**INTRODUCTION TO MATRICES**

**LAB # 02**



**Spring 2023**

**CSE301L Signals & Systems Lab**

Submitted by: **Ali Asghar**

Registration No. : **21PWCSE2059**

Class Section: **C**

“On my honor, as student of University of Engineering and Technology, I have neither given nor received unauthorized assistance on this academic work.”

Submitted to:

**Engr. Sumayyea Salahuddin**

Date:

**March 3, 2023**

**Department of Computer Systems Engineering**

**University of Engineering and Technology, Peshawar**

## Lab Objective(S):

Objectives of this Lab are:

* Built in Matrix Functions
* Indexing Matrices
* Sub Matrices
* Matrix element level operations
* Round Floating-Point numbers to Integers

Built-In Matrix Functions: MATLAB offers a variety of built-in capabilities for interacting with matrices. Common matrix operations including constructing matrices, transposing matrices, determining the determinant, and resolving linear systems of equations are made simple by these built-in functions. In this note, we'll look at a few of MATLAB's most popular built-in matrix functions.

Creating a Matrix: A number of functions, including zeros(), ones(), rand(), and eye, are available in MATLAB for constructing matrices (). When using the zeros(), ones(), rand(), and eye() functions, a matrix is created that is entirely composed of zeros, while an identity matrix is produced using the zeros() and ones() functions.

Matrix transposition: In MATLAB, you may use the transpose() function to determine a matrix's transposition. By switching the rows and columns in a matrix, you can create the transpose of that matrix.

Evaluating the Determinant: The MATLAB det() function can be used to determine the determinant of a matrix. It is possible to tell whether a matrix can be inverted or not using a scalar value called the determinant.

Solving Linear Systems of Equations: MATLAB offers the backslash operator (). Systems of equations of the form Ax = b, where A is a matrix, x is a vector of unknowns, and b is a vector of constants, can be solved using the backslash operator.

Matrix multiplication: In MATLAB, the \* operator can be used to multiply two matrices. The number of rows in the first matrix and the number of columns in the second matrix together determine the size of the final matrix.

Eigenvalues and Eigenvectors: The eig() function in MATLAB can be used to calculate the eigenvalues and eigenvectors of a matrix. In numerous fields, including principal component analysis, image processing, and control theory, the eigenvalues and eigenvectors play a crucial role.

## Task # 01:

Write a program to generate a new matrix B from the matrix A given below such that each

column in the new matrix except the first one is the result of subtraction of that column from

the previous one i.e., 2nd new column is the result of subtraction of 2nd column and 1st column

and so on. Copy the first column as it is in the new matrix.

13 6 9

A = 1 4 8

2 8 17

### Code:

### Output:

## Task # 02:

Generate two 2500 sampled random discrete time signals (1 dimensional) using rand() function

i.e. rand(1, 2500). Write a program to add the two such random signals together using simple

vector addition.

### Code:

### Output:

## Task # 03:

Using colon notation, generate the following sequence:

-65.25, -57.75, -50.25. . . . . . . . . . ., 54.75,62.25, 69.75

### Code:

### Output:

## Task # 04:

Given the matrices:

A=[-12,34,61,-9;65,78,90,12; 14,78,45,12; 60,25,3,8]

B=[34,67,8,9; 12,-91,12,9; 89,-8,0,2; 16,9,23,67]

Find the following:

1) Array addition; store the result in matrix C

2) Array subtraction; store the result in matrix D

3) Array multiplication using .\* operator; store the result in matrix E

4) Array division using ./ operator; store the result in matrix F

5) Array exponentiation using .^ operator; store the result in matrix G

6) Take sin of A and store the result in H, Take sqrt of B and store the result in I. Find H\*I

and store the result in J.

### Code:

### Output:

**Code Explanation:**

## Task # 01:

## Task # 02:

## Task # 03:

## Task # 04:

## Task # 05:

## Task # 06:

## Task # 07:

## Conclusion: