1. Following is the original image houses.bmp from the sample images.

Figure 1: Original image: houses.bmp

Following images obtained from the Non-Casual Moving Average Filter for the M values of 11, 31, 61.

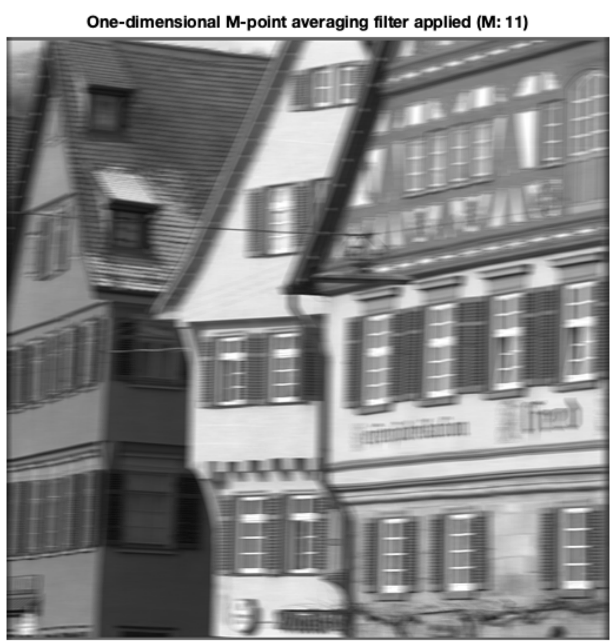

Figure 2: image: houses.bmp, M = 11

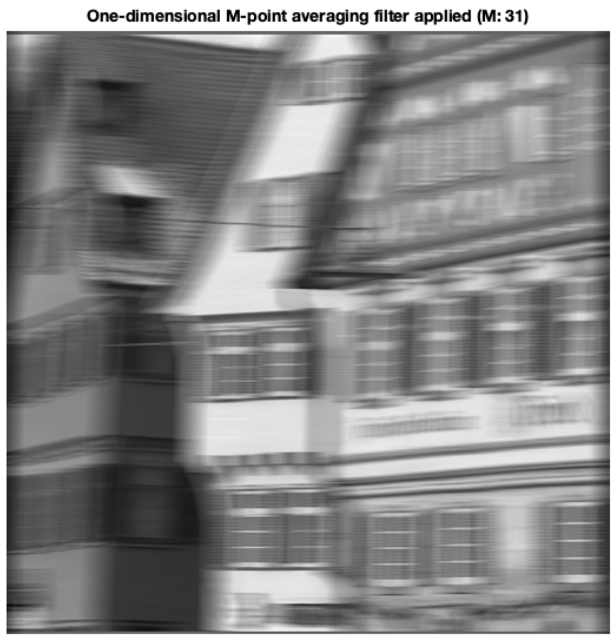

Figure 3: image: houses.bmp, M = 31

Figure 4: image: houses.bmp, M = 61

- The filter blurred the image in horizontal direction.
- The details of the image smoothed by the filter. As the value M increased the smoothing affect amplified since the image cells averaged over broader range of window horizontally.

- The visual effect observed in the horizontal direction (m) whereas the vertical dimension (n) no effect observed since the filter only takes input from 1D horizontal direction.
- As the frequency increase the magnitude decreases; thus the affect of the filter decreases. As the value M increase the affect of the filter decreases more rapidly as we increase the frequency having the same sampling frequency to obtain the frequency response plot.
- The edges of the image are smoothed when compared with the original image. Additionally, corner of the images are padded with the value 0. Additionally as the value M increases the window of padding also increases as more 0 values contribute to the final corner values.

The frequency response of the non-casual averaging filter found as using the relationship between the non-casual Moving Average (MA) filter and the casual MA filter. The relationship states that frequency response of the non-casual MA filter is that of casual MA filer is an added phase of $(M-1)/2$ to the casual MA filter.
Following are the filter's frequency response graphs for the M values of 11, 31, 61.

