البرمجة الإجرائية

Lecture No. 4

Matrix

ميكاترونيك-سنة أولى-فصل أول

Dr. Eng. Essa Alghannam
Ph.D. Degree in Mechatronics Engineering

2024

Matrices

A matrix has multiple rows and columns. For example, the matrix

$$\mathbf{M} = \begin{bmatrix} 2 & 4 & 10 \\ 16 & 3 & 7 \\ 8 & 4 & 9 \\ 3 & 12 & 15 \end{bmatrix}$$

has four rows and three columns.

Vectors are special cases of matrices having one row or one column.

Creating Matrix

```
>> y=[11 22 33 ; 55 88 7; 6,9,7]

y =

    11    22    33
    55    88     7
     6     9     7
```

Creating Matrix

Command Window

New to MATLAB? See resources for Getting Started.

```
>> clear all
>> x=[1:6]

x =

     1     2     3     4     5     6

>> y=[1 2; 3 4]

y =

     1     2
     3     4
```

Workspace

Name	Value
x	[1,2,3,4,5,6]
y	[1,2;3,4]

Variables - x

x | y

1x6 double

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

Variables - y

x | y

2x2 double

	1	2	3
1	1	2	
2	3	4	

Creating Matrices from Vectors

Suppose a = [1,3,5] and b = [7,9,11] (row vectors).

Note the difference between the results given by [a b] and [a;b] in the following session:

```
>>c = [a b];
c =
  1  3  5  7  9  11
>>D = [a;b]
D =
  1  3  5
  7  9  11
```

Creating Matrices from Vectors

You need not use symbols to create a new array.

For example, you can type

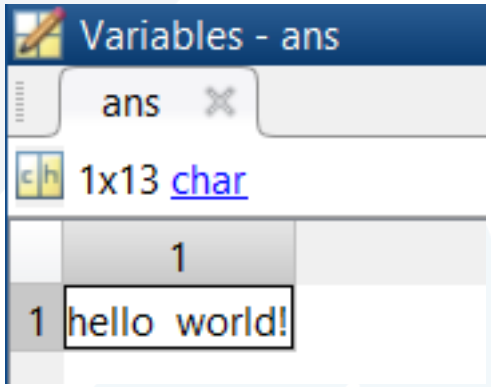
```
>> D = [[1,3,5];[7,9,11]];
```

D =

1	3	5
7	9	11

Creating Char Matrices

```
>> ['hello  world' '!']

ans =

  'hello  world!'
```



Concatenation of Matrices

x = [1 2], y = [4 5], z=[0 0]

A = [x y]

1 2 4 5

B = [x ; y]

1 2

4 5

>> C = [x y z]

C =

1 2 4 5 0 0

>> C = [x y ;z]

Error using vertcat

Dimensions of arrays being concatenated are not consistent.

Automatic Initialization
Generating Matrix from functions

- zeros(M,N) MxN matrix of zeros

```
>> zeros(n)
>> zeros(n,m)
```

يولد مصفوفة مربعة قيم عناصرها 0 أبعادها n*n

يولد مصفوفة قيم عناصرها 0 أبعادها n*m

```
x = zeros(1,3)

x =

     0     0     0
```

- ones(M,N) MxN matrix of ones

```
>> ones(n)
>> ones(n,m)
```

يولد مصفوفة مربعة قيم عناصرها 1 أبعادها n*n

يولد مصفوفة قيم عناصرها 1 أبعادها n*m

```
x = ones(1,3)

x =

     1     1     1
```

Automatic Initialization

Generating Matrix from functions

rand(M,N) MxN matrix of uniformly distributed random numbers on (0,1)

```
x = rand(1,3)
x =
 0.9501  0.2311 0.6068
```

try by yourself

ones(1,10)

eye(3)

zeros(23,1)

rand(1,25)

rand(6)

Automatic Initialization

Generating Matrix from functions

```
>> eye(n)
>> eye(n,m)

>> rand(n)
>> rand(n,m)
>> size(A)
```

يولد مصفوفة مربعة قطرها الأساسي واحد أبعادها n*n

يولد مصفوفة قطرها الأساسي (بإكمالها لمربعة) واحد أبعادها n*m

يولد مصفوفة مربعة قيم عناصرها عشوائية أبعادها n*n

يولد مصفوفة مربعة قيم عناصرها عشوائية أبعادها n*m

يعيد أبعاد المصفوفة

```
>> eye(2,4)

ans =

     1     0     0     0
     0     1     0     0
```

