Figures 1



Figure2:



Figure 3



Codes:

Running script: HW2.m:

%Create input signal (rectangular pulse)

n\_x1 = -5;

n\_x2 = 20;

n\_x = n\_x1:n\_x2;

n\_on = 0;

n\_off = 11;

x = u(n\_x-n\_on)-u(n\_x-n\_off);

%Define system:

a = 0.5; %exponential parameter

n\_0 = 0; %Time at which we start solving the equation.

k\_0 = find(n\_x==n\_0); %Index of time at which we start solving eqn.

n\_yA1 = n\_x1;

n\_yA2 = n\_x2;

n\_yA = n\_yA1 : n\_yA2;

y\_A = zeros(size(n\_yA));

a\_fo = [1 -a];

b\_fo = 1;% The a and b parameters from the general filter command.

y\_A(k\_0:end) = filter(b\_fo,a\_fo,x(k\_0:end));

figure(1);

subplot(211);

stem(n\_x,x);

lw = 1.5;

l = get(gca, 'children');

set(l,'linewidth',lw);

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

title('Input Signal');

subplot(212);

stem(n\_yA,y\_A);

lw = 1.5;

l = get(gca,'children');

set(l,'linewidth',lw);

xlabel('n');ylabel('y[n]');

title('Output Signal, Solution of Difference Equation');

c = 1e-5; % cut-off value

n\_h2 = ceil(log(c)/log(abs(a))); % cut-off index

n\_h1 = n\_x1; nh = n\_h1:n\_h2; % impulse response indexing

n\_yB1 = n\_x1+n\_h1; n\_yB2 = n\_x2+n\_h2; n\_yB = n\_yB1:n\_yB2; % output indexing

h\_fo\_trunc = hfo(nh,a);

y\_B = conv(x,h\_fo\_trunc);

figure(2);

subplot(211);

stem(n\_x,x);

lw = 1.5;

l = get(gca, 'children');

set(l,'linewidth',lw);

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

title('Input Signal');

subplot(212);

stem(n\_yB,y\_B);

lw = 1.5;

l = get(gca,'children');

set(l,'linewidth',lw);

xlabel('n');ylabel('y[n]');

title('Output Signal, Convolving Input with Impulse Response');

y\_C = sfo(n\_x-n\_on,a)-sfo(n\_x-n\_off,a);

figure(3);

subplot(211);

stem(n\_x,x);

lw = 1.5;

l = get(gca, 'children');

set(l,'linewidth',lw);

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

title('Input Signal');

subplot(212);

stem(n\_x,y\_C);

lw = 1.5;

l = get(gca,'children');

set(l,'linewidth',lw);

xlabel('n');ylabel('y[n]');

title('Output Signal, Combining Step Response');

u.m:

function y = u(x)

%unit step function

y = double(x>=0);

end

hfo.m:

function h = hfo(n,a)

h = (a.^n).\*u(n);

end

sfo.m:

function s = sfo(n,a)

s = 1/(1-a)\*(1-a.^(n+1)).\*u(n);

end