School of Engineering and Applied Science (SEAS)
Ahmedabad University

BTech(ICT) Digital Signal Processing (Section 1)

Laboratory Assignment-3

Enrollment No: AU1841145 **Name: Samarth Shah**

AIM : Understand different concepts of convolution along with its applications.

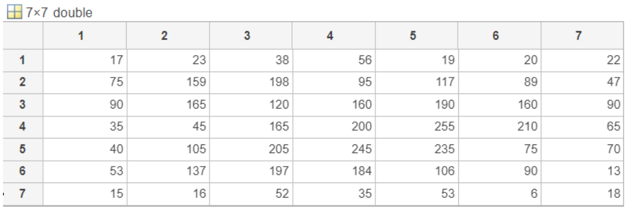
1. Solution Problem-1

   (a) Approach: In this question, first the order of the matrix was taken from the user and then the value of that matrix were taken. Total 2 times these step was repeated. After reshapping conv2 command used for 2d convolution between these two matrices and the designated output was assigned to variables.

   (b) Matlab Script:

```
% Name : Samarth Shah
% Roll No: AU1841145
% Lab3 (Question_1) Explore command conv2 in Matlab. Take input of 2 Matrix from
    user and Find 2D convolution of the same. Also explore the properties of conv2
     command and analyze the result.
clc ;
close all ;
clear ;
input_order = input ('Please enter order of your Square Matrix:'); % Enter order
    of matrix 1
for i =1: input_order ^2 %1 to n*n elements in for loop
    matrix1 (i)= input ('Please enter elements (One by One) -'); % input elements
    for square matrix
end
matrix1 = reshape ( matrix1 , input_order , input_order ); % proper shappig
input_order2 = input ('Please enter order of your Square Matrix:'); % Enter order
    of matrix 2
for i =1: input_order2 ^2 %1 to n*n elements in for loop
    matrix2 (i )= input ('Please enter elements (One by One) -'); % input elements
     for square matrix
end
matrix2 = reshape ( matrix2 , input_order2 , input_order2 ); % proper shapping
full_con = conv2 ( matrix1 , matrix2 ,'FULL CONV'); % full convolution
same = conv2 ( matrix1 , matrix2 ,'SAME CONV'); % same convolution
valid = conv2 ( matrix1 , matrix2 ,'VALID CONV'); % valid convolution
```

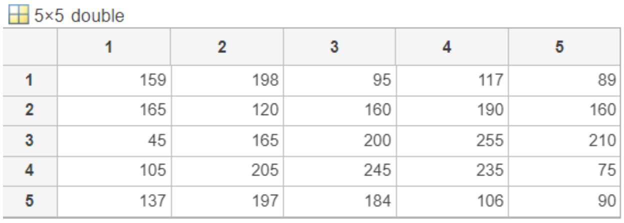
   (c) Simulation Output:



7×7 double

	1	2	3	4	5	6	7
1	17	23	38	56	19	20	22
2	75	159	198	95	117	89	47
3	90	165	120	160	190	160	90
4	35	45	165	200	255	210	65
5	40	105	205	245	235	75	70
6	53	137	197	184	106	90	13
7	15	16	52	35	53	6	18

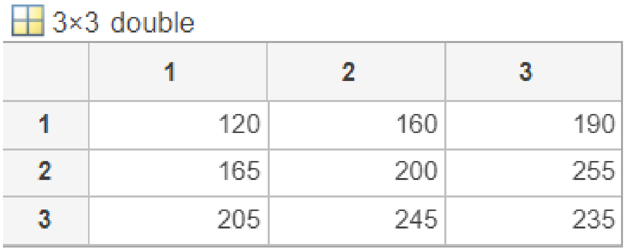
Figure:Full Covolution



5×5 double

	1	2	3	4	5
1	159	198	95	117	89
2	165	120	160	190	160
3	45	165	200	255	210
4	105	205	245	235	75
5	137	197	184	106	90

Figure: Same Convolution



3×3 double

	1	2	3
1	120	160	190
2	165	200	255
3	205	245	235

Figure: Valid Convolution

2. Solution Problem-2

   (a) Approach: In this question,there are two types of input. 1) Image 2) Kernels. Sample image was taken as a input for image filtration, it was converted to greyscale and different types of kernels were applied. Average, Sharpen, Edge, Edge Horizontal, Edge Vertical, Gradient Horizontal, Gradient Vertical, Sobel Horizontal, Sobel Vertical kernels were taken into consideration. Then the convolution of kernels with image done using conv2 and output was shown as a image filters.

