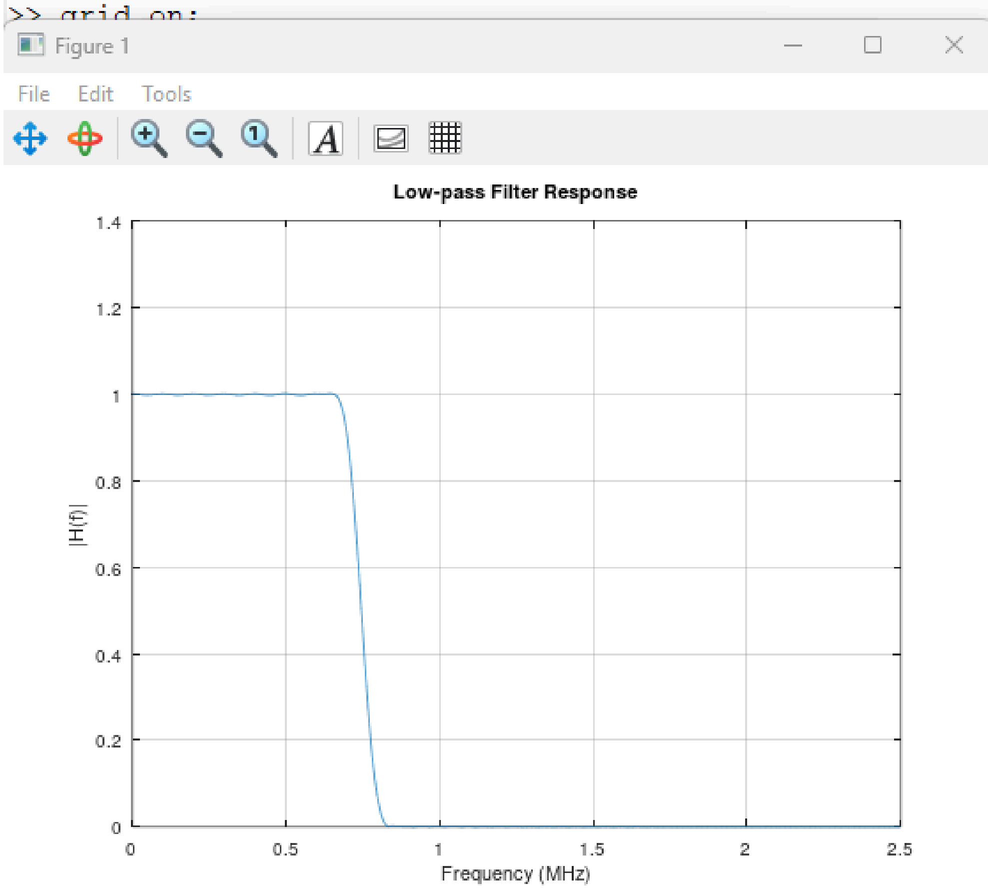


5d)

The moving average does show rapidly changing shifts in the economy. While the lines may dip slightly it doesn't properly show sharp dips in the economy. This is mainly because the the average only represents the data based on long term changes over time not immediate sudden changes.

