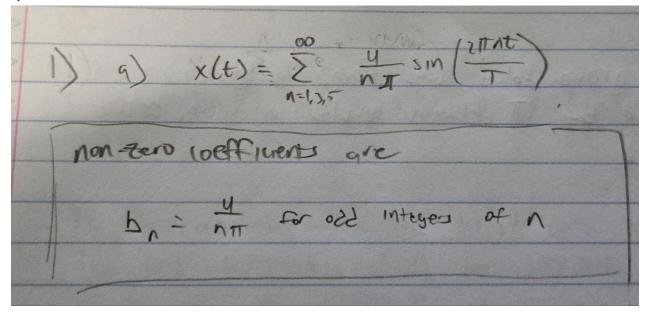
James Saw

Problem 1

a)



```
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')
                                                           ×
Figure 1
File Edit Tools
                            Stock Price of Google
     250
     200
     150
   Price (USD)
     100
      50
       0
                                          2000
                                                   2500
        0
                500
                        1000
                                 1500
                                                            3000
                                 Days
(35.945, 18.843)
```

```
>> 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');
      Figure 1
>> |
      File Edit Tools
      Magnitude of the Discrete Fourier Transform (DFT) of Google Alphabet Inc. Stock Data
         3e+05
          2e+05
          1e+05
            0
         -1e+05
```

Frequency (cycles per sample)

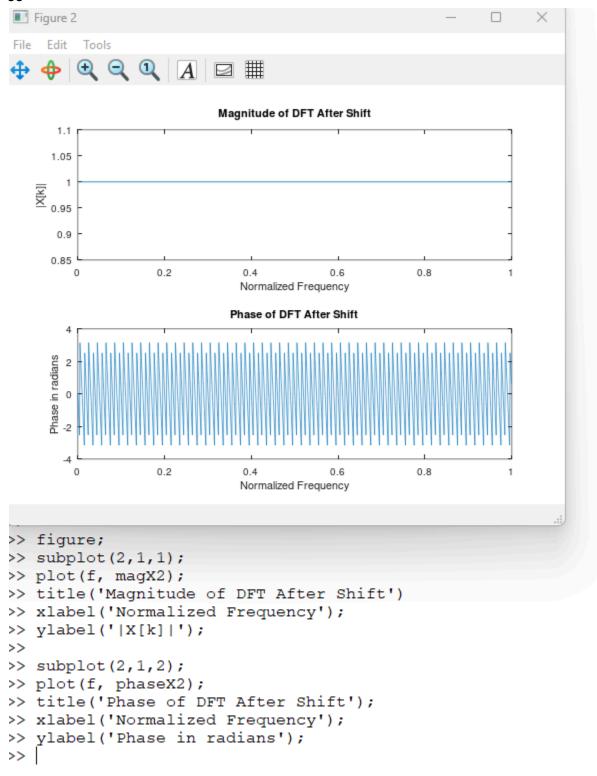
Problem 2

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')
Figure 1
                                                       File Edit Tools

♠ ♠ ♠ ♠ ♠ ♠ ■ ■ ■
                           Magnitude of DFT
     1.1
     1.05
   Š
    0.95
     0.9
                          Normalized Frequency
                            Phase of DFT
   phase in radians
     0.5
      0
     -0.5
                                              0.8
                          Normalized Frequency
```



```
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)');
 Figure 3
 File Edit Tools
    ⊕ ⊕ □ □
                    Unwrapped Phase of DFT (δ[n] shifted by 100)
       0
     -100
     -200
     -300
     -400
     -500
     -600
     -700
                  0.2
                            0.4
                                       0.6
                                                  0.8
                            Normalized Frequency
```

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/le6, abs(X));
>> title('Magnitude of DFT');
>> xlabel('Frequency (MHz)');
>> ylabel('|X(f)|');
Figure 1
                                                         Edit Tools

\mathbf{Q} \mathbf{Q} \mathbf{Q} \mathbf{A} \mathbf{B}

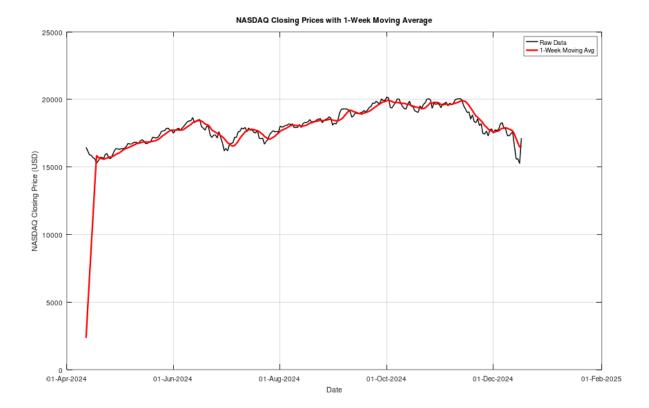
                           Magnitude of DFT
   12000
   10000
   8000
   6000
   4000
   2000
      0
                        0
                                             2
                            Frequency (MHz)
```

