



Computer science

second Stage

Lec4

Array&Matrix

Asst.Lect.Lubna Ali Jalil

أهداف المحاضرة: في نهاية يكون الطالب قادرا على

1- فهم مفهوم المصفوفات والمتجهات

2- التعرف على كيفية إنشاء المصفوفات والمتجهات في MATLAB

3- استخدام العمليات الأساسية على المتجهات



What is an Array?

An array is a collection of similar data elements stored at contiguous memory locations. It is the simplest data structure where each data element can be accessed directly by only using its index number.

Need of using Array:

In programming, most of the cases need to store a large amount of data of similar type. To store such a huge amount of data, we need to define numerous variables. It would be very tough to memorize all variable names while writing the programs. Instead, it is better to define an array and store all the elements into it.



Advantages of Array:

1. Arrays represent multiple data elements of the same type using a single name.
2. In an array, accessing or searching an element is easy by using the index number.
3. An array can be traversed easily just by incrementing the index by 1.
4. Arrays allocate memory in contiguous memory locations for all its data elements.

Accessing Elements in an Array:

To access any element of an array we need the following details:

1. Base Address of array.
2. Size of an element in bytes.
3. Which type of indexing, array follows.

Creating Vectors:

A vector is a one-dimensional array of numbers. MATLAB allows creating two types of vectors:

Row vectors

Column vectors

Row vectors are created by enclosing the set of elements in square brackets, using space or comma to delimit the elements.

For example,

```
r = [7 8 9 10 11]
```

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

r =

Columns 1 through 4

7 8 9 10

Column 5

11

Another example,

```
r = [7 8 9 10 11];
```

```
t = [2, 3, 4, 5, 6];
```

```
res = r + t
```

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

res =

Columns 1 through 4

9 11 13 15

Column 5

17

Column vectors are created by enclosing the set of elements in square brackets, using semicolon (;) to delimit the elements.

```
c = [7; 8; 9; 10; 11]
```

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

C =

7

8

9

10

11

There are many functions that are performed on vectors, which enhance their importance and applications. We will now explain most of these operations and functions through examples.

1-The Length function: It calculates the number of elements in the vector, as shown in the example.

```
>> v=[2 5 0 1 4 -1]

v =
    2     5     0     1     4    -1
>> length(v)
ans =
     6
```

The Sum function: This function calculates the total sum of the elements in the vector, as demonstrated in the example.

```
>> w=sum(v)

w =
    11
```

The Max function: This function identifies the largest element in the vector in terms of value, as illustrated in the example.

```
>> w=max(v)

w =
     5
```

The Min function: This function identifies the smallest element in the vector in

terms of value, as illustrated in the example.

```
>> w=min(v)
```

```
w =  
-1
```

The Size function: It returns the dimensions of a vector or matrix.

```
>> Matrix=[1,2,3,;4,5,6]  
Matrix =
```

```
1  2  3  
4  5  6
```

```
>> [Matrix]=size(Matrix)  
Matrix =
```

```
2  3
```

The Sort function: This function arranges the elements of the vector in ascending order.

```
>> r=[9 7 5 8 3]
```

```
    r =  
9    7    5    8    3
```

```
>> s=sort(r)
```

```
    s =  
3    5    7    8    9
```

The Range function: This function calculates the difference between the largest and smallest values in the vector.

```
>> range(r)
```

```
ans =  
6
```

Creating Matrices:

A matrix is a two-dimensional array of numbers.

It is a collection of data organized in the form of rows and columns.

In MATLAB, a matrix is created by entering each row as a sequence of space or comma separated elements, and end of a row is demarcated by a semicolon. For example, let us create a 3-by-3 matrix as:

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

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

```
m =
```

```
1 2 3
```

```
4 5 6
```

```
7 8 9
```

NOTE: We separate the row numbers using either a comma(,) or a space. and the elements of the first row are separated from the second row using(;).

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

```
Matrix =
```

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


If we have a matrix, we can retrieve the second or third row using this method.

```
>> Matrix(2,:)
ans =
    4    5    6
```

we can retrieve the second or third row using this method.

```
>> Matrix(:,2)
ans =
    2
    5
    8
```

HOMEWORK

let `v = [-3 0 4 6 8]`.

Use MATLAB functions to find:

`length(v)`

`sum(v)`

`max(v)`

`min(v)`

`sort(v)`

`range(v)`

