MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY

Course Title: DSP Lab

Course Code: CSE4104.

Report No: 01.

Report Title: Lab Report on Basic MATLAB Operations and Signal Visualization .

LAB REPORT

Submitted by	Submitted to
Name :Alamgir Hosain ID:CE21012 Session: 2020-21 Year: 4th Semester: 1st Department: Computer Science and Engineering. Mawlana Bhashani Science and Technology University.	Dr. Mahbuba Begum Associate Professor, Department of Computer Science and Engineering, Mawlana Bhashani Science and Technology University.

Lab Report: Image Read, Write, Display, Dimension Change (RGB to Gray), and Resize.

Problem Definition
To understand and implement basic image processing operations using MATLAB, including reading an image, writing an image, displaying it, converting it from RGB to grayscale, and resizing it.
Objective

1. To read and display an image using MATLAB.
2. To convert a color image (RGB) to grayscale.
3. To resize an image.
4. To save the processed image to a file.

Tools Used

Software: MATLAB R202x (any version with Image Processing Toolbox)
Hardware**:** Computer with MATLAB installed.

Theory
Image Processing Toolbox in MATLAB provides a comprehensive set of reference-standard algorithms and workflow apps for image processing, analysis, visualization, and algorithm development.

1. Grayscale images contain shades of gray rather than color.
2. A standard method to convert RGB to grayscale is the luminosity method.
3. Gray (R(i,i)*0.33+G(i,j)*0.33+B(i,j)*0.33).
4. The nearest-neighbor interpolation technique maps each pixel in the new image to the closest pixel in the original image using scaling factors .

Method / Procedure:

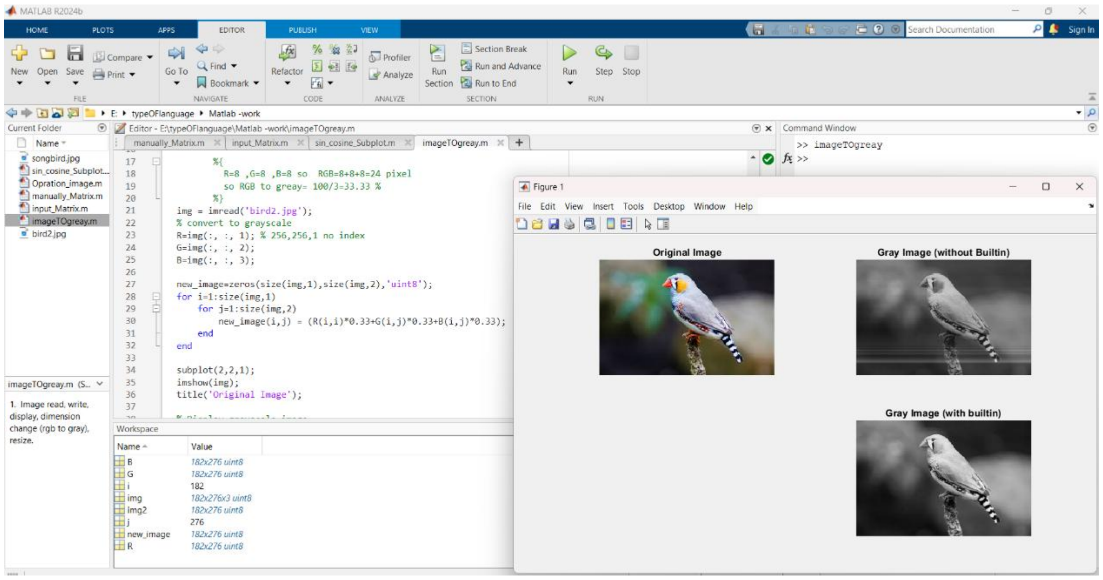
1. Read an image using \`imread\`.
2. Display the original image using imshow and subplot.
3. Manually convert the image to grayscale using nested loops and the luminosity formula.
4. 4.Display all processed images in a figure with subplots.