   (b) Matlab Script:

```
% Name : Samarth Shah
% Roll No: AU1841145
% Lab3 (Question_2) Application of 2D convolution on image processing applications
close all ;
clear ;
image_read = imread ("lenna.png") ; % Read the input image
rgb = rgb2gray (image_read); %Convert RGB image or colormap to grayscale
im2gray = im2double ( rgb ); % Convert image to double format

avg = [1/9 ,1/9 ,1/9;1/9 ,1/9 ,1/9;1/9 ,1/9 ,1/9]; % Average Filter
sharp = [0 -1 0; -1 5 -1 ; 0 -1 0]; % Sharpen Filter
edge = [0 -1 0 ; -1 4 -1 ; 0 -1 0]; %Edge Filter
e_horizontal = [0 0 0 ; -1 2 -1; 0 0 0]; %Edge Horizontal Filter
e_vertical = transpose ( e_horizontal ); %Edge Vertical Filter
g_horizontal = [ -1 -1 -1 ; 0 0 0 ; 1 1 1]; % Gradient Horizontal Filter
g_vertical = transpose ( g_horizontal ); % Gradient Vertical Filter
s_horizontal = [1 2 1;0 0 0 ; -1 -2 -1]; % Sobel Horizontal Filter
s_vertical = transpose ( s_horizontal ); % Sobel Horizontal Filter
conv_average = conv2 ( im2gray , avg ,'SAME'); % Convolution with Average
conv_sharpen = conv2 ( im2gray , sharp ,'SAME'); % Convolution with Sharpen
conv_edge = conv2 ( im2gray , edge ,'SAME'); % Convolution with Edge
conv_edgehorizon = conv2 ( im2gray , e_horizontal ,'SAME'); % Convolution with Eh
conv_gradhorizon = conv2 ( im2gray , g_horizontal ,'SAME'); % Convolution with Gh
conv_edgevertical = conv2 ( im2gray , e_vertical ,'SAME'); % Convolution with Ev
conv_gradvertical = conv2 ( im2gray , g_vertical ,'SAME'); % Convolution with Gv
conv_sobelhorizon = conv2 ( im2gray , s_horizontal ,'SAME'); % Convolution with Sh
conv_sobelvertical = conv2 ( im2gray , s_vertical ,'SAME'); % Convolution with Sv

figure ;
imshow ( conv_average ) ; % image show
title ('Filter- 1: Average');
figure ;
imshow ( conv_sharpen ) ; % image show
title ('Filter- 2: Sharpen');
figure ;
imshow ( conv_edge ); % image show
title ('Filter- 3: Edge Detection 1');
figure ;
imshow ( conv_edgehorizon ) ; % image show
title ('Filter- 3: Edge Detection 2');
figure ;
imshow ( conv_edgevertical ); % image show
title ('Filter- 3: Edge Detection 3');
```

```

figure ;
imshow ( conv_gradhorizon ) ; % image show
title ('Filter- 4: Gradient Detection 1');
figure ;
imshow ( conv_gradvertical ); % image show
title ('Filter- 4: Gradient Detection 2');

figure ;
imshow ( conv_sobelhorizon ); % image show
title ('Filter- 5: Sobel Detection 1') ;

figure ;
imshow ( conv_sobelvertical ); % image show
title ('Filter- 5: Sobel Detection 2') ;
```

(c) Simulation Output:

