```
% 19ucc023
% Mohit Akhouri
% Experiment 4 - Observation 3

% This code will verify the property that :
% if x[n] and h[n] are sequences then CIRCULAR CONVOLUTION
% of x[n] and h[n] is equal to IDFT(X(k)*H(k))

clc;
clear all;
close all;

% ALGORITHM : Initialize x[n] and h[n]
% First calculate DFT of x[n] and h[n] , let it be X(k) and H(k)
% Now , multiply X(k) and H(k) point by point and finally pass them
% thtough myIDFT function to calculate IDFT. IDFT obtained can now be
% compared with the circular convolution of x[n] and h[n]

% randperm(N,K) : returns a random permutation of K integers from 1 to
 N

x = randperm(8,8); % taking a random sequence x[n]
h = randperm(8,8); % taking a random sequence h[n]

% calculating the circular convolution of x[n] and h[n]

circ_conv_xh = my_Circular_Convolution(x,h);

% Finding the 8-point DFT sequence of x[n] and h[n]

[X,Dx] = myDFT(x,8); % DFT of input sequence x[n] , Dx is DFT matrix
 of x[n]
[H,Dh] = myDFT(h,8); % DFT of impulse response h[n] , Dh is DFT matrix
 of h[n]

% Multiplying X(k) and H(k) sample by sample ( point-by-point )

Y = X .* H; % point-by-point multiplication of X(k) and H(k)

% Finding the IDFT of Y(k)

y = myIDFT(Y,8); % IDFT of Y(k) to be compared with circ_conv_xh

% Plotting the results and comparing the IDFT y[n] with circular
% convolution circ_conv_xh

% plotting x[n],h[n],X(k) and H(k)
figure;
subplot(2,2,1);
stem(x,'Linewidth',1.5);
xlabel('samples(n) ->');
ylabel('x[n] ->');
```

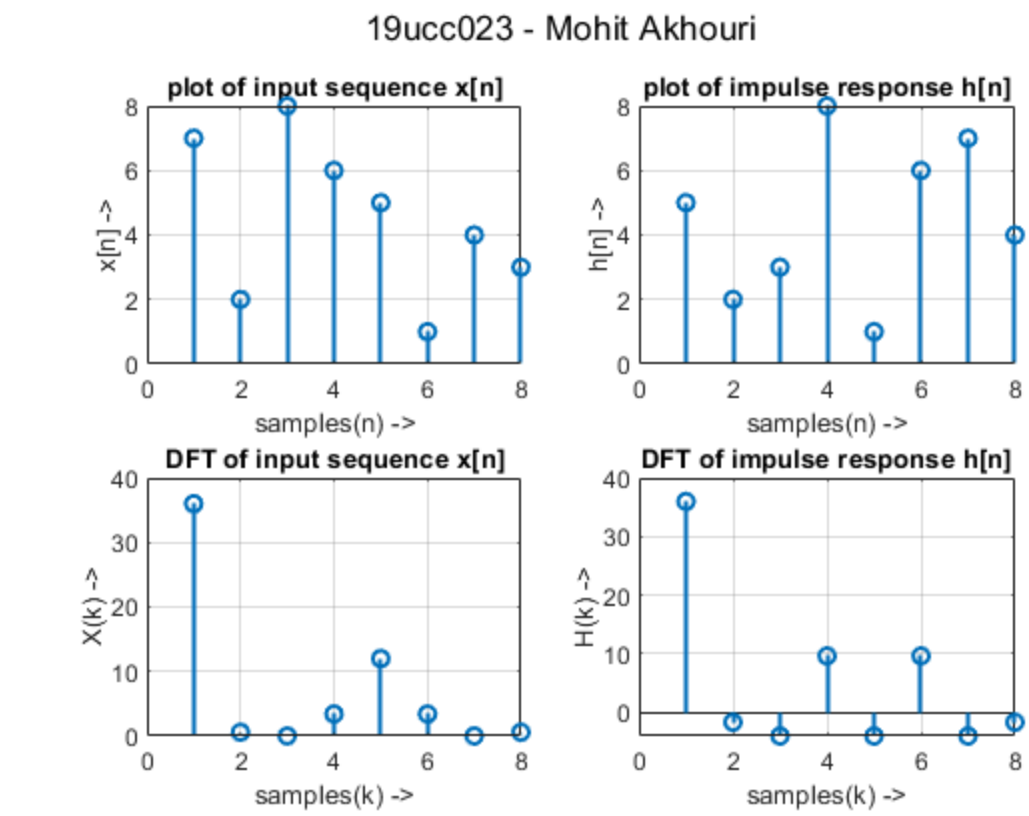
```
title('plot of input sequence x[n]');
grid on;
subplot(2,2,2);
stem(h,'Linewidth',1.5);
xlabel('samples(n) ->');
ylabel('h[n] ->');
title('plot of impulse response h[n]');
grid on;
subplot(2,2,3);
stem(X,'Linewidth',1.5);
xlabel('samples(k) ->');
ylabel('X(k) ->');
title('DFT of input sequence x[n]');
grid on;
subplot(2,2,4);
stem(H,'Linewidth',1.5);
xlabel('samples(k) ->');
ylabel('H(k) ->');
title('DFT of impulse response h[n]');
grid on;
sgtitle('19ucc023 - Mohit Akhouri');

% plotting Y(k) = X(k) .* H(k)
figure;
stem(Y,'Linewidth',1.5);
xlabel('samples(k) ->');
ylabel('Y(k) ->');
title('19ucc023 - Mohit Akhouri','Y(k) = X(k)*H(k) : MULTIPLICATION of
 X(k) and H(k) point-by-point in frequency domain');
grid on;

% plotting y[n] = IDFT(Y(k)) and circ_conv_xh for comparison of plots
figure;
subplot(2,1,1);
stem(y,'Linewidth',1.5);
xlabel('samples(n) ->');
ylabel('y[n] ->');
title('IDFT of seqeunce Y(k)');
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');
```

