Lab Report: 01

Title: Basic MATLAB

Course Title: *Data and Telecommunication Laboratory*

Course Code: *CSE-260*

2nd Year 2nd Semester Examination 2023

Date of Submission: 24/11/2024

Submitted to :-

Sarnali Basak

Associate Professor

Department of Computer Science and Engineering

Jahangirnagar University

Savar, Dhaka-1342

Class Roll	Name
371	Md. Ahad Siddiki

Department of Computer Science and Engineering

Jahangirnagar University

Savar, Dhaka, Bangladesh

Basic MATLAB

This lab report presents solutions to a series of MATLAB exercises, showcasing various applications such as matrix operations, numerical methods, data visualization, and signal processing. Each section provides an overview of the task, the MATLAB commands implemented, and the outcomes achieved.

Objective:

To explore fundamental MATLAB operations including:

- Matrix operations
- Linear equation solving
- Polynomial manipulation
- Signal processing
- Graphical visualization
- Graph theory algorithms

Experimental Equipment and Software

- MATLAB R2024a Software.
- Computer workstation.

Q1. Determine transpose, inverse, Eigen values, and Eigen vectors of a matrix.

$$A = \begin{bmatrix} 38 & -21 & 50 \\ 24 & 93 & 90 \\ -55 & 58 & 98 \end{bmatrix}$$

Code:

```
A = [38 -21 -50; -24 93 90; -55 58 98];
T = transpose(A);
Z = inv(A);
[X, Y] = eig(A);
A
T
Z
X
Y
```

Result:

```
Command Window
>> A = [38 -21 -50; -24 93 90; -55 58 98];
T = transpose(A);
Z = inv(A);
[X, Y] = eig(A);
A
T
Z
X
Y

A =

    38   -21   -50
   -24    93    90
   -55    58    98


T =

    38   -24   -55
   -21    93    58
   -50    90    98


Z =

    0.2377   -0.0514    0.1685
   -0.1586    0.0595   -0.1355
    0.2273   -0.0640    0.1850


X =

   -0.3141    0.6137   -0.4749
    0.6961   -0.4681   -0.8062
    0.6456    0.6358    0.3527


Y =

  187.2991         0         0
         0    2.2148         0
         0         0   39.4861

fx >>
```

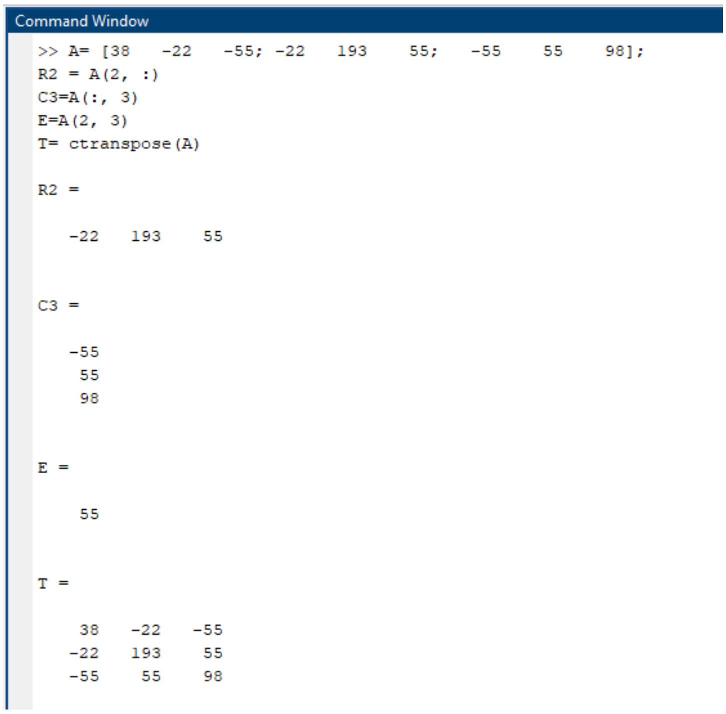
Q2. Show 2nd row, 3rd column, and conjugate transpose of matrix A.

$$A = \begin{bmatrix} 38 & -22 & -55 \\ -22 & 193 & 55 \\ -55 & 55 & 98 \end{bmatrix}$$

Codes:

```
A= [38   -22   -55; -22   193    55;   -55    55    98];
R2 = A(2, :)
C3=A(:, 3)
E=A(2, 3)
T= ctranspose(A)
```

Result:



Specified elements and conjugate transpose were computed.

