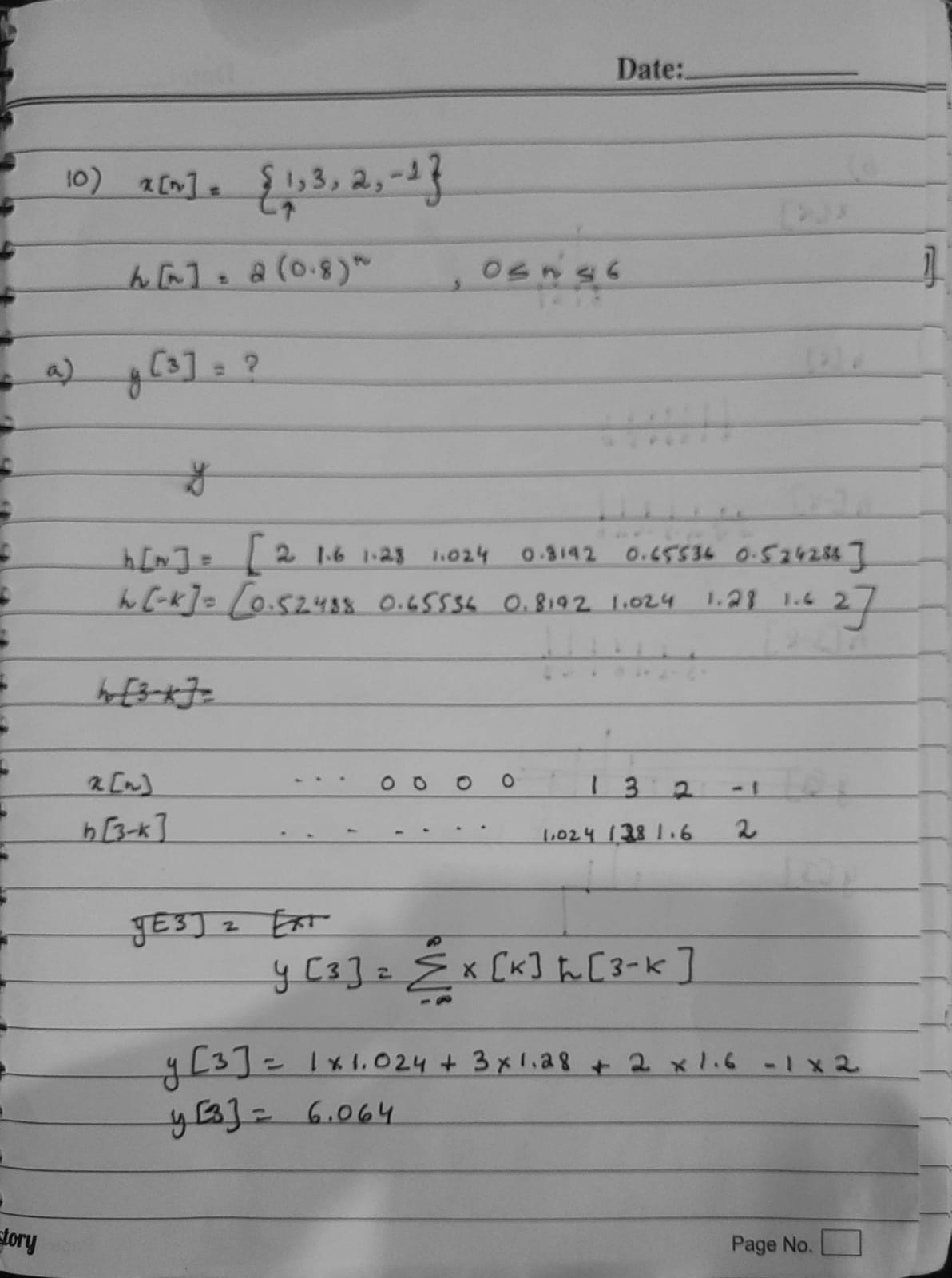