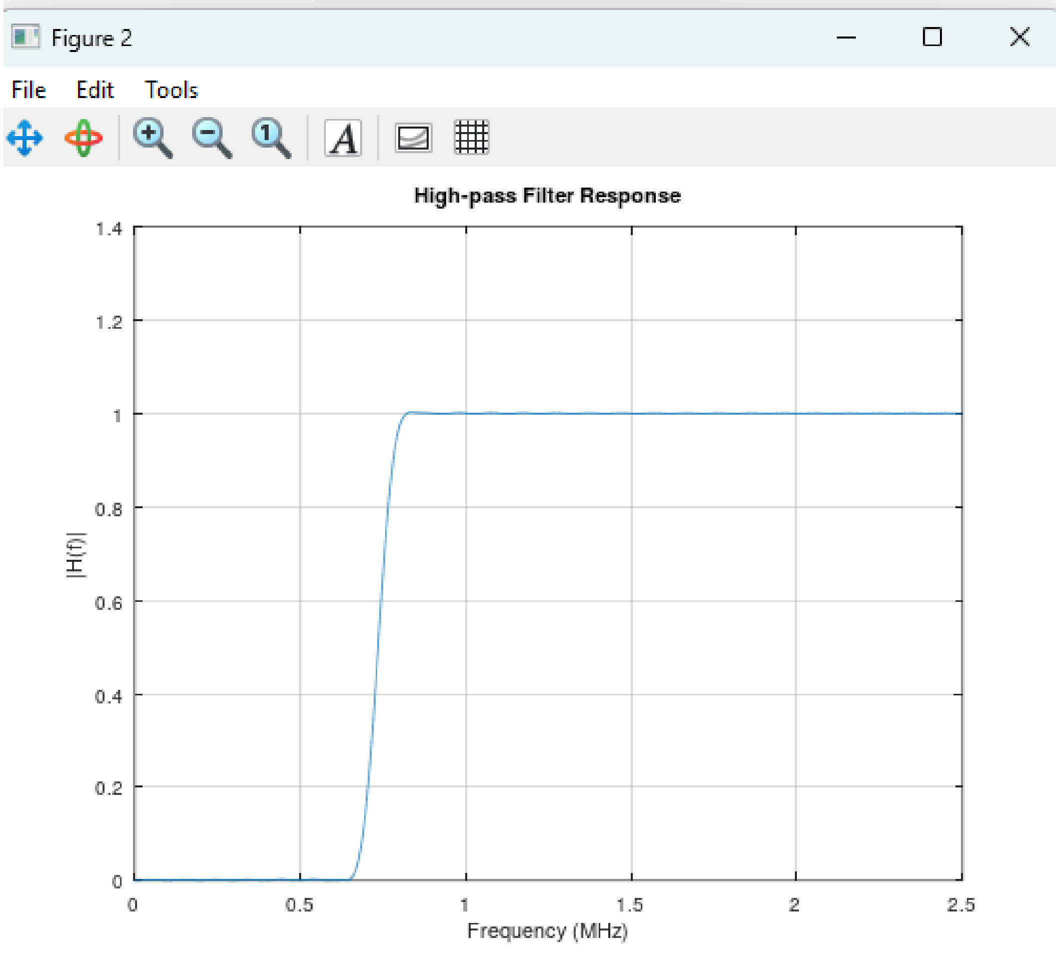
Problem 6a)

```
>> Fs = 5e6;
>> fc_low = 745e3;
>> M = 101;
>> n = 0:M-1;
>> fc_norm = fc_low / (Fs/2);
>>
>> h_low = fc_norm * sinc(fc_norm * (n - (M-1)/2));
>> h_low = h_low .* hamming(M)';
>> h_low = h_low / sum(h_low);
>>
>> [H_low, f] = freqz(h_low, 1, 1024, Fs);
>> figure;
>> plot(f/1e6, abs(H_low));
>> title('Low-pass Filter Response);
error: parse error:

  syntax error

>>> title('Low-pass Filter Response);
                                   ^
>> title('Low-pass Filter Response');
>> xlabel('Frequency (MHz)');
>> ylabel('|H(f)|');
>> grid on;
```



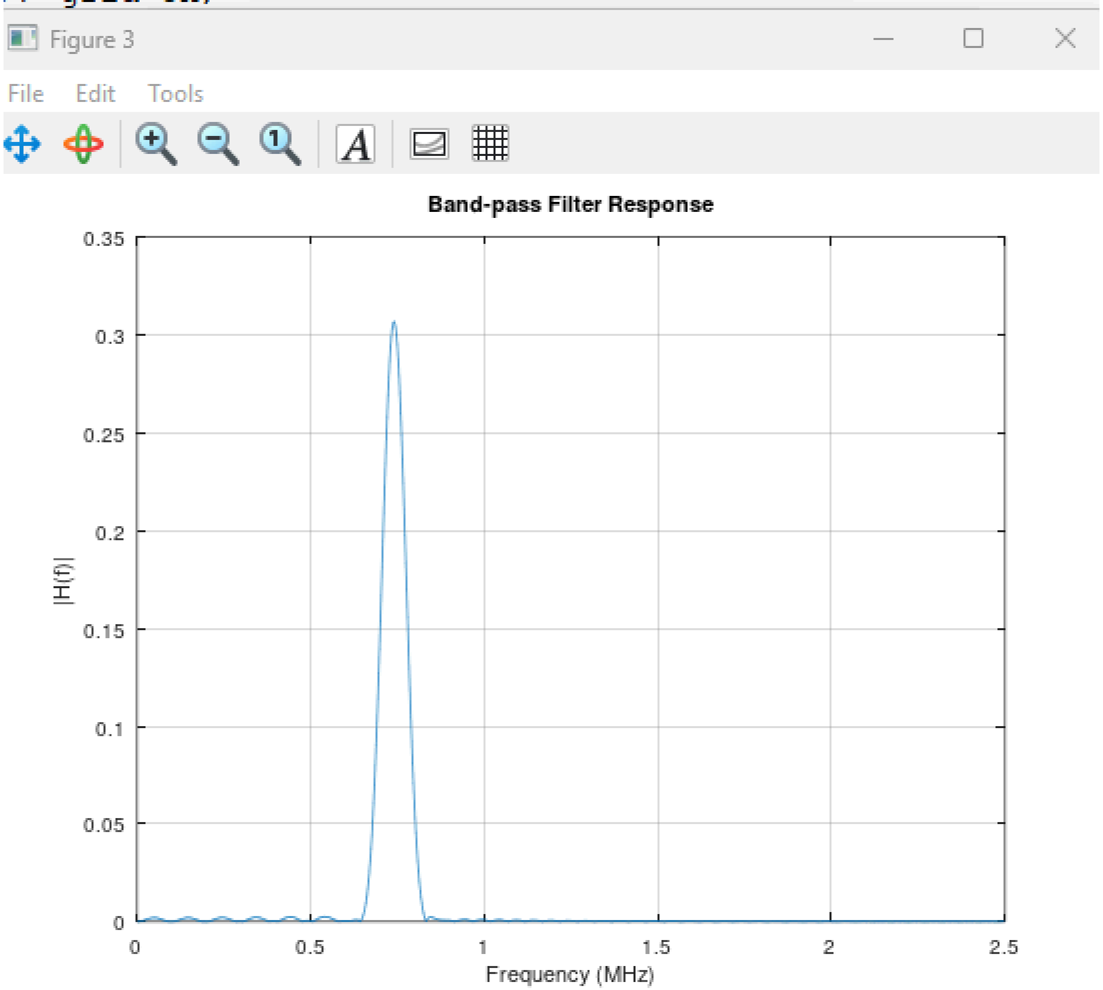
6b

```
>> fc_high = 735e3;
>> fc_norm_hp = fc_high / (Fs/2);
>>
>> %LP
>> h_lp = fc_norm_hp * sinc(fc_norm_hp * (n - (M-1)/2));
>> h_lp = h_lp .* hamming(M)';
>> h_lp = h_lp / sum(h_lp);
>>
>> %inversion
>> h_hp = -h_lp;
>> h_hp((M+1)/2) += 1;
>>
>> % Freq. Response
>> [H_hp, f] = freqz(h_hp, 1, 1024, Fs);
>> figure;
>> plot(f/1e6, abs(H_hp));
>> title('High-pass Filter Response');
>> xlabel('Frequency (MHz)');
>> ylabel('|H(f)|');
>> grid on;
```



