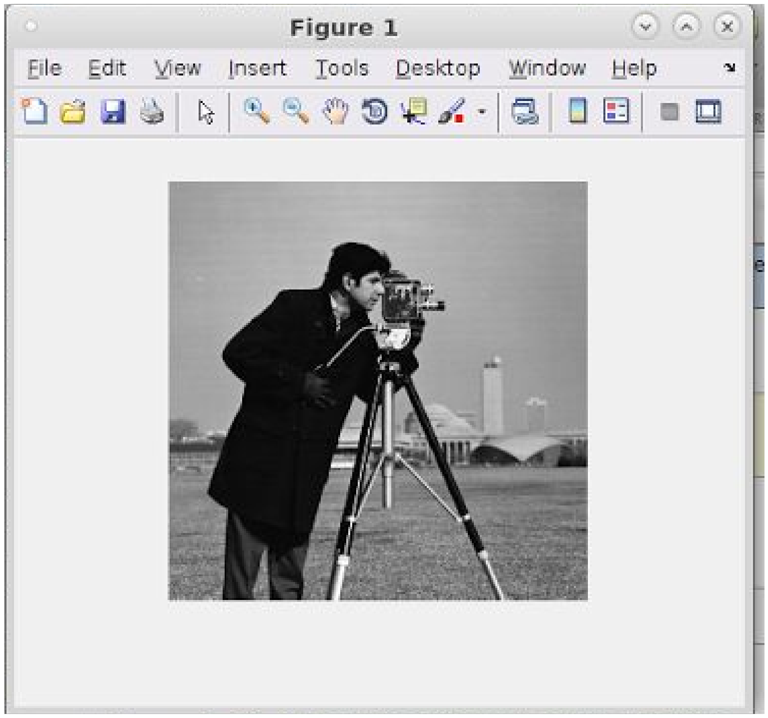
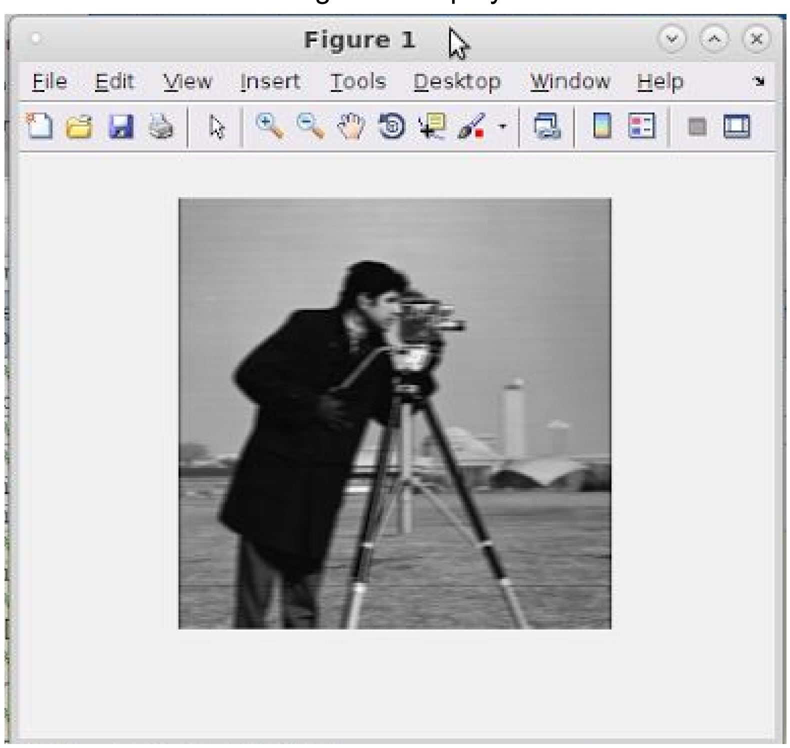
ECE311-Lab2

1. 1D Convolution for an Image

a). Load cameraman.tif image and display it using imshow



b). Filter the rows of cameraman.tif image and display it.

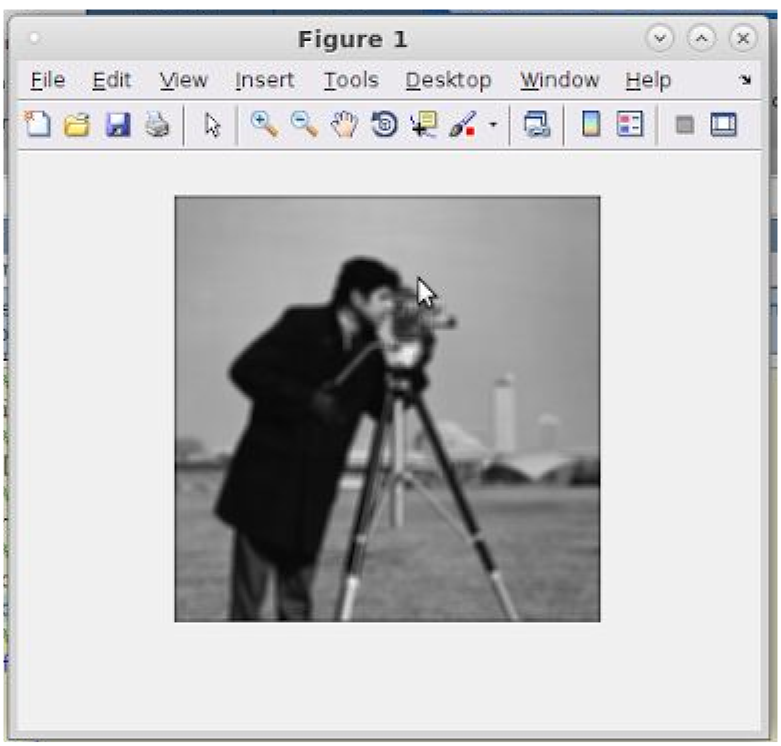


The image is blurred. Therefore, the filter is for blurring the image. The image is blurred horizontally, which looks like our car pass an object.

