

第二章 **MATLAB**基础

● 第一节 MATLAB的操作

一、MATLAB的工作窗口和指令行的操作

1. 若干通用操作命令

命令名称	指令功能
❖ cd、chdir	改变当前工作目录。
❖ <u>clear</u>	清除内存中的所有变量。 <i>clear + 指定变量</i>
❖ <u>clc</u>	擦除Matlab工作窗口中所有显示的内容。
❖ <u>clf</u>	擦除当前窗口中的图形。
❖ dir、ls	列出指定目录下的文件和子目录清单。
❖ disp	在运行中显示变量或文字内容。
❖ echo on/off	控制运行文字指令是否显示。
❖ pack	收集内存碎块以扩大内存空间。

命令名称	指令功能
❖ <u>quit</u>	! <u>关闭并退出Matlab。</u>
❖ <u>type test</u>	! <u>显示所指定的文件test.m的全部内容。</u>
❖ <u>what</u>	! <u>列出当前目录下所有的M文件。</u>
❖ <u>delete test</u>	! <u>删除文件test.m。</u>
❖ <u>which test</u>	! <u>列出指定名字文件test.m所在的目录。</u>
❖ <u>hold on/off</u>	! <u>控制当前图形窗口对象是否被刷新。</u>
❖ <u>exist</u>	! <u>检查指定名字的变量或函数文件的存在性。</u>
❖ <u>who</u>	! <u>列出工作内存中的变量。</u>
❖ <u>whos</u>	! <u>列出工作内存中的变量名称以及细节</u>

第二章 **MATLAB**基础

● 第二节 MATLAB基础

一、**MATLAB**的基本计算功能

1.变量与数值显示格式

变量规则:

- ❖ 每一个变量都具有一个名字
- ❖ 变量在内存中占据一定的空间
- ❖ 变量名必须以字母开头(不能超过19个字符), 之后可以是任意字母、数字, 或下划线(_);
- ❖ 变量区分字母的大小写, 同一名字的大写与小写被视为两个不同的变量;
- ❖ 变量中不能含有标点符号。

Matlab中含有的特殊变量

特殊变量	取 值	特殊变量	取 值
ans	缺省的结构变量名	i,j	
pi	圆周率	flops	浮点运算数
nargin	函数输入变量数目	nargout	函数输出变量数目
realmax	最大的可用正实数	realmin	最小的可用正实数
inf	无穷大 如1/0	NaN	不定量 如0/0
eps	计算机的最小数		

Matlab中含有的特殊变量

- `eps`、`realmax`、`realmin`三个常量具体的数值与运行MATLAB的计算机相关，不同的计算机系统可能具有不同的数值

```
>> eps
```

```
ans =
```

```
2.2204e-016
```

```
>> realmax
```

```
ans =
```

```
1.7977e+308
```

```
>> realmin
```

```
ans =
```

```
2.2251e-308
```

Matlab中含有的特殊变量

- MATLAB的常量数值是可以修改的

```
>> pi=100
```

```
pi =  
    100
```

```
>> clear
```

```
>> pi
```

```
ans =  
    3.1416
```


Matlab中含有的特殊变量

例：最小复数单位的使用

```
>> a=i
```

```
a =
```

```
0 + 1.0000i
```

```
>> i=1
```

```
i =
```

```
1
```

```
>> b=i+j
```

```
b =
```

```
1.0000 + 1.0000i
```

```
>> clear
```

```
>> c=i+j
```

```
c =
```

```
0 + 2.0000i
```

MATLAB的常量是可以赋予性的数值的。一旦被赋予了新的数值，则常量代表的就是新值，而不是原有的值，只有执行clear命令后，常量才会代表原来的值

数值显示格式：双字长浮点数

缺省：整数→用整数显示；

实数→小数点后4位

format命令改变 430/12显示结果

注释

❖ format short 35.833 3

缺省显示

❖ format long 35.833 333 333 333 34

16位

❖ format short e 35.8 3e+01

5位加指数

❖ format long e 3.583 333 333 333 334e+01

16位加指数

❖ format hex 4041eaaaaaaaaaab

16进制

❖ format bank 35.83

2个十进制位

❖ format + +

正、负、零

❖ format rat 215/6

有理数近似

2. MATLAB的基本数学函数

❖ <code>abs(x)</code>	纯量的绝对值或者复数的模
❖ <code>angle(z)</code>	复数 z 的相角
❖ <code>sqrt(x)</code>	开平方
❖ <code>real(z)</code>	复数 z 的实部
❖ <code>imag(z)</code>	复数 z 的虚部
❖ <code>conj(z)</code>	复数 z 的共轭复数
❖ <code>round(x)</code>	四舍五入到最近的整数
❖ <code>fix(x)</code>	无论正负，舍去小数至最近的整数

2. MATLAB的基本数学函数

- ❖ `floor(x)` | 舍去正小数至最近的整数
- ❖ `ceil(x)` | 加入正小数至最近的整数
- ❖ `rat(x)` | 将实数 x 化为分数表示
- ❖ `rats(x)` | 将实数 x 化为多项分数展开
- ❖ `sign(x)` | 符号函数-1; 0; +1。
- ❖ `rem(x,y)` | 求 x 除以 y 的余数
- ❖ `gcd(x,y)` | 整数 x 和 y 的最大公因数
- ❖ `lcm(x,y)` | 整数 x 和 y 的最小公倍数

2. MATLAB的基本数学函数

❖ $\exp(x)$	自然指数
❖ $\text{pow2}(x)$	2的指数
❖ $\log(x)$	以e为底的对数 $\ln(x)$
❖ $\log_2(x)$	以2为底的对数 $\log_2(x)$
❖ $\log_{10}(x)$	以10为底的对数 $\log_{10}(x)$

3. MATLAB的常用三角函数

- | | | | |
|-----------------------|--------|---------------------|---------|
| ❖ $\sin(x)$ | 正弦函数 | ❖ $\sinh(x)$ | 双曲正弦函数 |
| ❖ $\cos(z)$ | 余弦函数 | ❖ $\cosh(z)$ | 双曲余弦函数 |
| ❖ $\tan(x)$ | 正切函数 | ❖ $\tanh(x)$ | 双曲正切函数 |
| ❖ $\text{asin}(x)$ | 反正弦函数 | ❖ $\text{asinh}(x)$ | 反双曲正弦函数 |
| ❖ $\text{acos}(z)$ | 反余弦函数 | ❖ $\text{acosh}(z)$ | 反双曲余弦函数 |
| ❖ $\text{atan}(z)$ | 反正切函数 | ❖ $\text{atanh}(z)$ | 反双曲正切函数 |
| ❖ $\text{atan2}(x,y)$ | 四象限反正切 | | |

二、**MATLAB**矩阵和数组的创建和保存

★ 1. MATLAB的矩阵输入

- ❖ 直接输入创建矩阵
- ❖ 由矩阵编辑器创建和修改矩阵
- ❖ 由函数创建和修改矩阵

❖ 直接输入创建矩阵

整个矩阵以 “[”, “]” 作为首尾，按行方式输入每个元素。

- 同一行中的元素用逗号 “,” 或者用空格符来分隔，且空格个数不限；
- 行与行之间用 “;” 分隔，或按 Enter 键分割。

当矩阵是多维（三维以上），且方括号内的元素是维数较低的矩阵时，会有多重的方括号。

矩阵元素可以是任何 MATLAB 表达式，可以是实数，也可以是复数，复数用 i, j 输入

❖ 由矩阵编辑器创建和修改矩阵

步骤:

- 预先定义一个变量
- 在Workspace视图中选中该变量，双击或点击open selection打开矩阵编辑器。
- 改变维数、元素的值。可以是数值也可以是表达式

❖ 由函数创建和修改矩阵

(1). 特殊矩阵的生成

- | | |
|----------------------------|------------|
| ➤ <code>zeros(m,n)</code> | 零矩阵 |
| ➤ <code>ones(m,n)</code> | 全部元素都为1的矩阵 |
| ➤ <code>eye(m,n)</code> | 单位阵 |
| ➤ <code>rand(m,n)</code> | 0—1分布的随机矩阵 |
| ➤ <code>randn(m,n)</code> | 正态分布的随机矩阵 |
| ➤ <code>compan(A)</code> | 矩阵A的伴随矩阵 |
| ➤ <code>gallery</code> | 测试矩阵 |
| ➤ <code>hankel(m,n)</code> | n维Hankel矩阵 |