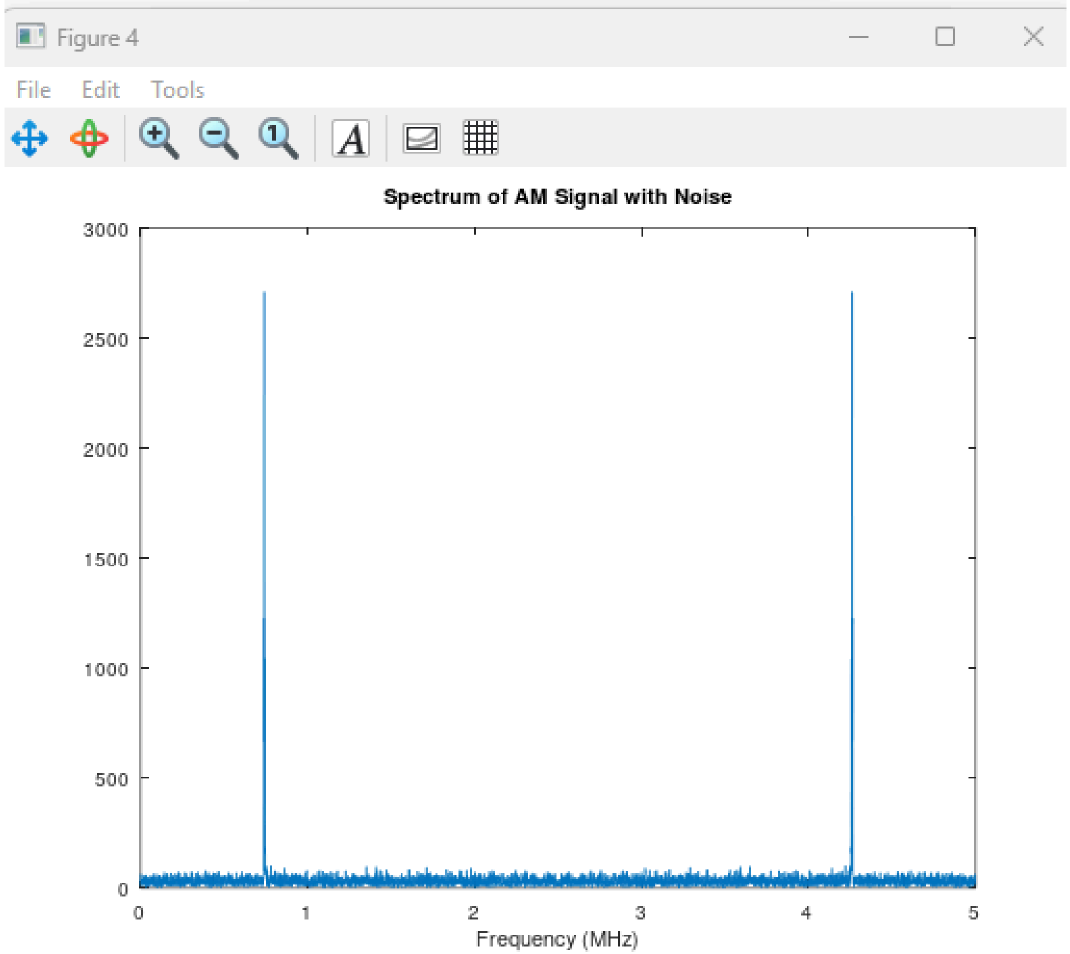
6c

```
>> h_bp = conv(h_low, h_hp);
>> [H_bp, f_bp] = freqz(h_bp, 1, 1024, Fs);
>>
>> figure;
>> plot(f_bp/1e6, abs(H_bp));
>> title('Band-pass Filter Response');
>> xlabel('Frequency (MHz)');
>> ylabel('|H(f)|');
>> grid on;
```



