SVKM'S NMIMS Deemed to be UNIVERSITY | NAVI MUMBAI

MATLAB
Unit 2-Lecture 3

BTech (CSBS) -Semester VII

19 July 2022, 09:35AM

*few figures/content have been prepared/referred from internet source/books

Creating 2D array(matrix)

Variable_name = [1st row elements; 2nd row elements;……;last row elements]

- Matrix are used in science & eng to describe many physical quantities.
- All rows must have same number of elements

Zeros , ones and eye commands

- **zeros(m,n):** *mxn* matrix of 0's.
- **ones(m,n):** *mxn* matrix of 1' s.
- **eye(n):** *nxn* identity matrix

Array

List of numbers arranged in row and/or columns.

Simplest array

-1D array

-usually to represent vectors.

Complex array

-2D array

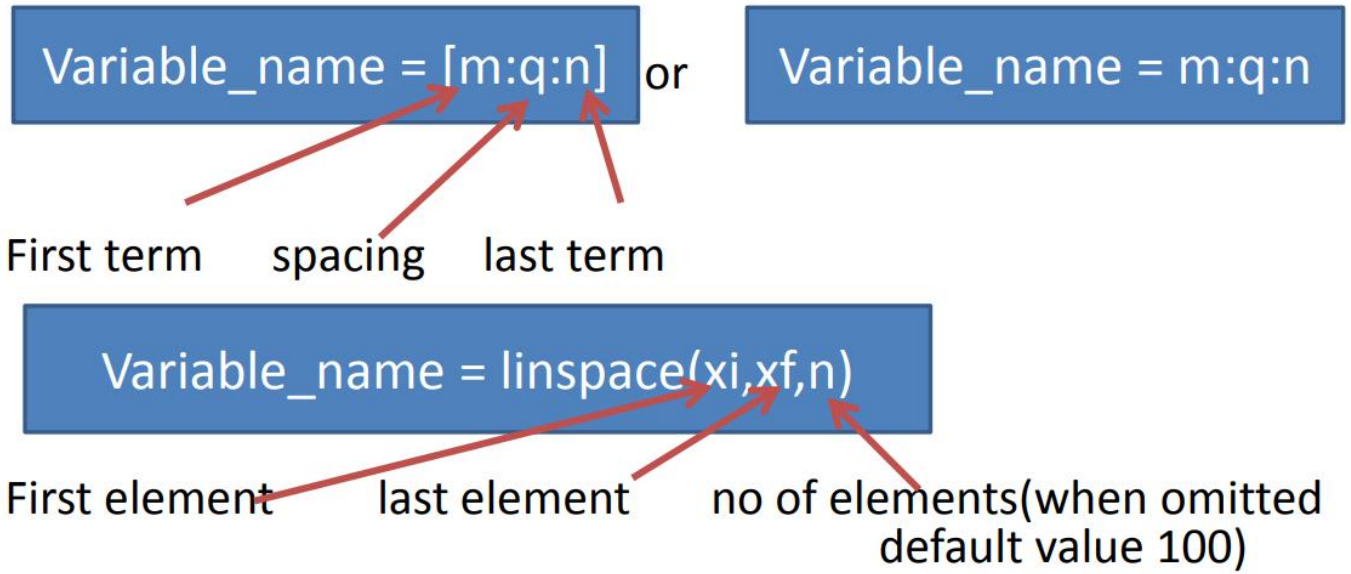
-represent matrixes

Creating vector from a known list of numbers

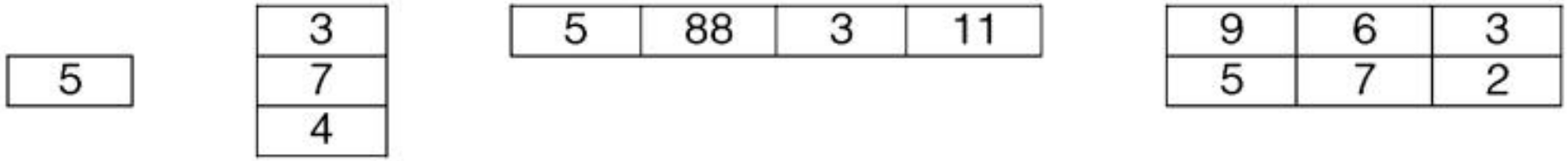
Variable_name = [type vector elements]

Row vector :- type elements with space or comma

Column vector :- type elements with semicolon (;) or press Enter key after each element



Vector and Matrix



The scalar is 1×1, the column vector is 3×1 (three rows by one column), the row vector is 1×4 (one row by four columns), and the matrix is 2×3 (two rows by three columns). All of the values stored in these matrices are stored in what are called ***elements***.

Creating row vector

There are several ways to create row vector variables. The most direct way is to put the values that you want in the vector in square brackets, separated by either spaces or commas. For example, both of these assignment statements create the same vector *v*:

```
>> v = [1   2   3   4]
   v =
     1   2   3   4

>> v = [1,2,3,4]
   v =
     1   2   3   4
```

Colon Operator

If, as in the preceding examples, the values in the vector are regularly spaced, the ***colon operator*** can be used to ***iterate*** through these values. For example, 2:6 results in all of the integers from 2 to 6 inclusive:

```
>> vec = 2:6
vec =
     2     3     4     5     6
```

In this vector, there are five elements; the vector is a 1×5 row vector. Note that in this case, the brackets [ ] are not necessary to define the vector.

With the colon operator, a ***step value*** can also be specified by using another colon, in the form (first:step:last). For example, to create a vector with all integers from 1 to 9 in steps of 2:

```
>> nv = 1:2:9
nv =
     1   3   5   7   9
```

Line space function

The **linspace** function creates a linearly spaced vector; **linspace(x,y,n)** creates a vector with *n* values in the inclusive range of *x* to *y*. If *n* is omitted, the default is 100 points. For example, the following creates a vector with five values linearly spaced between 3 and 15, including the 3 and 15:

```
>> ls = linspace(3,15,5)
ls =
     3     6     9    12    15
```

Concatenating Vector

Vector variables can also be created using existing variables. For example, a new vector is created here consisting first of all of the values from *nv* followed by all values from *ls*:

```
>> newvec = [nv ls]
newvec =
   1   3   5   7   9   3   6   9   12   15
```

Putting two vectors together like this to create a new one is called ***concatenating*** the vectors.

Logspace function

Similarly, the **logspace** function creates a logarithmically spaced vector; **logspace(x,y,n)** creates a vector with *n* values in the inclusive range from 10^x to 10^y. If *n* is omitted, the default is 50 points. For example, logspace (1,4,4) creates a vector with four elements, logarithmically spaced between 10^1 and 10^4, or in other words 10^1, 10^2, 10^3, and 10^4.

```
>> logspace(1,4,4)
ans =
          10         100        1000       10000
```