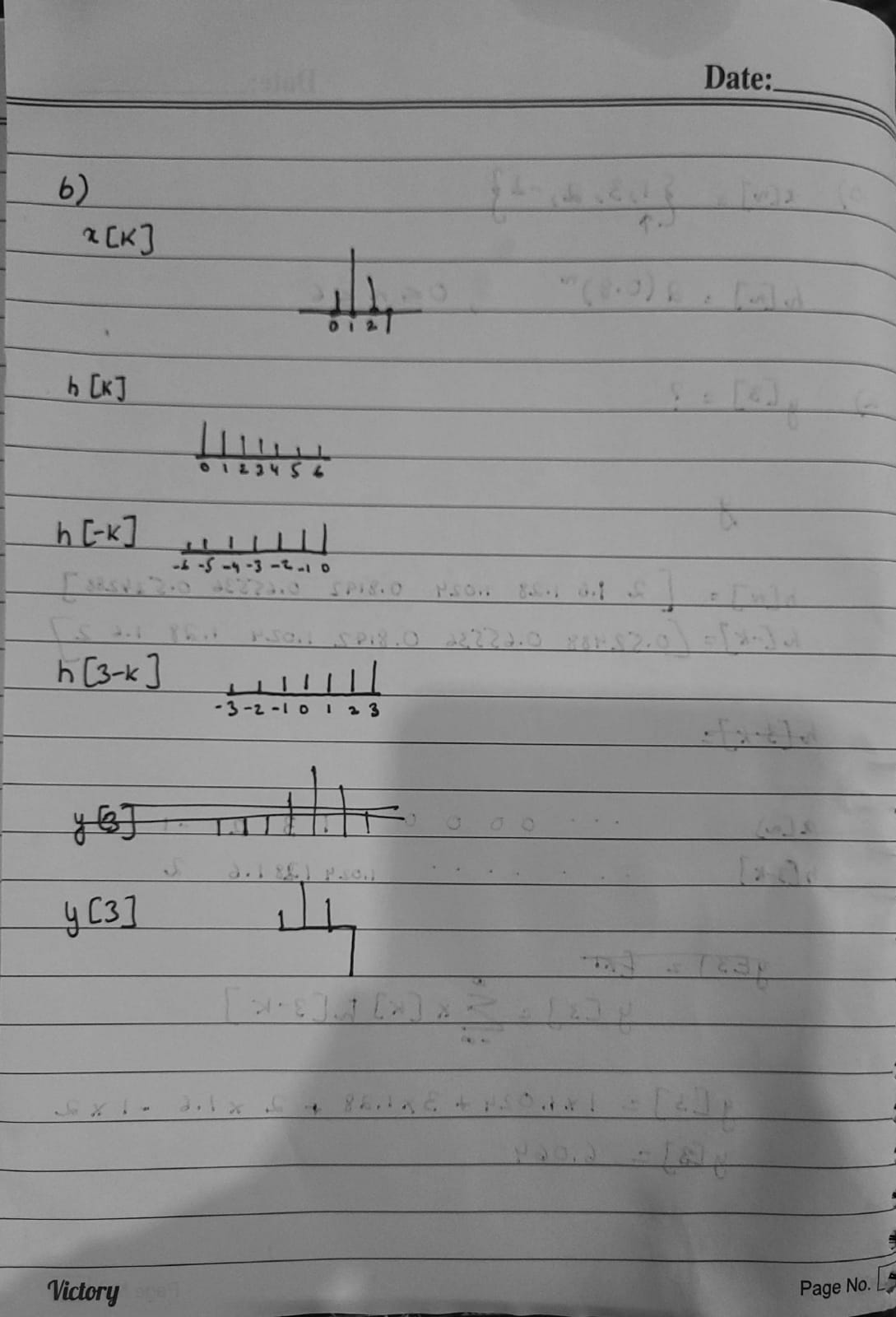
**2.10**