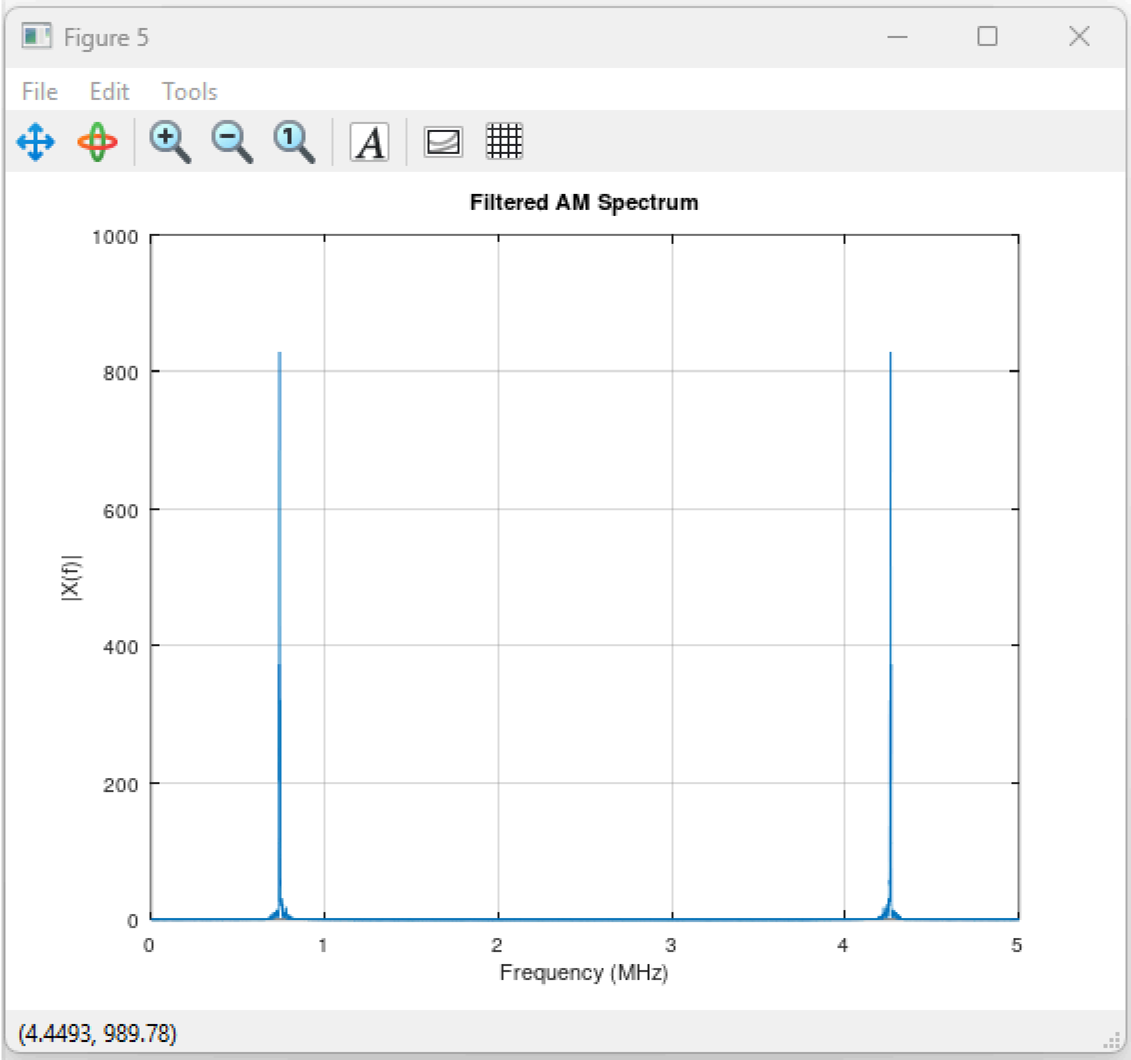
6d

```
>> T = 1e-3;
>> t = 0:1/Fs:T;
>> fc = 740e3;
>> f_audio = 2.5e3;
>> audio = sin(2*pi*f_audio*t);
>> noise = 0.5*randn(size(t));
>>
>> am_signal = (1 + audio) .* cos(2*pi*fc*t) + noise;
>>
>> %DFT
>> N = length(am_signal);
>> AM_spec = fft(am_signal);
>> f_axis = (0:N-1)*Fs/N;
>>
>> figure;
>> plot(f_axis/1e6, abs(AM_spec));
>> title('Spectrum of AM Signal with Noise');
>> xlabel('Frequency (MHz)');
>> ylabel('|X(f)|');
```