Q3. Solve the linear equations:

x+2y+4z = 8

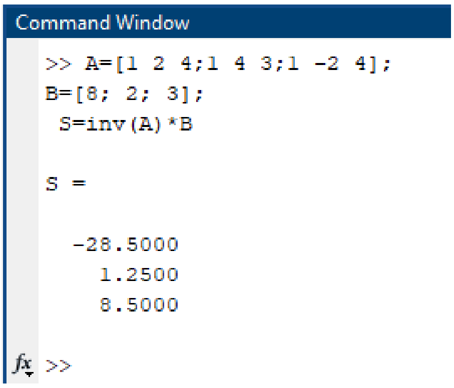
x+4y+3z = 2

x-2y+4z = 3

Codes:

```
A=[1 2 4;1 4 3;1 -2 4];
B=[8; 2; 3];
 S=inv(A)*B
```

Result:



Solutions for x, y, z were found.

Q4. Find the roots of the polynomial $x^3 + 6x^2 - x - 30 = 0$

Codes:
S=[1 6 -1 -30];
roots(S)

Result:

```
>> S=[1 6 -1 -30];
roots(S)

ans =

   -5.0000
   -3.0000
    2.0000

fx >> 
```

Computed the roots.

Q5. Determine the result of multiplying $(x + 5)(x + 3)(x - 2) = x^3 + 6x^2 - x - 30$

Codes:
s1 = [1 5];
s2 = [1 3];
s3 = [1 -2]
Result: conv(conv(s1, s2), s3)

Result:

```
>>  s1=[1 5];
>> s2=[1 3];
>> s3=[1 -2];
>>  conv(conv(s1,s2),s3)

ans =

     1     6    -1   -30

fx >> 
```

Polynomial multiplication was performed.

Q6. Find quotient and remainder of $x^2 - 1\left|\frac{(x^3+x-1)}{x^2-1}\right|x$

Codes:

s1 = [1 0 -1];
s2 = [1 0 1 -1];
[Q, R] = deconv(s2, s1)

Result:

```
>> s1=[1 0 -1];
 s2=[1 0 1 -1];
 [Q, R]=deconv(s2, s1)

Q =

     1     0


R =

     0     0     2    -1

>>
```

Calculated quotient and remainder.

Q7. Integrate $e^{-2x}sin(4x)$dx

Codes:

syms x
int(exp(-2*x)*sin(4*pi*x),x)

Result:

```
>> syms x
 int(exp(-2*x)*sin(4*pi*x),x)

ans =

-(exp(-2*x)*(2*sin(4*pi*x) + 4*pi*cos(4*pi*x)))/(16*pi^2 + 4)

>>
```

Integration result obtained.

Q8. Integrate $\int_0^{\pi} e^{-2x}sin(4x)\,dx$

Codes:

syms x;
int(exp(-2*x)*sin(4*x), x, pi,0)

Result:

```
>> syms x;
int(exp(-2*x)*sin(4*x), x, pi,0)

ans =

exp(-2*pi)/5 - 1/5

>>
```

Definite integral calculated.

Q9. Find the 5th derivative of sin(πx)

Codes:

syms x
diff(sin(pi*x),5)

Result:

```
>> syms x
diff(sin(pi*x),5)

ans =

pi^5*cos(pi*x)

>>
```

5th derivative result obtained.

Q10. Expand $(x + y)^7$

Codes:

syms x y;
expand((x + y)^7)

Result:

```
>> syms x y;
expand((x + y)^7)

ans =

x^7 + 7*x^6*y + 21*x^5*y^2 + 35*x^4*y^3 + 35*x^3*y^4 + 21*x^2*y^5 + 7*x*y^6 + y^7

>>
```

Binomial expansion calculated.

Q11. Compute Fourier transform of $sin^3(t$.

Codes:

syms t;
fourier(sin(t)^3)

Result:

```
>> syms t;
fourier(sin(t)^3)

ans =

- (pi*(dirac(w - 1) - dirac(w + 1))*3i)/4 + (pi*(dirac(w - 3) - dirac(w + 3))*1i)/4

>>
```

Fourier transform obtained.

Q12. Compute Laplace transform of sin^3(t).

Codes:

syms t s;
laplace(sin(t)^3)

Result:

```
>> syms t s;
laplace(sin(t)^3)

ans =

6/((s^2 + 1)*(s^2 + 9))

fx >>
```

Laplace transform obtained.

Q13. Determine convolution of $[2\ 3\ 7\ -2\ 4]$ and $[-5\ 0\ -3\ 1\ 6]$

Codes:

x = [2 3 7 -2 4];
y = [-5 0 -3 1 6];
f=conv(x, y);
f

Result:

```
>> x = [2 3 7 -2 4];
y = [-5 0 -3 1 6];
f=conv(x, y);
f

f =

   -10   -15   -41     3   -26    31    28    -8    24

fx >>
```

Convolution result computed.