c). Filter the columns of cameraman.tif image and display it.

Comparing with image filtered in part b, this image was blurred vertically. It is like the image is accelerating vertically while in part b, the image looks like accelerating horizontally.

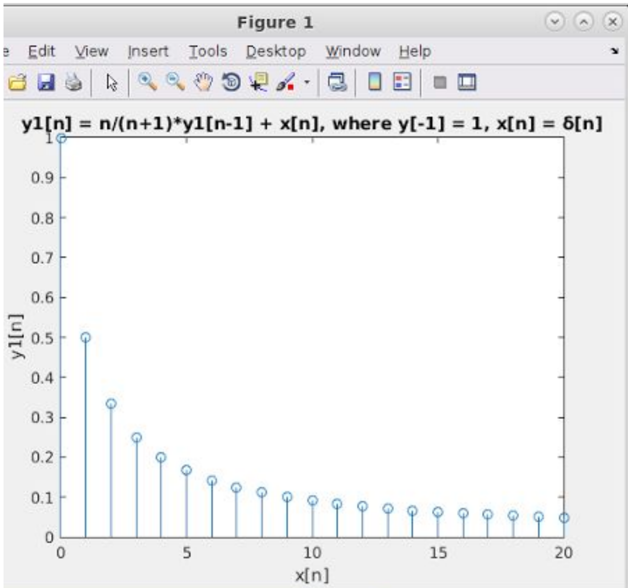
d). Perform filter on cols and rows

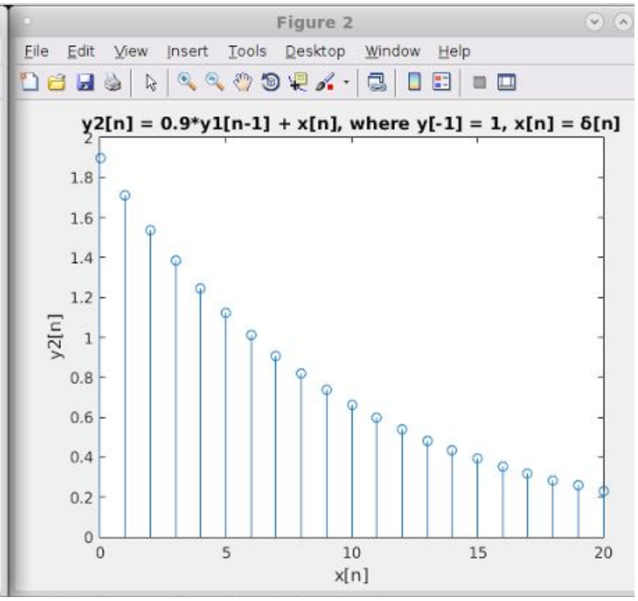


The picture is blurred vertically and horizontally. Comparing with images in b) and c), this one is less recognizable.

2. Zero-Input and Zero-State Response

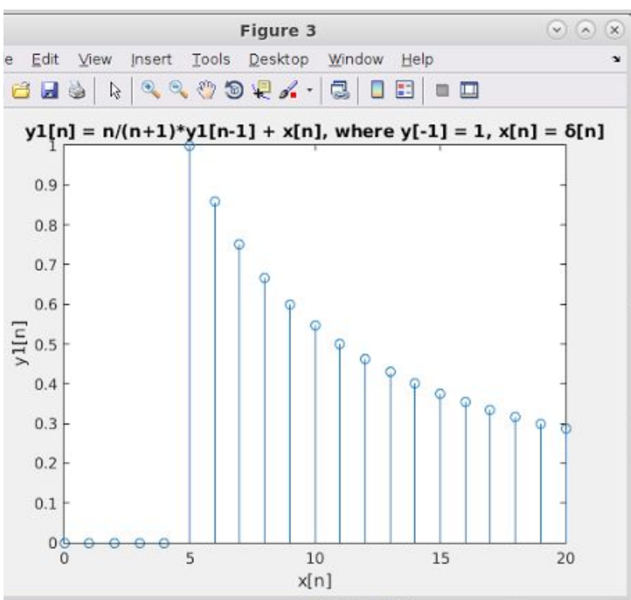
a). $x[n] = \delta[n]$, for $0 \le n \le 20$.