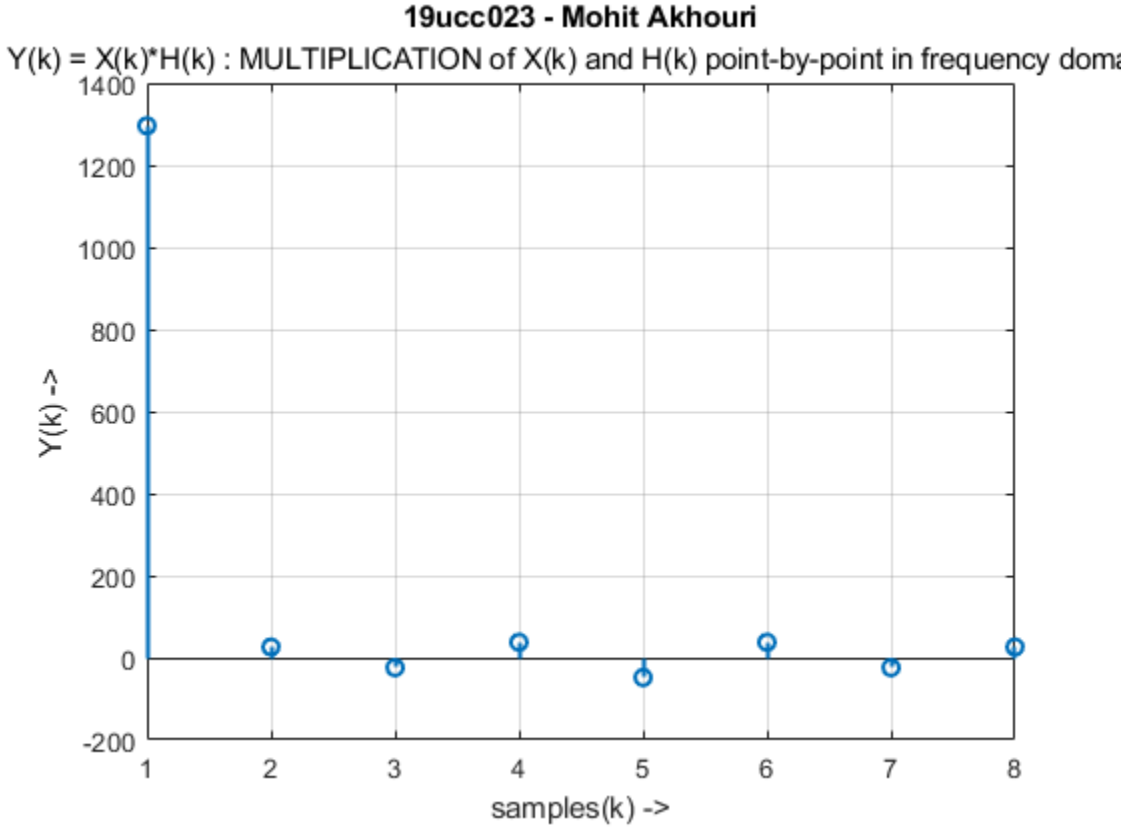
Circular convolution matrix is as follows :