Q14. Determine cross-correlation coefficient of $[2\ 3\ 7\ -2\ 4]$ and $[-5\ 0\ -3\ 1\ 6]$

Codes:

x=[2 3 7 -2 4] ;
y=[-5 0 -3 1 6];
f=corrcoef(x, y);
f

Result:

```
>> x=[2 3 7 -2 4] ;
y=[-5 0 -3 1 6];
f=corrcoef(x, y);
f

f =

    1.0000   -0.1126
   -0.1126    1.0000

fx >>
```

Correlation coefficient calculated.

Q15. Generate 5 random numbers in $[0, 1]$ and draw a bar graph.

Codes:

```
S = rand(1, 5)
bar(S)
```

Result:

```
>> S = rand(1, 5)
bar(S)

S =

    0.1419    0.4218    0.9157    0.7922    0.9595

>>
```

Random numbers plotted as a bar graph.

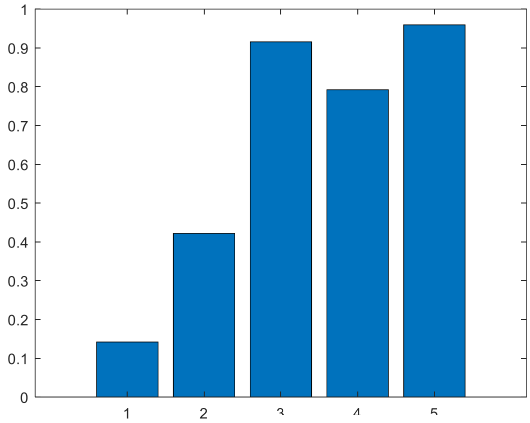


Fig : Bar Graph

Q16. Select 12 random points on (x, y) and draw a convex hull.

Codes:

```
x=rand(1,10)
y=rand(1,10)
K = convhull(x,y);
plot(x,y, 'd', 'markersize',10)
xlabel('Longitude')
ylabel('Latitude')
hold on
plot(x(K),y(K),'r')
```

Result:

```
>> x=rand(1,10)
y=rand(1,10)
K = convhull(x,y);
plot(x,y, 'd', 'markersize',10)
xlabel('Longitude')
ylabel('Latitude')
hold on
plot(x(K),y(K),'r')

x =

    0.6557    0.0357    0.8491    0.9340    0.6787    0.7577    0.7431    0.3922    0.6555    0.1712


y =

    0.7060    0.0318    0.2769    0.0462    0.0971    0.8235    0.6948    0.3171    0.9502    0.0344

>>
```

Convex hull plotted.

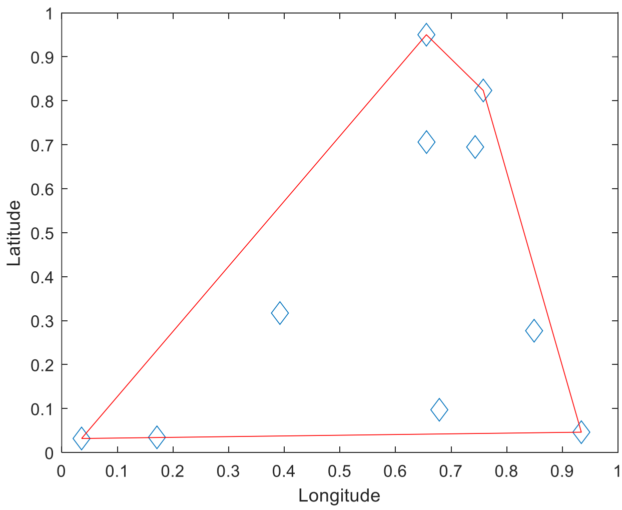


Fig : Convex Hull

Q17. Draw a directed graph and find shortest path between two nodes. Set of source node *s* and set of target node *t*.

Codes:

```
s = [1 1 2 3 3 4 4 6 6 7 8 7 5];
t = [2 3 4 4 5 5 6 1 8 1 3 2 8];
G = digraph(s, t);
plot(G)
P = shortestpath(G,7,8)
```

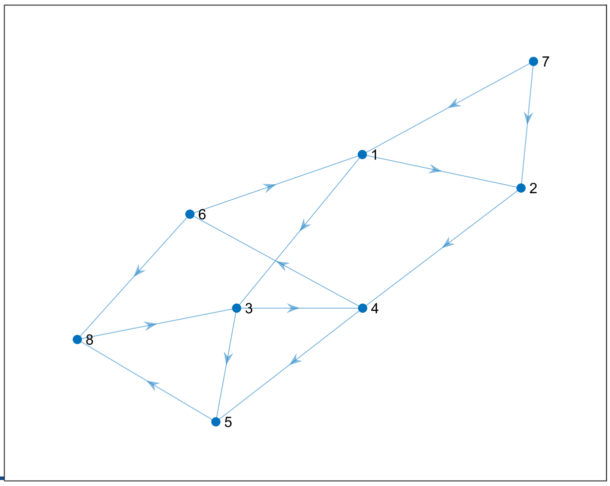
Result:

```
>> s = [1 1 2 3 3 4 4 6 6 7 8 7 5];
t = [2 3 4 4 5 5 6 1 8 1 3 2 8];
G = digraph(s, t);
plot(G)
P = shortestpath(G,7,8)

P =

     7     1     3     5     8

fx >>
```



Directed graph plotted; shortest path found.