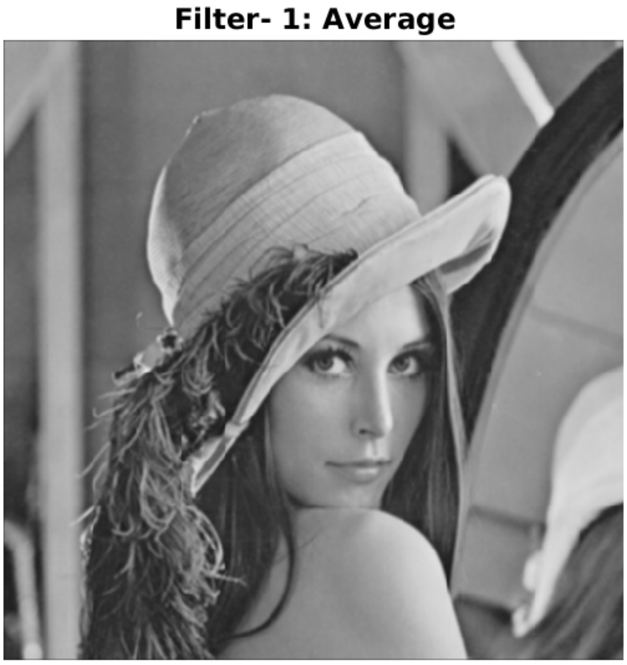


Figure:Average Filter

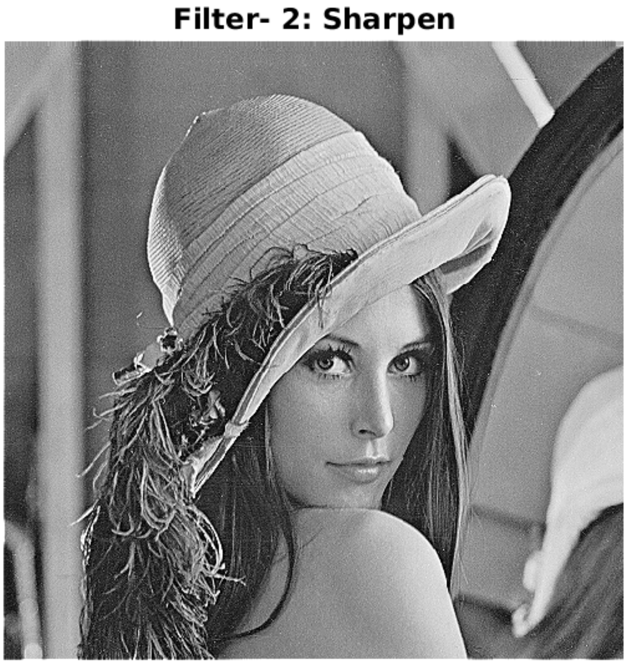


Figure: Sharpen Filter

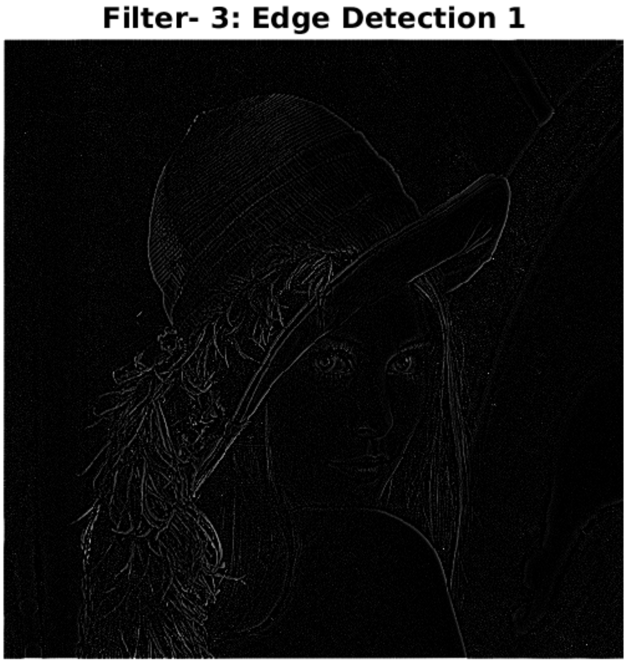


Figure:Edge Filter

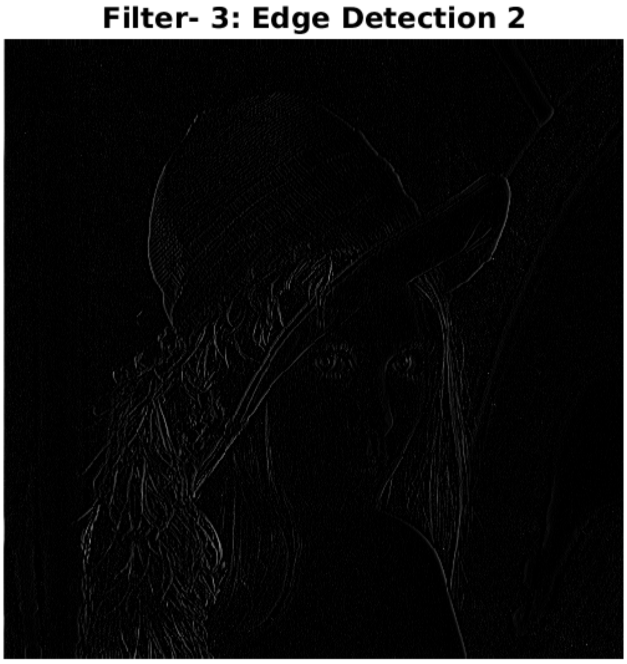


Figure: Edge Horizontal Filter

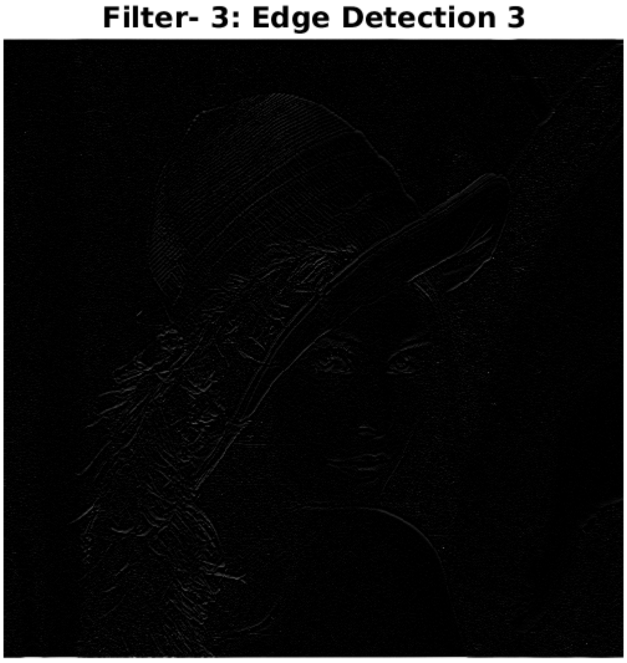


Figure:Edge Vertical Filter

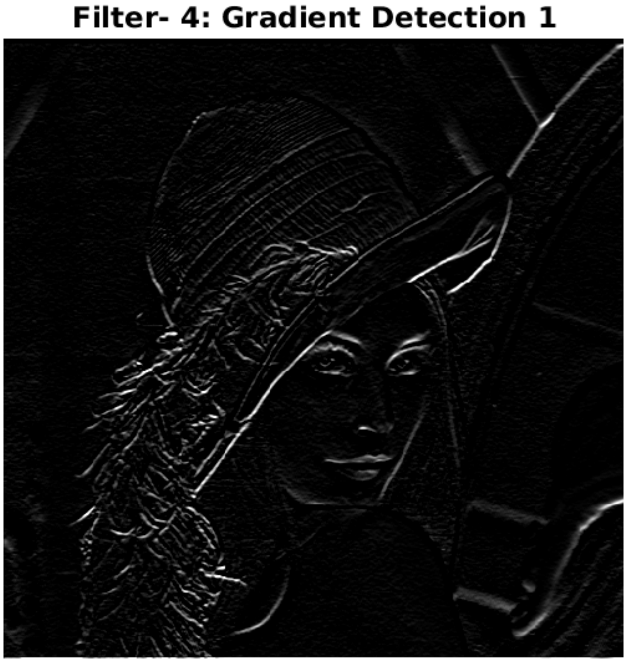


Figure: Gradient Horizontal Filter

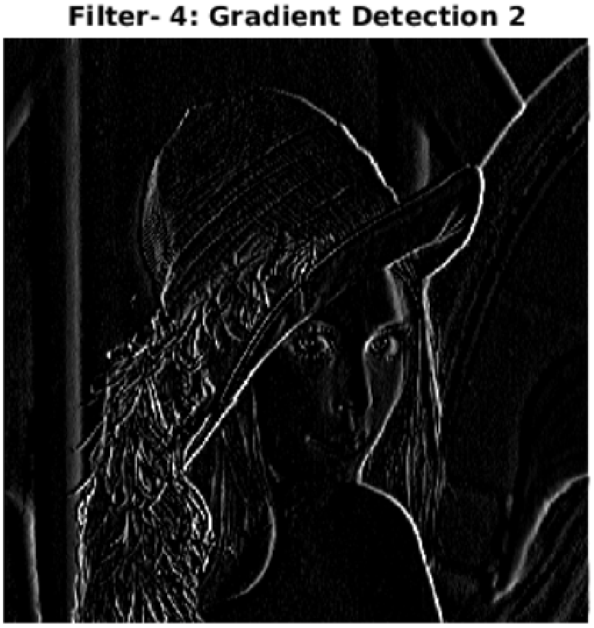


Figure:Gradient Vertical Filter

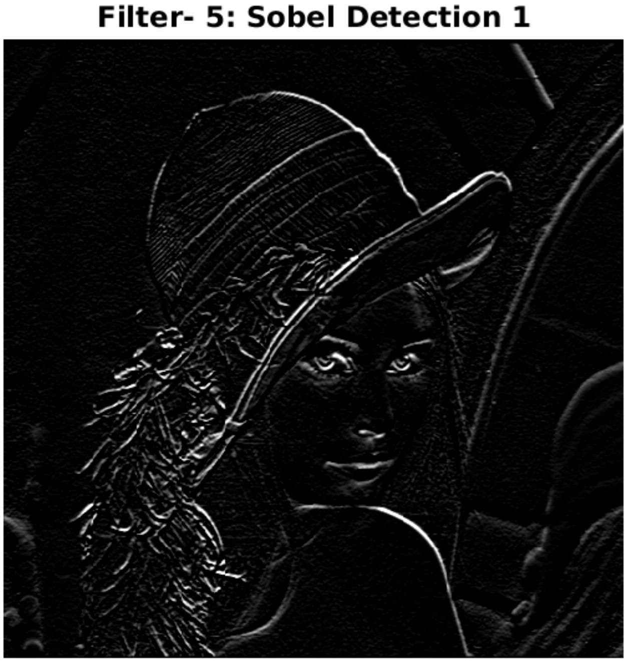


Figure:Sobel Horizontal Filter

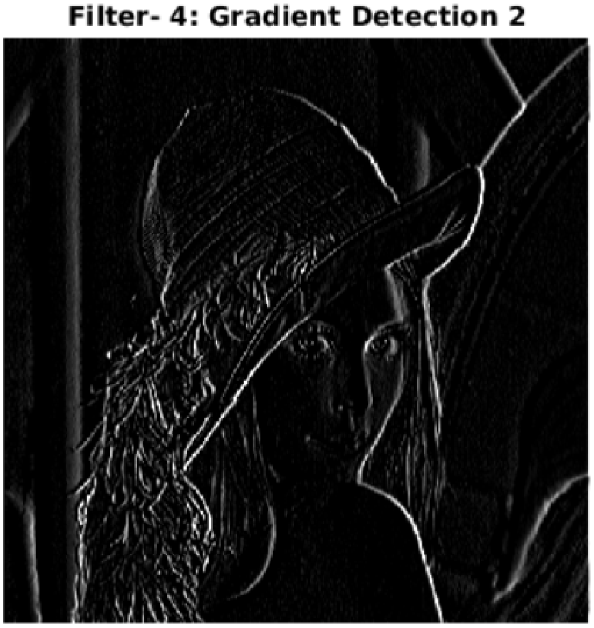


Figure:Sobel Vertical Filter

3. Solution Problem-3

   (a) Approach: First of all, both sequences were taken as a input from the user. Then found the maximum length of sequences and done circular convolution which was user defined function. Output was initialized to 0 and size to 