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;
2
3
   pkg load signal;
4
5
   filename = 'NASDAQ.txt';
   fid = fopen(filename, 'r');
   fgetl(fid);
   data = fscanf(fid, '%f %f', [2, Inf])'; % Read two columns
9
    fclose(fid);
.1
    days = data(:,1);
2
   prices = data(:,2);
.3
4
   % 1-week moving average filter (7-day window)
.5
   window_size = 7;
6
    b = ones(1, window_size)/window_size; % Simple averaging filter
.7
    a = 1;
.8
.9
    smoothed = filter(b, a, prices);
20
21
    & Convert day numbers to dates starting from April 10, 2024
   start date = datenum('10-Apr-2024');
    dates = start date + days;
23
24
    % Plot raw and smoothed data
25
   figure('position', [100, 100, 1000, 600]);
26
   plot(dates, prices, 'k', 'LineWidth', 1.2); hold on;
27
   plot(dates, smoothed, 'r', 'LineWidth', 2);
   datetick('x', 'dd-mmm-yyyy');
   xlabel('Date');
30
31
   ylabel('NASDAQ Closing Price (USD)');
   title('NASDAQ Closing Prices with 1-Week Moving Average');
33
   legend('Raw Data', 'l-Week Moving Avg');
   grid on;
35
```



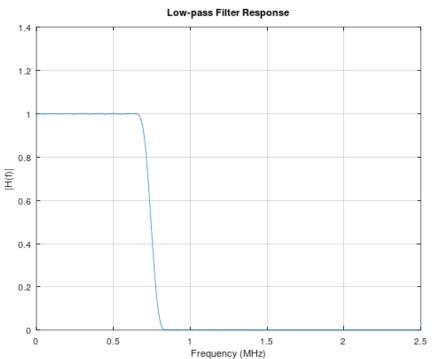
5d)

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)|');
>> arid on:
Figure 1
                                                  File Edit Tools

♠ ♠ ♠ Q Q Q A □ Ⅲ
                      Low-pass Filter Response
     1.4
```



```
6b
>> fc high = 735e3;
>> fc_norm_hp = fc_high / (Fs/2);
>>
>> %LP
\Rightarrow 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;
Figure 2
                                                     ×
File Edit Tools
       \bigcirc
                        High-pass Filter Response
     1.4
     1.2
      1
     0.8
   H(f)
     0.6
     0.4
     0.2
       0
                0.5
                                    1.5
                                              2
                                                       2.5
                           Frequency (MHz)
```

.. ----

```
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;
Figure 3
                                                     File Edit Tools
       \bigcirc
                       Band-pass Filter Response
    0.35
     0.3
    0.25
     0.2
  H(f)
    0.15
     0.1
```

1.5

Frequency (MHz)

2

2.5

0.05

0 6

0.5

```
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)|');
Figure 4
                                                          File Edit Tools
        oldsymbol{Q} oldsymbol{Q} oldsymbol{Q} oldsymbol{A} oldsymbol{B}
                        Spectrum of AM Signal with Noise
     3000
     2500
     2000
     1500
     1000
     500
       0
        0
                             Frequency (MHz)
```

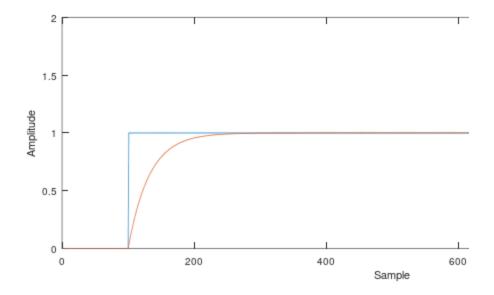
```
>> am_filtered = conv(am_signal, h_bp, 'same');
>>
>> AM_filt_spec = fft(am_filtered);
>> figure;
>> plot(f_axis/le6, abs(AM_filt_spec));
>> title('Filtered AM Spectrum');
>> xlabel('Frequency (MHz)');
>> ylabel('|X(f)|');
>> grid on;
>>
Figure 5
                                                        File Edit Tools
                           Filtered AM Spectrum
     1000
     800
     600
   (f)X
     400
     200
        0
                            Frequency (MHz)
(4.4493, 989.78)
```

```
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;
Figure 1
                                                     File Edit Tools
       \bigcirc
                       Convolution of Square Pulses
     50
     40
     30
     20
      10
      0
                  0.005
                              0.01
                                          0.015
                                                      0.02
                           Time in seconds
```

```
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
>>
Figure 2
                                                          File Edit Tools
                          Convolution of Sawtooth
     100
      50
      0
   Amplitude
     -50
     -100
       0
                   0.005
                                              0.015
                                                           0.02
                                 0.01
                             Time in Seconds
```