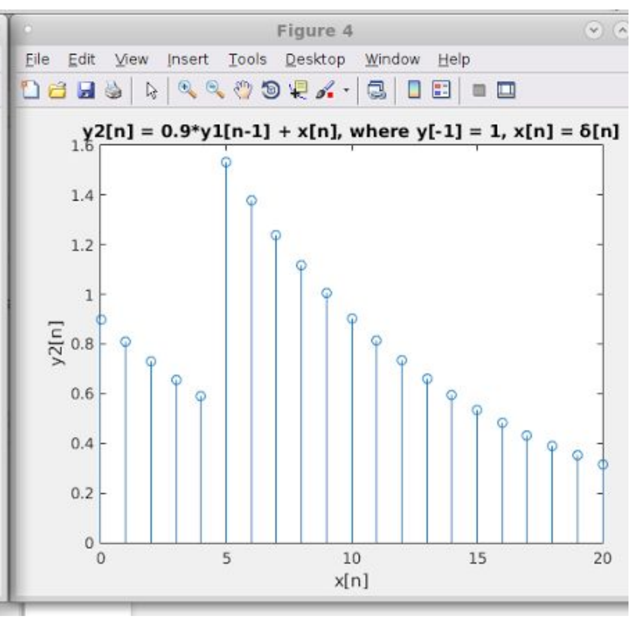




These two plots are impulse response for the system.

b). $x[n] = \delta[n-5]$, for $0 \le n \le 20$.





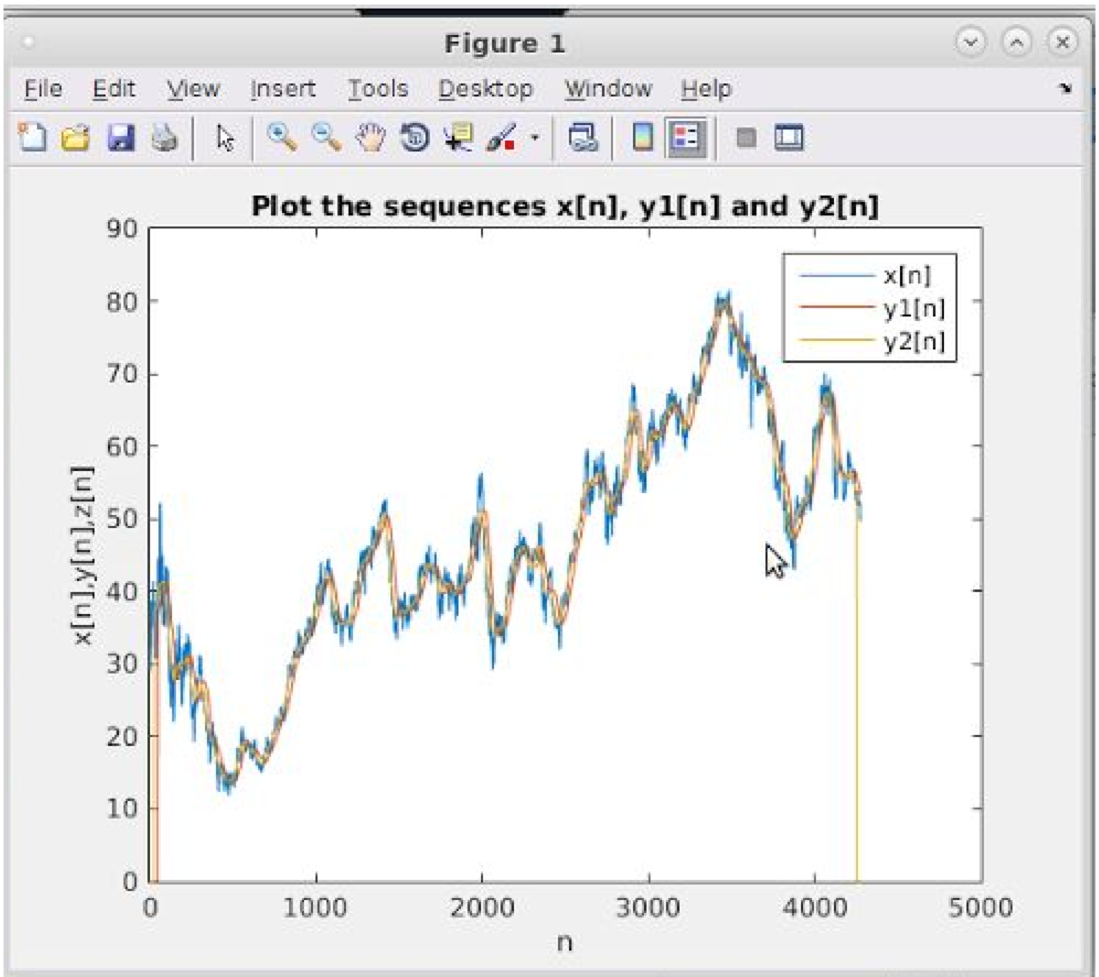
When the input impulse is shifted, two systems behave differently to the change. Y1 has no value before the impulse while y2 increases at the impulse.

3. Qualcomm Stock Data Analysis

a). Compute moving average.

The moving average calculated by the first equation is stored in y1.
The moving average calculated by the second equation is stored in y2.
I used conv(qcoms,h) verify the result, and the result is stored in y.
y1 and y2 are basically contains the same values, but their index are different. y1 starts at 51 and ended at 4276. y2 starts at 26 and ends at 4251. The result from the conv function also agrees with the calculated results(data from index 51 through 4251).

b). Plot and comment.