1 and then using for loop, size and output of function was calculated and then plotted that using stem command for discrete plot. In the 3rd part, we needed to calculated cos and sin values from 0 to 8. and then the same procedure was carried out as mentioned above. Both sequences were circually convoluted and plotted using stem function.

(b) Matlab Script For 1 and 2 sub-questions:

```
% Name : Samarth Shah
% Roll No: AU1841145
% Lab3 (Question_3) Develop a MATLAB function to obtain circular convolution of two sequences
clc ;
close all ;
seq_1 = input (’Enter Sequence 1’) ; % Taking an input sequence
seq_2 = input (’Enter Sequence 2’) ; % Taking another input sequence
size_seq_1 = length ( seq_1 );% Calculating length of sequence 1
size_seq_2 = length ( seq_2 );% Calculating length of sequence 2
size_y = max ( size_seq_1 , size_seq_2 );% Maximum length of output sequence
% For equating both sequence length
seq_1 =[ seq_1 , zeros(1 , size_y - size_seq_1 ) ]; % adjusting length same
seq_2 =[ seq_2 , zeros(1 , size_y - size_seq_2 ) ]; % adjusting length same
Y = c_conv ( seq_1 , seq_2 , size_y ); % User defined function circular
% Convolution
size_seq_2 =1: length (Y );% Range
stem ( size_seq_2 , Y); % Discrete Plot
xlabel (’Range’);
ylabel (’Output’);
title (’Circular Convoluted Sequence’); % Graph of output
grid on ;
function out = c_conv ( input_x , input_h , size_y ) % circular convolution
    for n =1: size_y
        Y(n ) =0; % initializing the output
        for range =1: size_y
            size =n - range +1; % initializing the size of output
            if(size <=0)
                size = size_y + size ; % calculating total size
            end
            Y(n )= Y(n) +( input_x ( range )* input_h ( size )); %output
        end
    end
    out = Y ;
end
```