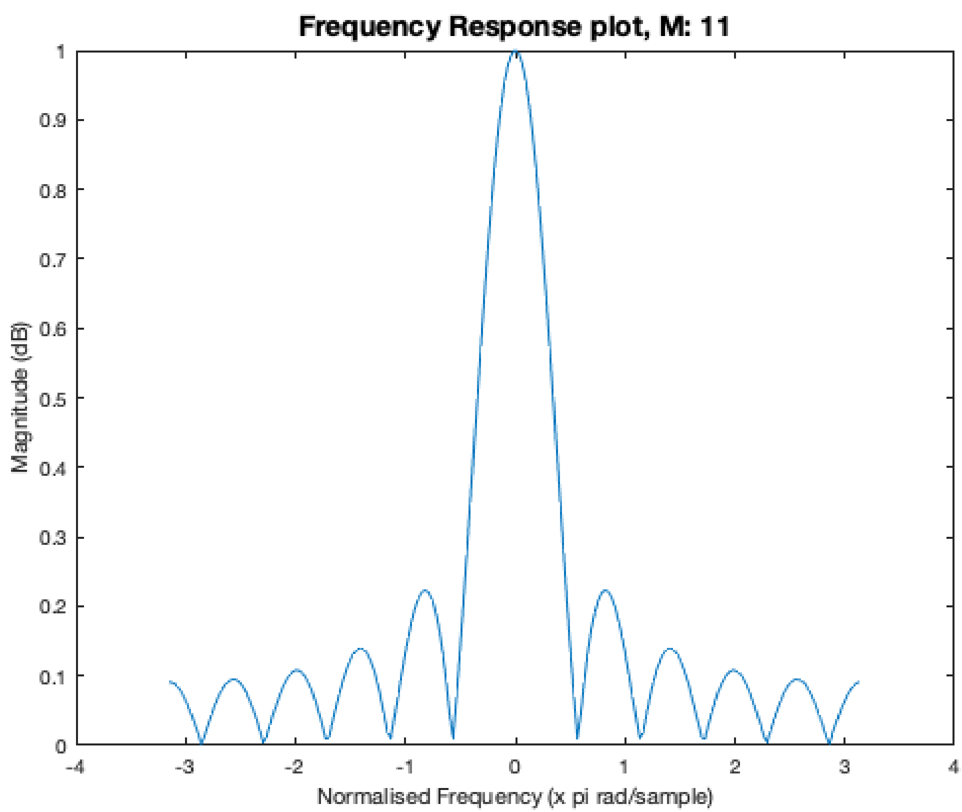

Figure 5: Frequency Response Plot, M = 11

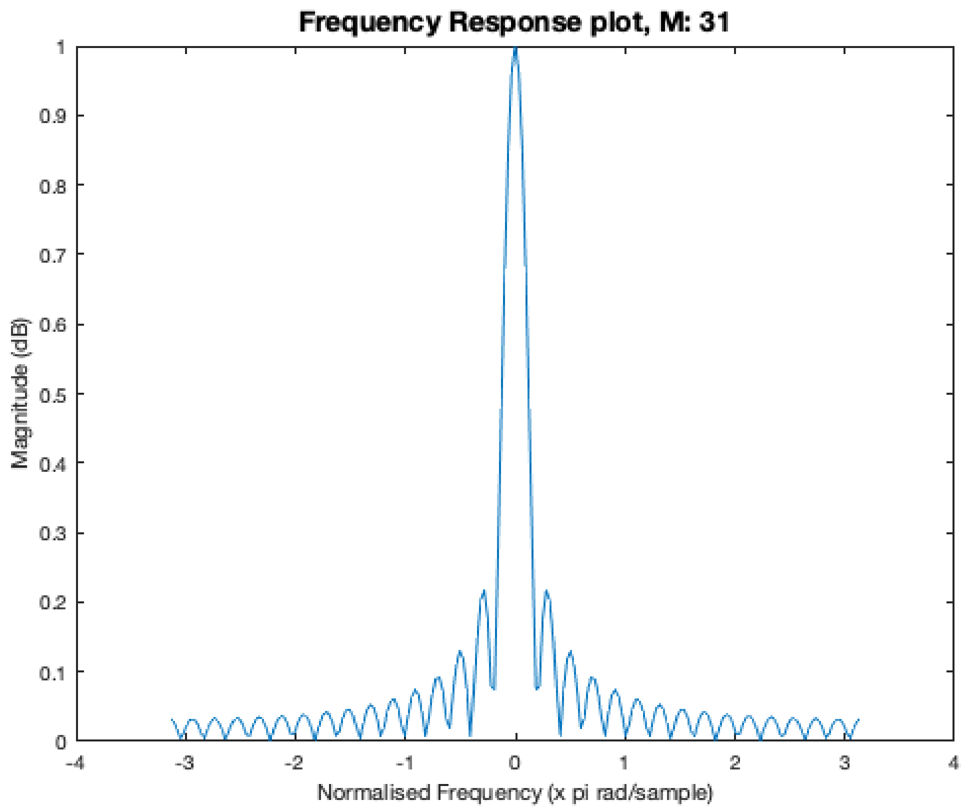

Figure 6: Frequency Response Plot, M = 31