Fig : Directed Graph

Q18. Use weighted graph to find the shortest path minimizing weights.

Codes:

```
s = [1 1 1 2 2 6 6 7 7 3 3 9 9 4 4 11 11 8];
t = [2 3 4 5 6 7 8 5 8 9 10 5 10 11 12 10 12 12];
weights = [10 10 10 10 10 1 1 1 1 1 1 1 1 1 1 1 1 1];
G = graph(s, t, weights);
plot(G,'EdgeLabel', G.Edges.Weight)
[P,d] = shortestpath(G,3,8)
```

Result:

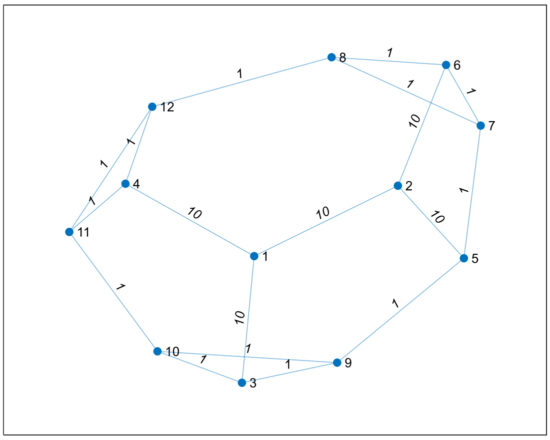
```
>> s = [1 1 1 2 2 6 6 7 7 3 3 9 9 4 4 11 11 8];
t = [2 3 4 5 6 7 8 5 8 9 10 5 10 11 12 10 12 12];
weights = [10 10 10 10 10 1 1 1 1 1 1 1 1 1 1 1 1 1];
G = graph(s, t, weights);
plot(G,'EdgeLabel', G.Edges.Weight)
[P,d] = shortestpath(G,3,8)

P =

     3     9     5     7     8


d =

     4
```



Minimum-weight path calcu lated.

Fig : Weighted Graph

Q19. Determine maximum flow from node 3 to 12.

Codes:

s = [1 1 1 2 2 6 6 7 7 3 3 9 9 4 4 11 11 8];
t = [2 3 4 5 6 7 8 5 8 9 10 5 10 11 12 10 12 12];
weights = [10 10 10 10 10 1 1 1 1 1 1 1 1 1 1 1 1 1];
G = graph(s, t, weights);
plot(G,'EdgeLabel', G.Edges.Weight)
mf = maxflow(G,3,12)

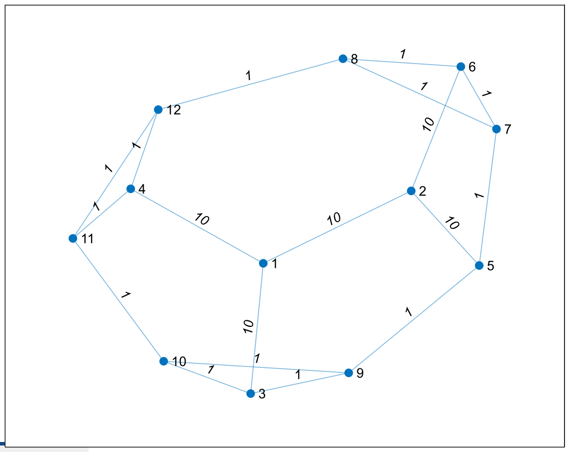
Result:

```
>> s = [1 1 1 2 2 6 6 7 7 3 3 9 9 4 4 11 11 8];
t = [2 3 4 5 6 7 8 5 8 9 10 5 10 11 12 10 12 12];
weights = [10 10 10 10 10 1 1 1 1 1 1 1 1 1 1 1 1 1];
G = graph(s, t, weights);
plot(G,'EdgeLabel', G.Edges.Weight)
mf = maxflow(G,3,12)

mf =

     3

>>
```



Maximum flow calculated.