(c) Matlab Script For 3rd Sub-Question:

```
% Name : Samarth Shah
% Roll No: AU1841145
% Lab3 (Question_3) Develop a MATLAB function to obtain circular convolution of two sequences
clc ;
close all ;
range_seq_1 = 0:1:7; %0 to N -1 where N = 8
range_seq_2 = 0:1:7; %0 to N -1 where N = 8
func_1 = cos (2* pi* range_seq_1 /8) ; % Funcion 1
func_2 = sin (2* pi* range_seq_2 /8) ; % Function 2

size__funct_1 = length ( func_1 );% length of sequence 1
size__func_2 = length ( func_2 );% length of sequence 2
size_ouput = max ( size__funct_1 , size__func_2 );% length of output sequence y(n)
% For equating both sequence length

func_1 =[ func_1 , zeros(1 , size_ouput - size__funct_1 ) ]; % length same
func_2 =[ func_2 , zeros(1 , size_ouput - size__func_2 ) ]; % length same
Y = circularconv ( func_1 , func_2 , size_ouput ); % user defined function circular
% conv
size__func_2 =1: length (Y );% Range of all Sequences
stem ( size__func_2 , Y); % discrete plot
xlabel (’Range’);
ylabel (’Output’);
title (’Circular Convoluted Sequence’); % graph of output y
grid on ;
function out = circularconv ( seq_1 , seq_2 , output_size ) % circular convolution
```

```
    for n =1: output_size
        output(n ) =0; % initializing the output
        for range =1: output_size
            total_size =n - range +1; % initializing the size of y
            if(total_size <=0)
                total_size = output_size + total_size ; % calculating total size
            end
            output(n )= output(n) +( seq_1 ( range )* seq_2 ( total_size )); %Y
        end
    end
    out = output ;
end
```

(d) Simulation Output:

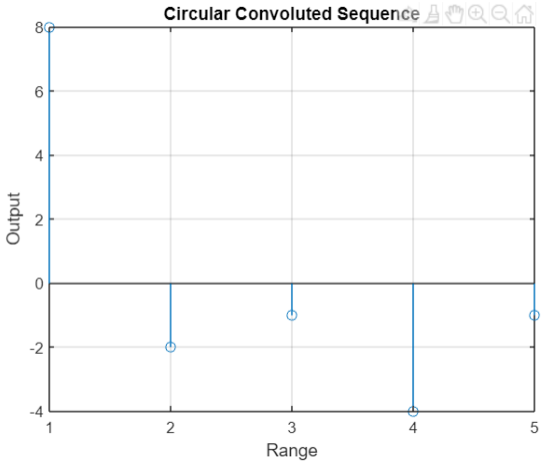


Figure:Sequence 1

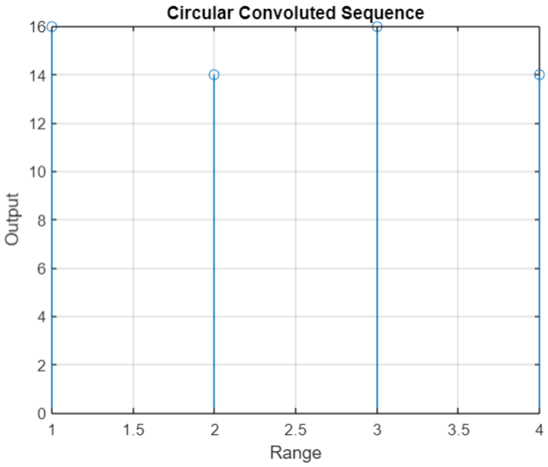
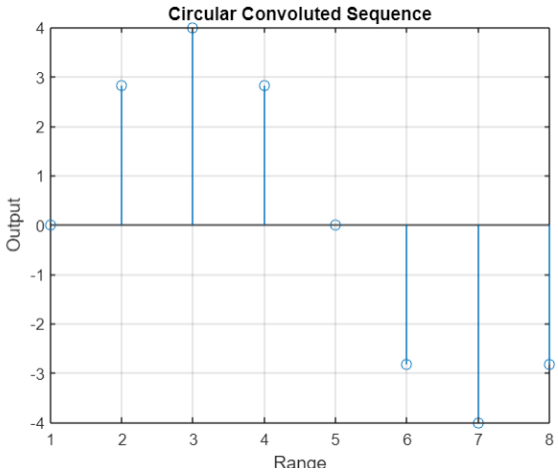


Figure:Sequence 2



4. Solution Problem-4

   (a) Approach: FBoth sequences were taken as a input from the user. Then found the maximum length of sequences(making the same length of sequences) and done circular convolution with was user defined function using circshift which is used for shifting circular right. After transpose of othe rmatrix, multiplication was carried out and the size of output from the function was calculated and then plotted that using stem command for discrete plot. In the 3rd part, we needed to calculated cos and sin values from 0 to 8. and then the same procedure was carried out as mentioned above. Both sequences were circually convoluted using transpose(to other matrix) and multiplication with each other and plotted using stem function.