(1).特殊矩阵的生成

- `diag(A)` A矩阵的对角阵
- `hilb(n)` n维Hilbert矩阵
- `invhilb(n)` n维逆Hilbert矩阵
- `magic(n)` n维Magic魔方阵 所有行、列相加为16
- `toeplitz(m,n)` toeplitz矩阵
- `eilkinson(n)` n维Wilkinson特征值测试矩阵
- `handamard(n)` n维Handamard矩阵
- `vander(A)` 由矩阵A产生的vandermonde矩阵

P48例: 2.1.2.7~2.1.2.10

例:

例2-14 矩阵生成函数示例

```
>> A=zeros(3)
```

A =

0	0	0
0	0	0
0	0	0

```
>> A=ones(3)
```

A =

1	1	1
1	1	1
1	1	1

```
>> A=eye(3)
```

A =

1	0	0
0	1	0
0	0	1

```
>> A=rand(3)
```

A =

0.9501	0.4860	0.4565
0.2311	0.8913	0.0185
0.6068	0.7621	0.8214

```
>> A=randn(3)
```

A =

-0.4326	0.2877	1.1892
-1.6656	-1.1465	-0.0376
0.1253	1.1909	0.3273

例:

例2-14 矩阵生成函数示例

```
>> A=magic(3)
```

```
A =
```

8	1	6
3	5	7
4	9	2

(15)

```
>> A=magic(4)
```

```
A =
```

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

(34)

例:

例2-15 矩阵生成函数示例

```
>>A=pascal(3)
```

```
ans =
```

1	1	1
1	2	3
1	3	6

```
>>tril(A)
```

```
ans =
```

1	0	0
1	2	0
1	3	6

```
>>diag(A)
```

```
ans =
```

1
2
6

```
>> diag(ans)
```

```
ans =
```

?

例:

例2-15 矩阵生成函数示例

```
>>A=pascal(3)
```

```
ans =
```

1	1	1
1	2	3
1	3	6

```
>>tril(A)
```

```
ans =
```

1	0	0
1	2	0
1	3	6

```
>>diag(A)
```

```
ans =
```

1
2
6

```
>> diag(ans)
```

```
ans =
```

1	0	0
0	2	0
0	0	6

例:

例2-15 矩阵生成函数示例

```
>>A=pascal(3)
```

```
ans =
```

1	1	1
1	2	3
1	3	6

```
>>tril(A)
```

```
ans =
```

1	0	0
1	2	0
1	3	6

```
>>diag(A)
```

```
ans =
```

1
2
6

```
>> diag(ans)
```

```
ans =
```

1	0	0
0	2	0
0	0	6

```
>> diag(ans)
```

```
ans =
```

?

例:

例2-15 矩阵生成函数示例

```
>>A=pascal(3)
```

```
ans =
```

1	1	1
1	2	3
1	3	6

```
>>tril(A)
```

```
ans =
```

1	0	0
1	2	0
1	3	6

```
>>diag(A)
```

```
ans =
```

1
2
6

```
>> diag(ans)
```

```
ans =
```

1	0	0
0	2	0
0	0	6

```
>> diag(ans)
```

```
ans =
```

1
2
6

例:

例2-15 矩阵生成函数示例

```
>>A=pascal(3)
```

```
ans =
```

1	1	1
1	2	3
1	3	6

```
>>tril(A)
```

```
ans =
```

1	0	0
1	2	0
1	3	6

```
>>diag(A)
```

```
ans =
```

1
2
6

```
>> diag(ans)
```

```
ans =
```

1	0	0
0	2	0
0	0	6

```
>> diag(ans)
```

```
ans =
```

1
2
6

diag(*)

*是向量，则执行该指令生成对角矩阵

*是矩阵，则执行该指令获取矩阵的对角线元素

(2). 矩阵的结构变换

❖ $B = \text{rot90}(A)$

A逆时针旋转 90° 得B;

❖ $B = \text{rot90}(A, k)$

A逆时针旋转 $k \times 90^\circ$ 得B;

❖ $B = \text{fliplr}(A)$

B由A左右翻转而得;

❖ $B = \text{flipud}(A)$

B由A上下翻转而得;

❖ $B = \text{reshape}(A, m, n)$

B阵的维数为 $(m \times n)$

$m \times n$ 等于A的列维和行维之积

(3). 矩阵的提取和保存

❖ 保存矩阵:

```
save mymatrix A B
```

Mymatrix是用户自定义的文件名;
系统默认的路径为matlabr11/bin, 可以在前面加上路径而改变。

❖ 提取矩阵:

```
load mymatrix
```

load命令不能指定变量名, 系统仍将A、B作为矩阵的名称。

2. MATLAB的数组的建立和保存

- ❖ 在Matlab中数组可以看做是行向量，即只有一列的矩阵。
- ❖ 前面介绍的所有矩阵的建立和保存方法对于数组都可以适用。
- ❖ Matlab系统还提供了一些创建数组的特殊命令

例1. 创建等差数列

➤ 方法一、 $a=0:0.5:10$

说明:

- ✓ 以“:”间隔起始值(=0)、增量值(=0.5)、终止值(=10);
- ✓ 如果增量值省略不写, 默认增量值为1;

➤ 方法二、利用函数 $x = \text{linspace}(0, 1, 75)$

说明:

- ✓ 利用函数 linspace 以间隔 起始值(=0)、终止值(=1) 和 元素数目(=75);
- ✓ 常用于绘图中区间的分割;

➤ 方法三、从原来的数组创建新的数组

^{1, 2, 3, 4}
a=1:4; ^{1 3 5 7} b=1:2:7;

c=1 3 5 7 1 2 3 4

c=[b,a];

d=1 3 4 .2 8

d=[a(1:2:4), 4 0.2 8];

^{1, 3}
说明:

- ✓ 数组的元素和矩阵一样可以通过下标来访问。

例2. 利用函数logspace创建等比数列

```
logspace(0,2,11)
```

```
ans =
```

```
Columns 1 through 7
```

```
1.0000 1.5849 2.5119 3.9811 6.3096 10.0000 15.8489
```

```
Columns 8 through 11
```

```
25.1189 39.8107 63.0957 100.0000
```

说明：

- ✓ 该函数产生一个起点为 10^0 、终点为 10^2 、包含11个数据的等比数列。

3. MATLAB的矩阵运算和数组运算

□ 矩阵运算

❖ + 加

$$A+B$$

❖ - 减

$$A-B$$

❖ * 乘

$$A*B$$

❖ ^ 幂

$$A^n$$

❖ \ 左除

$$A \setminus B$$

❖ / 右除

$$B/A$$

❖ ' 转置

$$A'$$

→ A, B 具有相同的行列数。

→ A 的列数和 B 的行数相同

→ A 是方阵。

$$A \setminus B = (A^{-1}) * B$$

$$B \setminus A = (A' / B')'$$

矩阵求逆函数 $\text{inv}(A)$

❖ 矩阵的乘方 A^p \Rightarrow A 是方阵。

➤ p 是整数时

$$A^p = A * A * A * \dots * A * A \quad p > 0;$$

$$A^p = (A * A * A * \dots * A * A)^{-1} \quad p < 0;$$

$$A^p = \text{与 } A \text{ 同维的单位阵} \quad p = 0;$$

➤ p 是非整数时

若 A 可以被分解成 $A = WDW^{-1}$, D 为对角阵

$$\text{则 } A^p = WD^pW^{-1}$$

说明:

- ✓ 如果 A 的特征值有重根, 以上指令不适合。
- ✓ 有些矩阵的非整数次方有多个解, Matlab 只是给出一个解。
- ✓ Matlab 有一个专门计算矩阵平方根的函数 $\text{sqrtm}(A)$ 算法同上。

3. MATLAB的矩阵运算和数组运算

□ 矩阵运算

1. 矩阵加、减运算 ($A+B$ 、 $A-B$)

规则:

