```
% 19ucc023
% Mohit Akhouri
% Experiment 4 - Observation 4

% In this code , we will perform matrix multiplication of Convolution
% matrix, DFT matrix and inverse of DFT matrix, next we will compare
 the
% results obtained with circular convolution of x[n] and h[n]

clc;
clear all;
close all;

% ALGORITHM : Initialize 8-point x[n] and h[n]
% First calculate the convolution matrix H and DFT matrix D
% Now calculate 8-point DFT of x[n]
% Then we perform the matrix multiplication and compare the results
% with the circular convolution of x[n] and h[n]

x = randperm(8,8); % input sequence x[n]
h = randperm(8,8); % impulse response h[n]
circ_conv_xh = my_Circular_Convolution(x,h); % Circular convolution of
 x[n] and h[n]

length_x = length(x); % length of input sequence x[n]
length_h = length(h); % length of impulse response h[n]

rows = 8; % rows of matrix ( both convolution matrix and DFT matrix )
columns = 8; % columns of matrix ( both convolution matrix and DFT
 matrix )
n = 8; % parameter for passing to myDFT function

H = myCirConvMat(h,length_x); % Circular convolution matrix H
[Xf , D_mat_x] = myDFT(x,8); % Calculating DFT matrix D_mat_x and DFT
 (Xf) of input sequence x[n]

D_mat_x_inverse = inv(D_mat_x); % Inverse of DFT matrix D_mat_x of
 x[n]

Hf = (D_mat_x * H)*(D_mat_x_inverse); % calculating Hf matrix

% calculating Yf = Hf * Xf
Yf = zeros(1,n); % initializing Yf vector

for i=1:rows
    sum = 0;
    for j=1:columns
        sum = sum + (Hf(i,j)*Xf(j));
    end
    Yf(i) = sum;
end
```

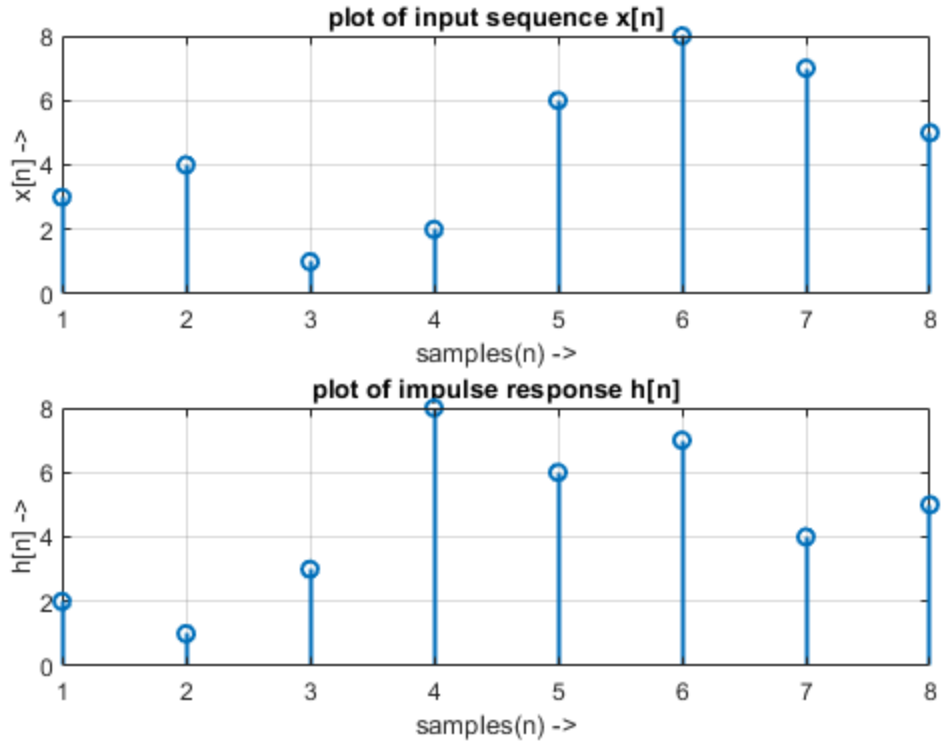
```
% calculating y = D8(-1) . Yf to be compared with circ_conv_xh
y = zeros(1,n); % initializing vector 'y'
for i=1:rows
    sum = 0;
    for j=1:columns
        sum = sum + (D_mat_x_inverse(i,j)*Yf(j));
    end
    y(i) = sum;
end

% plotting x[n] and h[n]
figure;
subplot(2,1,1);
stem(x,'Linewidth',1.5);
xlabel('samples(n) ->');
ylabel('x[n] ->');
title('plot of input sequence x[n]');
grid on;
subplot(2,1,2);
stem(h,'Linewidth',1.5);
xlabel('samples(n) ->');
ylabel('h[n] ->');
title('plot of impulse response h[n]');
grid on;
sgtitle('19ucc023 - Mohit Akhouri');

% plotting y[n] and circ_conv_xh for comparison of Circular
 convolution
figure;
subplot(2,1,1);
stem(y,'Linewidth',1.5);
xlabel('samples(n) ->');
ylabel('y[n] ->');
title('y[n] = D_{8}^{-1}.Y_{F} : circular convolution obtained via
 MATRIX MULTIPLICATION');
grid on;
subplot(2,1,2);
stem(circ_conv_xh,'Linewidth',1.5);
xlabel('samples(n) ->');
ylabel('x[n] \otimes h[n] ->');
title('Circular Convolution of x[n] and h[n] using my\_Circular
\_Convolution');
grid on;
sgtitle('19ucc023 - Mohit Akhouri');
```

Circular convolution matrix is as follows :