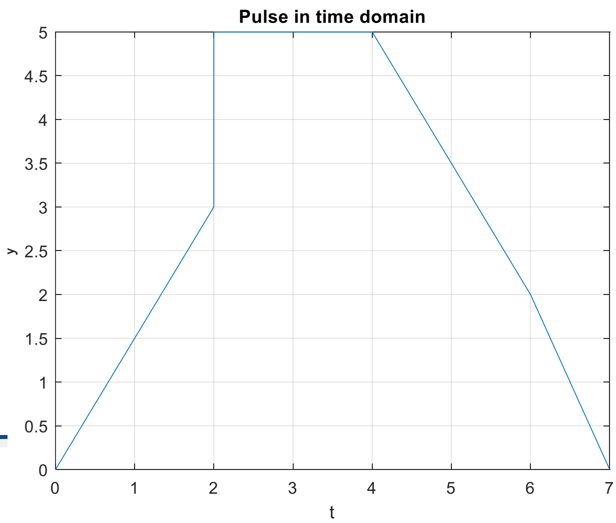
Q20. Plot pulse signal in time domain.

Codes:

t = [0 2 2 4 6 7];
y =[0 3 5 5 2 0];
plot(t, y)
xlabel('t')
ylabel('y')
title('Pulse in time domain')
grid on

Result:

```
>> t = [0   2   2   4   6   7];
y =[0   3   5   5   2   0];
plot(t, y)
xlabel('t')
ylabel('y')
title('Pulse in time domain')
grid on
>>
```



Pulse signal plotted.

Fig : Pulse

Q21. Plot sinusoidal wave $y(t) = 2sin(3\pi t + \pi/3)$

Codes:

```
t=0: 0.02:pi;
y=2*sin(3*pi*t+pi/3);
plot(t, y)
xlabel('Time')
ylabel('Amplitude')
title('Sinusodial wave')
grid on
```

Result:

```
>> t=0: 0.02:pi;
y=2*sin(3*pi*t+pi/3);
plot(t, y)
xlabel('Time')
ylabel('Amplitude')
title('Sinusodial wave')
grid on
fx >>
```

Sinusoidal wave plotted.

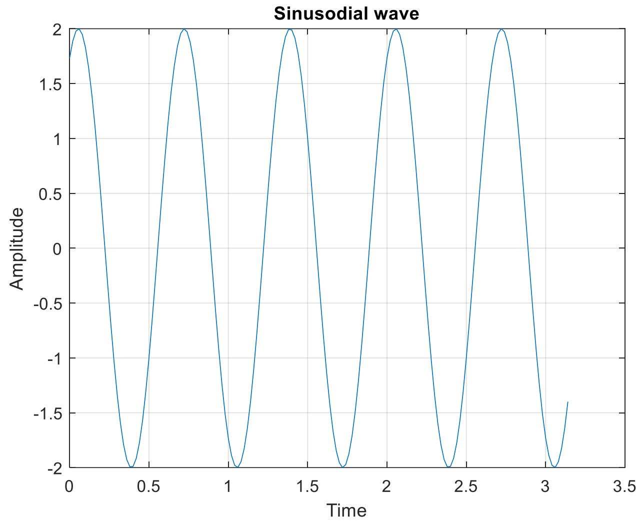


Fig : Sinusoidal Wave

Q22. Plot damped sinusoidal wave $y(t) = e^{-0.6t}sin(3\pi t + \pi/3)$

Codes:

```
t=0: 0.02:pi;
y=exp(-0.6*t).*sin(3*pi*t+pi/3);
plot(t, y)
xlabel('t')
ylabel('y')
title('Damped Sinusoidal wave')
grid on
```

Result:

```
>> t=0: 0.02:pi;
y=exp(-0.6*t).*sin(3*pi*t+pi/3);
plot(t, y)
xlabel('t')
ylabel('y')
title('Damped Sinusoidal wave')
grid on
fx >>
```

Damped wave plotted.