- 相加、减的两矩阵必须有相同的行和列，两矩阵对应元素相加减。
- **MATLAB**允许参与运算的两矩阵之一是标量，标量与矩阵的所有元素分别进行加减操作。

例: $A=[1\ 2\ 3;\ 4\ 5\ 6]$ $B=[3\ 4\ 5;\ 7\ 8\ 9]$ $C=3$

$$A+B=[4\ 6\ 8;\ 11\ 13\ 15]$$

$$A+C=[4\ 5\ 6;\ 7\ 8\ 9]$$

$$B+C=[6\ 7\ 8;\ 10\ 11\ 12]$$

3. MATLAB的矩阵运算和数组运算

□ 矩阵运算

2. 矩阵乘运算

- $A*B$: A 矩阵的列数必须等于 B 矩阵的行数。
- $s*A$ 或 $A*s$: 标量可与任何矩阵相乘, 标量 s 分别与矩阵 A 每个元素相乘。

例: $A = [1\ 2\ 3; 4\ 5\ 6; 7\ 8\ 0]$; $B = [1; 2; 3]$;

$C = A*B$

$C = 14$

32

23

$D = [-1; 0; 2]$;

$F = \pi*D$

$F = -3.1416$

0

6.2832

3. MATLAB的矩阵运算和数组运算

□ 矩阵运算

3. 矩阵除运算及线性方程组的解

在线性代数中没有矩阵的除运算，只有矩阵逆的运算，在MATLAB中有两种矩阵除运算。

A/B — 矩阵右除，相当于 A*inv(B) $A \cdot B^{-1}$

A\B — 矩阵左除，相当于 inv(A)*B $A^{-1} \cdot B$

因此， $x = A \setminus B$ 是线性方程组 $Ax=B$ 的解。

3. MATLAB的矩阵运算和数组运算

□ 矩阵运算

3. 矩阵除运算及线性方程组的解

例：求解方程组
$$\begin{cases} 3x_1 + x_2 - x_3 = 3.6 \\ x_1 + 2x_2 + 4x_3 = 2.1 \\ -x_1 + 4x_2 + 5x_3 = -1.4 \end{cases}$$

```
>> A = [ 3 1 -1 ; 1 2 4 ; -1 4 5 ];
```

```
>> B = [ 3.6 ; 2.1 ; -1.4 ];
```

```
>> x = A\B
```

```
x =
```

```
1.4818
```

```
-0.4606
```

```
0.3848
```


3. MATLAB的矩阵运算和数组运算

□ 矩阵运算

4. 矩阵乘方

A^n —— A自乘n次幂

方阵

>1的整数

例

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

```
>>a^2
```

```
ans = 30 36 42
```

```
66 81 96
```

```
102 126 150
```


3. MATLAB的矩阵运算和数组运算

□数组运算

- ❖ $+. \text{加法}$ $A+B$
 - ❖ $.- \text{减法}$ $A-B$
 - ❖ $.* \text{乘法}$ $A*B$
 - ❖ $.^ \text{幂}$ A^n
 - ❖ $.\backslash \text{左除}$ $A\backslash B$
 - ❖ $./ \text{右除}$ B/A
 - ❖ $.' \text{共轭}$ A'
- ❖ 数组乘方 \rightarrow 所得结果与A同维
- $A.^p$ 数组A对应元素的p次方
- $p.^A$ p的数组A对应元素次方
- 作业: $A=[3,6,7;9,2,5;1,6,3]$,
 $p=0.4$ 。
求 $A.^p$ 以及 $p.^A$

说明: 无论那种运算都是对元素逐个进行的。

3. MATLAB的矩阵运算和数组运算

□ 数组运算

- 数组运算指元素对元素的算术运算，与通常意义上的由符号表示的线性代数矩阵运算不同。

(1) 数组加减 (+, -) 运算

规则：

- 相加、减的两数组必须有相同的行和列，两数组对应元素相加减。
- **MATLAB**允许参与运算的两数组之一是标量，标量与数组的所有元素分别进行加减操作

$A+B$

$A-B$

} 与矩阵加减运算等效，数组之一也可
为标量。

3. MATLAB的矩阵运算和数组运算

□ 数组运算

(2) 数组乘 (.*) 运算

■ $A.*B$

A , B 两数组必须有相同的行和列，两数组相应元素相乘。

● $s.*A$ 或 $A.*s$

标量与数组相乘，标量 s 分别与数组 A 每个元素相乘，与 $s*A$ 或 $A*s$ 相同。

3. MATLAB的矩阵运算和数组运算

□ 数组运算

例16: >>A = [1 2 3; 4 5 6; 7 8 9];
>>B = [2 4 6; 1 3 5; 7 9 10];
>>A.*B
ans =
2 8 18
4 15 30
49 72 90

```
>>A = [ 1 2 3; 4 5 6; 7 8 9 ];  
>>B = [ 2 4 6; 1 3 5; 7 9 10 ];  
>>A*B  
ans =  
25 37 46  
55 85 109  
85 133 172
```

