SVKM'S NMIMS Deemed to be UNIVERSITY | NAVI MUMBAI

MATLAB
Unit 3-Lecture 5

BTech (CSBS) -Semester VII

26 July 2022, 09:35AM

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

Unit 3

- Matrix
- Array
- Basic mathematical functions

An array is MATLAB's basic data structure

Can have any number of dimensions. Most common are

- vector - one dimension (a single row or column)
- matrix - two or more dimensions
- Scalar - matrices with only one row and one column.

Arrays can have numbers or letters

Creating Matrices

In MATLAB, a vector is created by assigning the elements of the vector to a variable. This can be done in several ways depending on the source of the information.

—Enter an explicit list of elements

—Load matrices from external data files

—Using built-in functions

—Using own functions in M-files

A matrix can be created in MATLAB by typing the elements (numbers) inside square brackets [ ]

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

Creating Matrices

```
>> A = [2 -3 5; -1 4 5]  % Note MATLAB displays column vector vertically
A=
2 -3 5
-1 4 5
>> x = [1 4 7] % Note MATLAB displays row vector horizontally
x=
4
1 4 7
>> x = [1; 4; 7]  %Optional commas may be used between the elements.Type the semicolon (or
                                press Enter) to move to the next row
x=
1
4
7
```

Creating Matrices

```
>> cd=6; e=3; h=4;

>> Mat=[e cd*h cos(pi/3);h^2 sqrt(h*h/cd) 14]

Mat =
    3.0000   24.0000    0.5000
   16.0000    1.6330   14.0000
```

Concatenation of Matrices

```
Command Window
>>  a=[1 2;3 4];
b=[4 6 ;8 9];
A=[a,b]

A =

     1     2     4     6
     3     4     8     9

>> B=[a;b]

B =

     1     2
     3     4
     4     6
     8     9

fx >>
```

Row wise concate

Coloumn wise concate

Colon operator

The colon operator can be used to create a vector with constant spacing

```
x = m:q:n
```

- `m` is first number
- `n` is last number
- `q` is difference between consecutive numbers

```
>>x=[1:2:10]
x =
  1   3   5   7   9
```

If omit q, spacing is one

```
y = m:n
```

```
>> y=1:5
y =
  1 2 3 4 5
```

Colon operator

```
>> x=1:5:50

x =

     1     6    11    16    21    26    31    36    41    46

>> 1:5:50

ans =

     1     6    11    16    21    26    31    36    41    46
```

Question 1

How can you use the colon operator to generate the vector shown below?

```
9   7   5   3   1
```

linspace function

v = linspace(x_i, x_f, n)

- x_i is first number
- x_f is last number
- n is number of terms

(= 100 if omitted)

```
>> linspace (4,8,50)

ans =

  Columns 1 through 11

    4.0000    4.0816    4.1633    4.2449    4.3265    4.4082    4.4898    4.5714    4.6531    4.7347    4.8163

  Columns 12 through 22

    4.8980    4.9796    5.0612    5.1429    5.2245    5.3061    5.3878    5.4694    5.5510    5.6327    5.7143

  Columns 23 through 33

    5.7959    5.8776    5.9592    6.0408    6.1224    6.2041    6.2857    6.3673    6.4490    6.5306    6.6122

  Columns 34 through 44

    6.6939    6.7755    6.8571    6.9388    7.0204    7.1020    7.1837    7.2653    7.3469    7.4286    7.5102

  Columns 45 through 50

    7.5918    7.6735    7.7551    7.8367    7.9184    8.0000
```

Misc Matrix

- zeros(r,c) - makes matrix of r rows and c columns, all with zeros
- ones(r,c) - makes matrix of r rows and c columns, all with ones
- rand(r,c) - makes matrix of r rows and c columns, with random numbers
- eye(n) - makes square matrix of n rows and columns. Main diagonal (upper left to lower right) has ones, all other elements are zero
- magic(n) - makes a special square matrix of n rows and c columns, called Durer's matrix

Misc Matrix

```
>> a=zeros(4,3)

a =

     0     0     0
     0     0     0
     0     0     0
     0     0     0
```

```
>> b=ones(4,3)

b =

     1     1     1
     1     1     1
     1     1     1
     1     1     1
```

```
>> c=rand(4,3)

c =

    0.8147    0.6324    0.9575
    0.9058    0.0975    0.9649
    0.1270    0.2785    0.1576
    0.9134    0.5469    0.9706
```

```
>> d=eye(4)

d =

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

```
>> e=magic(4)

e =

    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
```

Creating Matrix variable

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

```
>> mat = [3 5 7; 1 2]
Error using vertcat
Dimensions of matrices being concatenated are not consistent.
```

```
>> mat = [2:4; 3:5]
mat =
    2   3   4
    3   4   5
```

Linear indexing

```
>> intmat = [100 77; 28 14]
intmat =
    100     77
     28     14
>> intmat(1)
an =
    100
>> intmat(2)
ans =
     28
>> intmat(3)
ans =
     77
>> intmat(4)
ans =
     14
```

Dimension

```
>> vec = -2:1
vec =
   -2   -1   0   1
>> length(vec)
ans =
   4
>> size(vec)
ans =
   1   4
```

```
>> mat = [1:3; 5:7]'
mat =
   1   5
   2   6
   3   7

>> [r, c] = size(mat)
r =
  3
c =
  2
```

```
>> size(mat)
ans =
   3   2
```

Question

How could you create a matrix of zeros with the same size as another matrix?

numel function

```
>> v=9:-2:1

v =

     9     7     5     3     1

>> numel(v)

ans =

     5
```

For vectors, **numel** is equivalent to the **length** of the vector. For matrices, it is the product of the number of rows and columns.

Question

```
mat = [1:3; 44 9  2; 5:-1:3]
mat(3,2)
mat(2,:)
size(mat)
mat(:,4) = [8;11;33]
numel(mat)
v = mat(3,:)
v(v(2))
v(1) = []
reshape(mat,2,6)
```