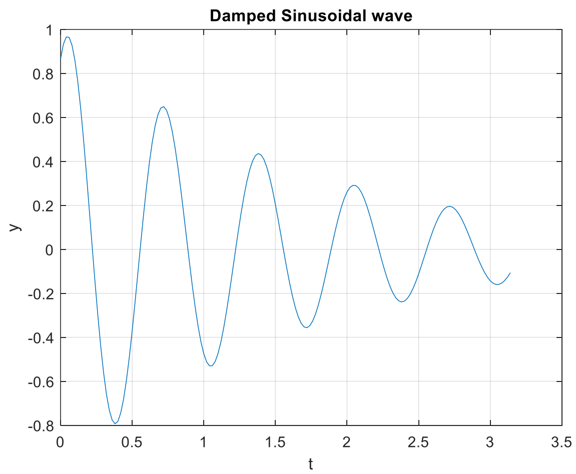


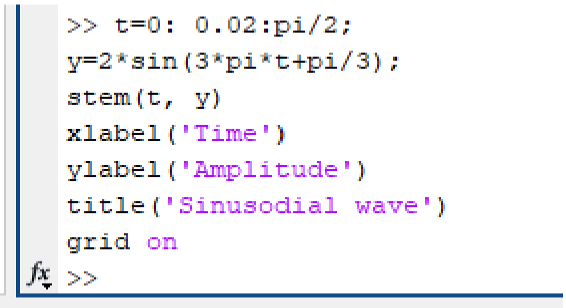
Fig : Damped Sinusoidal Wave

Q23. Create discrete plot of sinusoidal wave $y(t) = 2sin(3\pi t + \pi/3)$

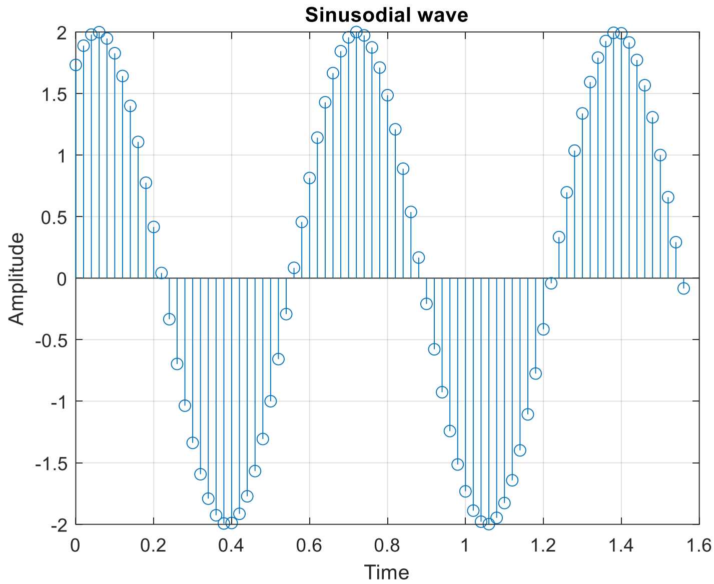
Codes:

```
t=0: 0.02:pi/2;
y=2*sin(3*pi*t+pi/3);
stem(t, y)
xlabel('Time')
ylabel('Amplitude')
title('Sinusodial wave')
grid on
```

Result:



```
>> t=0: 0.02:pi/2;
y=2*sin(3*pi*t+pi/3);
stem(t, y)
xlabel('Time')
ylabel('Amplitude')
title('Sinusodial wave')
grid on
>>
```



Discrete sinusoidal wave plotted.

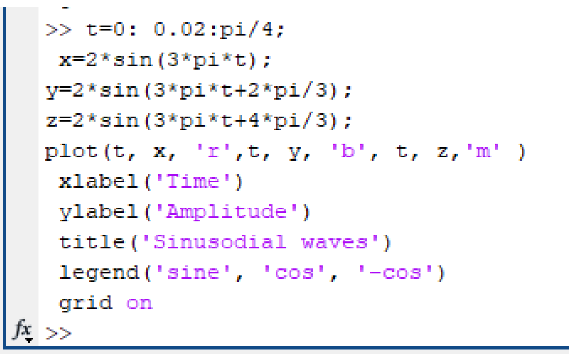
Fig : Sinusoidal Wave

Q24. Plot three sinusoidal waves, $x(t) = 2sin(3\pi t)$, $y(t) = 2sin(3\pi t + 2\pi/3)$ and $y(t) = 2sin(3\pi t + 4\pi/3)$ with legends.

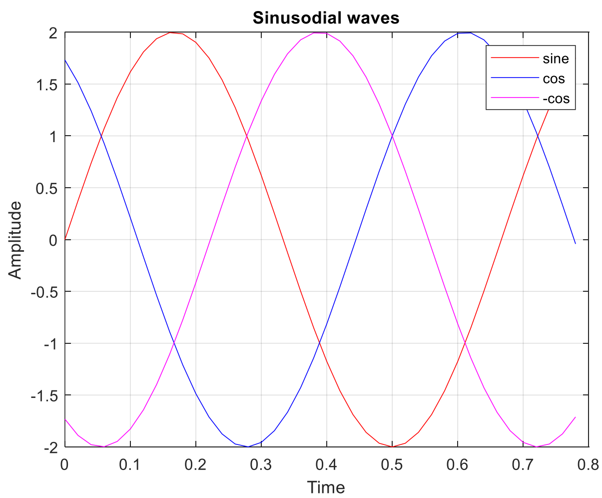
Codes:

```
t=0: 0.02:pi/4;
 x=2*sin(3*pi*t);
y=2*sin(3*pi*t+2*pi/3);
z=2*sin(3*pi*t+4*pi/3);
plot(t, x, 'r',t, y, 'b', t, z,'m' )
 xlabel('Time')
 ylabel('Amplitude')
 title('Sinusodial waves')
 legend('sine', 'cos', '-cos')
 grid on
```

