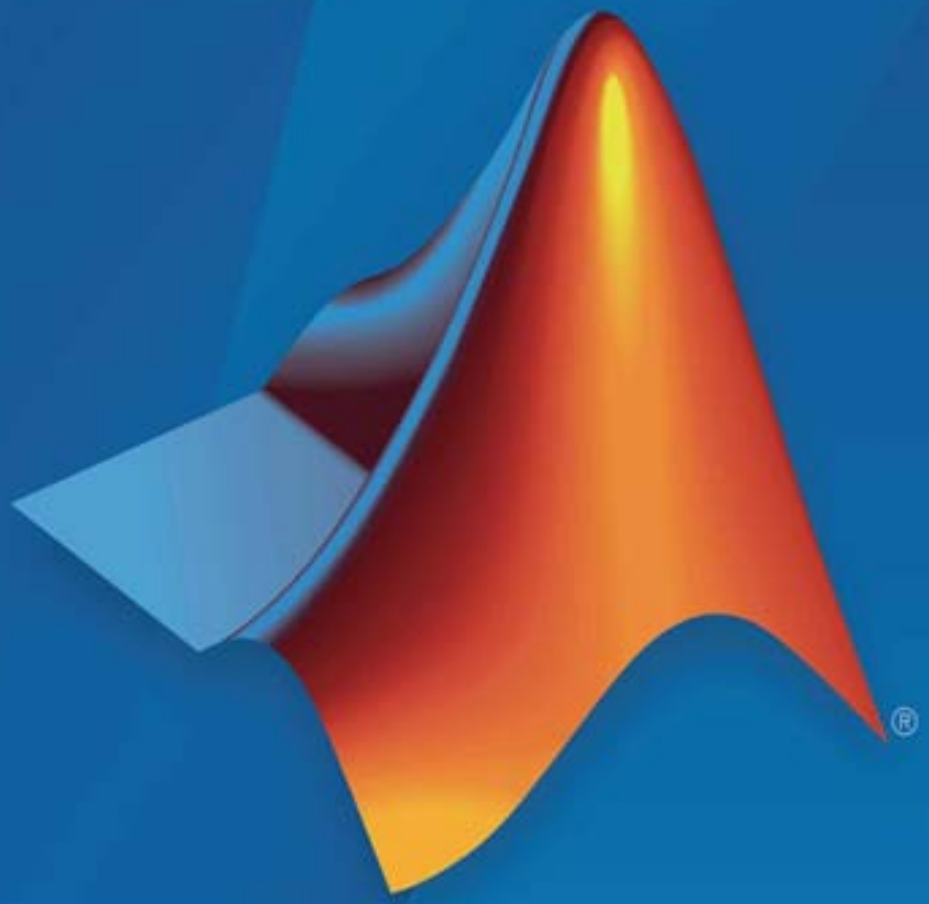
الجامعة المستنصرية
كلية الهندسة
قسم هندسة الموارد المائية



Lecture 3: Matrices in MATLAB

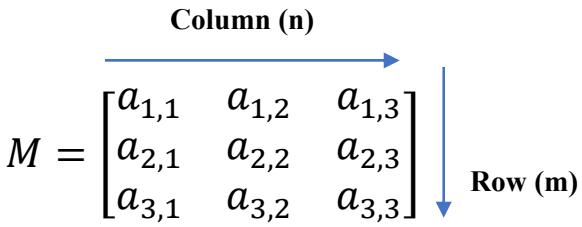
المحاضرة الثالثة: المصفوفات في ماتلاب

المنهاج الدراسي لمادة البرمجه والتطبيقات (ماتلاب)
الكورس الدراسي الاول / المرحلة الثانية

من اعداد د. صباح حسن لفته

1. Matrices in MATLAB:

In MATLAB, a matrix is a two-dimensional array of numbers. The creation of matrices cab be done by entering elements in each row as comma or space delimited numbers and using semicolons to mark the end of each row.



So, each element references with a unique two subscript numbers ($M_{n,,m}$) which defines its location inside the matrix.

Example (1): To create a matrix with 4 x 4 size, we are entering the elements of the matrix in command window as follow:

```
>> M=[1, 2, 3, 4; 5, 6, 7, 8; 9, 10, 11, 12; 13, 14, 15, 16]
```

```
M =

     1     2     3     4
     5     6     7     8
     9    10    11    12
    13    14    15    16
```

To call element with row number (2) and column (4), we type:

```
>> M(2,4)
ans =     8
```

To reference all the elements in the m^{th} column we type M(:,m).

```
>> M(:,3)
ans =
```

```
     3
     7
    11
    15
```

Example (2): Let us create a smaller matrix taking the elements from the second and third columns of matrix (M):

```
>> N=M(:,2:3)
```

MATLAB will execute the above statement and return the following result:

```
 N =
     2     3
     6     7
    10    11
    14    15
```

Also, we can create a matric from different vectors.

Example (3): Create two normal vectors and use them to create a matrix with size (2,4):

```
>> A=[ 4; 6; 3; 7];
>> B=[ 1; 8; 0; 2];
>> C=[A B]
```

```
   C =
     4     1
     6     8
     3     0
     7     2
```

Another way to create matrices in MATLAB using built-in functions (***ones and zeros***).