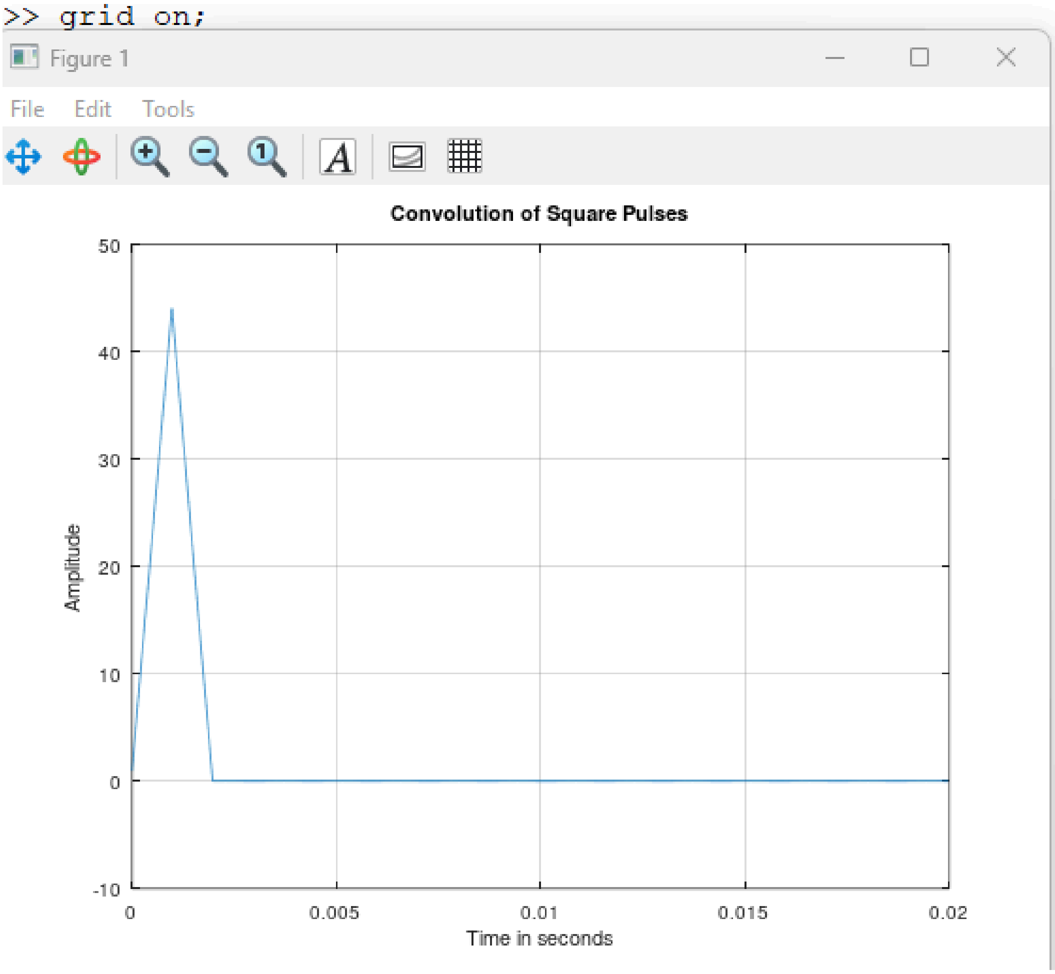
6e

```
>> am_filtered = conv(am_signal, h_bp, 'same');
>>
>> AM_filt_spec = fft(am_filtered);
>> figure;
>> plot(f_axis/1e6, abs(AM_filt_spec));
>> title('Filtered AM Spectrum');
>> xlabel('Frequency (MHz)');
>> ylabel('|X(f)|');
>> grid on;
>>
```