Result:



```
>> t=0: 0.02:pi/4;
 x=2*sin(3*pi*t);
y=2*sin(3*pi*t+2*pi/3);
z=2*sin(3*pi*t+4*pi/3);
plot(t, x, 'r',t, y, 'b', t, z,'m' )
 xlabel('Time')
 ylabel('Amplitude')
 title('Sinusodial waves')
 legend('sine', 'cos', '-cos')
 grid on
>>
```



Three sinusoidal waves plotted.

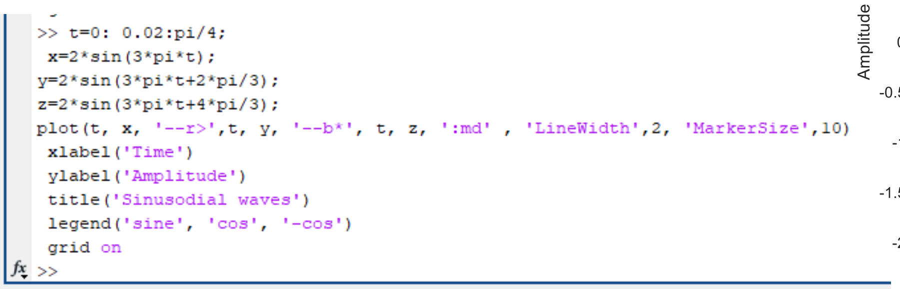
Fig : Sinusoidal Wave

Q25. Plot sinusoidal waves $x(t) = 2sin(3\pi t)$, $y(t) = 2sin(3\pi t + 2\pi/3)$ *and* $y(t) = 2sin(3\pi t + 4\pi/3)$ with custom line styles and markers.

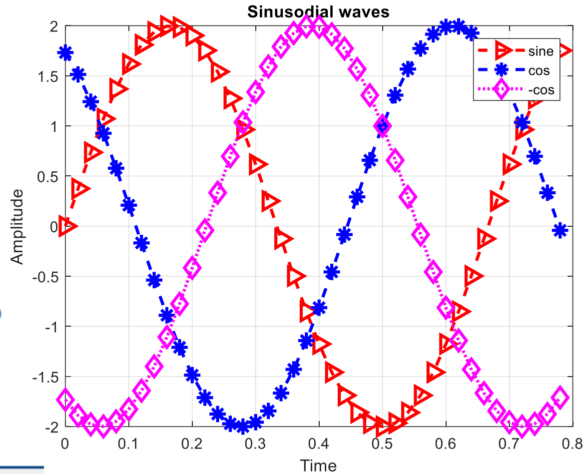
Codes:

```
t=0: 0.02:pi/4;
 x=2*sin(3*pi*t);
y=2*sin(3*pi*t+2*pi/3);
z=2*sin(3*pi*t+4*pi/3);
plot(t, x, '--r>',t, y, '--b*', t, z, ':md' , 'LineWidth',2, 'MarkerSize',10)
 xlabel('Time')
 ylabel('Amplitude')
 title('Sinusodial waves')
 legend('sine', 'cos', '-cos')
 grid on
```

Result:



```
>> t=0: 0.02:pi/4;
 x=2*sin(3*pi*t);
y=2*sin(3*pi*t+2*pi/3);
z=2*sin(3*pi*t+4*pi/3);
plot(t, x, '--r>',t, y, '--b*', t, z, ':md' , 'LineWidth',2, 'MarkerSize',10)
 xlabel('Time')
 ylabel('Amplitude')
 title('Sinusodial waves')
 legend('sine', 'cos', '-cos')
 grid on
>>
```



Customized sinusoidal waves plotted.

