

1.5 矩阵元素的引用

- 矩阵元素的引用方式
- 利用冒号表达式获得子矩阵
- 利用空矩阵删除矩阵的元素
- 改变矩阵的形状

1. 矩阵元素的引用方式

(1) 通过下标来引用矩阵的元素

$A(3, 2)$ 表示A矩阵第3行第2列的元素。

```
>> A(3, 2)=200
```



1. 矩阵元素的引用方式

(1) 通过下标来引用矩阵的元素

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

```
>> A(4, 5)=10
```

```
A =
```

1	2	3	0	0
4	5	6	0	0
0	0	0	0	0
0	0	0	0	10

注意：如果给出的行下标或列下标大于原来矩阵的行数和列数，那么MATLAB将自动扩展原来的矩阵，并将扩展后没有赋值的矩阵元素置为0。

(2) 通过序号来引用

- ❑ 在MATLAB中，矩阵元素按列存储，即首先存储矩阵的第一列元素，然后存储第二列元素， \dots ，一直到矩阵的最后一列元素。
- ❑ 矩阵元素的序号就是矩阵元素在内存中的排列顺序。

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

```
A =
```

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

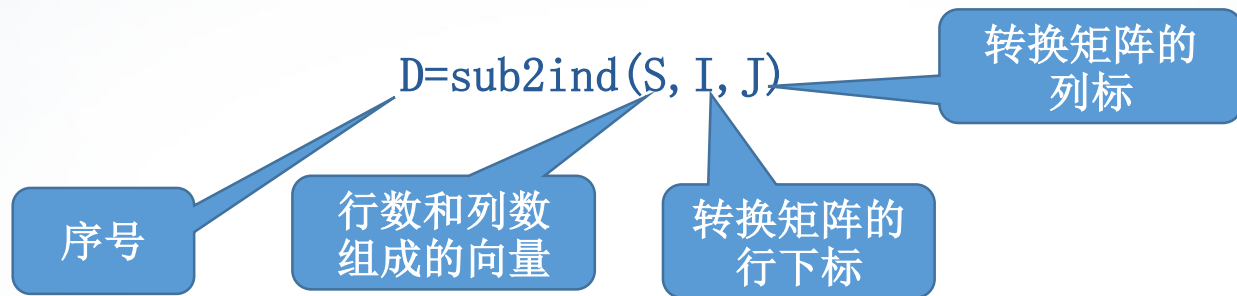
```
>> A(3)
```

```
ans =
```

```
     2
```

序号与下标是一一对应的，以 $m \times n$ 矩阵A为例，矩阵元素 $A(i, j)$ 的序号为 $(j-1) \times m + i$

sub2ind函数：将矩阵中指定元素的行、列下标转换成存储的序号。调用格式为：





sub2ind函数举例。

```
>> A=[1:3;4:6]
```

```
A =
```

```
     1     2     3
```

```
     4     5     6
```

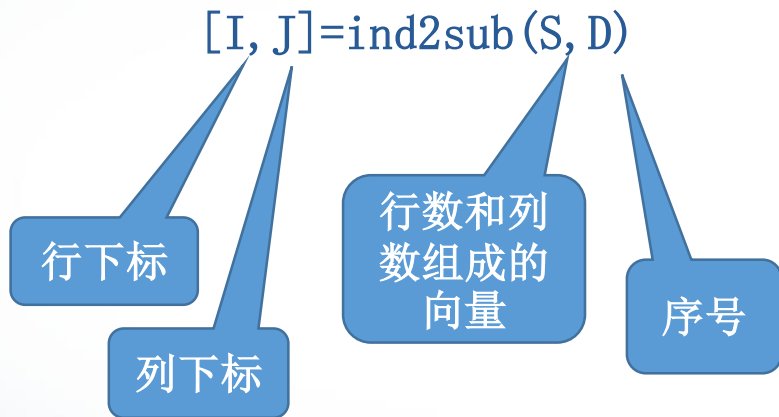
```
>> D=sub2ind(size(A), [1,2;2,2], [1,1;3,2])
```

```
D =
```

```
     1     2
```

```
     6     4
```

ind2sub函数：将把矩阵元素的序号转换成对应的下标，其调用格式为：



ind2sub函数举例。

```
>> [I,J]=ind2sub([3,3],[1,3,5])
```

```
I =
```

```
     1     3     2
```

```
J =
```

```
     1     1     2
```

2. 利用冒号表达式获得子矩阵

子矩阵是指由矩阵中的一部分元素构成的矩阵。

$A(i,:)$	第 <i>i</i> 行的全部元素
$A(:,j)$	第 <i>j</i> 列的全部元素
$A(i:i+m,k:k+m)$	第 <i>i</i> ~ <i>i</i> + <i>m</i> 行内且在第 <i>k</i> ~ <i>k</i> + <i>m</i> 列中的所有元素
$A(i:i+m,:)$	第 <i>i</i> ~ <i>i</i> + <i>m</i> 行的全部元素



```
>> A=[1, 2, 3, 4, 5;6, 7, 8, 9, 10;11, 12, 13, 14, 15]
```

```
A =
```

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

```
>> A(1:2, :)
```

```
ans =
```

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

```
>> A(2:3, 1:2:5)
```

```
ans =
```

```
     6     8    10
    11    13    15
```

end运算符：表示某一维的末尾元素下标。

```
>>A=[1, 2, 3, 4, 5;6, 7, 8, 9, 10;11, 12, 13, 14, 15;16, 17, 18, 19, 20];
```

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

```
ans =
```

```
    16    17    18    19    20
```

```
>> A([1, 4], 3:end)
```

```
ans =
```

```
     3     4     5  
    18    19    20
```

3. 利用空矩阵删除矩阵的元素

空矩阵是指没有任何元素的矩阵。

```
>> x=[]
```

```
x =
```

```
 []
```

X是一个空矩阵。



```
>> A=[1, 2, 3, 0, 0;7, 0, 9, 2, 6;1, 4, -1, 1, 8]
```

```
A =
```

1	2	3	0	0
7	0	9	2	6
1	4	-1	1	8

```
>> A(:, [2, 4])=[]
```

```
A =
```

1	3	0
7	9	6
1	-1	8

4. 改变矩阵的形状

`reshape(A, m, n)`：在矩阵总元素保持不变的前提下，将矩阵A重新排成 $m \times n$ 的二维矩阵。

注意：`reshape`函数只是改变原矩阵的行数和列数，但并不改变原矩阵元素个数及其存储顺序。

```
>> x=[23, 45, 65, 34, 65, 34, 98, 45, 78, 65, 43, 76];
```

```
>> y=reshape(x, 3, 4)
```

y

23	34	98	65
45	65	45	43
65	34	78	76

$A(:)$: 将矩阵A的每一列元素堆叠起来, 成为一个列向量。

```
>> A = [-45, 65, 71; 27, 35, 91]
```

```
A =
```

```
    -45     65     71  
     27     35     91
```

```
>> B=A(:)
```

```
B =
```

```
    -45  
     27  
     65  
     35  
     71  
     91
```

$A(:)$ 等价于 $\text{reshape}(A, 6, 1)$ 。