****

****

**2.17**

A discrete-time system is described by the following difference equation

**y[n] = 1.15y[n − 1] − 1.5y[n − 2] + 0.7y[n − 3] − 0.25y[n − 4] + 0.18x[n] + 0.1x[n − 1] + 0.3x[n − 2] + 0.1x[n − 3] + 0.18x[n − 4]**

with zero initial conditions.

a = [1 -1.15 1.5 -0.7 0.25];

b = [0.18 0.1 0.3 0.1 0.18];

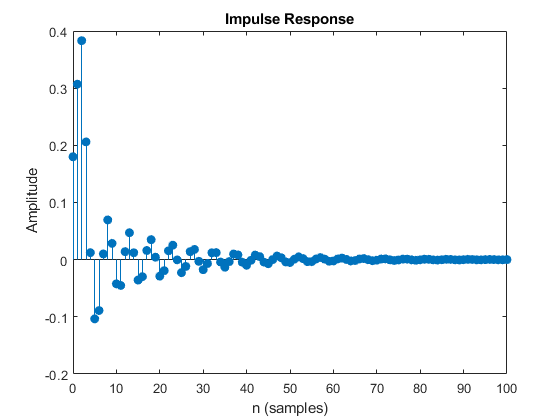
n = [0:100];

(a) Compute and plot the impulse response h[n], 0 ≤ n ≤ 100 using the function

**h=impz(b,a,N)**.

h = impz(b,a,length(n));

impz(b,a,length(n))



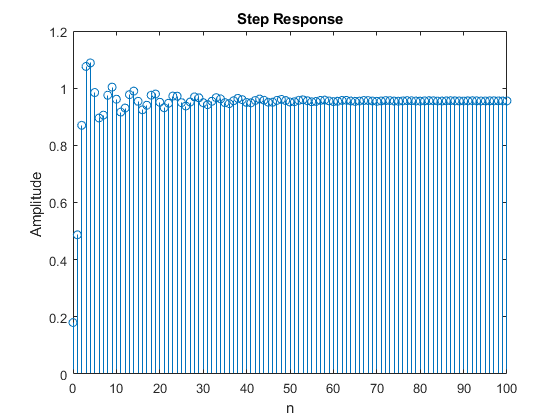
(b) Compute and plot the output y[n], if x[n] = u[n], 0 ≤ n ≤ 100 using the function

**y=filter(b,a,x)**.

y = filter(b,a,ones(1,length(n)));

stem(n,y)

title('Step Response') ; xlabel('n') ; ylabel('Amplitude ')



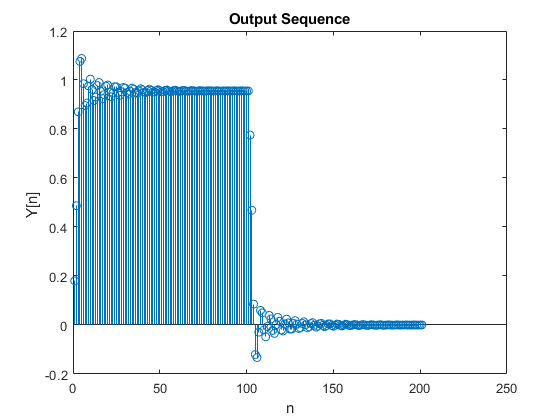
(c) Compute and plot the output y[n], if x[n] = u[n], 0 ≤ n ≤ 100 using the function

**y=conv(h,x).**

y = conv(h,ones(1,length(n)));

stem(y)

title('Output Sequence') ; xlabel('n') ; ylabel('Y[n] ')



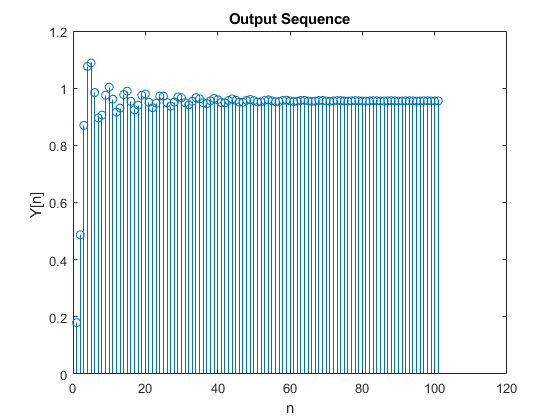
(d) Compute and plot the output y[n], if x[n] = u[n], 0 ≤ n ≤ 100 using the function

**y=filter(h,1,x)**

y = filter(h,1,ones(1,length(n)));

stem(y)

title('Output Sequence') ; xlabel('n') ; ylabel('Y[n] ')



Part A gives the Impulse Response of the System

Part B gives the Step Response of the System

Part C by convolving gives both the Step and Impulse Response of the system

Part D gives the Zero State Response of the System, this mean the response of the system when initial conditions are set to Zero which is ame as Step Response of the System.

2.22

A downsampler system is defined in (2.24). Consider the sequence x[n] = cos(0.1πn)

for −30 ≤ n ≤ 30. Using the stem function plot

(a) x[n] versus n.

(b) A down sampled signal y[n] for M = 5.

(c) A down sampled signal y[n] for M = 20.

(d) How does the downsampled signal appear? Compressed or expanded.