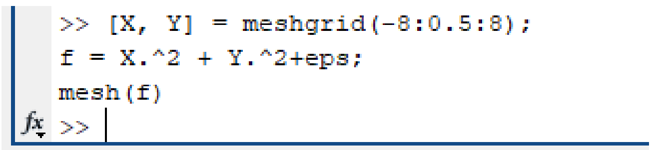
Fig : Sinusoidal Wave

Q26. Create mesh plot for $f(x, y) = x^2 + y^2$

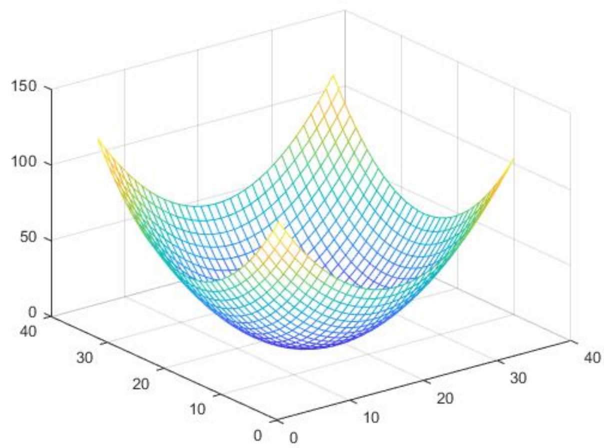
Codes:

```
[X, Y] = meshgrid(-8:0.5:8);
f = X.^2 + Y.^2+eps;
mesh(f)
```

Result:



```
>> [X, Y] = meshgrid(-8:0.5:8);
f = X.^2 + Y.^2+eps;
mesh(f)
>>
```



3D mesh plot created.

Fig : Mesh Plot

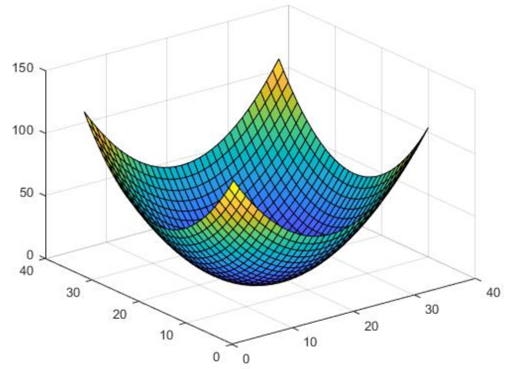
Q27. Create surface plot for $f(x, y) = x^2 + y^2$

Codes:

[X, Y] = meshgrid(-8:0.5:8);
f = X.^2 + Y.^2+eps;
surf(f)

Result:

```
>> [X, Y] = meshgrid(-8:0.5:8);
f = X.^2 + Y.^2+eps;
surf(f)
fx >>
```



3D surface plot created.

Fig : Surface Plot

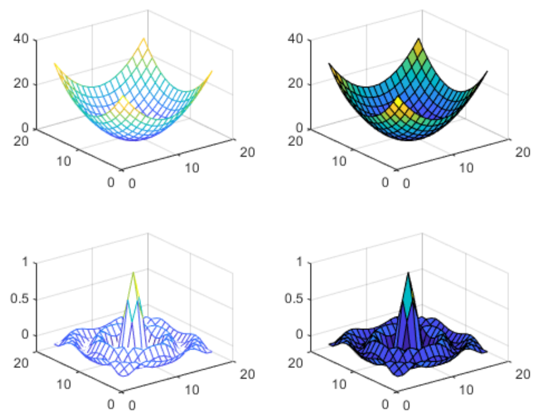
Q28. Demonstrate subplots with mesh and surface plots.

Codes:

[X, Y]=meshgrid(-4:0.5:4);
f1=X.^2+Y.^2+eps;
f2=sinc(sqrt(X.^2+Y.^2))+eps;
subplot(2,2,1)
mesh(f1)
subplot(2,2,2)
surf(f1)
subplot(2,2,3)
mesh(f2)
subplot(2,2,4)
surf(f2)

Result:

```
>> [X, Y]=meshgrid(-4:0.5:4);
 f1=X.^2+Y.^2+eps;
 f2=sinc(sqrt(X.^2+Y.^2))+eps;
 subplot(2,2,1)
 mesh(f1)
 subplot(2,2,2)
 surf(f1)
 subplot(2,2,3)
 mesh(f2)
 subplot(2,2,4)
 surf(f2)
fx >>
```



Subplots for mesh and surface plots created.

Fig : Sub Plots

Conclusion

This MATLAB lab exercise covered a diverse range of computational and analytical tasks, demonstrating the software's versatility in solving mathematical problems, visualizing data, and simulating systems. The topics ranged from basic matrix operations, solving equations, and polynomial manipulation to advanced techniques such as Fourier and Laplace transforms, signal processing, and graph theory. Each task emphasized the importance of MATLAB as a powerful tool for engineers and scientists.

Through this lab, the following skills were reinforced:

1. Efficient handling of matrices for operations like transposition, inversion, eigenvalues, and eigenvectors.
2. Solving linear systems and analyzing polynomial roots and operations.
3. Visualizing functions through 2D and 3D plots, including sinusoidal waves, surface plots, and subplots.
4. Applying symbolic computations for differentiation, integration, and transformations.
5. Working with graphs and networks to determine paths and flows.

The practical application of these concepts enhances problem-solving abilities and provides insights into real-world systems, making MATLAB an indispensable resource in computational fields.