Code

```
  %{
 R=8 ,G=8 ,B=8 so  RGB=8+8+8=24 pixel
 so RGB to greay= 100/3=33.33 %
%}
img = imread('bird2.jpg');
% convert to grayscale

R=img(:, :, 1); % 256,256,1 no index
G=img(:, :, 2);
B=img(:, :, 3);
new_image=zeros(size(img,1),size(img,2),'uint8');
```

```
for i=1:size(img,1)
    for j=1:size(img,2)
        new_image(i,j) = (R(i,i)*0.33+G(i,j)*0.33+B(i,j)*0.33);
    end
end

subplot(2,2,1);
imshow(img);
title('Original Image');

% Display grayscale image
subplot(2,2,2);
imshow(new_image);
title('Gray Image (without Builtin)');

% Display grayscale image
img2=im2gray(img);
subplot(2,2,4);
imshow(img2);
title('Gray Image (with builtin)');
```

Result/Output (Screenshot)



Editor - E:\typeOFlanguage\Matlab -work\imageTOgreay.m

```
        %{
          R=8 ,G=8 ,B=8 so  RGB=8+8+8=24 pixel
          so RGB to greay= 100/3=33.33 %
        %}
img = imread('bird2.jpg');
% convert to grayscale
R=img(:, :, 1); % 256,256,1 no index
G=img(:, :, 2);
B=img(:, :, 3);

new_image=zeros(size(img,1),size(img,2),'uint8');
for i=1:size(img,1)
    for j=1:size(img,2)
        new_image(i,j) = (R(i,i)*0.33+G(i,j)*0.33+B(i,j)*0.33);
    end
end

subplot(2,2,1);
imshow(img);
title('Original Image');
```

```
>> imageTOgreay
>>
```

1. Image read, write, display, dimension change (rgb to gray), resize.

Workspace

Name	Value
B	182x276 uint8
G	182x276 uint8
i	182
img	182x276x3 uint8
img2	182x276 uint8
j	276
new_image	182x276 uint8
R	182x276 uint8

Result Discussion

The experiment successfully demonstrated basic image processing operations in MATLAB. The original image was read and displayed without error. The grayscale image accurately represented the luminance of the original image.

Lab Report: Create two matrices manually and perform operations: add, sub, multiplication, division.

Problem Definition

To manually create two matrices in MATLAB and perform basic matrix operations: addition, subtraction, multiplication, and division. The aim is to understand how MATLAB handles these operations and observe the differences between element-wise and matrix operations.

Objective

1. To create two matrices manually in MATLAB.
2. To perform and understand the results of matrix addition, subtraction, multiplication, and division.
3. To analyze element-wise operations versus matrix operations where applicable.

Tools Used

1. Software: MATLAB R202x (any version with Image Processing Toolbox)
2. Hardware**:** Computer with MATLAB installed.

Theory

In MATLAB, matrices are a fundamental part of the language. Matrix operations can be broadly classified into:

1. Addition/Subtraction (+, -): Performed element-wise and require matrices of the same dimensions.
2. Multiplication (*): Standard matrix multiplication; number of columns of the first matrix must match the number of rows of the second.
3. Division (/ or .\):Element-wise division (./): Divides corresponding elements.