**y[n] = H{x[n]} = x[nM] (2.24)**

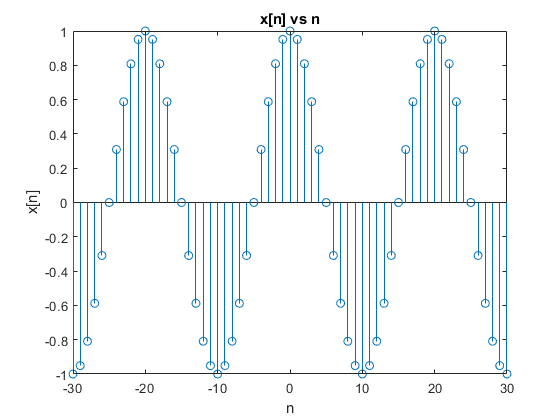
**a)**

n = [-30:30];

xn = cos(0.1\*pi\*n);

stem(n,xn)

title('x[n] vs n'); xlabel('n') ; ylabel('x[n]');



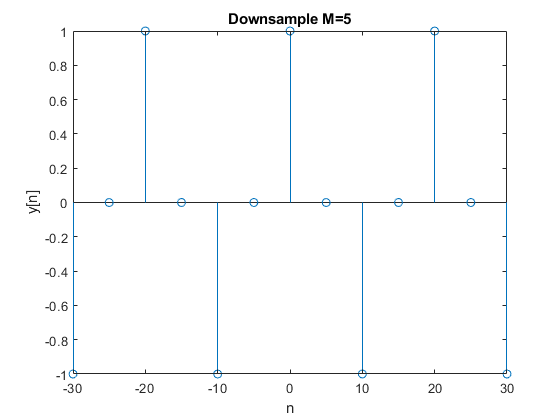
**b)**

n5d = downsample(n,5);

xn5d = downsample(xn,5);

stem(n5d,xn5d)

title('Downsample M=5'); xlabel('n') ; ylabel('y[n]');



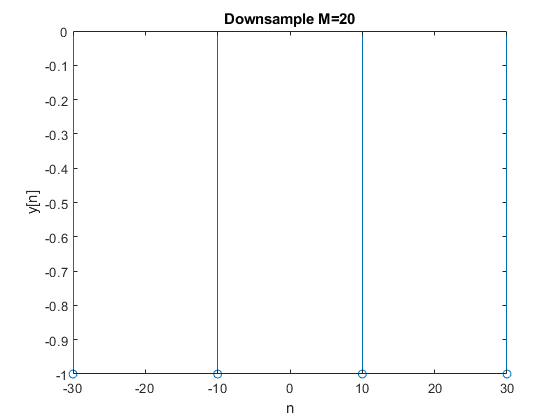
**c)**

n20d = downsample(n,20);

xn20d = downsample(xn,20);