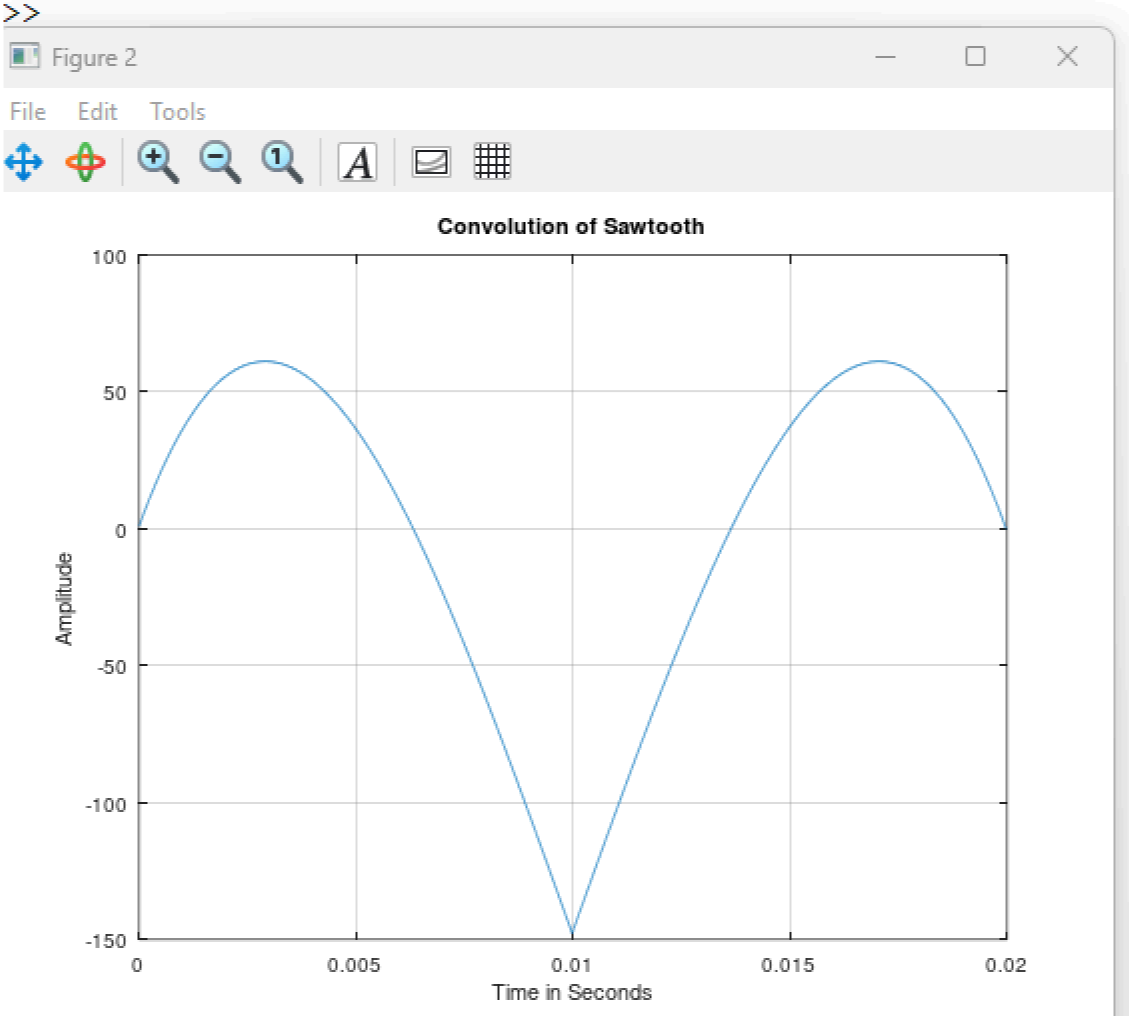
Problem 7a)

```
>> Fs = 44100;
>> T = 0.01;
>> N = round(Fs * T);
>>
>> % Square Pulse
>>
>> pulse = zeros(1, N);
>> pulse(1:N/10) = 1;
>>
>> % Convulsion
>> L = 2*N;
>> y = ifft(fft(pulse, L) .* fft(pulse, L));
>>
>> t = (0:L-1)/Fs;
>> figure;
>> plot(t, y);
>> title('Convolution of Square Pulses');
>> xlabel('Time in seconds');
>> ylabel('Amplitude');
>> grid on;
```



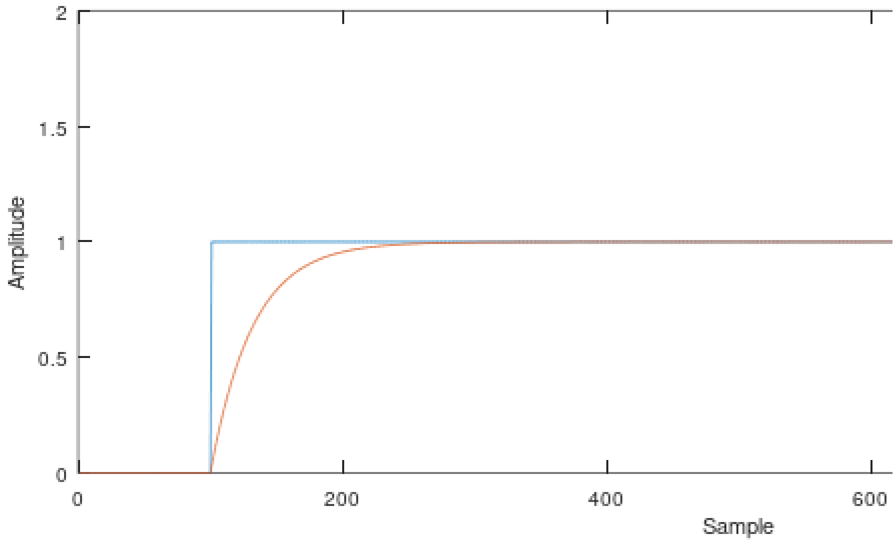
7b)

```
>> saw = linspace(-1, 1, N);
>>
>> y2 = ifft(fft(saw, L) .* fft(saw, L));
>>
>> figure;
>> plot(t, y2);
>> title('Convolution of Sawtooth')
>> xlabel('Time in Seconds');
>> ylabel('Amplitude');
>> grid on
>>
```