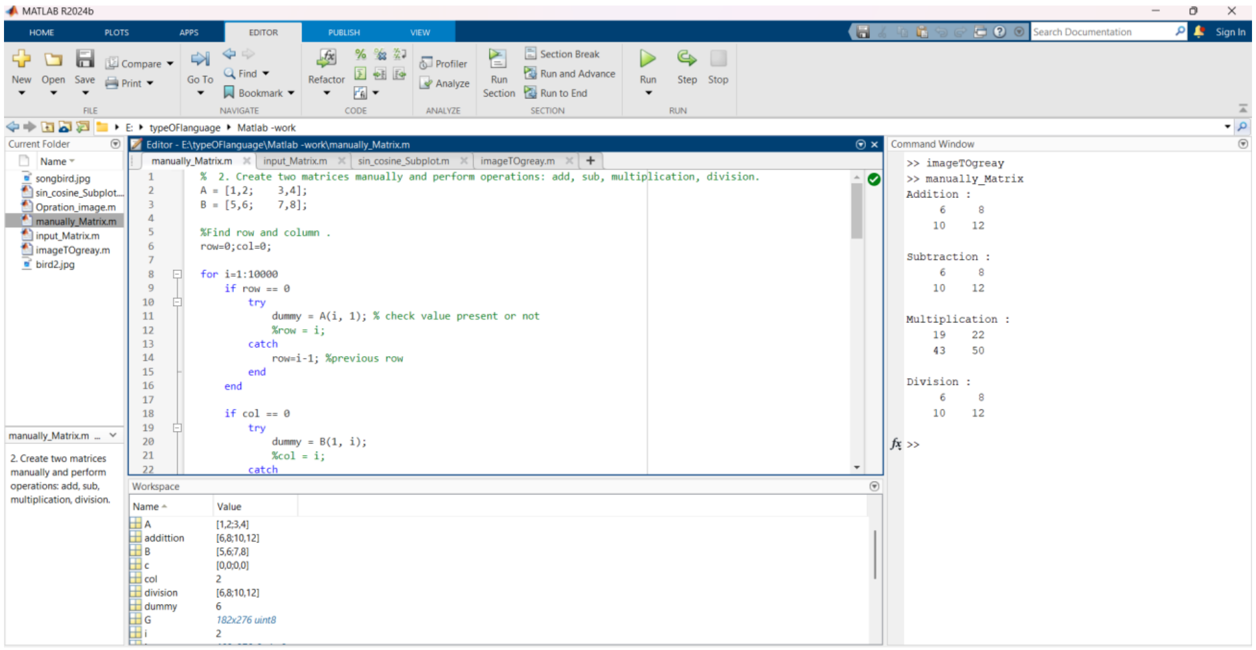
Method/Procedure

1. Open MATLAB and create a new script file.
2. Manually define two matrices A and B of the same size.
3. Perform addition, subtraction, element-wise multiplication, and division using MATLAB operators.
4. Display all matrices and results using disp() function.
5. Run the script and observe the output in the command window.

Code

```
A = [1,2;    3,4];
B = [5,6;    7,8];
%Find row and column .
row=0;col=0;

for i=1:10000
    if row == 0
        try
            dummy = A(i, 1); % check value present or not
            %row = i;
        catch
            row=i-1; %previous row
        end
    end
    if col == 0
        try
            dummy = B(1, i);
            %col = i;
        catch
            col=i-1; %previous column
        end
    end
     if row > 0 && col > 0
        break;
    end
end
addittion=zeros(row,col);
subtraction=zeros(row,col);
multiplication=zeros(row,col);
division=zeros(row,col);

for i = 1:row
    for j = 1:col
        addittion(i,j) = A(i,j)+B(i,j);
        subtraction(i,j) = A(i,j)+B(i,j);
        division(i,j) = A(i,j)+B(i,j);
    end
end
c=zeros(row,col);
for i = 1:row %row of A
    for j = 1:col %col of B
        sum_val = 0;
        for k = 1:col %col of A
            sum_val = sum_val + A(i, k) * B(k, j);
        end
        multiplication(i, j) = sum_val;
    end
end
disp('Addition :');
disp(addittion);
disp('Subtraction :');
disp(subtraction);
disp('Multiplication :');
disp(multiplication);
disp('Division :')
disp(division);
```

Result/Output



```matlab
%  2. Create two matrices manually and perform operations: add, sub, multiplication, division.
A = [1,2;    3,4];
B = [5,6;    7,8];

%Find row and column .
row=0;col=0;

for i=1:10000
    if row == 0
        try
            dummy = A(i, 1); % check value present or not
            %row = i;
        catch
            row=i-1; %previous row
        end
    end

    if col == 0
        try
            dummy = B(1, i);
            %col = i;
        catch
```

Command Window

```
>> imageTOgreay
>> manually_Matrix
Addition :
     6     8
    10    12

Subtraction :
     6     8
    10    12

Multiplication :
    19    22
    43    50

Division :
     6     8
    10    12

>>
```

Workspace

Name	Value
A	[1,2;3,4]
addittion	[6,8;10,12]
B	[5,6;7,8]
c	[0,0;0,0]
col	2
division	[6,8;10,12]
dummy	6
G	182x276 uint8
i	2

Result Discussion

1. Addition and subtraction results were calculated by adding/subtracting each corresponding element.
2. Element-wise multiplication resulted in each element of A being multiplied with the corresponding element in B.
3. Similarly, element-wise division divided each element of A by the corresponding element in B.

Lab Report: Create two matrices by taking user input and perform add, sub, multiplication, division operations.

Problem Definition

The purpose of this lab is to create two matrices by taking user input and perform basic arithmetic operations addition, subtraction, multiplication, and division element-wise between them using MATLAB.

Objective

4. To create two matrices manually in MATLAB.
5. To perform and understand the results of matrix addition, subtraction, multiplication, and division.
6. To analyze element-wise operations versus matrix operations where applicable.