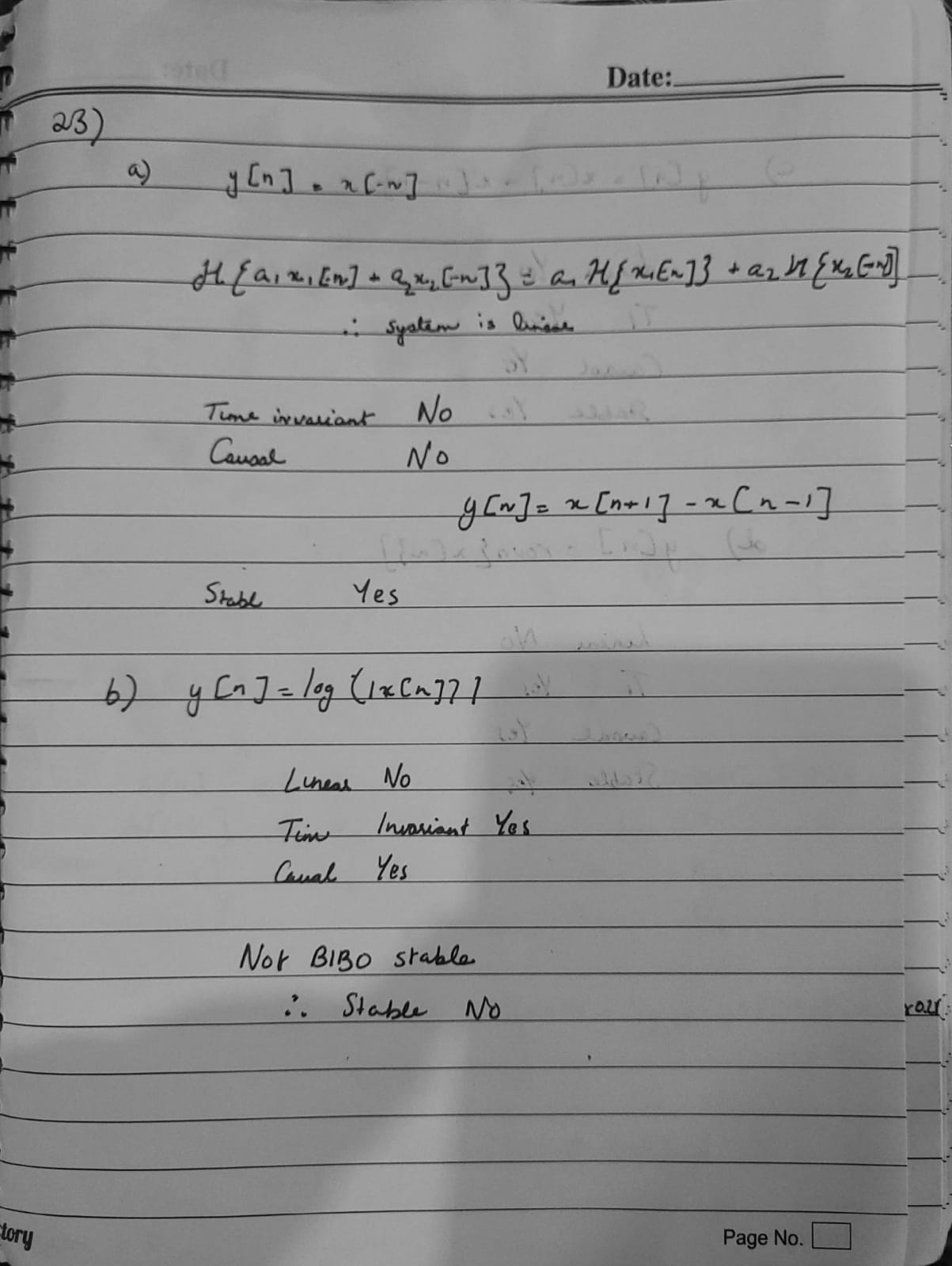
stem(n20d,xn20d)