Example (4): Create two matrices using ones and zeros functions with size (4x4):

```
>> Mn=ones(4,4)
```

```
Mn =
     1     1     1     1
     1     1     1     1
     1     1     1     1
     1     1     1     1
```

```
>> Mo=zeros(4,4)
```

```
Mn =
     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
```

MATLAB allows to delete an entire row or column of a matrix by assigning an empty set of square braces [] to that row or column. Basically, [] denotes an empty array. For example, let us delete the fourth row of M:

```
>> M(4,:)=[]
```

MATLAB will execute the above statement and return the following result:

```
M =
     1     2     3     4
     5     6     7     8
     9    10    11    12
```

Again, this time will delete the entire 4^{th} of M:

```
>> M(:,4)=[]
```

```
M =
     1     2     3
```

5 6 7

9 10 11

The statement size can use to know the number of rows and column in any matrix, for example:

>> [n,m]=size(M) ↵

n=4 m=4

in which, n is number of rows and m is number of columns.

2. Matrix Operations:

MATLAB offers several operations that can be performed with matrices as follows:

a- scalar operations.

b- inverse of a matrix.

c- transpose of a matrix.

d- determinant of a matrix.

e- joining of matrices.

f- addition and subtraction.

g- division of matrices

h- multiplication of matrices.

a- Scalar Operations of Matrices

Adding, subtracting, multiplying or dividing a matrix by a number is called the ***scalar operation***. Scalar operations produce a new matrix with same number of rows and columns with each element of the original matrix added to, subtracted from, multiplied by or divided by the number.

Example: Create a code script which can do scalar operations for matrix (a) with size (3x3) and scalar value of (3).

```
>> x= [ 9, 6, 3; 10, 2, 7; 11, 4, 5];

>> y=3;

>> m1=x-y;

>> m2=x+y;

>> m3=x*y;

>> m4=x/b;
```

After the execution of the above script, we get the following results,

```
m1 =
    6    3    0
    7   -1    4
    8    1    2
m2 =
   12    9    6
   13    5   10
   14    7    8
m3 =
   27   18    9
   30    6   21
   33   12   15
m4 =
   3.0000   2.0000   1.0000
   3.3333   0.6667   2.3333
   3.6667   1.3333   1.6667
```

b- Inverse of a Matrix

The inverse of a matrix X is represented by X^{-1} such that the following relationship refers:

$$X\,X^{-1}=X^{-1}\,X=1$$

A matrix inverse is not always existing. If the determinant of the matrix is zero, then the inverse does not exist, and the matrix is singular. Inverse of a matrix in MATLAB is calculated using the ***inv*** function. Inverse of a matrix A is given by inv(X).

Example: Write a code to calculate the inverse of the matrix (A) with size (3x3).

```
>> A=[ 1, 2, 3; 2, 3, 4; 1, 2, 5];
>> Ainv=inv(A)
```

```
Ainv =
   -3.5000    2.0000    0.5000
    3.0000   -1.0000   -1.0000
   -0.5000         0    0.5000
```

c- Transpose of a Matrix

A switching process of rows and columns in a matrix is called transpose operation. It is denoted by a single quote (').

Example: Write a code to find the transpose of the matrix (A).

```
>> A=[ 1, 2, 3; 2, 3, 4; 1, 2, 5];
>> Atrn=A'
```

```
Atrn =
     1     2     1
     2     3     2
     3     4     5
```

d- Determinant of a Matrix

Determinant of a matrix is determined using the ***det*** function of MATLAB. Determinant of a matrix X is given by ***det(X)***.

Example: Write a code to find the determinant of the matrix (A).

```
>> A=[ 1, 2, 3; 2, 3, 4; 1, 2, 5];
>> DetA=det(A)
DetA =
    -2
```

e- Joining of Matrices

We can join two matrices to create a larger matrix. The pair of square brackets '**[]**' is the joining operator. MATLAB allows two types of joining:

- Horizontal joining.
- Vertical joining.

This can be achieved by matrices using commas, they are just arranged horizontally. It is called horizontal joining. On the other side, if we join two matrices by separating those using semicolons, they are arranged vertically. It is called vertical joining.

Example: Write a code to join matrix (A, size 3x3) with matrix (B, size 3x3) horizontally and vertically to produce matrices (C and D).

```
>> A=[ 10, 12, 23; 12, 7, 6; 21, 9, 8];
```

```
>> B=[ 31, 9, 30; 9, 0, 8; 24, 3, 12];
>>  C=[ A, B]
>>  D=[ A; B]
```

```
C =
    10    12    23    31     9    30
    12     7     6     9     0     8
    21     9     8    24     3    12
D =
    10    12    23
    12     7     6
    21     9     8
    31     9    30
     9     0     8
    24     3    12
```

f- Addition and Subtraction of Matrices

We can add or subtract matrices in MATLAB. The operand matrices must have the same number of rows and columns in order to do addition and subtraction operations.

Example: Write a code to do addition and subtraction operations between matrices (A and B).

```
>> A=[ 10, 12, 23; 12, 7, 6; 21, 9, 8];
>> B=[ 31, 9, 30; 9, 0, 8; 24, 3, 12];
>>  C=A+B
>>  D=A-B
```

```
C =
    41    21    53
    21     7    14
    45    12    20
D =
   -21     3    -7
     3     7    -2
    -3     6    -4
```

g- Division of Matrices

Division of two matrices can be performed using left (\) or right (/) which are called division operators. Both the operand matrices must have the same number of rows and columns.

Example: Write a code to make division between matrices (A and B) using left and right operators.

```
>> A=[ 10, 12, 23; 12, 7, 6; 21, 9, 8];
>> B=[ 31, 9, 30; 9, 0, 8; 24, 3, 12];
>>  C= A / B
>>  D= A \ B
C =
    1.6898   -1.8577   -1.0693
    0.5620   -2.3285    0.6472
    0.5693   -3.0730    1.2920
D =
    2.0793    0.4816    0.2321
   -4.8104   -1.7756   -0.5184
```

```
    2.9536    1.1083    1.4739
```