Tools Used

3. Software: MATLAB R202x (any version with Image Processing Toolbox)
4. Hardware**:** Computer with MATLAB installed.

Theory

In MATLAB, matrices are a fundamental part of the language. Matrix operations can be broadly classified into:

4. Addition/Subtraction (+, -): Performed element-wise and require matrices of the same dimensions.
5. Multiplication (*): Standard matrix multiplication; number of columns of the first matrix must match the number of rows of the second.
6. Division (/ or .\):Element-wise division (./): Divides corresponding elements.

Method/Procedure

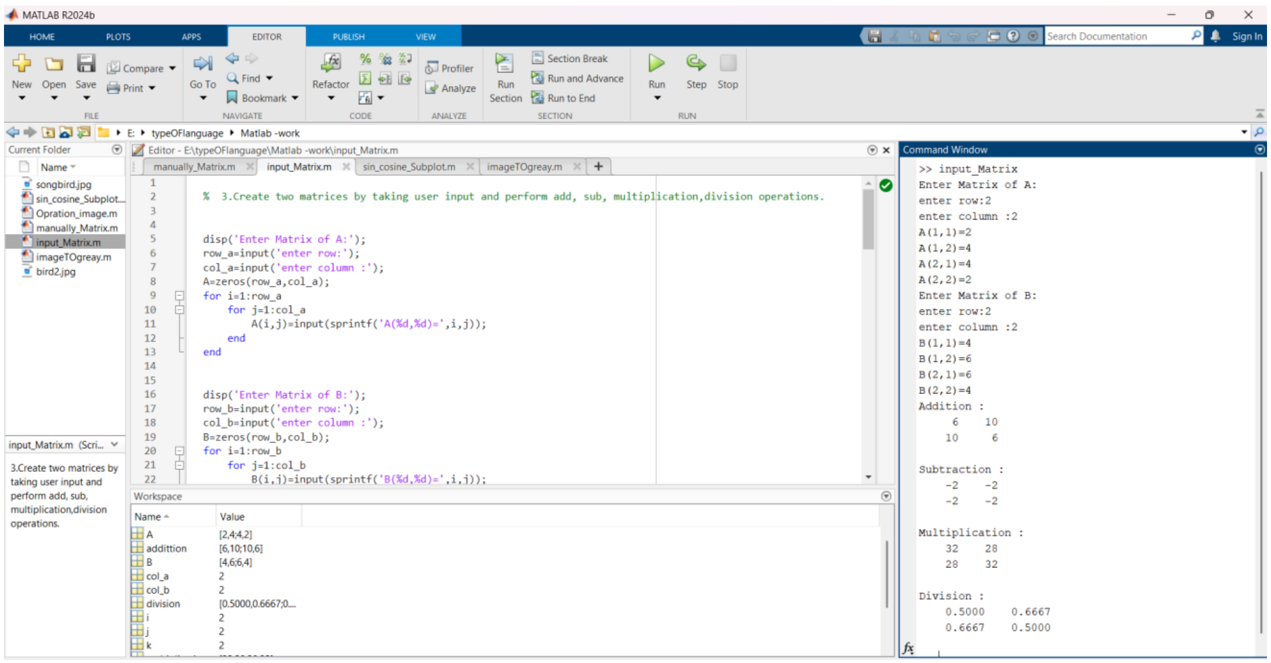
6. Open MATLAB and create a new script file.
7. Manually define two matrices A and B of the same size.
8. Perform addition, subtraction, element-wise multiplication, and division using MATLAB operators.
9. Display all matrices and results using disp() function.
10. Run the script and observe the output in the command window.