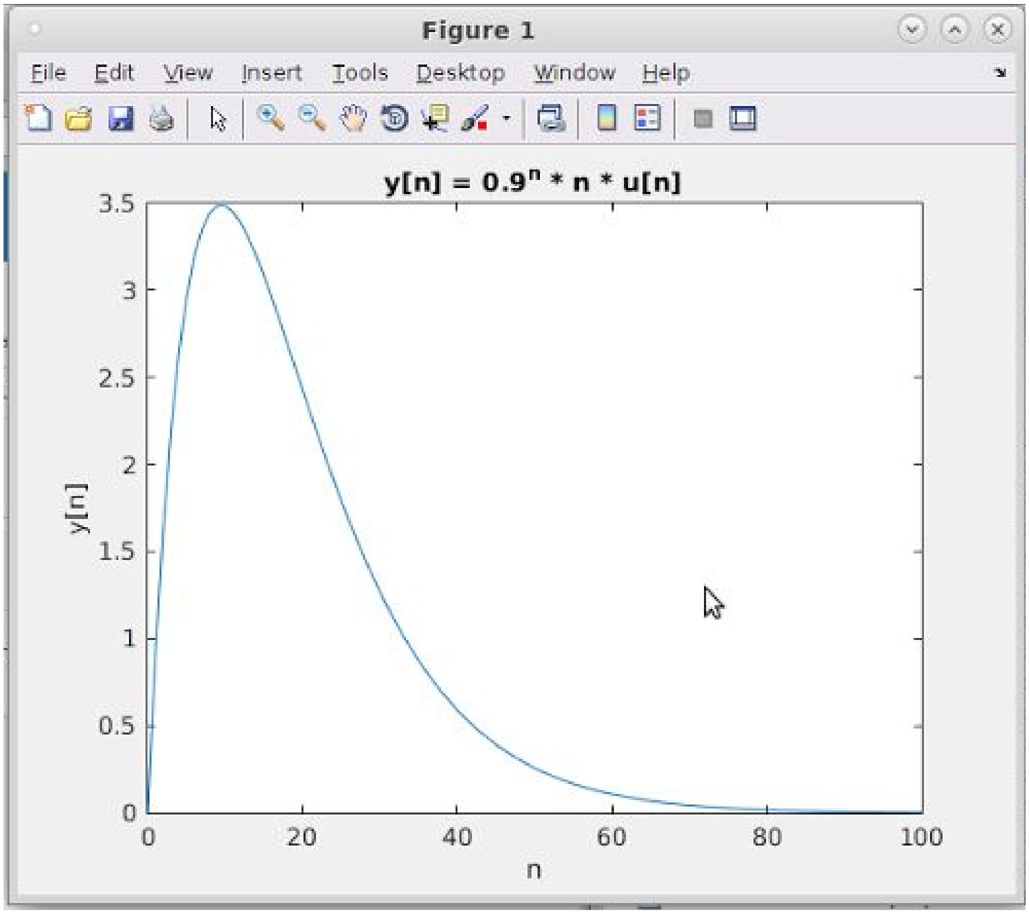
X[n] represents the actual data from qcoms.mat. y2 shift right by 25 gives y1. However, it is hardly to see this difference from the plot.

4. Analytical Expression and Finite Length Representation of a Convolution

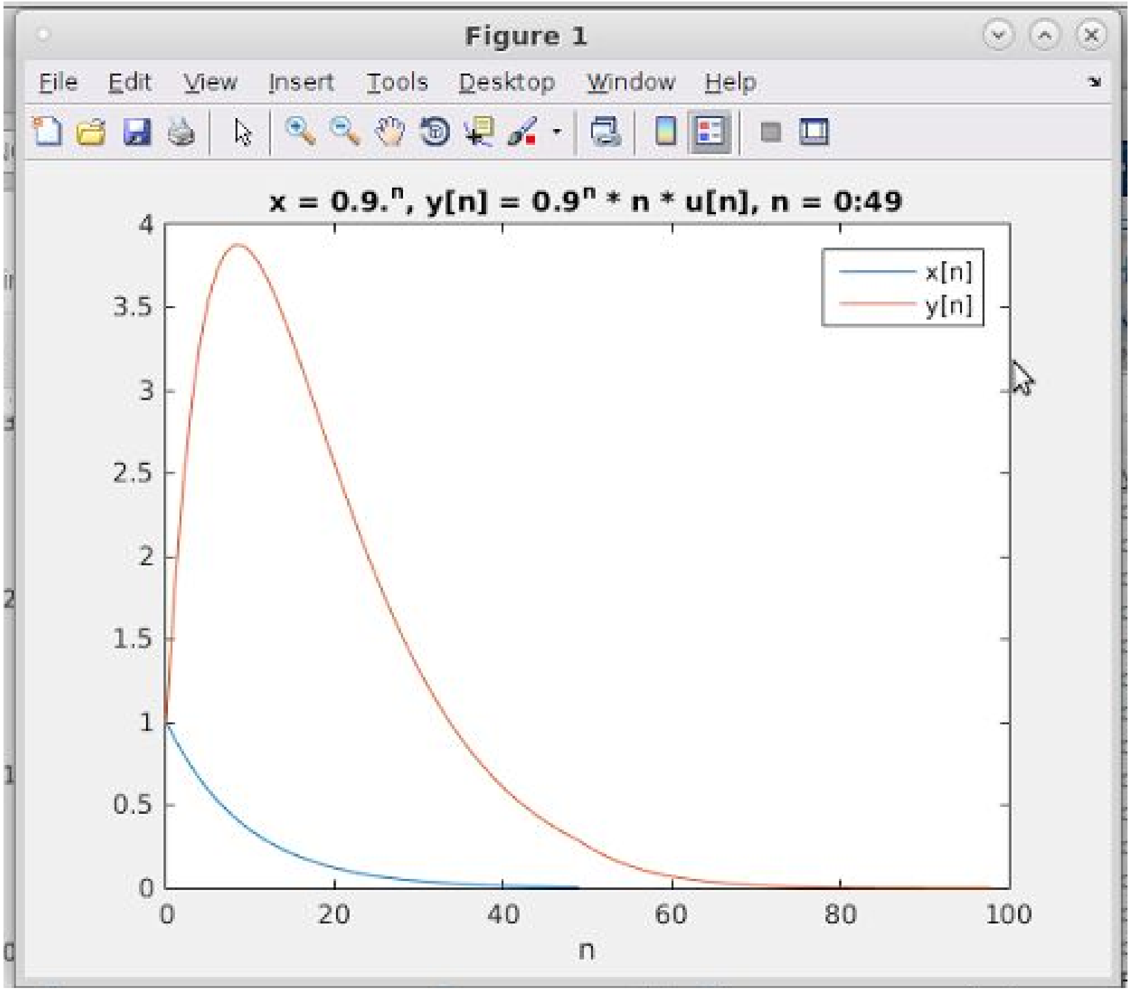
a). Determine y[n] and Plot.

