Al-Mustaqbal University College
Department of Medical Instrumentation Techniques Engineering
Class:Third
Subject: Computer Applications
Lecturer: Dr_Feryal Althefery
Lecture: (3)

Matrix

Useful Matrix Generators

MATLAB provides four easy ways to generate certain simple matrices. These are:

zeros	a matrix filled with zeros
ones	a matrix filled with ones
rand	a matrix with uniformly distributed random elements
randn	a matrix with normally distributed random elements
eye	identity matrix

To generate matrices in MATLAB should be you give the functions the number of rows and columns.

For examples

```
>> a = zeros(2,3)

a =

    0   0   0

    0   0   0

>> b = ones(2,2)/2
b =

    0.5000   0.5000
```

Email : feryal.ibrahim@uomus.edu.iq

```
    0.5000   0.5000

>> u = rand(1,5)

u =

0.9218  0.7382  0.1763  0.4057  0.9355

>> n = randn(5,5)

n =

-0.4326   1.1909   -0.1867   0.1139   0.2944
-1.6656   1.1892    0.7258   1.0668  -1.3362
 0.1253  -0.0376   -0.5883   0.0593   0.7143
 0.2877   0.3273    2.1832  -0.0956   1.6236
-1.1465   0.1746   -0.1364  -0.8323  -0.6918

>> eye(3)

and =

     1   0   0
     0   1   0
     0   0   1
```

Subscripting

Individual elements in a matrix are denoted by a row index and a column index.

To pick out the third element of the vector u type:

```
>> u(3)
ans =

    0.1763

>> u([1 2 3])

ans =

    0.9218    0.7382    0.1763

>> u(1:3)

ans =

    0.9218    0.7382    0.1763

>> a = [1 2 3; 4 5 6]

a =

   1  2  3

   4  5  6

>> a(2,3)

ans = 6

>> a(1:2,3)

ans = 3

      6
```

Colon Operator

The symbol colon (:) is used as an operator in MATLAB programming and is a commonly used operator. This operator is used to construct a vector that is defined by a simple expression, iterate over or subscribe to an array, or access a set of elements of an existing vector.

```
>> a(2 , :)
ans =
     2   5   6
>> a(: , 3)
ans =
      3
      6
      9
>> a(4)
ans =
      2
```

The colon symbol can be used as a single index to a matrix.

```
>> a(:)
ans =1
      4
      7
      2
```

5

8

3

6

9

End as a subscript

To access the last element of a matrix along a given dimension, use end as a subscript.

For example:

```
>> q = 4 : 10
q =
    4  5  6  7  8  9  10
>> q(end)
ans =
     10
>> q(end-4:end)
ans =
     6  7  8  9  10
```

End as a subscript

```
>> q = [ 7 8 9 10; 6 1 2 20; 5 4 3 30]
q =
    7  8  9  10
    6  1  2  20
    5  4  3  30
>> q(end , end)
ans =
      30
>> q(2,end-1:end)
ans =
      2   20
>> q(end-2:end,end-1:end)
ans =
      9   10
      2   20
      3   30
>> q(end-1,:)
ans =
      6  1  2  20
```

Transpose

Transpose method is a mathematical operation where:

1- The elements of the row are converted into the elements of the column. (or)

2- The elements of the column are converted into the elements of the row.

The transpose of the Matrix is represented by an apostrophe or single quote (') or power of T like as [ ]' Or [ ]^{T}.

Note: The representation of the transpose of the Matrix is the same as the representation of the transpose of the vector.

How to determine the transpose of the given below matrix D?

For example:

```
>> D = [2 0 1; 9 6 8; 4 1 1]

D =

     2   0   1

     9   6   8

     4   1   1
```

By using the transpose method,

```
>>D'

D =

     2   9   4

     0   6   1

     1   8   1
```

Deleting Rows or Columns

To get rid of a row or column set it equal to the empty matrix [].

```
>> a = [1 2 3;4 5 6;7 8 9]

a =

   1  2  3

   4  5  6

   7  8  9

>> a(:,2) = []

a =

   1  3

   4  6

   7  9
```