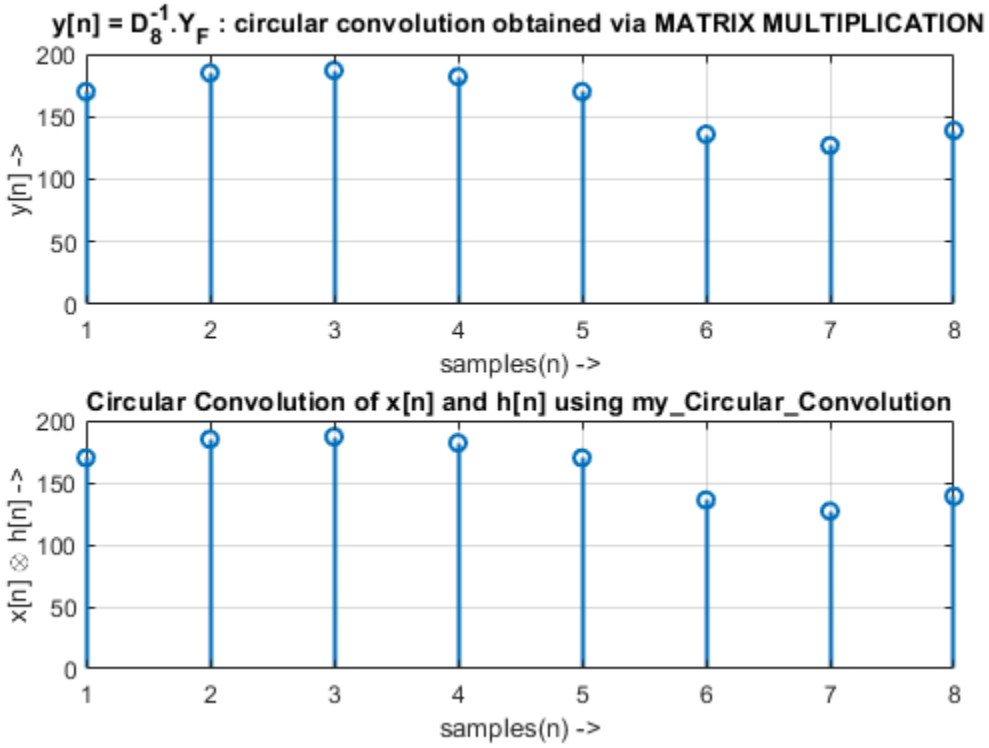
2	*5*	*4*	*7*	*6*	*8*	*3*	*1*
1	*2*	*5*	*4*	*7*	*6*	*8*	*3*
3	*1*	*2*	*5*	*4*	*7*	*6*	*8*
8	*3*	*1*	*2*	*5*	*4*	*7*	*6*
6	*8*	*3*	*1*	*2*	*5*	*4*	*7*
7	*6*	*8*	*3*	*1*	*2*	*5*	*4*
4	*7*	*6*	*8*	*3*	*1*	*2*	*5*
5	*4*	*7*	*6*	*8*	*3*	*1*	*2*

```
Circular convolution matrix is as follows :
     2     5     4     7     6     8     3     1
     1     2     5     4     7     6     8     3
     3     1     2     5     4     7     6     8
     8     3     1     2     5     4     7     6
     6     8     3     1     2     5     4     7
     7     6     8     3     1     2     5     4
     4     7     6     8     3     1     2     5
     5     4     7     6     8     3     1     2

The DFT matrix is given as :
  Columns 1 through 4

   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 +
 0.0000i
   1.0000 + 0.0000i   0.7071 - 0.7071i   0.0000 - 1.0000i  -0.7071 -
 0.7071i
   1.0000 + 0.0000i   0.0000 - 1.0000i  -1.0000 - 0.0000i  -0.0000 +
 1.0000i
   1.0000 + 0.0000i  -0.7071 - 0.7071i  -0.0000 + 1.0000i   0.7071 -
 0.7071i
   1.0000 + 0.0000i  -1.0000 - 0.0000i   1.0000 + 0.0000i  -1.0000 -
 0.0000i
   1.0000 + 0.0000i  -0.7071 + 0.7071i   0.0000 - 1.0000i   0.7071 +
 0.7071i
   1.0000 + 0.0000i  -0.0000 + 1.0000i  -1.0000 - 0.0000i   0.0000 -
 1.0000i
   1.0000 + 0.0000i   0.7071 + 0.7071i  -0.0000 + 1.0000i  -0.7071 +
 0.7071i

  Columns 5 through 8

   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 +
 0.0000i
  -1.0000 - 0.0000i  -0.7071 + 0.7071i  -0.0000 + 1.0000i   0.7071 +
 0.7071i
   1.0000 + 0.0000i   0.0000 - 1.0000i  -1.0000 - 0.0000i  -0.0000 +
 1.0000i
  -1.0000 - 0.0000i   0.7071 + 0.7071i   0.0000 - 1.0000i  -0.7071 +
 0.7071i
   1.0000 + 0.0000i  -1.0000 - 0.0000i   1.0000 + 0.0000i  -1.0000 -
 0.0000i
  -1.0000 - 0.0000i   0.7071 - 0.7071i  -0.0000 + 1.0000i  -0.7071 -
 0.7071i
   1.0000 + 0.0000i  -0.0000 + 1.0000i  -1.0000 - 0.0000i  -0.0000 -
 1.0000i
  -1.0000 - 0.0000i  -0.7071 - 0.7071i  -0.0000 - 1.0000i   0.7071 -
 0.7071i

Warning: Using only the real component of complex data.
```

19ucc023 - Mohit Akhouri



19ucc023 - Mohit Akhouri



Published with MATLAB® R2020b