

به نام خدا

جزوه جلسه دوم درس برنامه نویسی با متلب

۱- ماتریس ها و آرایه ها

در نرم افزار متلب همه متغیرها بصورت ماتریس هستند. ماتریس یک سطری و یا یک ستونی را بردار می گوئیم. برای تعریف ماتریس از براکت [] استفاده می کنیم بین عناصر یک سطر فاصله می گذاریم و برای رفتن به سطر بعدی از سمیکالن ; استفاده می کنیم.

```
>> A = [ 1 2 3 ; 4 5 6 ; 7 8 9]
A =
    ماتریس
     1     2     3
     4     5     6
     7     8     9
>> v = [ 1 2 3]   بردار یک سطری
v =
     1     2     3
>> w = [ 5 ; 6 ; 7]   بردار یک ستونی
w =
     5
     6
     7
```

برای دسترسی به عناصر یک ماتریس یا بردار از پرانتز استفاده می کنیم. دو نقطه برای تعیین چند سطر و ستون است

```
A =
     1     2     3
     4     5     6
     7     8     9
>> A(2:3,2:3)   سطر دوم و سوم ، ستون دوم تا سوم
ans =
     5     6
     8     9
>> A(1,2)   سطر اول ستون دوم
ans =
     2
>> A(2:3)   سطر دوم تا سوم ستون اول
ans =
     4     7
>> A(3,1:2)   سطر سوم ستونهای یک تا دو
ans =
     7     8
```

نکته $A(:,2:3)$ یعنی همه سطرهای A ستون های ۲ تا ۳

ماتریس ها را می توان در داخل براکت کنار یکدیگر قرار داد و ماتریس های جدیدی ساخت

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

```
>> [ A(:,2) B(2:3,2:3); B(3:4,1) [B(1,3:4);C(2,4:5)] ]
ans =
     2     6     3
     4     5     7
     4     7     8
     5     9     6
سطر اول
سطر دوم
```

به این مثال توجه کنید $B(2:4)$ از ستون یک سطرهای ۲ الی چهار ماتریس را بصورت یک بردار سطری نمایش می دهد.

```
B =
     5     6     7     8
     9     6     3     2
     4     5     7     9
     5     2     1     4
>> B(2:4,1)
ans =
     9
     4
     5
>> B(1,2:4)
ans =
     6     7     8
>> B(2:4)
ans =
     9     4     5
```

```

A =
    11    12    13    14    15
    21    22    23    24    25
    31    32    33    34    35
    41    42    43    44    45
    51    52    53    54    55

>> A(3:4,2:3)
ans =
     32     33
     42     43

>> A(1:3,2)
ans =
     12
     22
     32

>> A([1,3],2:3)
ans =
     12     13
     32     33

>> A(2:3)
ans =
     21     31

>> A(1,2:3)
ans =
     12     13

>> A(2:3,1)
ans =
     21
     31

>> A([1,2,4])
ans =
     11     21

>> A([1,2,4])
ans =
     11     21     41

>> A([1,3],2:3)
ans =
     12     13
     32     33

>> A([1,4],2:3)
ans =
     12     13
     42     43

```

عملیات ماتریس‌ها را می‌توان به دو دسته تقسیم نمود دسته اول عملیات نقطه به نقطه است عملیات دسته دوم عملیات عادی ماتریس‌ها می‌باشد. عملیات نقطه به نقطه روی دو عنصر متناظر عمل می‌کند

```

>> A = [ 1 2 3 4]
A =
     1     2     3     4

>> B = [9 8 7 6]
B =
     9     8     7     6

>> A.*B
ans =
     9    16    21    24

>> a = [ 1 2; 3 4];
>> a
a =
     1     2
     3     4

>> b = [ 7 ; 4]
b =
     7
     4

>> a*b
ans =
    15
    37

>> inv(a)
ans =
   -2.0000    1.0000
    1.5000   -0.5000

>> transpose(a)
ans =
     1     3
     2     4

```

matrix and vector operations/functions

<code>x=[1, 2, 3]</code>	1x3 (Row) vector
<code>x=[1; 2; 3]</code>	3x1 (Column) vector
<code>x=[1, 2; 3, 4]</code>	2x2 matrix
<code>x(2)=4</code>	change index value nr 2
<code>x(:)</code>	All elements of x (same as x)
<code>x(j:end)</code>	j'th to last element of x
<code>x(2:5)</code>	2nd to 5th element of x
<code>x(j,:)</code>	all j row elements
<code>x(:,j)</code>	all j column elements
<code>diag(x)</code>	diagonal elements of x
<code>x.*y</code>	Element by element multiplication
<code>x./y</code>	Element by element division
<code>x+y</code>	Element by element addition
<code>x-y</code>	Element by element subtraction
<code>A^n</code>	normal/Matrix power of A
<code>A.^n</code>	Elementwise power of A
<code>A'</code>	Transpose
<code>inv(A)</code>	Inverse of matrix
<code>size(x)</code>	Rows and Columns
<code>eye(n)</code>	Identity matrix
<code>sort(A)</code>	sorts vector from smallest to largest
<code>eig(A)</code>	Eigenvalues and eigenvectors
<code>numel(A)</code>	number of array elements
<code>x(x>5)=0</code>	change elemnts >5 to 0
<code>x(x>5)</code>	list elements >5
<code>find(A>5)</code>	Indices of elements >5
<code>find(isnan(A))</code>	Indices of NaN elements
<code>[A,B]</code>	concatenates horizontally
<code>[A;B]</code>	concatenates vertically

For functions on matrices, see `bsxfun`, `arrayfun` or `repmat`
*if `arrayfun/bsxfun` is passed a `gpuArray`, it runs on GPU.
*Standard operations: `rank`, `rref`, `kron`, `chol`
*Inverse of matrix `inv(A)` should almost never be used, use RREF through `\` instead: $inv(A)b = A \setminus b$.