**B.TECH. (2020-24)**

**Artificial Intelligence**

Lab File

**Experiment 1**

on

**Basic Simulation Lab**

**[ES204]**

**Logo

Description automatically generated**

Submitted To

**Ms Shikha Bathla**

Submitted By

**Hitesh**

**A023119820027**

**4CSE11 (AI)**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AMITY SCHOOL OF ENGINEERING AND TECHNOLOGY

AMITY UNIVERSITY UTTAR PRADESH

NOIDA (U.P)

**EXPERIMENT- 1**

**AIM**

(A) Creating a One-Dimensional Array (Row / Column Vector)

(B) Creating a Two-Dimensional Array (Matrix of given size)

(C) Performing Arithmetic Operations - Addition, Subtraction, Multiplication and Exponentiation

(D) Performing Matrix operations - Inverse, Transpose and Rank.

**SOFTWARE USED**

Octave Online - <https://octave-online.net/>

**THEORY**

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation.

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning.

MATLAB is an abbreviation for "**mat**rix **lab**oratory." While other programming languages mostly work with numbers one at a time, MATLAB is designed to operate primarily on whole matrices and arrays.

All MATLAB variables are multidimensional arrays, no matter what type of data. A matrix is a two-dimensional array often used for linear algebra.

(I) Creating a One-Dimensional Array (Row / Column Vector)

To create a 1D Array, you need to assign a list of numbers separated with comma (,) or space ( ), to any variable say (A) for a row vector or separated with semi-colon (;) for a column vector.

For instance, to create an array with four elements in a single row, separate the elements with either a comma (,) or a space.

|  |
| --- |
| A = [1 2 3 4] |

This type of array is a row vector.

To create a column vector with 3 elements, separate the (row) elements with semicolons.

|  |
| --- |
| B = [1; 2; 7] |

(II) Creating a Two-Dimensional Array (Matrix of given size)

To create a 2D array or matrix of given size (say 3x3) that has multiple rows, separate the rows with semicolons.

|  |
| --- |
| C = [1 3 5; 2 4 6; 7 8 10] |

(III) Performing Arithmetic Operations - Addition, Subtraction, Multiplication and Exponentiation

MATLAB allows you to process all the values in a matrix using a single arithmetic operator or function.

Consider that A and B are two 3x3 matrices,

1. **Addition**: Using operator (+) as such , adds arrays A and B by adding corresponding elements. If one input is a string array, then plus appends the corresponding elements as strings. The sizes of A and B must be the same or be compatible.

Alternatively, using MATLAB command plus() as such to execute A + B.

1. **Subtraction**: Using operator (-) as such , subtracts array B from array A by subtracting corresponding elements. The sizes of A and B must be the same or be compatible.

Alternatively, using MATLAB command minus() as such to execute A – B.

1. **Multiplication**: Using operator (\*) as such for the matrix product of A and B. If A is an m-by-p and B is a p-by-n matrix, then C is an m-by-n matrix defined by

A picture containing text, watch

Description automatically generated

Alternatively, using MATLAB command mtimes() as such to execute A\*B.

1. **Hadamard Product (Element-wise Multiplication):** Using operator (.\*) as such , multiplies arrays A and B by multiplying corresponding elements. The sizes of A and B must be the same or be compatible.

Alternatively, using MATLAB command times() as such to execute A.\*B.

1. **Division**: Using operator (./) as such , divides each element of A by the corresponding element of B. The sizes of A and B must be the same or be compatible.

Alternatively, using MATLAB commands rdivide() for right divide and ldivide() for left divide as such and to divide A by B.

1. **Element-by-element Exponentiation**: Using MATLAB command exp() as such , returns the exponential ex for each element in array X.
2. **Matrix Exponentiation**: Using MATLAB command expm() as such , computes the matrix exponential of X. Although it is not computed this way, if X has a full set of eigenvectors V with corresponding eigenvalues D, then [V,D] = eig(X) and expm(X) = V\*diag(exp(diag(D)))/V.

It returns the result of matrix exponential eAt, i.e,

A picture containing text, watch, clock

Description automatically generated

(IV) Performing Matrix operations - Inverse, Transpose and Rank.

1. **Inverse**: Using operator {^(-1)} as such , computes the inverse of square matrix X.

Alternatively, using MATLAB command inv() as such to get the inverse of square matrix X.