Automatic Initialization

Generating Matrix from functions

```
>> diag(3)

ans =

     3


>> diag([1 2 3])

ans =

     1     0     0
     0     2     0
     0     0     3
```

```
>> x=diag(ans)

x =

     1
     2
     3
```

```
>> diag([1 2 3],2)

ans =

     0     0     1     0     0
     0     0     0     2     0
     0     0     0     0     3
     0     0     0     0     0
     0     0     0     0     0
```

```
>> diag([1 2 3],-2)

ans =

     0     0     0     0     0
     0     0     0     0     0
     1     0     0     0     0
     0     2     0     0     0
     0     0     3     0     0
```

Automatic Initialization

Generating Matrix from functions

```
>> ones(3)-eye(2)
Matrix dimensions must agree.
 >> ones(3)-eye(3)

ans =

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

```
>> pi*(ones(3)-eye(3))

ans =

    0.0000    3.1416    3.1416
    3.1416         0    3.1416
    3.1416    3.1416         0
```

- يفضل تقليل العمليات المنفذة على المصفوفات الكبيرة لتقليل أزمنة تنفيذ البرامج عمليتي الجمع والطرح أسرع بكثير من الضرب والقسمة
- ادرس التوابع التركيبية في الدارت المنطقية

> pi*(ones(3)-eye(3))

ans =

0	3.1416	3.1416
3.1416	0	3.1416
3.1416	3.1416	0

>> a=pi*eye(4)

a =

3.1416	0	0	0
0	3.1416	0	0
0	0	3.1416	0
0	0	0	3.1416

>> diag(a,2)

ans =

0

0

>> diag(a,1)

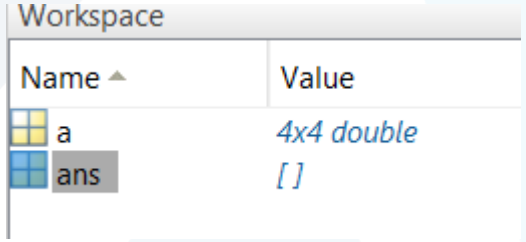
ans =

0

0

0

```
>> diag(a,3)

ans =
    0

>> diag(a,4)

ans =
 0×1 empty double column vector
```



```
>> a=ones(3,4);
>> b=zeros(3,4);
>> c=eye(3,4);
>> abc=[a;b;c]
```

```
abc =

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

```
>> size(abc)

ans =

     9     4
```

يكبر MATLAB المصفوفات اذا دعت الحاجة.

عمليا طريقة التكبير تتم من خلال تشكيل مصفوفة جديدة بالأبعاد الجديدة ثم نسخ محتوى القديمة اليها والمعطيات الجديدة قبل حذف القديمة من الذاكرة.

النتيجة بطء التنفيذ.

يفضا إنشاء مصفوفة بالأبعاد المطلوبة بدلا من انشاء مصفوفة صغيرة ثم توسيعها

```
>> a=eye(3)

a =

     1     0     0
     0     1     0
     0     0     1

>> size(a)

ans =

     3     3
```

```
>> a(5,6)=2e5

a =

     1     0     0     0     0     0
     0     1     0     0     0     0
     0     0     1     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0200000
```

Array Addressing

The colon operator selects individual elements, rows, columns, or "subarrays" of arrays. Here are some examples:

- v(:) represents all the row or column elements of the vector v.
- v(2:5) represents the second through fifth elements; that is v(2), v(3), v(4), v(5).
- A(:,3) denotes all the elements in the third column of the matrix A.
- A(:,2:5) denotes all the elements in the second through fifth columns of A.
- A(2:3,1:3) denotes all the elements in the second and third rows that are also in the first through third columns.