   (b) Matlab Script For 1 and 2 sub-questions:

```
% Name : Samarth Shah
% Roll No: AU1841145
% Lab3 (Question_4) MATLAB program to find circular convolution of two sequences using Matrix Multiplication method.
clc ;
clear all ;
seq_1 = input (’Enter the First sequence :’) ; % Input Seqence 1
seq_2 = input (’Enter the Second sequence :’); % Input Sequence 2
output_convo = circular_conv (seq_1 , seq_2);
output_convo = transpose ( output_convo);
stem (0:1: length ( output_convo) -1 , output_convo);
xlabel (’Range’);
ylabel (’Output’);
title (’Circular Convolution’);

function out = circular_conv (seq_1 , seq_2)
    seq_1_length= length (seq_1 ); % length of first sequence
    seq_2_length = length (seq_2); % length of second sequence
    max_of_2 = max (seq_1_length , seq_2_length); % maximum of both
    % this if loop is for making x and h same
    if(seq_2_length == seq_1_length)
        seq1 = seq_1 ; % length same
        seq2 = seq_2 ; % length same
    elseif ( seq_2_length > seq_1_length)
        seq1 = [ seq_1 zeros(1 ,max_of_2 - seq_1_length) ];
        seq2 = seq_2 ;
    else
        seq1 = seq_1 ;
        seq2 = [ seq_2 zeros(1 ,max_of_2 - seq_2_length) ];
    end
    output = ones (max_of_2); % initializing to ones
    output (: ,1) = transpose (seq1) ; % transposing matrix
    for k = 2 : max_of_2
        seq1 = circshift (seq1 ,[1 1]) ; % circular shift right
        output (: , k ) = transpose (seq1) ; % transposing matrix
    end
    out = output * transpose (seq2) ; % matrix multiplication
end
```

   (c) Matlab Script For 3rd Sub-Question:

```
% Name : Samarth Shah
% Roll No: AU1841145
% Lab3 (Question_4) MATLAB program to find circular convolution of two sequences using Matrix Multiplication method.
clc ;
close all ;
clear all ;
range = 0:1:7; %deifning range
seq1 = cos (2* pi*range /8) ; %seq_1
seq2 = sin (2* pi*range /8) ; %seq_2
out1 = circular_conv (seq1 , seq2); % output
```

```
out1 = transpose (out1); % transpose
stem (0:1: length (out1) -1 , out1);% size and output plot in discrete form
xlabel ('Range ');
ylabel ('Output ');
title ('Circular Convolution ');

function out = circular_conv (seq_1 , seq_2)
    seq_1_length= length (seq_1 ); % length of first sequence
    seq_2_length = length (seq_2); % length of second sequence
    max_of_2 = max (seq_1_length , seq_2_length); % maximum of both
    % this if loop is for making x and h same
    if(seq_2_length == seq_1_length)
        seq1 = seq_1 ; % length same
        seq2 = seq_2 ; % length same
    elseif ( seq_2_length > seq_1_length)
        seq1 = [ seq_1 zeros(1 ,max_of_2 - seq_1_length) ];
        seq2 = seq_2 ;
    else
        seq1 = seq_1 ;
        seq2 = [ seq_2 zeros(1 ,max_of_2 - seq_2_length) ];
    end
    output = ones (max_of_2); % initializing to ones
    output (: ,1) = transpose (seq1) ; % transposing matrix
    for k = 2 : max_of_2
        seq1 = circshift (seq1 ,[1 1]) ; % circular shift right
        output (: , k ) = transpose (seq1) ; % transposing matrix
    end
    out = output * transpose (seq2) ; % matrix multiplication
end
```

(d) Simulation Output:

