**Lab Report:** 01

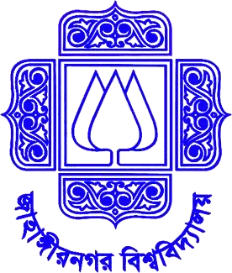
**Title: Basic MATLAB**

***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.

### Code:

A = [38 -21 -50; -24 93 90; -55 58 98];

T = transpose(A);

Z = inv(A);

[X, Y] = eig(A);

A

T

**Result:**

Z

X

Y

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

### 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.

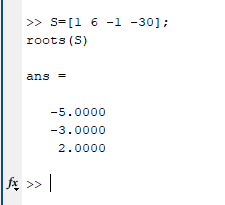
## Q4. Find the roots of the polynomial

### Codes:

### S=[1 6 -1 -30];

### roots(S)

### Result:



Computed the roots.

## Q5. Determine the result of multiplying

### Codes:

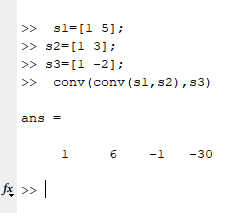
s1 = [1 5];

s2 = [1 3];

s3 = [1 -2]

Result: conv(conv(s1, s2), s3)

### Result:



Polynomial multiplication was performed.

## Q6. Find quotient and remainder of

### Codes:

s1 = [1 0 -1];

s2 = [1 0 1 -1];

[Q, R] = deconv(s2, s1)

### Result:

Calculated quotient and remainder.

## Q7. Integrate )dx

### Codes:

syms x

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

### Result:

Integration result obtained.

## Q8. Integrate

### Codes:

syms x;

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

### Result:

Definite integral calculated.

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

### Codes:

syms x

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

### Result:

5th derivative result obtained.

## Q10. Expand

### Codes:

syms x y;

expand((x + y)^7)

### Result:

Binomial expansion calculated.

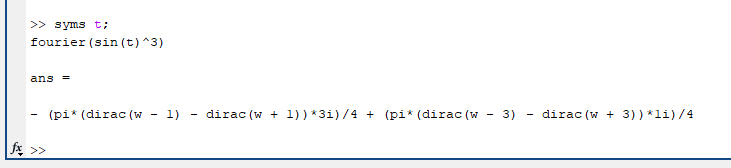
## Q11. Compute Fourier transform of .

### Codes:

syms t;

fourier(sin(t)^3)

### Result:



Fourier transform obtained.

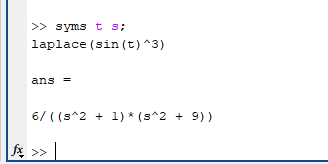
## Q12. Compute Laplace transform of sin^3(t).

### Codes:

syms t s;

laplace(sin(t)^3)

### Result:



Laplace transform obtained.

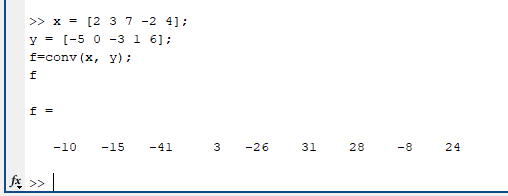
## Q13. Determine convolution of and

### Codes:

x = [2 3 7 -2 4];

y = [-5 0 -3 1 6];

f=conv(x, y);

f

### Result:

Convolution result computed.

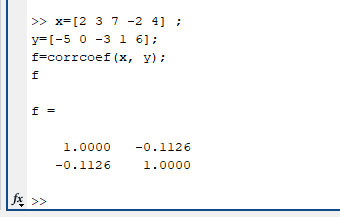
## Q14. Determine cross-correlation coefficient of and

### Codes:

x=[2 3 7 -2 4] ;

y=[-5 0 -3 1 6];

f=corrcoef(x, y);

f

### Result:

Correlation coefficient calculated.

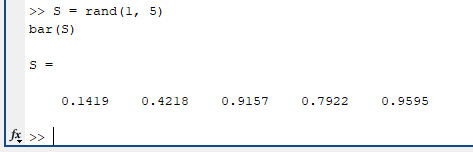
## Q15. Generate 5 random numbers in and draw a bar graph.

### Codes:

S = rand(1, 5)

bar(S)

### Result:



Random numbers plotted as a bar graph.

Fig : Bar Graph

## Q16. Select 12 random points on 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:



Convex hull plotted.

Fig : Convex Hull

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

### 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:

Fig : Directed Graph

Directed graph plotted; shortest path found.

## 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:

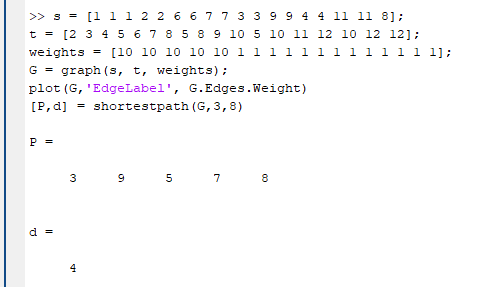


Fig : Weighted Graph

Minimum-weight path calcu lated.