By computing, we got: y[n] = (0.9^n)*n*u[n]

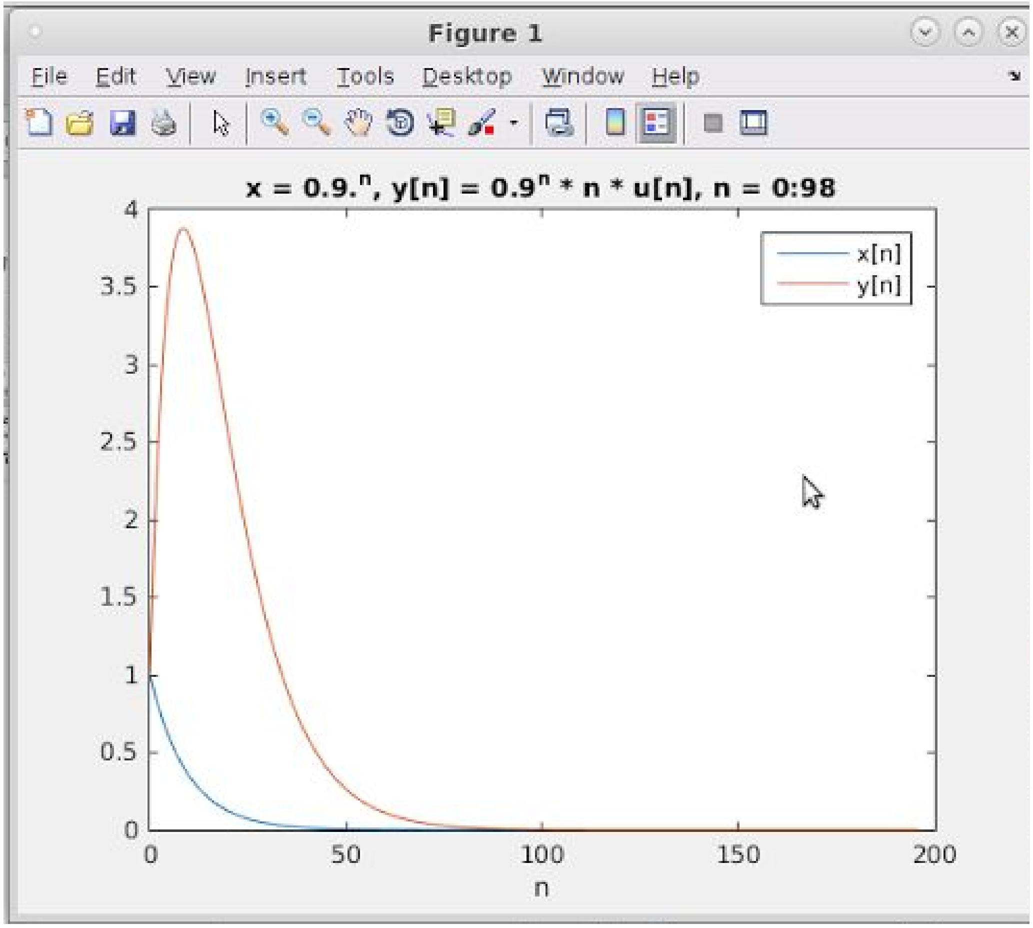
By plotting y[n], we got:



b). Compute and plot the first 50 samples.