3. MATLAB的矩阵运算和数组运算

□ 数组运算

(3) 数组除 (./, .\) 运算

$C=A./B$ —— 数组右除

$$C(i,j) = A(i,j)/B(i,j)$$

$C=A.\backslash B$ —— 数组左除

$$C(i,j) = B(i,j)/A(i,j)$$

$$A./B = B.\backslash A$$

$A./s = s.\backslash A$ — A 的元素分别被标量 s 除

$s./A = A.\backslash s$ — 标量 s 分别被 A 的元素除

3. MATLAB的矩阵运算和数组运算

□ 数组运算

例: >>A = [1 2 3];

>>B = [4 5 6];

>>C1 = A./B

C1 = 0.2500 0.4000 0.5000

>>C2 = B.\A

C2 = 0.2500 0.4000 0.5000

>>C3 = A.\B

C3 = 4.0000 2.5000 2.0000

>> A = [1 2 3]; B = [4 5 6];

>> A/B

ans =

0.4156

>> A\B

ans =

0

0

0

0

0

0

1.3333

1.6667

2.0000

3. MATLAB的矩阵运算和数组运算

□ 数组运算

(4) 数组乘方 (.[^])

$A.^n$ —— A 的每个元素自乘 n 次

$A.^p$ —— 对 A 各元素分别求非整数幂

$p.^A$ —— 以 p 为底,分别以 A 的元素为指数求幂值

$C = A.^B$ —— 元素对元素的幂

$C(i,j) = A(i,j) .^ B(i,j)$

3. MATLAB的矩阵运算和数组运算

□ 数组运算

(5) 数组转置 (.')

例: >> A=[1 3 5;2 4 6]

A =

1	3	5
2	4	6

>> A'

ans =

1	2
3	4
5	6

>> A.'

ans =

1	2
3	4
5	6

结论: 对于实数矩阵, 矩阵转置和数组转置的计算结果是一致的。

3. MATLAB的矩阵运算和数组运算

□ 数组运算

例: >> A=A*i

A =

0 + 1.0000i

0 + 3.0000i

0 + 5.0000i

0 + 2.0000i

0 + 4.0000i

0 + 6.0000i

>> A'

ans =

0 - 1.0000i

0 - 2.0000i

0 - 3.0000i

0 - 4.0000i

0 - 5.0000i

0 - 6.0000i

>> A.'

ans =

0 + 1.0000i

0 + 2.0000i

0 + 3.0000i

0 + 4.0000i

0 + 5.0000i

0 + 6.0000i

结论:

对于复数矩阵, 矩阵转置
和数组转置的计算结果不一致。

{ 矩阵转置运算——共轭转置
数组转置运算——非共轭转置

三、**MATLAB**矩阵和数组的索引

❖ 访问和操作向量或矩阵元素的方法——利用矩阵或向量元素的索引完成相应的操作。

注意：**MATLAB**的矩阵或数组的索引起始数值为1

❖ 介绍的内容

- 向量元素的访问
- 矩阵元素的访问

三、**MATLAB**矩阵和数组的索引

1 向量元素的访问

- 访问向量的元素只要使用相应元素的索引即可
- 访问向量元素的结果是创建新的向量
- 访问向量的元素直接给出元素在向量中的序号
 - 元素的序号可以是单一的整数
 - 元素的序号可以是元素序号组成的向量
- 关键字end在访问向量元素时，表示向量中最后一个元素的序号
- 访问向量元素时，序号的数值必须介于1~end之间
- 可以通过访问元素的方法，对具体的元素赋值

三、MATLAB矩阵和数组的索引

1 向量元素的访问

例2-11 B=[3 2 7 4 9 6 1 8 0 5]