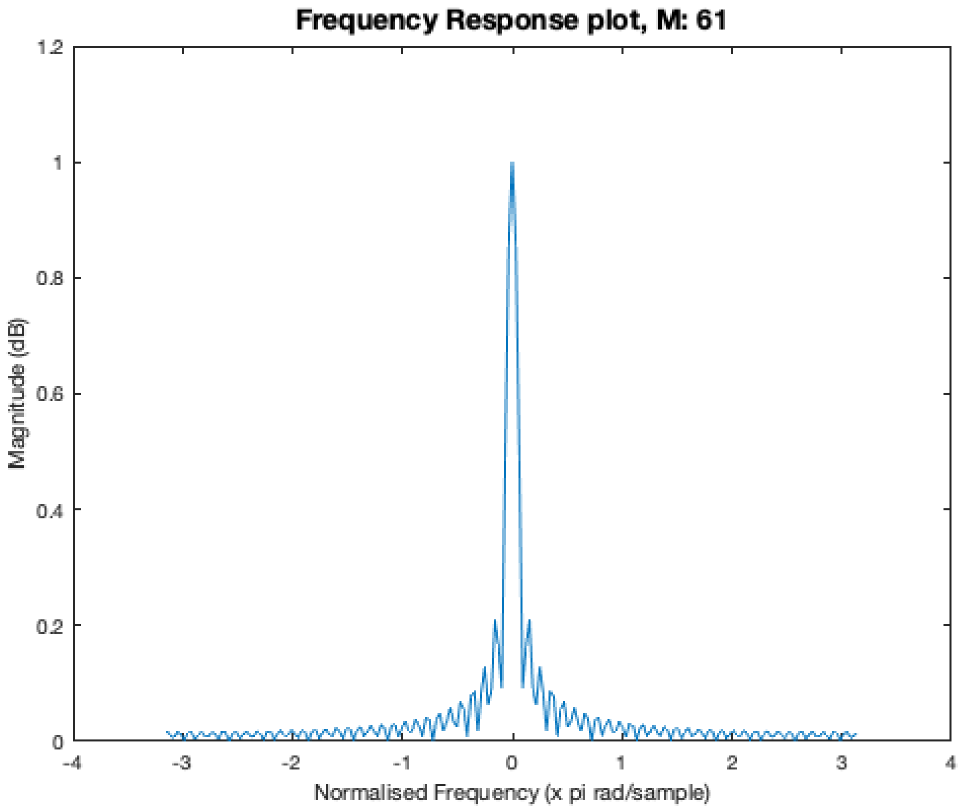

Figure 7: Frequency Response Plot, M = 61

Following 6 images obtained for applying noise with c values of c = 0.2 and c = 1 on filters with M = 11, M = 31, M = 61.

