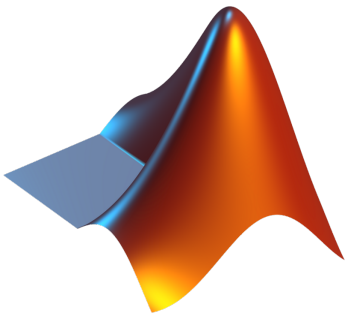
جامعة المستقبل
AL MUSTAQBAL UNIVERSITY
كلية العلوم



1st class

2024- 2025

Mathematics

Practical

MATLAB

Lecture 2

Asst. Lect. Mohammed Jabbar

Mohammed.Jabbar.Obaid@uomus.edu.iq

الرياضيات الاساسية: المرحلة الاولى

مادة الرياضيات

ماتلاب

المحاضرة الثانية

استاذ المادة: م.م محمد جبار

Cybersecurity Department

قسم الأمن السيبراني

Contents

1 Matrices and Vectors

In MATLAB, matrices and vectors are fundamental data structures for numerical computing.

1.1 Creating Vectors

```
%Row Vector:
R= [1, 2, 3, 4, 5]

R =

     1   2   3   4   5

  %Column Vector:
C = [1; 2; 3]

C =

     1
     2
     3
```

1.1.1 linspace function

In MATLAB, the linspace function generates a row vector of linearly spaced elements between two specified limits. It is especially useful for creating vectors when you know the number of points you want between a start and an end value, rather than specifying the step size (as you do with the colon operator :).

$$y = \text{linspace}(a,\ b,\ n)$$

- a: Start value.
- b: End value.

- n: Number of points to generate

Default Number of Points (100 points if n is omitted):

```
y = linspace(a, b)
```

```
y = linspace(1,50,5)
```

```
y =

    1.00   13.25   25.50   37.75   50.00
```

1.1.2 Colon Operator

Creates a vector from a to b with a step of s

```
v = a:s:b
```

Creates a vector from 1 to 10 with a step of 2

```
v = 1:2:10 = 1 3 5 7 9
```

1.2 Creating Matrices

```
%Manually:
A= [1, 2, 3; 4, 5, 6; 7, 8, 9]

A =

    1   2   3
    4   5   6
    7   8   9

  %Zero Matrix:
B = zeros(3, 3) %3x3 matrix of zeros

B =
```

0 0 0

0 0 0

0 0 0

%Identity Matrix:

E = eye(3) %3x3 identity matrix

E =

1 0 0

0 1 0

0 0 1

%Random Matrix:

R= rand(3, 3); %3x3 matrix of random values between 0 and 1

1.3 Matrix Functions

A= [1, 2, 3; 4, 5, 6; 7, 8, 9]

A =

1 2 3

4 5 6

7 8 9

%Determinant:

DA = det(A)

DA =

6.6613e-16

%Inverse:

IA = inv(A)

$$\text{IA} = \begin{matrix} -0.4504 & 0.9007 & -0.4504 \\ 0.9007 & -1.8014 & 0.9007 \\ -0.4504 & 0.9007 & -0.4504 \end{matrix}$$

Transpose

Function:

A^T (Transpose of matrix A)

Example:

$$A' = \begin{matrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{matrix}$$

2 Indexing of Matrix Values

In MATLAB, indexing is a way to access specific elements of a matrix. MATLAB uses 1-based indexing, meaning that the first element of any array is indexed by 1.

1. Accessing Individual Elements

To access an individual element in a matrix A, use the syntax $A(i, j)$, where i is the row index and j is the column index.

Example:

$$A = \begin{matrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix}$$

$A(2, 3) = 6$ (Accessing the element in the 2nd row and 3rd column)

2. Accessing Entire Rows or Columns

To access an entire row, use $A(i,:)$, and to access an entire column, use $A(:,j)$.

Example:

- Accessing the 2nd row:

$$A(2,:) = \begin{matrix} 4 & 5 & 6 \end{matrix}$$

- Accessing the 3rd column:

$$A(:,3) = \begin{matrix} 3 \\ 6 \\ 9 \end{matrix}$$

3. Logical Indexing

Logical indexing allows you to access elements based on conditions.

Example: To access elements greater than 5:

$$C = A(A > 5) = \begin{matrix} 6 \\ 7 \\ 8 \\ 9 \end{matrix}$$

5. Modifying Elements

You can modify elements in a matrix using the same indexing methods.

Example: To set the element in the 1st row and 2nd column to 10:

$$A(1,2) = 999 \quad \Rightarrow \quad A = \begin{matrix} 1 & 999 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix}$$