Lecture three: Vectors and Matrices

1. Introduction

Arrays: an array is a mathematical structure that has a collection of numerical elements, each of these elements referenced by an index which represents the location of that element within the array. This index has integer sequence and any element can be accessed by this sequence.

Arrays can be one Dimension, two dimensions or even multi-dimension depending on their structure and nature.

A vector is a one dimension array; it can be either row vector which consists of one row and several columns or (n×1) elements size. Column vector is also one dimensional array but with one column and several rows or (1×n) elements size.

A matrix is normally a two dimensional array with (n×m) elements in which n represent the number of rows while m represents the number of columns. In some cases when (n=m) then such a matrix is called a square matrix.

2. Defining and entering vectors and matrices elements

There are several methods to enter elements to vectors and matrices in Matlab:-

1- Entering an explicit list of elements.

2- Loading data from external file (or importing from an application such as Excel).

3- Generating elements using function (such as (rand) function for random data).

4- Building from other arrays elements.

Array elements in rows are represented by semi columns (;), while columns in either spaces or commas (,) in between.

3. Matrix Generation

Matrices are fundamental to MATLAB. Therefore, we need to become familiar with matrix generation and manipulation. Matrices can be generated in several ways.

3.1 Entering a vector

A vector is a special case of a matrix. The purpose of this section is to show how to create vectors and matrices in MATLAB. As discussed earlier, an array of dimension 1×n is called a row vector, whereas an array of dimension m×1 is called a column vector. The elements of vectors in MATLAB are enclosed by square brackets and are separated by spaces or by commas. For example, to enter a row vector, v, type

```
>> v = [1 4 7 10 13]
v =
1 4 7 10 13
```

Column vectors are created in a similar way; however, semicolon (;) must separate the components of a column vector,

```
>> w = [1;4;7;10;13]
w =
1
4
7
10
13
```

On the other hand, a row vector is converted to a column vector using the transpose operator.

The transpose operation is denoted by an apostrophe or a single quote (').

```
>> w = v'
w =
1
4
7
10
13
```

Thus, v(1) is the first element of vector v, v(2) its second element, and so forth. Furthermore, to access blocks of elements, we use MATLAB's colon notation (:). For example, to access the first three elements of v, we write,

```
>> v(1:3)
ans =
1 4 7
```

Or, all elements from the third through the last elements,

```
>> v(3,end)
ans =
7 10 13
```

where end signifies the last element in the vector. If v is a vector, writing >> v(:) produces a column vector, whereas writing >> v(1:end) produces a row vector.

3.2 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,].
- Here is a typical example. To enter a matrix A, such as,

A = [ ]

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

Note that the use of semicolons (;) here is different from their use mentioned earlier tom 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)
ans =
4
```

A(2,1) is an element located in the second row and first column. Its value is 4.