访问向量中的元素

```
>> B(3)
```

```
ans =
```

```
7
```

```
>> B([1 3 7])
```

```
ans =
```

```
3    7    1
```

```
>> B([1:3:5])
```

```
ans =
```

```
3    4
```

```
>> B([end-3:end])
```

```
ans =
```

```
1    8    0
```

```
>> B([1:5,5:-1:1])
```

```
ans =
```

```
?
```

```
>> B([1:5;5:-1:1])
```

```
ans =
```

```
?
```

7 : 10 \Rightarrow 7 8 9 10

B(1 5 2 3 4 5 5 4 3 2 1)

三、**MATLAB**矩阵和数组的索引

1 向量元素的访问

例2-11 B=[3 2 7 4 9 6 1 8 0 5]

访问向量中的元素

```
>> B(3)
```

```
ans =
```

```
7
```

```
>> B([1 3 7])
```

```
ans =
```

```
3    7    1
```

```
>> B([1:3:5])
```

```
ans =
```

```
3    4
```

```
>> B([end-3:end])
```

```
ans =
```

```
1    8    0    5
```

```
>> B([1:5,5:-1:1])
```

```
ans =
```

```
3    2    7    4    9    9    4    7    2    3
```

```
>> B([1:5;5:-1:1])
```

```
ans =
```

```
?
```

三、**MATLAB**矩阵和数组的索引

1 向量元素的访问

例2-11 B=[3 2 7 4 9 6 1 8 0 5]

访问向量中的元素

```
>> B(3)
```

```
ans =
```

```
7
```

```
>> B([1 3 7])
```

```
ans =
```

```
3    7    1
```

```
>> B([1:3:5])
```

```
ans =
```

```
3    4
```

```
>> B([end-3:end])
```

```
ans =
```

```
1    8    0    5
```

```
>> B([1:5,5:-1:1])
```

```
ans =
```

```
3    2    7    4    9    9    4    7    2    3
```

```
>> B([1:5;5:-1:1])
```

```
ans =
```

```
3    2    7    4    9
```

```
9    4    7    2    3
```

三、**MATLAB**矩阵和数组的索引

1 向量元素的访问

例2-12 对向量的元素进行赋值

```
>> B(3)=-3
```

```
B =
```

3	2	-3	4	9	6	1	8	0	5
---	---	----	---	---	---	---	---	---	---

```
>> B(15)= -15
```

```
B =
```

Columns 1 through 13

3	2	-3	4	9	6	1	8	0	5	0	0	0
---	---	----	---	---	---	---	---	---	---	---	---	---

Columns 14 through 15

0	-15
---	-----

三、**MATLAB**矩阵和数组的索引

2 矩阵元素的访问

- 访问矩阵的元素需要使用矩阵元素的索引
 - 使用矩阵元素的行列全下标形式 $A(i,j)$
 - 使用全下标形式访问矩阵元素的方法简单、直接，同线性代数的矩阵元素的概念一一对应
 - 使用矩阵元素的单下标形式 $A(k)$
 - 矩阵元素的单下标是矩阵元素在内存中存储的序列号，一般地，同一个矩阵的元素在连续的内存单元中（元素的排列以列元素优先）

全下标形式的矩阵标识和子矩阵

矩阵的子阵可以通过标量、向量、冒号的标识来引用和赋值

❖ 子阵的序号向量标识方式 $A(v,w)$

v,w 可以是任何排列的向量。 v,w 中任一个可以是冒号“:”，它表示全部行（在 v 的位置）或列（在 w 的位置）。

例:

(1)产生一个五阶魔方阵B;

$$B_1 = B([1,2], [1,3,5])$$

(2)提取B阵的第1行, 第2行的第1,3,5个元素;

$$B_2 = B([3,1], :)$$

(3)提取B阵的第三行和第一行全部元素;

$$B([1,3], [2,4]) = \begin{cases} [0,0; 0,0] \\ \text{zeros}(0) \\ \text{zeros}[0,0; 0,0] \end{cases}$$

(4)使得B阵的第一行和第三行第2, 4个元素为0;

$$L = (B(1, :) < 5)$$

(5)标出B阵的第一行中小于5的元素;

$$B_6 = B(1, L)$$

(6)获得B阵的第一行中小于5的子向量;

三、**MATLAB**矩阵和数组的索引

2 矩阵元素的访问

例13 A=

4	1	10	5	1	9	6	13	2	17
8	2	2	6	9		4	14	7	18
7	3	5	7	7		1	15	5	19
0	4	3	8	4		5	16	8	20