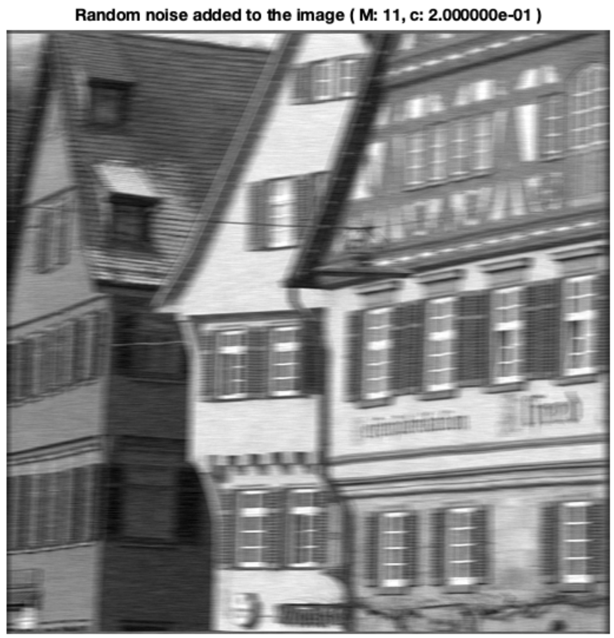

Figure 8: image: houses.bmp, houses, M = 11, c = 0.2

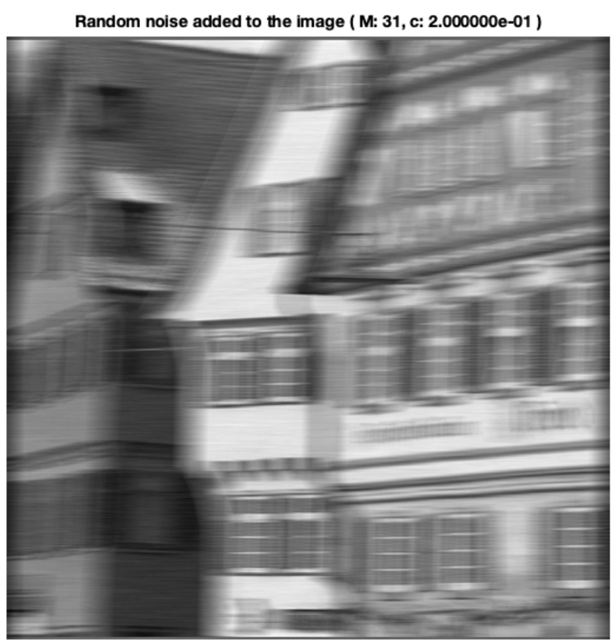

Figure 9: image: houses.bmp, houses, M = 31, c = 0.2

Figure 10: image: houses.bmp, houses, M = 61, c = 0.2

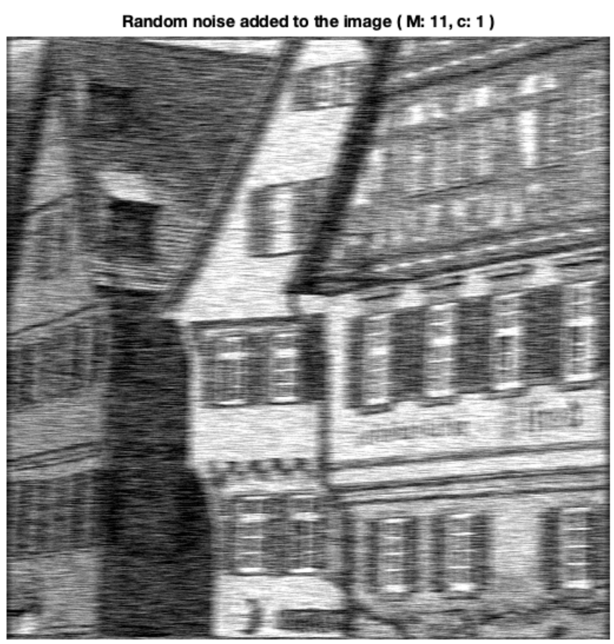

Figure 11: image: houses.bmp, houses, M = 11, c = 1

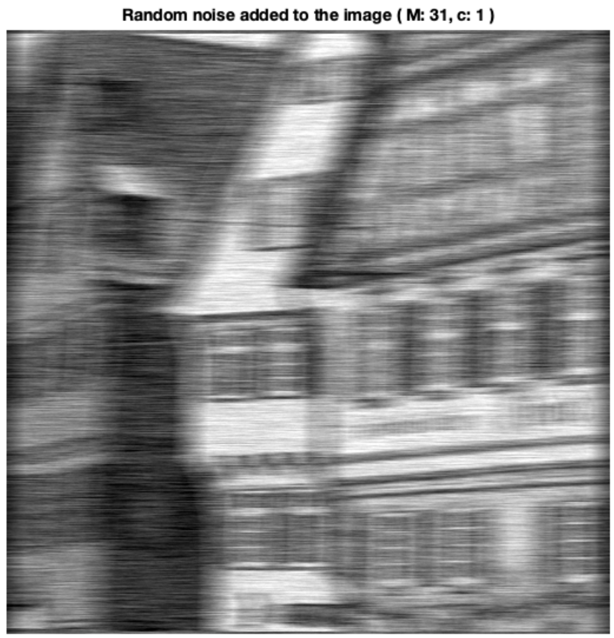

Figure 12: image: houses.bmp, houses, M = 31, c = 1

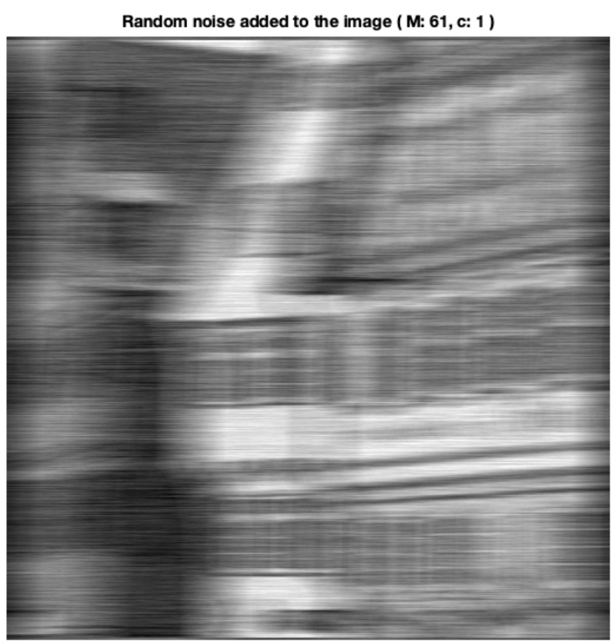

Figure 13: image: houses.bmp, houses, M = 61, c = 1

- The noise seemed to be reduced with the help of averaging. As M increase the images looked similar to the noise free counterparts where filter applied. The noise reducing affect increased as we increase M.

- The undesirable effect is the blurring effect due to loss of exact information at each point of the image due to the horizontal non-casual moving averaging filter.
- It depends on the amount of noise reduction we want to achieve in general. If we want to achieve highest noise reduction we should pick the highest M value which is 61 from the sample. We must beware the choice of M is not over-shooting the image dimension.