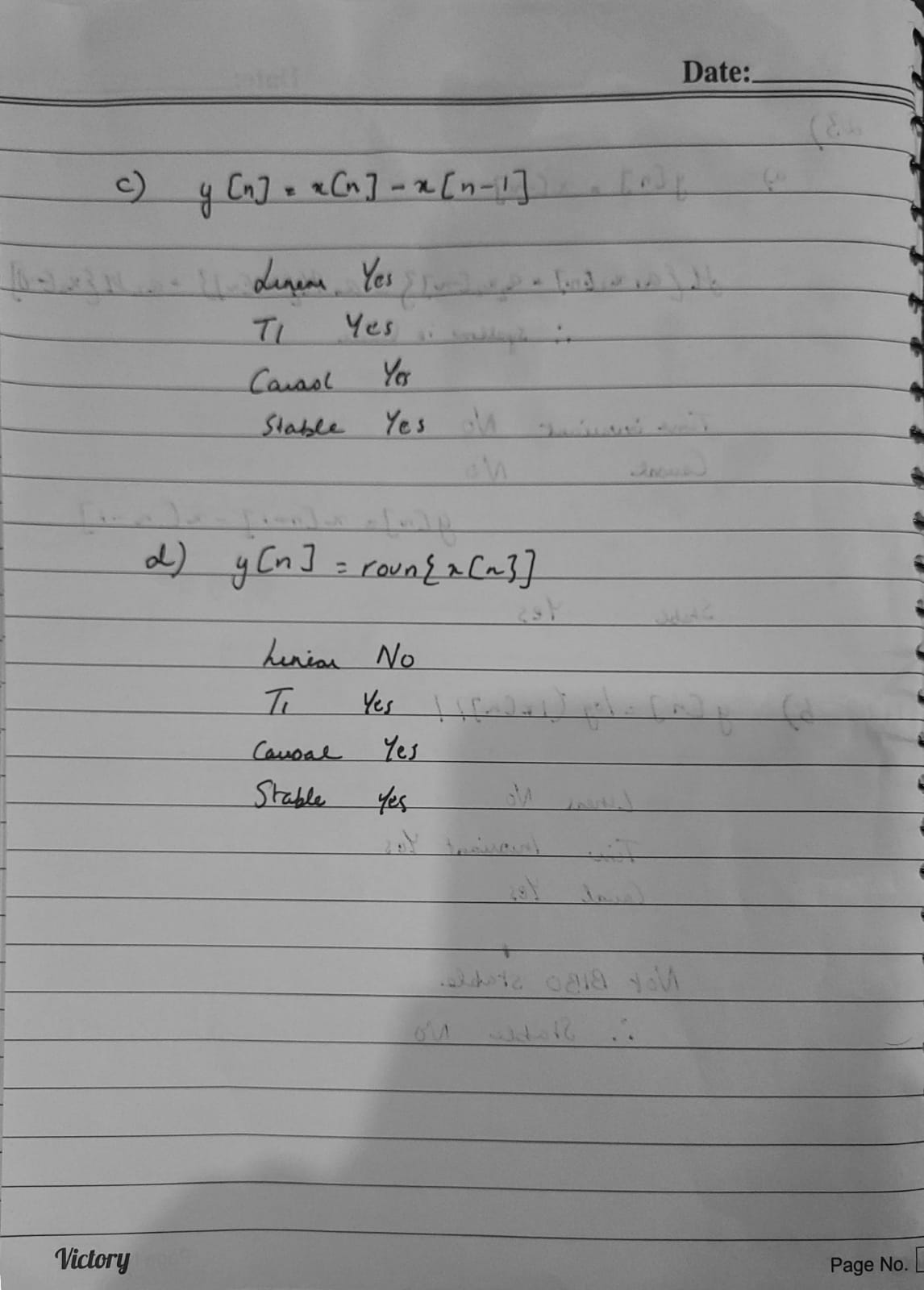
title('Downsample M=20'); xlabel('n') ; ylabel('y[n]');