You can use array indices to extract a smaller array from another array. For example, if you first create the array **B**

$$\mathbf{B} = \begin{bmatrix} 2 & 4 & 10 & 13 \\ 16 & 3 & 7 & 18 \\ 8 & 4 & 9 & 25 \\ 3 & 12 & 15 & 17 \end{bmatrix}$$

then type C = B(2:3,1:3), you can produce the following array:

$$\mathbf{C} = \begin{bmatrix} 16 & 3 & 7 \\ 8 & 4 & 9 \end{bmatrix}$$

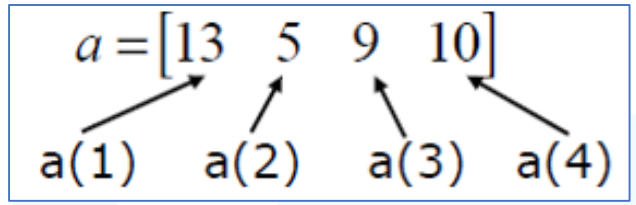
C(-2), C(0), C(1) , C(3)

- The matrix indices begin from 1 (not 0 (as in C))
- The matrix indices must be positive integer

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

>> a(1)
ans =
   1

>> a(2)
ans =
   4
```

```
>> a(6)

ans =

   8

>> a(10)
Index exceeds the number of array elements (9).

>> a(9)

ans =

   9
```

```
>> a(3,3)

ans =

   9
```



>> a(0)

Array indices must be positive integers or logical values.

>> a(-1)

Array indices must be positive integers or logical values.

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

a =

     1     2     3
     4     5     6
     7     8     9

>> a(1:end-1,:)

ans =

     1     2     3
     4     5     6
```

```
>> a(1:end-1)

ans =

     1     4     7     2     5     8     3     6
```

```
>> A=ones(2,4)

A =

     1     1     1     1
     1     1     1     1

>> A(1,1)=0

A =

     0     1     1     1
     1     1     1     1
```

```
>> A(1,4)=0

A =

     0     1     1     0
     1     1     1     1

>> A(2,3)=0

A =

     0     1     1     0
     1     1     0     1
```

```
>> A(1,:)

ans =

    0    1    1    0

>> A(:,4)

ans =

    0
    1
```

```
>> A(:,3)=[3;6]

A =

    0    1    3    0
    1    1    6    1
```

```
>> A(:,3)=[7 9]

A =

    0    1    7    0
    1    1    9    1
```

```
>> S=[ 1 2 3 4 5 6;7 8 9 10 11 12; 13 14 15 16 17 18;
      19 20 21 22 23 24; 25 26 27 28 29 30]

S =

    1    2    3    4    5    6
    7    8    9   10   11   12
   13   14   15   16   17   18
   19   20   21   22   23   24
   25   26   27   28   29   30

>> S(:, 1,3)
Index in position 3 exceeds array bounds (must not exceed 1).
```

```
>> S(:, [1,3])
ans =
    1    3
    7    9
   13   15
   19   21
   25   27

>> S([4:-1:2],[end:-1:3])
ans =
   24   23   22   21
   18   17   16   15
   12   11   10    9
```

```
>> S([4,2],[end,3])
ans =
   24   21
   12    9
```

```
>> S([4,2],[end,3])=zeros(1,2)
Unable to perform assignment because the size of the left side
is 2-by-2 and the size of
the right side is 1-by-2.
```

```
>> S([4,2],[end,3])=zeros(2,2)
S =
    1    2    3    4    5    6
    7    8    0   10   11    0
   13   14   15   16   17   18
   19   20    0   22   23    0
   25   26   27   28   29   30
```

يمكن تعديل أبعاد مصفوفة بـ m سطر و n عمود ليصبح بـ o سطر و p عمود شرط أن يكون m*n = o*p

S =

1	2	3	4	5	6
7	8	0	10	11	0
13	14	15	16	17	18
19	20	0	22	23	0
25	26	27	28	29	30

>> size(S)

ans =

5 6

>> X=reshape(S,10,10)

Error using reshape

To RESHAPE the number of elements must not change.

>> X=reshape(S,6,5)

X =

1	8	15	22	29
7	14	0	28	6
13	20	27	5	0
19	26	4	11	18
25	3	10	17	0
2	0	16	23	30

>> X=reshape(S,3,10)

X =

1	19	8	26	15	4	22	11	29	18
7	25	14	3	0	10	28	17	6	0
13	2	20	0	27	16	5	23	0	30

>> X=reshape(S,10,3)

X =

1	3	5
7	0	11
13	15	17
19	0	23
25	27	29
2	4	6
8	10	0
14	16	18
20	22	0
26	28	30

جَامعة
المَنارة
MANARA UNIVERSITY

Thanks .