2. Following image obtained after applying the first difference filter on the original image.

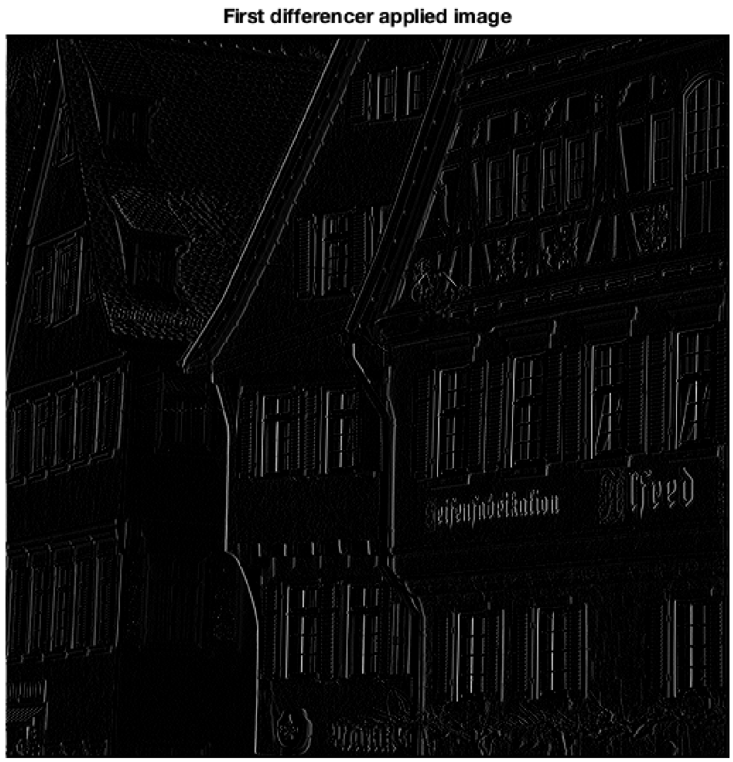

Figure 14: image: houses, after first difference filter

The frequency response of the first difference filter found from the following system: $y[n] = x[n] - x[n-1]$ via the help of *freqz* MATLAB commend which able to return Magnitude of the frequency response of the system from sample of frequencies.
Following is the filter's frequency response plots after applying first difference filter.

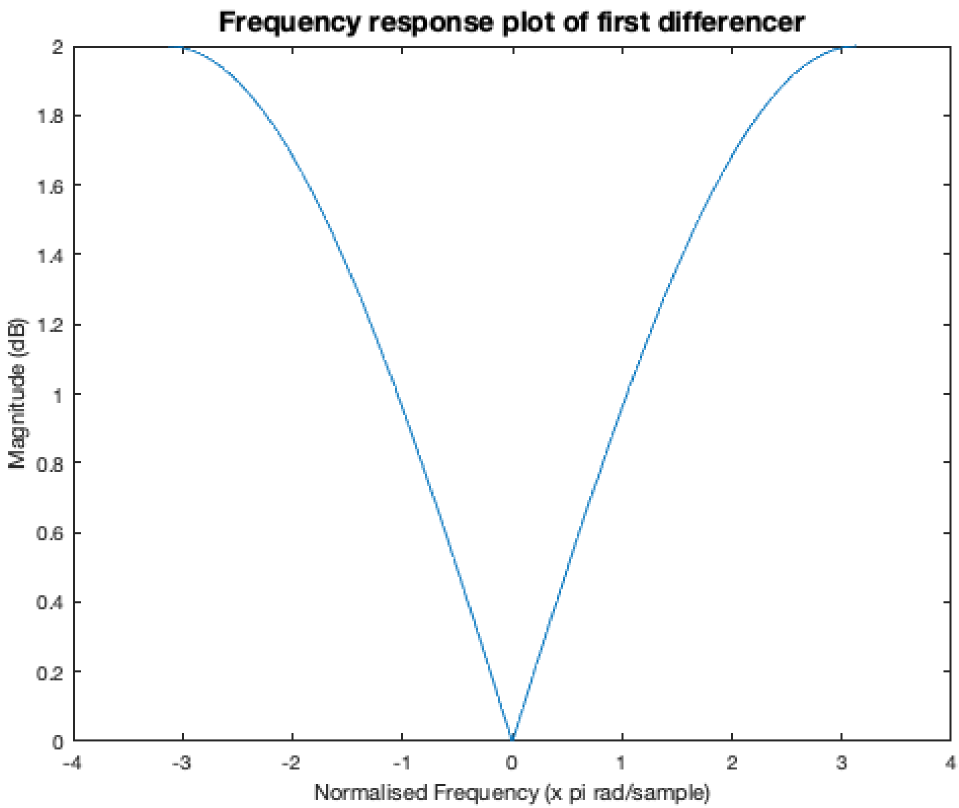

Figure 15: Frequency Response of the first differencer system

- The filter sharpened the edges of the image.
- I would expect it to be similar to the output image since we take difference of the pixel's immediate left neighbor and the current pixel value itself. In the images except the edges I wouldn't expect sudden changes which puts the result of amplified edges into perspective.
- The filter affect on horizontal direction since it is differencing with the value of it's immediate left neighbor.
- The affect of the filter increased together with the increase in frequency as we can see the magnitude of the frequency response plot increases with frequency.