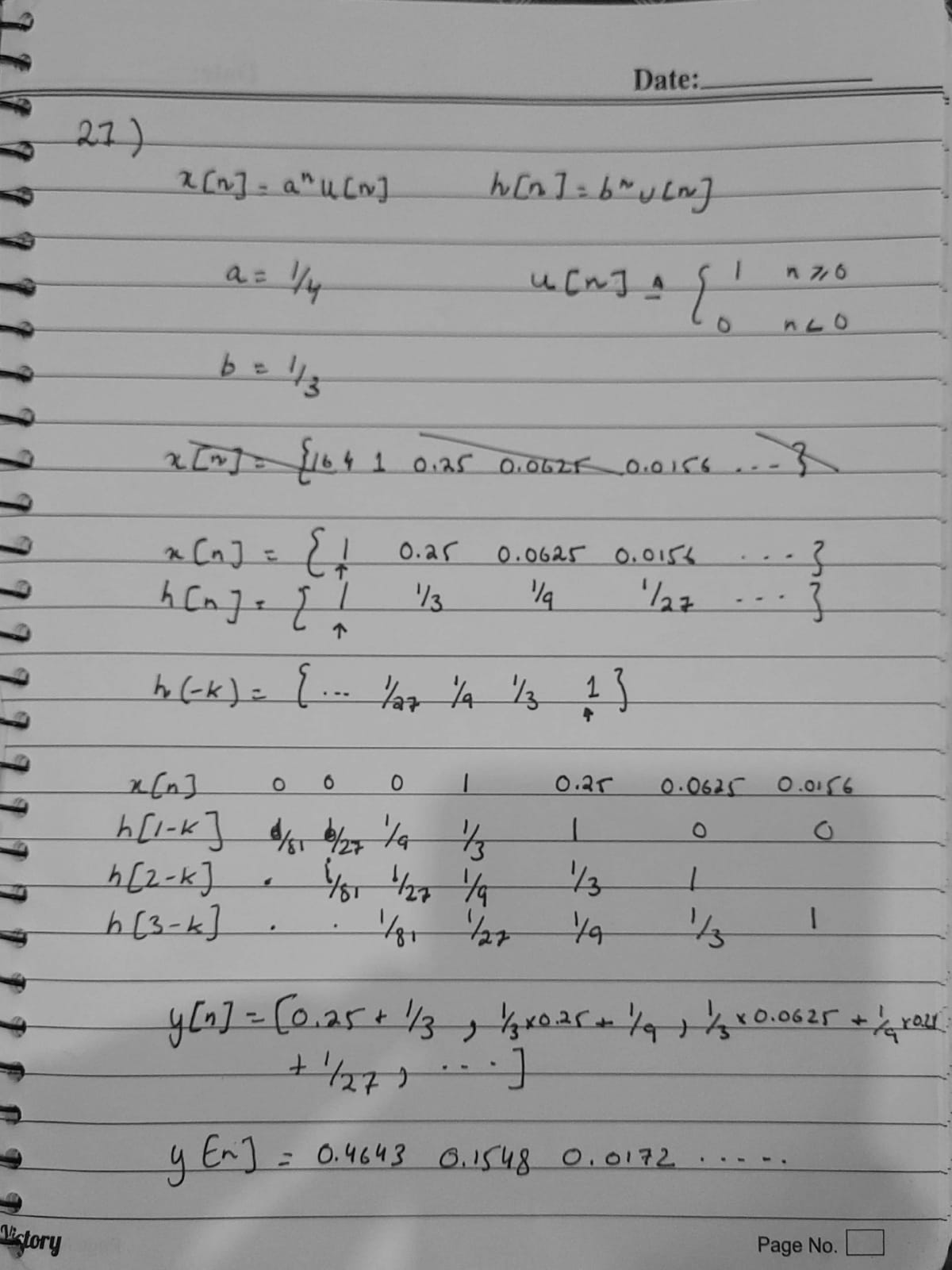
**d) Compressed**

2.23





2.27



n = 1:100;

un = ones(1,length(n));

an = (1/4) \* ones(1,length(n));

an = an.^n;

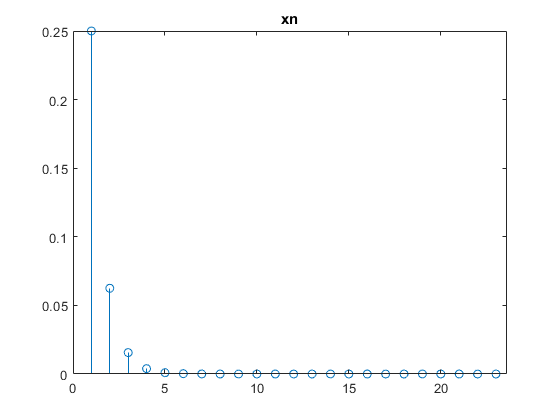
xn = an.\*un;

stem(xn)

title('xn')

xlim([0.0 23.6])

ylim([0.000 0.250])



bn = (1/3) \* ones(1,length(n));

bn = bn.^n;

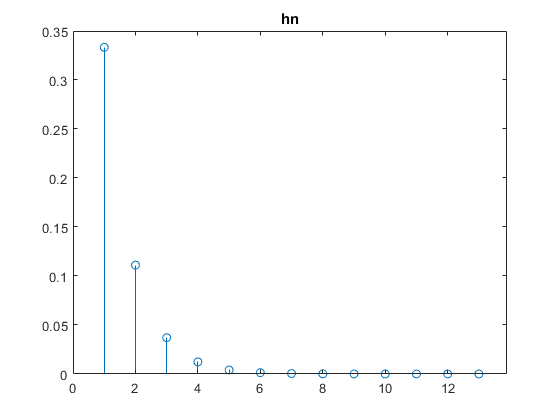
hn = bn.\*un;

stem(hn)

title('hn')

xlim([0.0 13.9])

ylim([0.000 0.350])

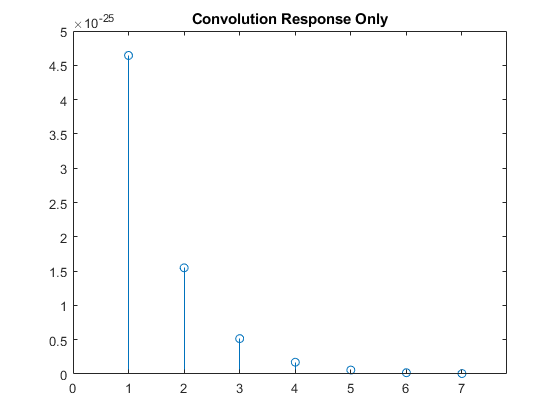


yn = conv(xn,hn,'same');

stem(yn)

title('Convolution Response Only')

xlim([0.00 7.81])



stem(yn)

hold on

stem(xn)

stem(hn)