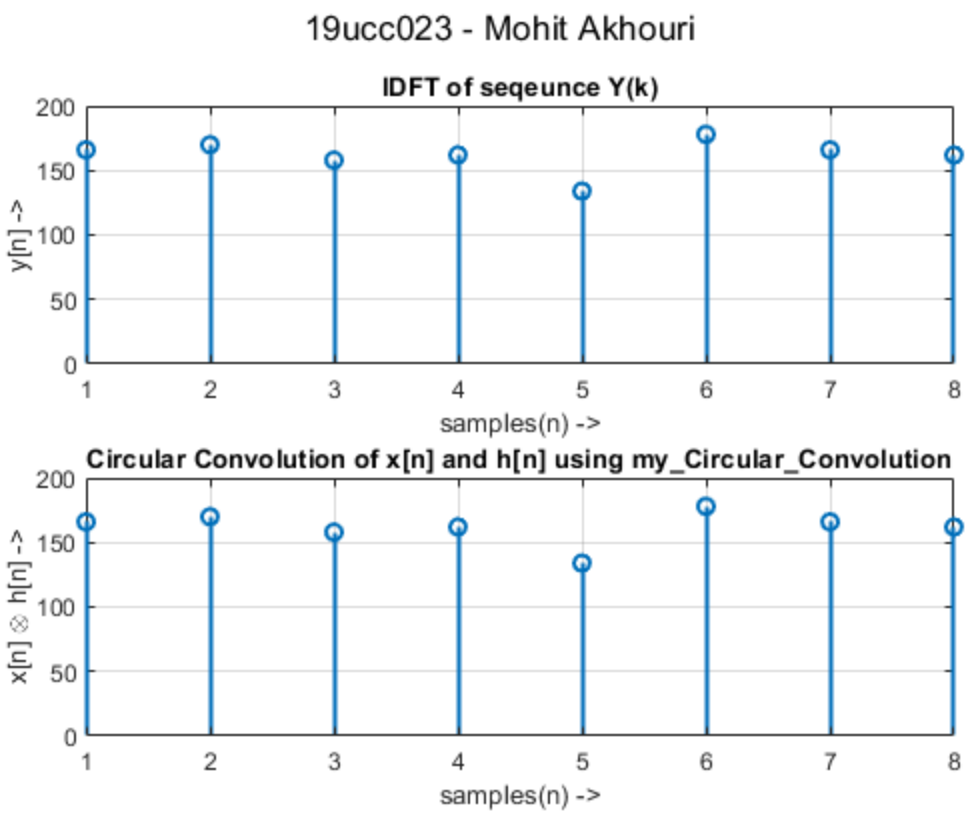
5	*4*	*7*	*6*	*1*	*8*	*3*	*2*
2	*5*	*4*	*7*	*6*	*1*	*8*	*3*
3	*2*	*5*	*4*	*7*	*6*	*1*	*8*
8	*3*	*2*	*5*	*4*	*7*	*6*	*1*
1	*8*	*3*	*2*	*5*	*4*	*7*	*6*

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

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

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


19ucc023 - Mohit Akhouri
plot of input sequence x[n]
x[n] ->
samples(n) ->
plot of impulse response h[n]
h[n] ->
samples(n) ->
DFT of input sequence x[n]
X(k) ->
samples(k) ->
DFT of impulse response h[n]
H(k) ->
samples(k) ->


19ucc023 - Mohit Akhouri
Y(k) = X(k)*H(k) : MULTIPLICATION of X(k) and H(k) point-by-point in frequency doma
Y(k) ->
samples(k) ->



Published with MATLAB® R2020b