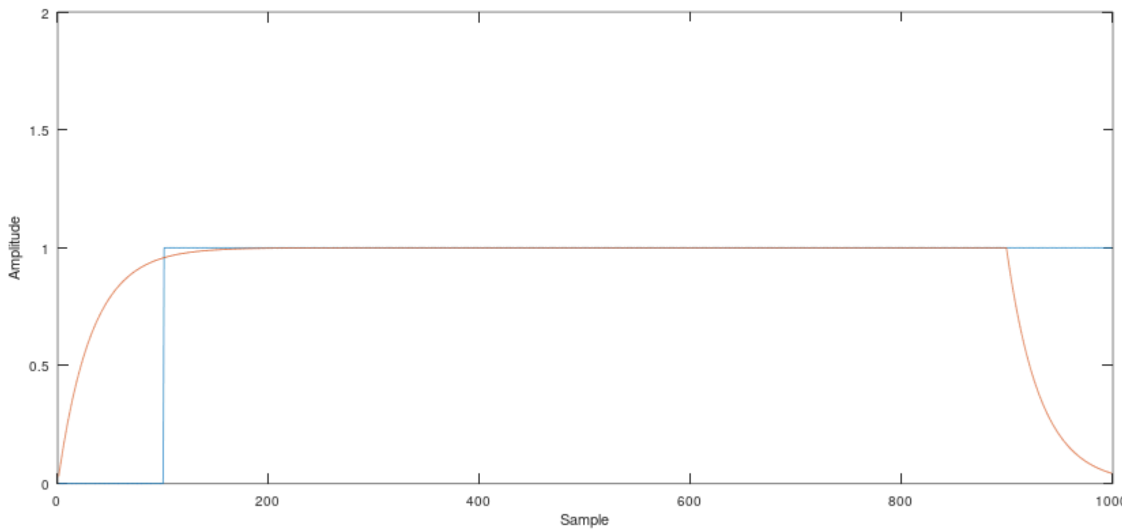
Problem 8abc)

```
clear;
close;
home;
pkg load signal;

%dsp parameters for x-axis:
fs = 1.0e4; %Hz
fc = 0.5e2; %Hz
fc = fc/fs;
x = exp(-2*pi*fc);
a0 = 1-x;
b1 = x;
N = 1000;
M = 100;

%Low-pass filtering
st = [zeros(M,1); ones(N-M,1);];
y = zeros(size(st));
y(1) = a0*st(1);
for i = 2:N
  y(i) = a0*st(i)+b1*y(i-1);
endfor

%Plotting section
figure(1, 'position',[0,0,1000,1000]);
subplot(2,1,1);
plot(st);
hold on;
plot(y);
axis([0 N+1 0 2]);
xlabel('Sample');
ylabel('Amplitude');
set(gca());
```


2
1.5
1
0.5
0
Amplitude
0
200
400
600
Sample

Problem 9ab

```
%Problem 9a
clear;
close;
home;
pkg load signal;

%dsp parameters for x-axis:
fs = 1.0e4; %Hz
fc = 0.5e2; %Hz
fc = fc/fs;
x = exp(-2*pi*fc);
a0 = 1-x;
b1 = x;
N = 1000;
M = 100;

%Step Pulse
st = [zeros(M,1); ones(N-M,1);];

%Reverse Step Pulse
st_reversed = flipud(st);

% Low Pass filter w/reversed step pulse
y = zeros(size(st_reversed));
y(1) = a0*st_reversed(1);
for i = 2:N
    y(i) = a0*st_reversed(i) + b1*y(i-1);
endfor

% Plotting section
figure(1, 'position',[0,0,1000,1000]);
subplot(2,1,1);
plot(st);  % Plot the original step pulse
hold on;
plot(y);  % Plot the filtered output of the reversed pulse
axis([0 N+1 0 2]);
xlabel('Sample');
ylabel('Amplitude');
set(gca());
```