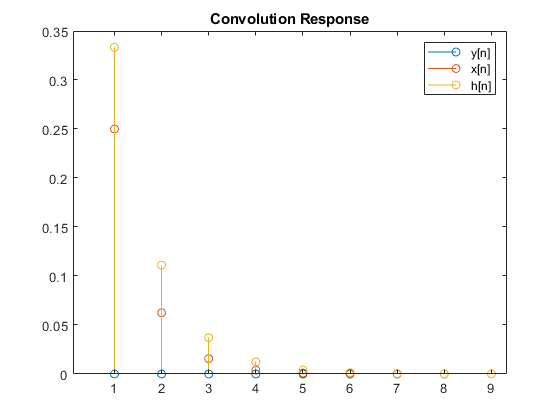
title('Convolution Response')

legend('y[n]','x[n]','h[n]')

xlim([0.12 9.33])

ylim([0.000 0.350])

hold off



2.31

Consider the system y[n] = y[n − 1] + y[n − 2] + x[n], y[−1] = y[−2] = 0.

(a) Compute and plot the impulse response, for 0 ≤ n ≤ 100, using function filter.

(b) Can you draw any conclusions about the stability of this system from the results

in (a)?

(c) Determine the output y[n], if the input is x[n] = an, −∞ < n < ∞, and comment

upon the result.

**a)**

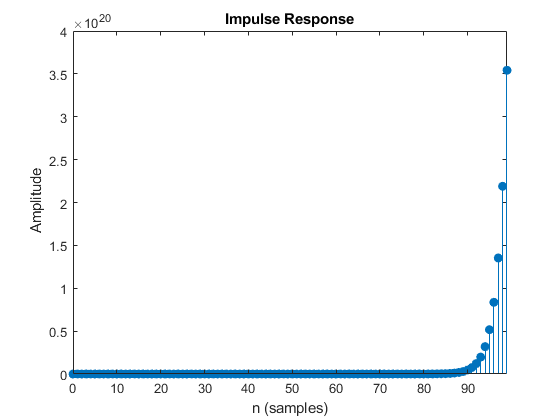
a = [1 -1 -1];

b = [1];

n = [1;100];

figure;

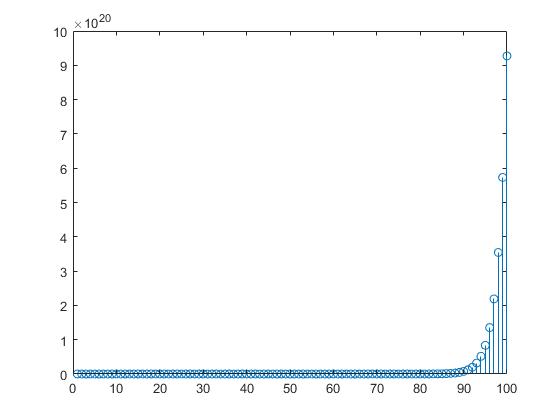
impz(b,a,100)



y = filter(b,a,ones(1,100));

figure;

stem(y)



**b) The Function is not decaying and continues and hence is not stable.**

**c)**

c = [2];

d = [10];

e = [50];

figure;

subplot(3,1,1)

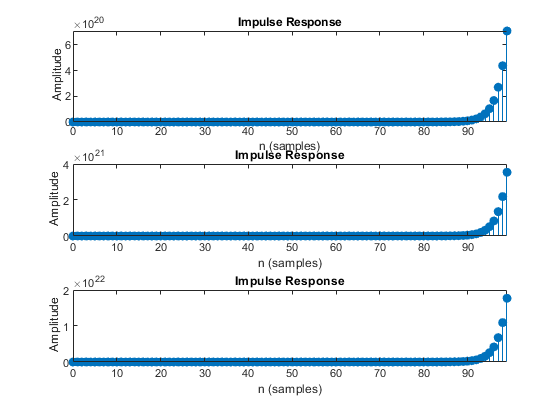
impz(c,a,100)

subplot(3,1,2)

impz(d,a,100)

subplot(3,1,3)

impz(e,a,100)



The shape of impulse response will remain the same, however. the amplitude now increases at a greater exponential scale.