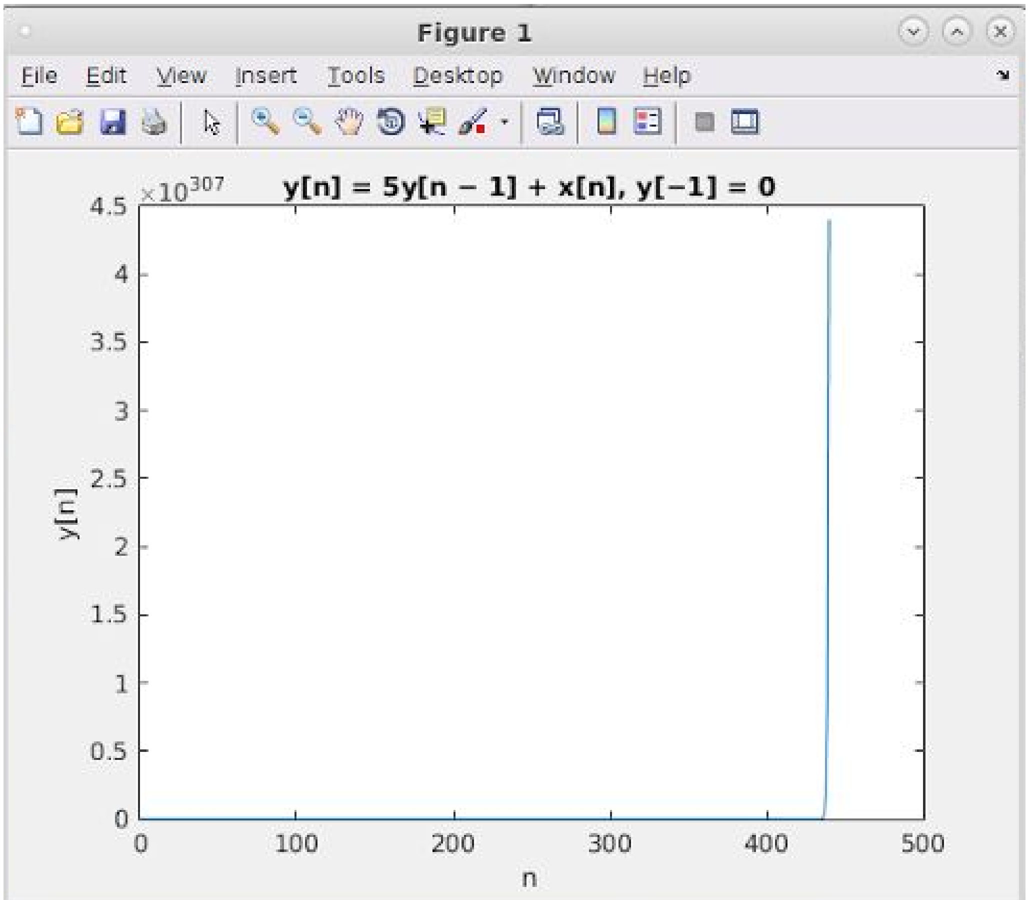
c). Use conv function to determine and plot first 90 samples.



d). Compare and comment.

Plot with first 99 samples are closer to the real plot since it takes more samples when plotting. We can see the difference from the plot. In part b, which 50 samples are plotted, it is not smooth at approximately n=50. However in part c, the curve is smooth.

5. Stable or Unstable



The system is not stable. Given an input u[n] that is bounded, the output of the signal goes to infinity. Therefore, the system is not stable.

6. Convolution Sum

a). Write convolution sum.

$$A_y = \sum_{n=-\infty}^{\infty} y[n]$$

$$A_x = \sum_{n=-\infty}^{\infty} x[n]$$

$$A_h = \sum_{n=-\infty}^{\infty} h[n]$$

$$A_x A_h = \sum_{n=-\infty}^{\infty} x[n] \cdot \sum_{n=-\infty}^{\infty} h[n]$$

$$= \sum_{k=-\infty}^{\infty} x[k] \cdot \sum_{n=-\infty}^{\infty} h[n]$$

$$= \sum_{n=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} x[k] \cdot h[k-n]$$

Since $y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[k-n]$

$$\therefore A_x \cdot A_h = \sum_{n=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} x[k] \cdot h[k-n]$$