Code

```
disp('Enter Matrix of A:');
row_a=input('enter row:');
col_a=input('enter column :');
A=zeros(row_a,col_a);
for i=1:row_a
    for j=1:col_a
        A(i,j)=input(sprintf('A(%d,%d)=',i,j));
    end
end
```

```
disp('Enter Matrix of B:');
row_b=input('enter row:');
col_b=input('enter column :');
B=zeros(row_b,col_b);
for i=1:row_b
    for j=1:col_b
        B(i,j)=input(sprintf('B(%d,%d)=',i,j));
    end
end

addittion=zeros(row_a,col_a);
subtraction=zeros(row_a,col_a);
multiplication=zeros(row_a,col_a);
division=zeros(row_a,col_a);

for i = 1:row_a
    for j = 1:col_a
        addittion(i,j) = A(i,j)+B(i,j);
        subtraction(i,j) = A(i,j)-B(i,j);
        division(i,j) = A(i,j)/B(i,j);

    end
end
for i = 1:row_a%row of A
    for j = 1:col_b %col of B
        sum_val = 0;
        for k = 1:col_a %col of A
            sum_val = sum_val + A(i, k) * B(k, j);
        end
        multiplication(i, j) = sum_val;
    end
end
disp('Addition :');
disp(addittion);

disp('Subtraction :');
disp(subtraction);

disp('Multiplication :');
disp(multiplication);

disp('Division :')
disp(division);
```

Result/Output



```
Editor - E:\typeOFlanguage\Matlab -work\input_Matrix.m

%  3.Create two matrices by taking user input and perform add, sub, multiplication,division operations.


disp('Enter Matrix of A:');
row_a=input('enter row:');
col_a=input('enter column :');
A=zeros(row_a,col_a);
for i=1:row_a
    for j=1:col_a
        A(i,j)=input(sprintf('A(%d,%d)=',i,j));
    end
end


disp('Enter Matrix of B:');
row_b=input('enter row:');
col_b=input('enter column :');
B=zeros(row_b,col_b);
for i=1:row_b
    for j=1:col_b
        B(i,j)=input(sprintf('B(%d,%d)=',i,j));
```

```
>> input_Matrix
Enter Matrix of A:
enter row:2
enter column :2
A(1,1)=2
A(1,2)=4
A(2,1)=4
A(2,2)=2
Enter Matrix of B:
enter row:2
enter column :2
B(1,1)=4
B(1,2)=6
B(2,1)=6
B(2,2)=4
Addition :
     6    10
    10     6

Subtraction :
    -2    -2
    -2    -2

Multiplication :
    32    28
    28    32

Division :
    0.5000    0.6667
    0.6667    0.5000
```

Name	Value
A	[2,4;4,2]
addittion	[6,10;10,6]
B	[4,6;6,4]
col_a	2
col_b	2
division	[0.5000,0.6667;0....
i	2
j	2
k	2

3.Create two matrices by taking user input and perform add, sub, multiplication,division operations.

Result Discussion

This experiment was successful in demonstrating user interaction with matrix operations in MATLAB. The program dynamically accepts matrix size and elements, making it adaptable for different dimensions.

Lab Report: Create a sine wave and cosine wave; Plot sinusoidal wave in a single graph using Subplot.

Problem Definition

To generate and visualize the sine and cosine waves using MATLAB. The goal is to plot both waves in a single figure using subplots for better comparison and analysis of their behavior.

Objective

1. To understand how to create sine and cosine waves in MATLAB.
2. To use subplots for displaying multiple plots within a single figure window.
3. To compare the phase and amplitude differences between sine and cosine waves

Tools Used

5. Software: MATLAB R202x (any version with Image Processing Toolbox)
6. Hardware**:** Computer with MATLAB installed.

Theory

Sinusoidal waves are periodic waves that can be described mathematically by sine and cosine functions. They are fundamental in signal processing, communication systems, and physics.
The sine wave is given by: $y(t)=A\cdot\sin(2\pi ft)$
The cosine wave is given by: $y(t)=A\cdot\cos(2\pi ft+\phi)$

Where:

- A = Amplitude
- f = Frequency (Hz)
- t = Time (seconds)

