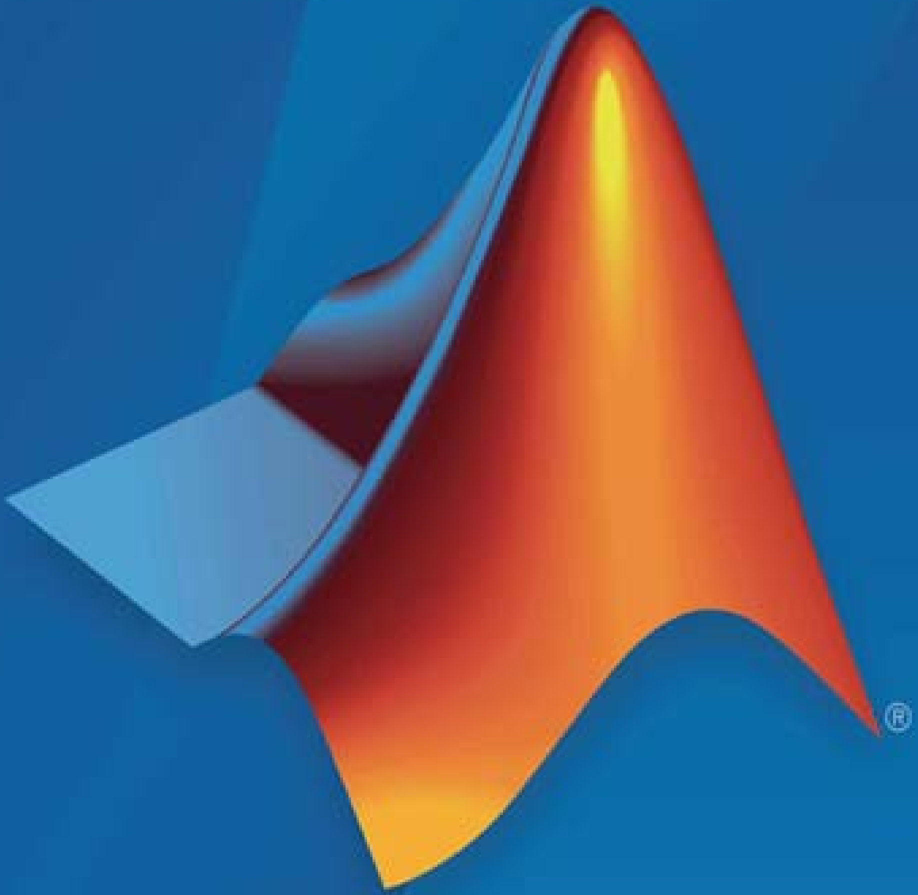
الجامعة المستنصرية

كلية الهندسة

قسم هندسة الموارد المائية



المحاضرة الثانية: المتجهات في ماتلاب

Lecture 2: Vectors in MATLAB

المنهاج الدراسي لمادة البرمجه والتطبيقات (ماتلاب)

الكورس الدراسي الاول / المرحلة الثانية

من اعداد د. صباح حسن لفته

1. Vectors in MATLAB:

In MATLAB, a vector is a matrix with either one row (*called horizontal vector, elements separated by coma ,*) or one column (*called normal vector, elements separated by semi-colon ;*). The distinction between row vectors and column vectors is **essential**. MATLAB vectors are used in many situations, e.g., creating x-y plots, that do not fall under the rubric of linear algebra. In these contexts, a vector is just a convenient data structure. MATLAB still enforces the rules of linear algebra so paying attention to the details of vector creation and manipulation is always important.

2. Creating Vectors in MATLAB

In MATLAB, its easy to create a vector by simply introduce it on the left-hand side of an equal sign (V=). While the expression on the right side of the equal sign must evaluate to a vector. There are many ways to create a vector in MATLAB, each one having advantages in particular situations:

a. using the built-in functions ones, zeros, linspace, and logspace,

b. assigning a mathematical expression involving vectors,

c. using colon notation (:),

Each of the above-mentioned methods is useful. These methods will be described in the following section.

a- Creating Vectors Using Built-In Functions

The following statements ***ones, zeros linspace, and logspace*** are built-in functions in MATLAB and helpful to create vectors. They can be used in creation of vectors of a specific size and with a prescribed spacing between the elements. These functions will be demonstrated by examples.

To create a vector with one of these functions you must (at least initially) decide how long do you want the vector to be. You must also decide whether the vector is a row or column vector. The ***ones and zeros*** functions have two arguments. The first is the number of rows in the *matrix* you wish to create. The second is the number of columns. To create a row or a column vector set the appropriate argument of ones and zeros to one.