$$= \sum_{n=-\infty}^{\infty} y[n] = A_y$$

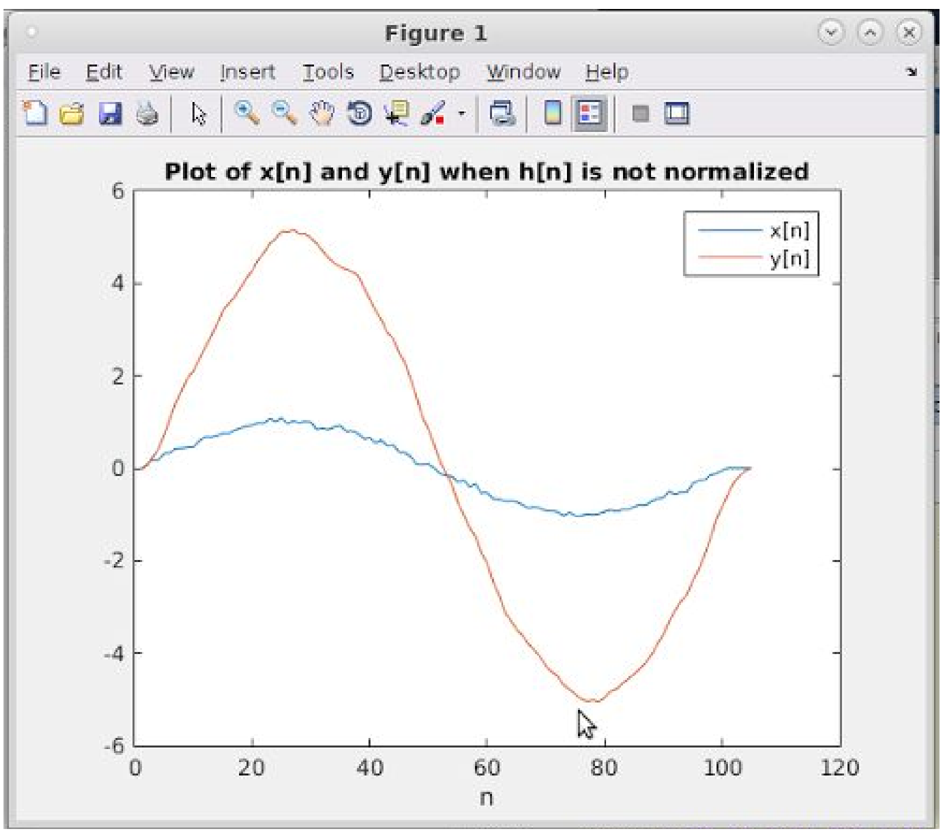
$$\therefore A_y = A_x \cdot A_h$$

b). Compute, Check and Plot.

Use conv function to calculate the convolution of h[n] and x[n]. The result Ay and AxAh is computed using the sum function in Matlab.

```
AxAh =

   -0.4478


Ay =

   -0.4478


AxAh =

   -2.6725


Ay =

   -2.6725
```

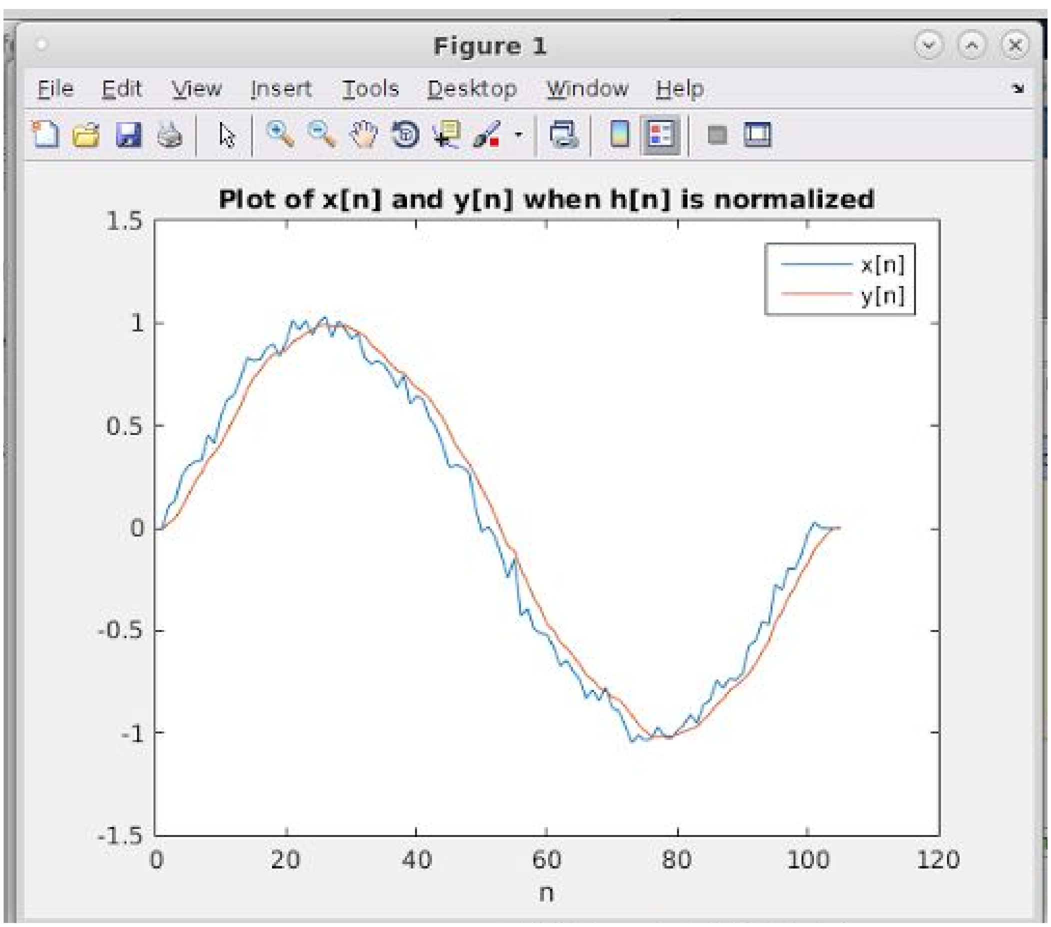
The plotted x[n] and y[n] is:



c). Normalize and repeat b).

```
AxAh =

    0.2993


Ay =

    0.2993


AxAh =

   -0.1428


Ay =

   -0.1428
```

The plotted x[n] and y[n] is:



d). Explain difference in b) and c).