Method/Procedure

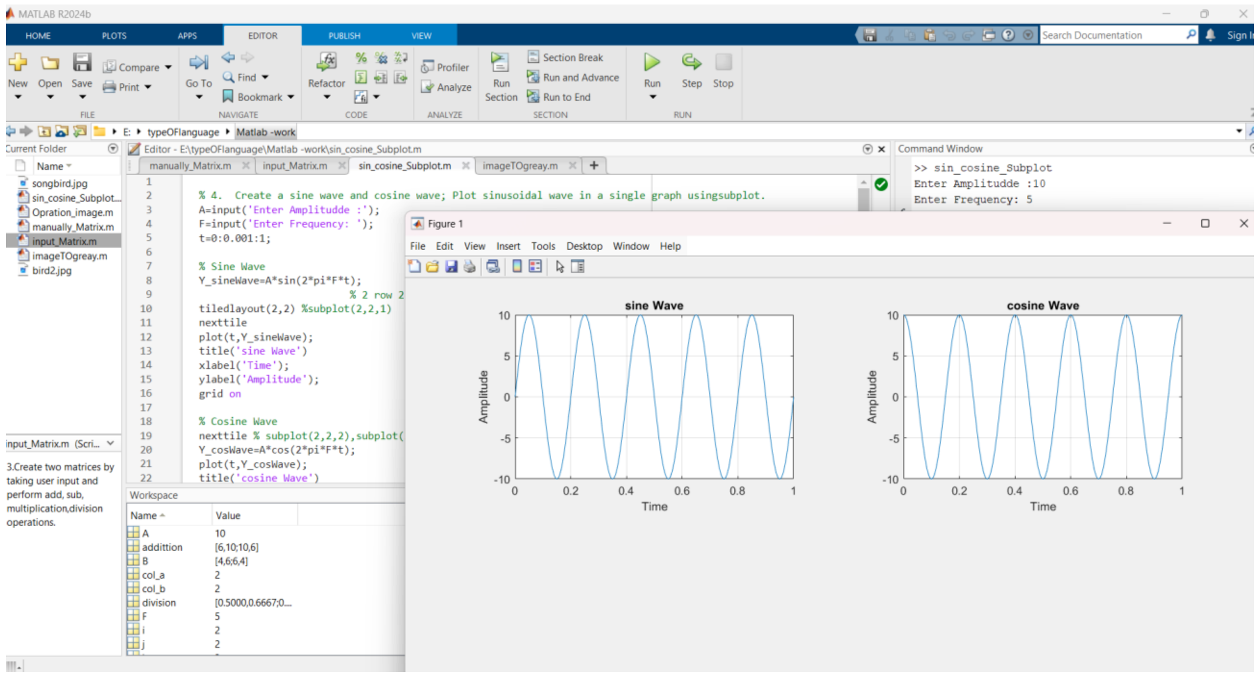
1. Open MATLAB.
2. Define the sine and cosine functions.
3. Use the subplot function to create a 2-row layout for separate plotting.
4. Plot the sine wave in the first subplot and the cosine wave in the second subplot.
5. Label the axes and add titles for clarity.

Code

```
A=input('Enter Amplitudde :');
F=input('Enter Frequency: ');
t=0:0.001:1;

% Sine Wave
Y_sineWave=A*sin(2*pi*F*t);
                         % 2 row 2 col and 1 meand 1 point
tiledlayout(2,2) %subplot(2,2,1)
nexttile
```

```
plot(t,Y_sineWave);
title('sine Wave')
xlabel('Time');
ylabel('Amplitude');
grid on

% Cosine Wave
nexttile % subplot(2,2,2),subplot(2,2,3),subplot(2,2,4)
Y_cosWave=A*cos(2*pi*F*t);
plot(t,Y_cosWave);
title('cosine Wave')
xlabel('Time');
ylabel('Amplitude');
grid on
```

Result/Output



3.Create two matrices by taking user input and perform add, sub, multiplication,division operations.

```
% 4.  Create a sine wave and cosine wave; Plot sinusoidal wave in a single graph usingsubplot.
A=input('Enter Amplitudde :');
F=input('Enter Frequency: ');
t=0:0.001:1;

% Sine Wave
Y_sineWave=A*sin(2*pi*F*t);
                                          % 2 row 2
tiledlayout(2,2) %subplot(2,2,1)
nexttile
plot(t,Y_sineWave);
title('sine Wave')
xlabel('Time');
ylabel('Amplitude');
grid on

% Cosine Wave
nexttile % subplot(2,2,2),subplot(
Y_cosWave=A*cos(2*pi*F*t);
plot(t,Y_cosWave);
title('cosine Wave')
```

```
>> sin_cosine_Subplot
Enter Amplitudde :10
Enter Frequency: 5
```

Name	Value
A	10
addittion	[6,10;10,6]
B	[4,6;6,4]
col_a	2
col_b	2
division	[0.5000,0.6667;0....
F	5
i	2
j	2

Result Discussion

The experiment successfully generated and visualized both sine and cosine waves. The sine wave and cosine wave are plotted correctly.The sine wave starts at zero, while the cosine wave starts at its maximum amplitude, showing a phase shift of 90 degrees.