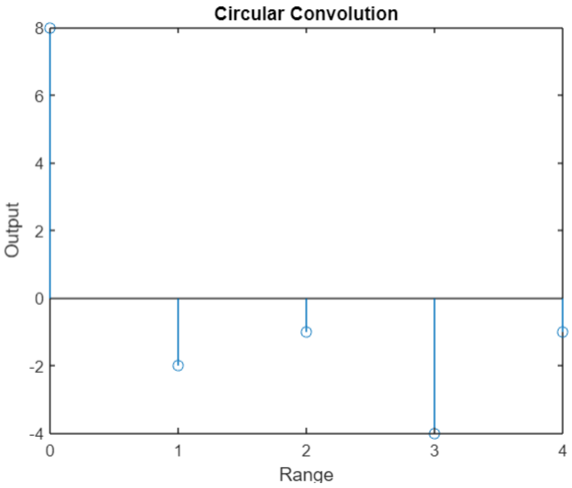


Figure:Sequence 1

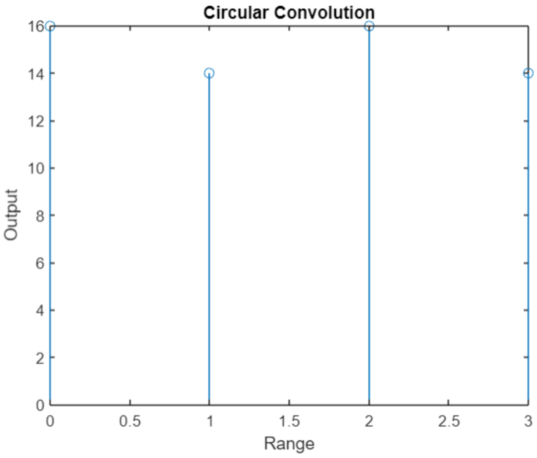
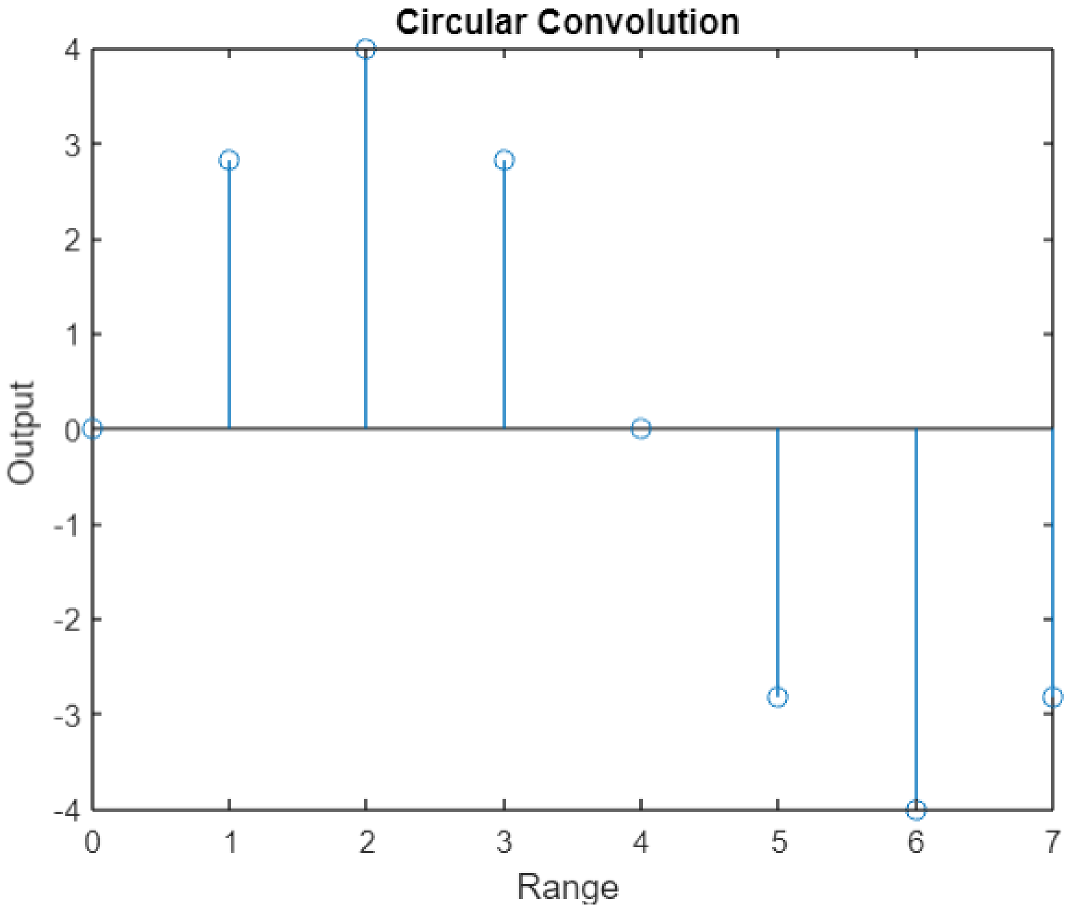


Figure:Sequence 2


Circular Convolution
4
3
2
1
0
-1
-2
-3
-4
Output
0
1
2
3
4
5
6
7
Range