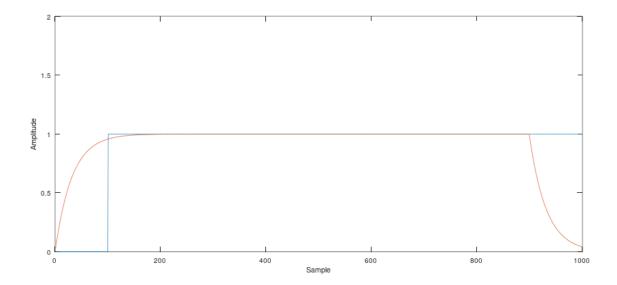
Problem 8abc)

```
1 clear;
 2 close;
 3 home;
 4 pkg load signal;
 5
 6 %dsp parameters for x-axis:
7 fs = 1.0e4; %Hz
8 fc = 0.5e2; %Hz
9 fc = fc/fs;
10 x = exp(-2*pi*fc);
11 a0 = 1-x;
12 b1 = x;
13 N = 1000;
14 M = 100;
15
16 %Low-pass filtering
17 st = [zeros(M,1); ones(N-M,1);];
18 y = zeros(size(st));
19 y(1) = a0*st(1);
20 for i = 2:N
21 y(i) = a0*st(i)+b1*y(i-1);
22 endfor
23
24 %Plotting section
25 figure(1, 'position',[0,0,1000,1000]);
26 subplot(2,1,1);
27 plot(st);
28 hold on;
29 plot (y);
30 axis([0 N+1 0 2]);
31 xlabel('Sample');
32 ylabel('Amplitude');
33 set(gca());
```



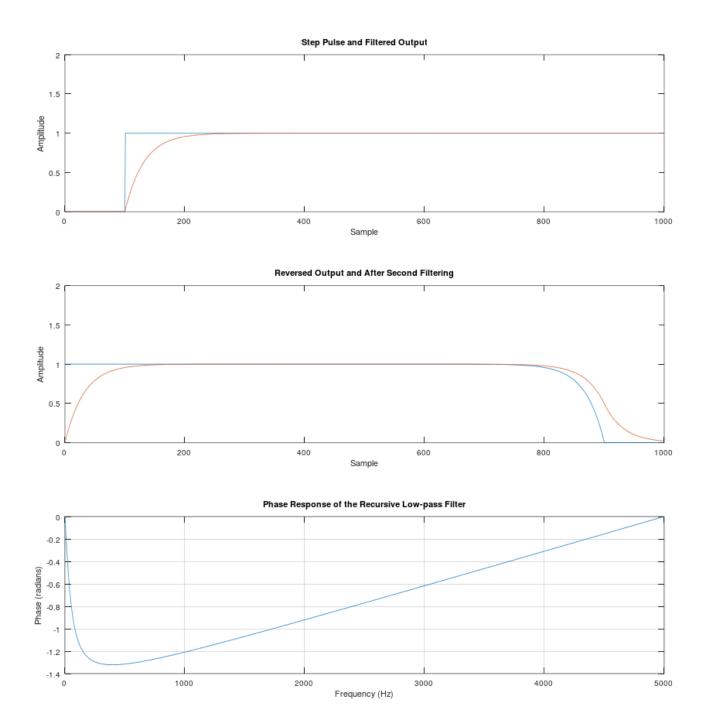
Problem 9ab

```
%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());
```



```
1 % Problem 9c
2 clear;
3
   close;
4 home;
5 pkg load signal;
6
7 %dsp parameters for x-axis:
8
   fs = 1.0e4; %Hz
9 fc = 0.5e2; %Hz
l0 fc = fc/fs;
11 \quad x = \exp(-2*pi*fc);
12 \quad a0 = 1-x;
L3
   b1 = x;
14 N = 1000;
M = 100;
16
17 % Step Pulse
L8
   st = [zeros(M,1); ones(N-M,1);];
ا 19
   % Low-pass filtering (using the original step pulse)
20
21 y = zeros(size(st));
y(1) = a0*st(1);
23 For i = 2:N
24
       y(i) = a0*st(i) + b1*y(i-1);
25
   endfor
26 L
27
   % Reverse the filtered output
28 y_reversed = flipud(y);
29
$ Run the reversed output through the recursive low-pass filter again
31 y_final = zeros(size(y_reversed));
32  y_final(1) = a0*y_reversed(1);
33 - for i = 2:N
34
       y_final(i) = a0*y_reversed(i) + b1*y_final(i-1);
35
   endfor
36 L
37 % Plotting section
38 figure(1, 'position',[0,0,1000,1000]);
```

```
39
40
    % First subplot: Original step pulse and the filtered output
41 subplot(3,1,1);
42 plot(st);
43 hold on;
44 plot(y);
45 axis([0 N+1 0 2]);
46 xlabel('Sample');
47
    ylabel('Amplitude');
48 title('Step Pulse and Filtered Output');
49 set (gca());
50
51 % Second subplot: Reverse the filtered output and run through filter again
52 subplot (3,1,2);
53 plot (y reversed);
54 hold on;
55 plot(y final);
56 axis([0 N+1 0 2]);
57 xlabel('Sample');
58 ylabel('Amplitude');
59
    title('Reversed Output and After Second Filtering');
60 set(gca());
61
   % Third subplot: Phase Response of the original filter
62
63 subplot(3,1,3);
64
    [H, f] = freqz([a0], [1, -b1], N, fs);
65 plot(f, angle(H));
66 xlabel('Frequency (Hz)');
67 ylabel('Phase (radians)');
68 title('Phase Response of the Recursive Low-pass Filter');
69 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