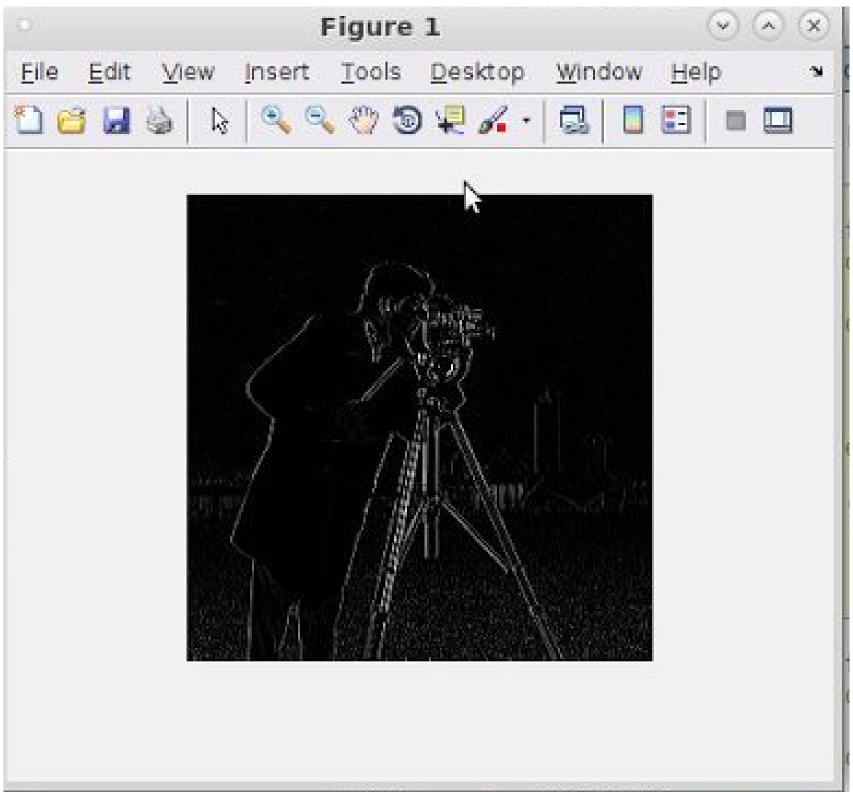
When h[n] is not normalized (as in part b), x[n] and y[n] are in the same shape but they have different magnitude for the same n. However, when h[n] is normalized (as in part c), they are almost the same in shape and magnitude according to the picture.

7. Edge Detector for an Image

a). Determine impulse response.

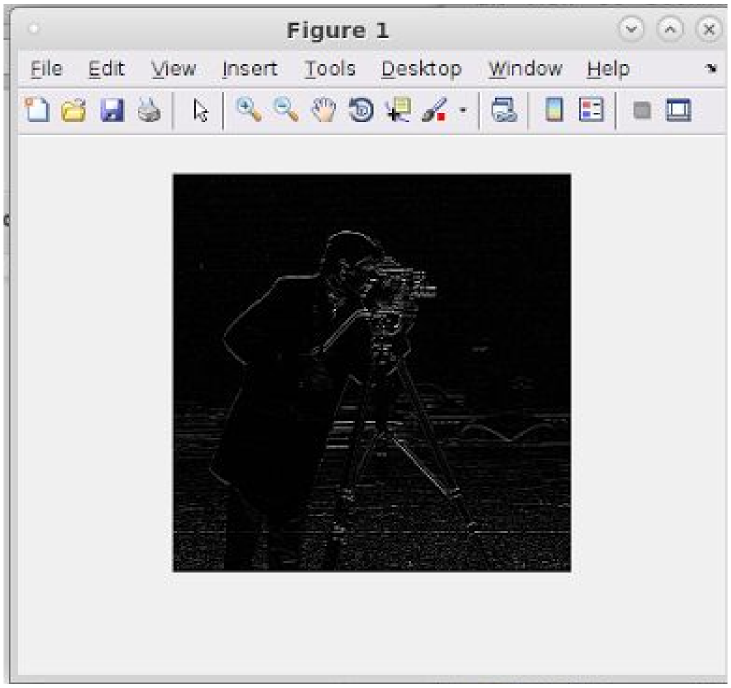
The impulse response for this edge detector is: h = delta[n+1] - 2*delta[n] + delta[n-1];

b). Manipulate rows of cameraman.tif using the determined impulse response.



The plot only contains white and black colors, wherea in the previous plot there are grey colors with different tensity of grey. Therefore, it is more blurred than the previous image in question 1. Also, the image is blurred vertically when the convolution is operated on the rows of the image matrix. This is different than the effect in question 1 part b.

c). Manipulate cols of cameraman.tif using the determined impulse response.



The plot only contains white and black colors, wherea in the previous plot there are grey colors with different tensity of grey. Therefore, it is more blurred than the previous image in question 1. Also, the image is blurred horizontally when the convolution is operated on the cols of the image matrix. This is different than the effect in question 1 part c.