## 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];

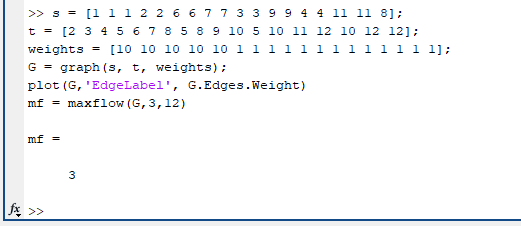
### weights = [10 10 10 10 10 1 1 1 1 1 1 1 1 1 1 1 1 1];

### G = graph(s, t, weights);

### plot(G,'EdgeLabel', G.Edges.Weight)

### mf = maxflow(G,3,12)

### Result:



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)

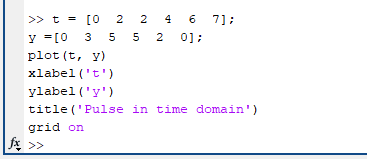
### xlabel('t')

### ylabel('y')

### title('Pulse in time domain')

### grid on

### Result:



Pulse signal plotted.

Fig : Pulse

## Q21. Plot sinusoidal wave

### Codes:

### t=0: 0.02:pi;

### y=2\*sin(3\*pi\*t+pi/3);

### plot(t, y)

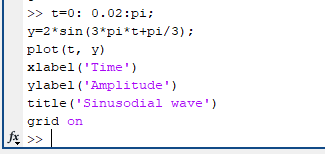
### xlabel('Time')

### ylabel('Amplitude')

### title('Sinusodial wave')

### grid on

### Result:



Sinusoidal wave plotted.

Fig : Sinusoidal Wave

## Q22. Plot damped sinusoidal wave

### Codes:

### t=0: 0.02:pi;

### y=exp(-0.6\*t).\*sin(3\*pi\*t+pi/3);

### plot(t, y)

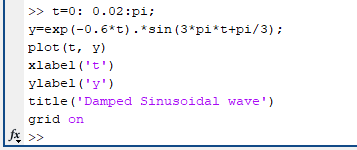
### xlabel('t')

### ylabel('y')

### title('Damped Sinusoidal wave')

### grid on

### Result:



Damped wave plotted.

Fig : Damped Sinusoidal Wave

## Q23. Create discrete plot of sinusoidal wave

### Codes:

### t=0: 0.02:pi/2;

### y=2\*sin(3\*pi\*t+pi/3);

### stem(t, y)

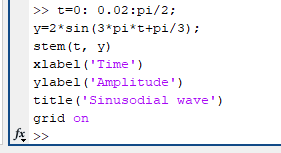
### xlabel('Time')

### ylabel('Amplitude')

### title('Sinusodial wave')

### grid on

### Result:



Discrete sinusoidal wave plotted.

Fig : Sinusoidal Wave

## Q24. Plot three sinusoidal waves, 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:

Three sinusoidal waves plotted.

Fig : Sinusoidal Wave

## Q25. Plot sinusoidal waves 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:

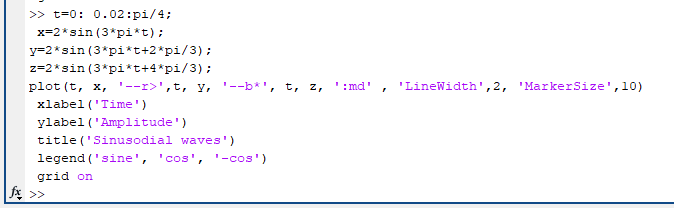


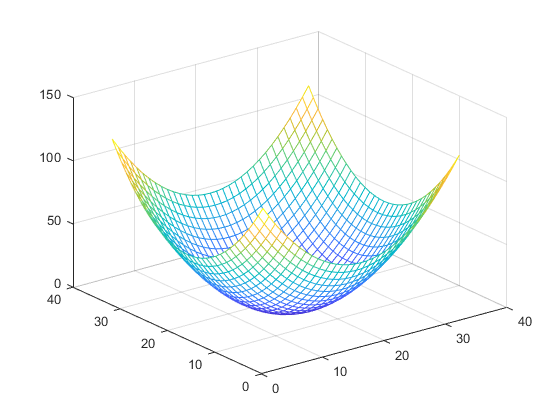
Fig : Sinusoidal Wave

Customized sinusoidal waves plotted.

## Q26. Create mesh plot for

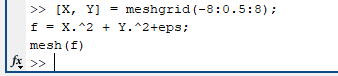
### Codes:

[X, Y] = meshgrid(-8:0.5:8);

f = X.^2 + Y.^2+eps;

mesh(f)

### Result:



3D mesh plot created.

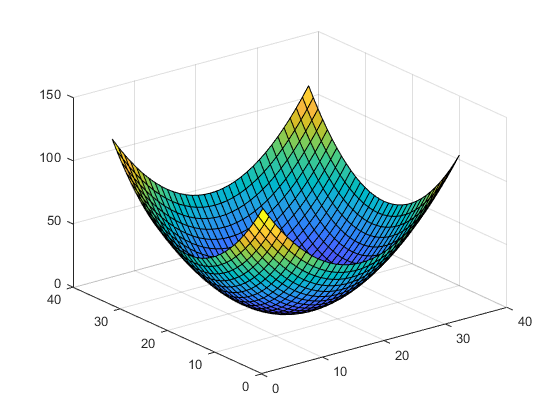
Fig : Mesh Plot

## Q27. Create surface plot for

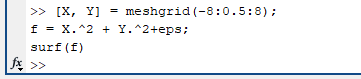
### Codes:

[X, Y] = meshgrid(-8:0.5:8);

f = X.^2 + Y.^2+eps;

surf(f)

### Result:



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:

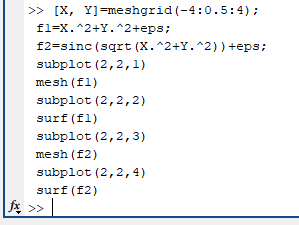


Fig : Sub Plots

Subplots for mesh and surface plots created.

**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.