1. **Transpose**: Using operator (.') as such , returns the nonconjugate transpose of A, that is, interchanges the row and column index for each element.

Alternatively, using MATLAB command transpose() as such to compute the nonconjugate transpose of A.

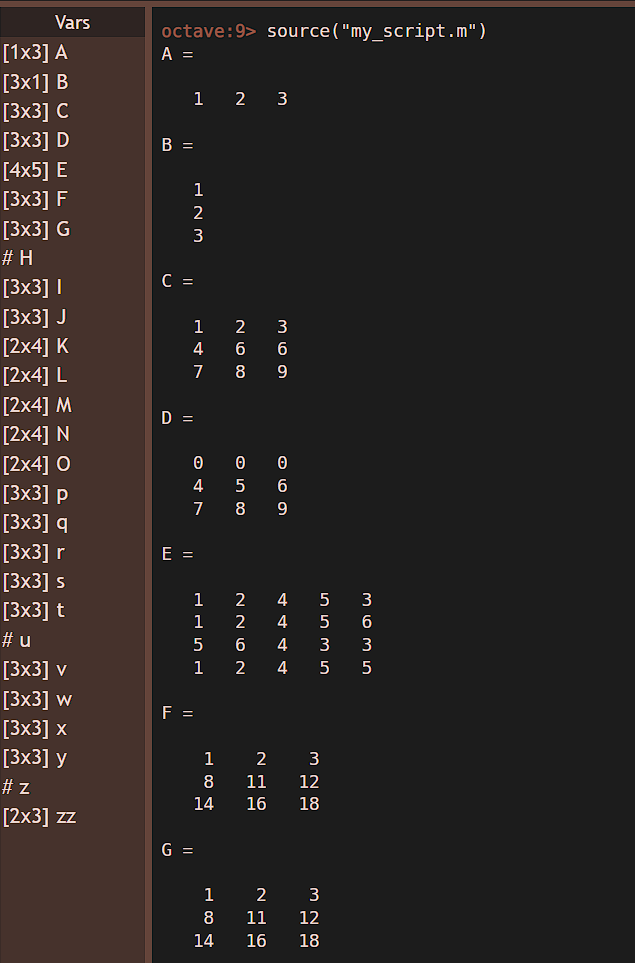
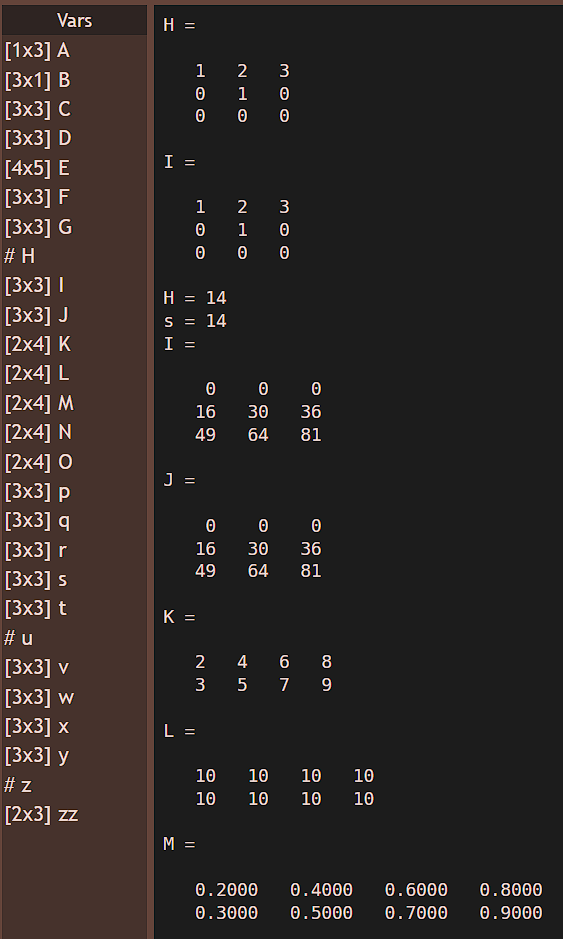
1. **Rank**: Using MATLAB command rank() as such , returns the rank of matrix A.