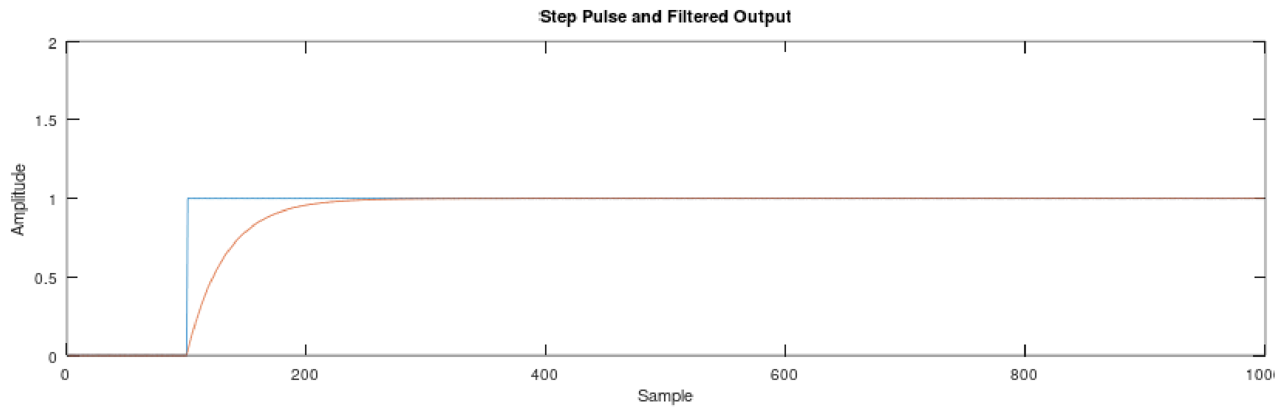


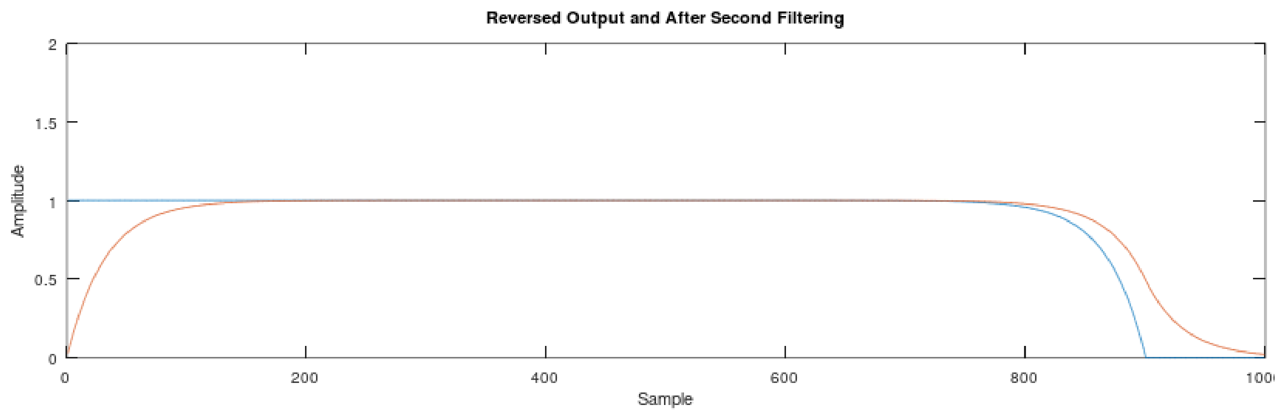
2
1.5
1
0.5
0
Amplitude
0
200
400
600
800
1000
Sample

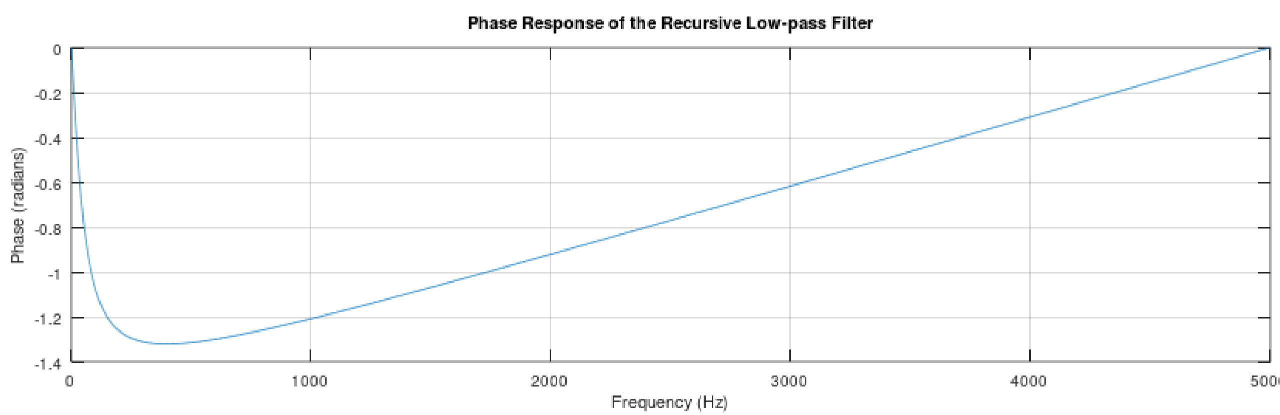
9c

```
% Problem 9c
clear;
close;
home;
pkg load signal;

%dsp parameters for x-axis:
fs = 1.0e4; %Hz
fc = 0.5e2; %Hz
fc = fc/fs;
x = exp(-2*pi*fc);
a0 = 1-x;
b1 = x;
N = 1000;
M = 100;

% Step Pulse
st = [zeros(M,1); ones(N-M,1);];

% Low-pass filtering (using the original step pulse)
y = zeros(size(st));
y(1) = a0*st(1);
for i = 2:N
    y(i) = a0*st(i) + b1*y(i-1);
endfor

% Reverse the filtered output
y_reversed = flipud(y);

% Run the reversed output through the recursive low-pass filter again
y_final = zeros(size(y_reversed));
y_final(1) = a0*y_reversed(1);
for i = 2:N
    y_final(i) = a0*y_reversed(i) + b1*y_final(i-1);
endfor

% Plotting section
figure(1, 'position',[0,0,1000,1000]);
```

```
% First subplot: Original step pulse and the filtered output
subplot(3,1,1);
plot(st);
hold on;
plot(y);
axis([0 N+1 0 2]);
xlabel('Sample');
ylabel('Amplitude');
title('Step Pulse and Filtered Output');
set(gca());

% Second subplot: Reverse the filtered output and run through filter again
subplot(3,1,2);
plot(y_reversed);
hold on;
plot(y_final);
axis([0 N+1 0 2]);
xlabel('Sample');
ylabel('Amplitude');
title('Reversed Output and After Second Filtering');
set(gca());

% Third subplot: Phase Response of the original filter
subplot(3,1,3);
[H, f] = freqz([a0], [1, -b1], N, fs);
plot(f, angle(H));
xlabel('Frequency (Hz)');
ylabel('Phase (radians)');
title('Phase Response of the Recursive Low-pass Filter');
grid on;
```







9c)

Based on the phase response of the recursive low-Pass Filter, I observed that the line becomes linear so the phase response is linear or becomes linear