Ministry of Higher Education and
Scientific Research – Iraq
AL-Mustaqbal University
Department of Electrical Engineering
techniques

Stage 2

Lecture 2

Lecture: matrix

Assist lecture: Salwan Saud

Mathematical Processes on Matrices in MATLAB

Introduction

MATLAB (Matrix Laboratory) is a powerful tool for numerical computing, particularly well-suited for matrix operations. This lecture will cover fundamental mathematical processes on matrices in MATLAB, including addition, multiplication, transposition, and inversion.

---

1. Creating Matrices in MATLAB

In MATLAB, matrices are defined using square brackets [], with elements separated by spaces (or commas) in a row and semicolons; to separate rows.

```
A = [1 2; 3 4]; % 2x2 matrix
B = [5, 6; 7, 8]; % Another 2x2 matrix
```

---

2. Matrix Addition and Subtraction

Matrix addition and subtraction in MATLAB are performed element-wise.

```
C = A + B; % Matrix addition
D = A - B; % Matrix subtraction
```

Example Output:

```
     6     8
    10    12


D =
```

```
   -4    -4
   -4    -4
```

3. Scalar Multiplication

Multiplying a matrix by a scalar multiplies each element by that scalar.

```
k = 3;
E = k * A; % Multiply each element of A by 3
```

Example Output:

```
E =
    3     6
    9    12
```

---

4. Matrix Multiplication

Matrix multiplication follows the rule of dot products between rows and columns.

```
F = A * B; % Standard matrix multiplication
```

Example Output:

```
F =
   19    22
   43    50
```

---

5. Element-wise Multiplication (Hadamard Product)

To perform element-wise multiplication, use the. * operator.

```
G = A .* B;  % Element-wise multiplication
```

Example Output:

```
G =
     5    12
    21    32
```

---

6. Matrix Transposition

The transpose of a matrix swaps its rows and columns.

```
H = A'; % Transpose of A
```

Example Output:

```
H =
     1     3
     2     4
```

---

7. Matrix Inversion

A square matrix has an inverse if its determinant is non-zero.

```
if det(A) ~= 0
    A_inv = inv(A); % Inverse of A
end
```

Example Output:

```
A_inv =
    -2      1
     1.5   -0.5
```

---

8. Determinant of a Matrix

The determinant helps determine if a matrix is invertible.

```
det_A = det(A);
```

Example Output:

det_A = -2

Entering a matrix

A matrix is an array of numbers. To type a matrix into MATLAB you must
† begin with a square bracket, [
† separate elements in a row with spaces or commas (,)
† use a semicolon (;) to separate rows † end the matrix with another square bracket,].
20
Here is a typical example. To enter a matrix A, such as,
A = 2 4 1 2 3
4 5 6 7 8 9 35
(2.1)
type,
>> A = [1 2 3; 4 5 6; 7 8 9]
MATLAB then displays the 3 £ 3 matrix as follows,

A =
1 2 3
4 5 6
7 8 9

Note that the use of semicolons (;) here is difierent from their use mentioned earlier to suppress output or to write multiple commands in a single line.
Once we have entered the matrix, it is automatically stored and remembered in the *Workspace*. We can refer to it simply as matrix A. We can then view a particular element in a matrix by specifying its location. We write,
>> A(2,1)
an =
s 4
A(2,1) is an element located in the second row and flrst column. Its value is 4

Colon operator in a matrix

The colon operator can also be used to pick out a certain row or column. For example, the statement A(m:n,k:l specifles rows *m* to *n* and column *k* to *l*.
Subscript expressions refer to portions of a matrix. For example,
>> A(2,:)
ans =
4 5 6
is the second row elements of A.

The colon operator can also be used to extract a sub-matrix from a matrix A.

```
>> A(:,2:3)
ans =
2 3
5 6
8 0
```

A(:,2:3) is a sub-matrix with the last two columns of A.

A row or a column of a matrix can be deleted by setting it to a *null* vector, [ ].

```
>> A(:,2)=[]
ans =
1 3
4 6
7 0
```

Deleting row or column

To delete a row or column of a matrix, use the *empty vector* operator, [ ].

```
>> A(3,:) = []
A =
1 2 3
4 5 6
```

Third row of matrix A is now deleted. To restore the third row, we use a technique for
creating a matrix

```
>> A = [A(1,:);A(2,:);[7 8 0]]
A =
1 2 3
4 5 6
7 8 0
```

Matrix A is now restored to its original form