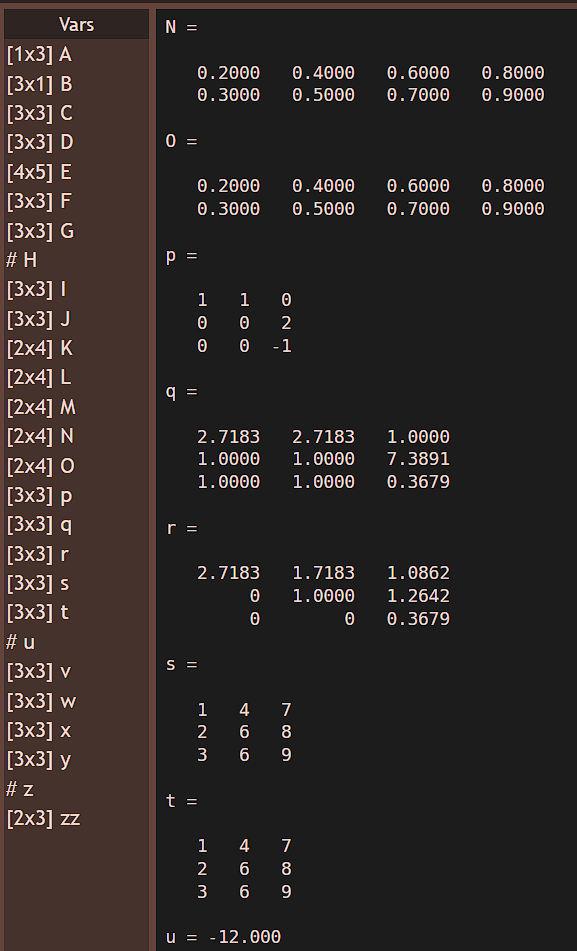
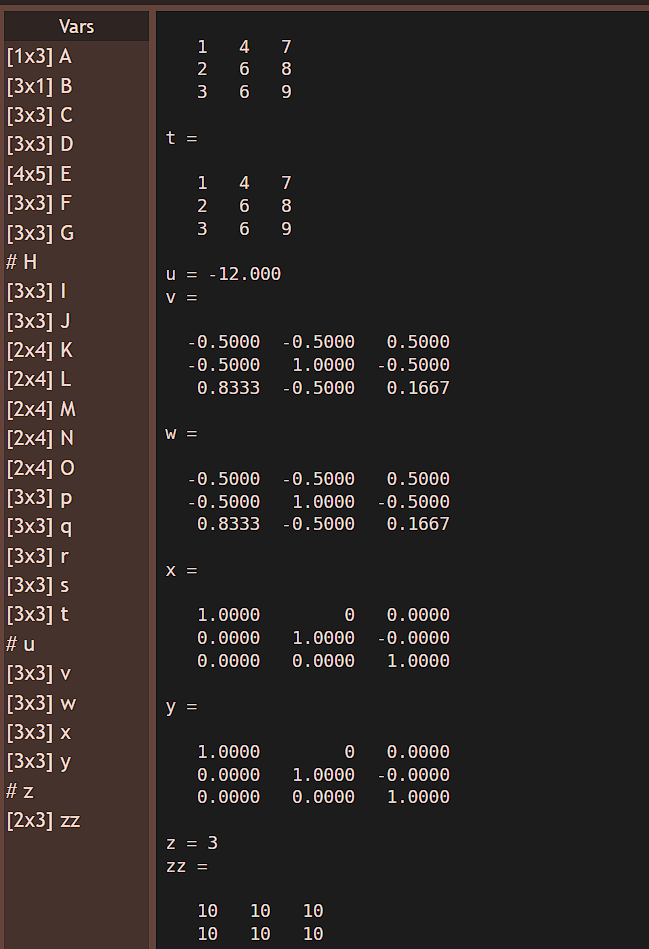
**PROGRAM CODE**

Text

Description automatically generated

**RESULTS**

** **

** **

**CONCLUSION**

The several MATLAB commands have been explored and successfully used to create a One-Dimensional Array (Row / Column Vector), a Two-Dimensional Array (Matrix of given size) and perform the required Arithmetic Operations and Matrix Operations on the Octave Online platform.

**PRECAUTIONS**

* Don’t forget to save the code after every change you make.
* Use MATLAB properly.
* MATLAB requires a stable network connection.
* Save the file after compiling the code and take the required notes and screenshots, so that you don’t have to open octave and do everything again.

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Total Marks** | **Marks Obtained** | **Comments** |
| Concept (A) | 2 |  |  |
| Implementation (B) | 2 |  |  |
| Performance (C) | 2 |  |  |
| Total | 6 |  |  |