A(1,2)
A(5)

A(1:4,5)
A(:,5)
A(:,end)
A(17:20)'

A(2:4,2:3)
A([2 3 4],[2 3])

三、**MATLAB**矩阵和数组的索引

2 矩阵元素的访问

- 矩阵元素的单下标与全下标之间的转换关系

- 以 $m \times n$ 的矩阵为例

第 i 行第 j 列的元素全下标转换为单下标

$$l = (j-1) \times m + i$$

例: $A(1,2) \rightarrow A(5)$

$$m=4, n=5, i=1, j=2$$

$$l = (j-1) \times m + i = (2-1) \times 4 + 1 = 5$$

- MATLAB提供的两个函数

- sub2ind: 根据全下标计算单下标

- ind2sub: 根据单下标计算全下标

三、**MATLAB**矩阵和数组的索引

2 矩阵元素的访问

例: >> A=[4 10 1 6 2;8 2 9 4 7;7 5 7 1 5;0 3 4 5 8]

A =

4	10	1	6	2
8	2	9	4	7
7	5	7	1	5
0	3	4	5	8

>> sub2ind(size(A),2,2)

ans =

6

>> [i,j]=ind2sub(size(A),7)

i =

3

j =

2

三、MATLAB矩阵和数组的索引

2 矩阵元素的访问

使用索引访问矩阵元素的方法

矩阵元素的访问	说 明
$A(i,j)$	访问矩阵A的第i行第j列上的元素，其中i和j为标量
$A(I,J)$	访问由向量I和J指定的矩阵A中的元素
$A(i,:)$	访问矩阵A中第i行的所有元素
$A(:,j)$	访问矩阵A中第j列的所有元素
$A(:)$	访问矩阵A中的所有元素，将矩阵看成一个向量
$A(I)$	使用单下标的方式访问矩阵元素，其中I为标量
$A(L)$	访问由向量L指定的矩阵A的元素，向量L中的元素为矩阵元素的单下标数值

在索引矩阵或数组的元素时，若直接用冒号运算符且不给任何的参数，则表示选择该行或列，或维中的所有元素

三、MATLAB矩阵和数组的索引

2 矩阵元素的访问

例：用不同的方法访问矩阵的元素

```
>>A=1:25
```

```
>>A=reshape(A,5,5)
```

A=

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

```
>>A(3,1)或A(3)
```

ans =

3

```
>> A(3,:)
```

ans =

3 8 13 18 23

```
>> A(:,4)
```

ans =

16
17
18
19
20

```
>> A(end,:)
```

ans =

5 10 15 20 25

```
>>I=[1 3 5];J=[2 4];
```

```
>>A(I,J)
```

```
>>A([1 3 5],[2 4])
```

ans =

6 16
8 18
10 20

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

四、**MATLAB**矩阵和数组的操作函数

用于矩阵（数组）操作的常用函数

函数	说明
size	获取矩阵的行、列数，对于多维数组，获取数组的各个维的尺寸
length	获取向量长度，若输入参数为矩阵或多维数组，则返回各个维尺寸的最大值
ndims	获取矩阵或多维数组的维数
numel	获取矩阵或数组的元素个数
disp	显示矩阵或者字符串内容
cat	合并不同的矩阵或数组
reshape	保持矩阵元素的个数不变，修改矩阵的行数和列数
repmat	复制矩阵元素并扩展矩阵
fliplr	交换矩阵左右对称位置上的元素
flipud	交换矩阵上下对称位置上的元素
flipdim	获取指定的方向翻转交换矩阵元素
find	获取矩阵或者数组中非零元素的索引

四、**MATLAB**矩阵和数组的操作函数

例2-18: reshape函数使用示例

```
>> A=1:8
```

```
A =
```

```
    1    2    3    4    5    6    7    8
```

```
>> B=reshape(A,2,4) ← 将矩阵A改成2行4列，也可写成  
B =                               B=reshape(1:8,2,4)
```

```
    1    3    5    7
```

```
    2    4    6    8
```

```
>> C=reshape(A,3,3)
```

```
??? Error using ==> reshape
```

```
To RESHAPE the number of elements must not change.
```

不能改变矩阵包含元素的个数