3. Appendix

The MATLAB codes

```
% EEE391 Basics of Signals and Systems: Computer Assignment 2
% Author: Zeynep Cankara

%% 1) Implementing 1D M-point averaging filter

% chosen image: "houses.bmp"
```

```
A=imread("./images/houses.bmp");
J=mat2gray(A, [0 255]);
imageno = 1;

% Original image
figure;
imshow(J);
caption = sprintf('Original image: No filter applied');
title(caption, 'FontSize', 14);
pause(0.5);


for M = [11, 31, 61]
    % iterate over columns and perform average filtering
    Y = zeros(512,512);
    for col = 1:size(J,2)
        y = zeros(512,1);
        % find window range
        mid = (M-1)./2;
        lower = -1 * mid;
        upper = mid;
        % iterate over M window
        for i = lower:upper
            if (((i  + col) >= 1) && ((i + col) <= 512))
                y(:,1) = y(:,1)+ J(:,(i+col));
            else
            % pass
            end
        end
        % assign the averaged column
        Y(:,col) = y./M;
    end
    % display the image
    figure(imageno);
    imshow(Y);
    m = M;
    caption = sprintf('One-dimensional M-point averaging filter applied (M: %d)', m);
    title(caption, 'FontSize', 14);
    imageno = imageno + 1;
end

%% 1) Introducing a random noise

for c = [0.2, 1]
    noise = rand(512,512);
    noise = noise - (0.5 .* ones(512,512));
    noise = noise .* c;
    new_J = J + noise;
```

```
    for M = [11, 31, 61]
        % iterate over columns and perform average filtering
        Y = zeros(512,512);
        for col = 1:size(J,2)
            y = zeros(512,1);
            % find window range
            mid = (M-1)./2;
            lower = -1 * mid;
            upper = mid;
            % iterate over M window
            for i = lower:upper
                if (((i  + col) >= 1) && ((i + col) <= 512))
                    y(:,1) = y(:,1)+ new_J(:,(i+col));
                else
                % pass
                end
            end
            % assign the averaged column
            Y(:,col) = y./M;
        end
        % display the image
        figure(imageno);
        imshow(Y);
        m = M;
        caption = sprintf('Random noise added to the image ( M: %d, c: %d )', m, c);
        title(caption, 'FontSize', 14);
        imageno = imageno + 1;
    end
end


%% 1.1) Frequency Response plot of Moving Average FIR filter in Q1

for M = [11, 31, 61]
    L = M;
    w = -pi:(pi/100):pi; %to plot frequency response
    H = [ones(1,L)]/L;
    H_old = H;
    [H,W] = freqz(H,1,w);
    % add phase of (M-1)/2 to connect property of non-casual moving average
    H = H .* exp(((M-1)/2) .* 1j .* w);
    H_new = H;
    figure(imageno);
    plot(W,abs(H));
    caption = sprintf('Frequency Response plot, M: %d', M);
    title(caption, 'FontSize', 14);
    xlabel('Normalised Frequency (x pi rad/sample)');
    ylabel('Magnitude (dB)');
```

```
    imageno = imageno + 1;
end


%% 2) Implementing First Differencer Filter


% iterate over columns and perform average filtering
Y = zeros(512,512);
for col = 1:size(J,2)
    y = zeros(512,1);
    % find window range
    lower = -1;
    upper = 0;
    % iterate over M window
    for i = lower:upper
        if (((i  + col) >= 1) && ((i + col) <= 512))
            if (i == 0)
                y(:,1) = y(:,1)+ J(:,(i+col));
            else
                y(:,1) = y(:,1) - J(:,(i+col));
            end
        else
        % pass
        end
    end
    % assign the averaged column
    Y(:,col) = y;
end

% display the image
figure(imageno);
imshow(Y);
caption = sprintf('First differencer applied image');
title(caption, 'FontSize', 14);
imageno = imageno + 1;

%% 2.2) Frequency response plot of first difference filter

w = -pi:(pi/10):pi; %sampled w values for frequency response plot
b = [1 -1];
a = [1];
figure(imageno);
[H,W] = freqz(b,a,w); %  magnitude response  graph
plot(W,abs(H));
xlabel('Normalised Frequency (x pi rad/sample)');
ylabel('Magnitude (dB)');
caption = sprintf('Frequency response plot of first differencer');
```

```
title(caption, 'FontSize', 14);
imageno = imageno + 1;
```