Example (1): To create a row vector of length 5, filled with number one, we write the expression in MATLAB:

```
>> x = ones(1,5)          ↵
      x = 1   1   1   1   1
```

Example (2): To create a column vector of length 5, filled with number one, we write the expression in MATLAB:

```
      >> y = zeros(5,1)        ↵
```

```
y =
      0
      0
      0
      0
      0
```

On the other hand, the functions ***linspace and logspace*** create vectors with linearly spaced or logarithmically spaced elements, respectively. Following are some examples including the MATLAB outputs:

Example (1): Creating a vector contains five elements with number value increased by one,

```
 >> x = linspace(1,5,5)

        x =

           1     2    3    4    5
```

Example (2): Creating a vector contains four elements with number value increased logarithmically,

```
  >> y = logspace(1,4,4)


        y =

               10        100        1000       10000
```

b. Assigning a mathematical expression involving vectors

The creation of vector can be done by entering the values of its elements directly. Also, this vector may be assigned to another vector. If the vector on the left of the equal sign does not exist it is created to fit the expression on the right-hand side of the equal sign.

Example (1):

```
>> x = zeros(1,5);

>> y = x;
```

Example (2):

```
>> x = [1, 2, 3, 4, 5];      or     x= [1 2 3 4 5] ;

>> y = x;
```

Example (3):

```
>> x = [1; 2; 3; 4; 5];

>> y = x;        ↵

   y=

       1
       2
       3
       4
       5
```

a. Using colon notation

MATLAB colon notation is a perfect method to refer to ranges of vector elements. It is often used in copy operations and in the creation of vectors and matrices (next lecture will be discussed). Colon notation method can be used to create a vector as follows:

```
  >> x= start: increment:end   or  x= start:end
```

Example (1): Create a vector with 6 elements spaced by 3

```
>> x= 0:3:15

x     0     3    6     9    12    15
```

Example (2): Create a vector with 9 elements spaced by 1

```
>> x= 1:9

x     1     2    3     4    5    6     7     8     9
```

3. Vector Operations

Mathematical operations involving vectors follow the rules of linear algebra. In order to do the mathematical operations between vectors, they must be in same size and arrangement. The following vector operations can be performed in MATLAB:

***a**- Addition and Subtraction*

Example (1): Create a vector (c) by adding vector x with five elements to vector y with five elements also,

```
  >> x=1:5;

>> y=[3 4 5 6 7];

>> c=x+y;          ↵
```

```
c     4    6   8   10    12
```

b*- Vector multiplication and division*

Example (1): Create a vector (v) by multiplying vector x with six elements by vector d with the same size,

```
>> x= [1, 4, 5, 3, 9, 8];
>> d= [2 0 4 1 7 3];
>> v= x.*d;          ↵
 v =
    2    0   20    3   63   24
```

Hint:
Remember always to place a dot before multiplication symbol (.*).

Example (2): Create a vector (n) by dividing vector x with six elements to vector d with the same size,

```
>> x= [1, 4, 5, 3, 9, 8];
>> d= [2 1 4 1 7 3];
>> n= x./d;          ↵
 n =
   0.5000    4.0000    1.2500    3.0000    1.2857    2.6667
```

c*- Vector transpose*

Example (1): Create a vector by moving element of vector x with four elements from rows to columns,

```
>> x=[11, 3, 41, 5];
>> V= x′;

 V =

    11
     3
    41
     5
```

4. Useful Vector Operations

In order to know the number of elements in the vector (usually known as length of the vector), there is a helpful tool in MATLAB called *length* which can be used as follow:

```
>> X=[1,2,3,4,5,6,7,8];

>> length(X)      ↵

     8
```

Above value refers to vector length or number of elements in the vector.

To call any element in the vector, this process can be achieved by:

```
>> t= [1,0,4,5,6] ;

>> t(2)      ↵

    0
```

Here, the element number 2 is called by placing its location inside two parentheses.

Also, MATLAB provides another useful way to erase any element inside the vector using two square paratheses [] :

```
>> y=[1, 5, 6, 12, 14, 20, 11];
>> y(4)=[];
>> y          ↵
    1  5   6  14   20   11
```

In this example, the element number 4 has been deleted, this operation can be used for any deletion process to any element(s) of any vector.

Sort is another tool used in MATLAB to arrange vector elements. This built-in function is designed to arrange the elements inside the vector in ascending order (from small value until larger value).

```
>> n=[4, 3, 1, 5, 6, 7, 11, 0];

>> sort(n)          ↵

 ans =
     0  1  3  4  5  6  7  11
```

Display some vector element can be performed using following procedure in MATLAB:

```
>> x=[ 4; 2; 1; 6 ;5, 9; 8; 10];

>> x(3:6)          ↵

ans =
         1
         6
         5
         9
```

Here:
The elements from 3 until 6 are displayed only using this procedure.