



# Chapter 4: Loops (Cont.)

---

Yunong Zhang (张雨浓)

Email: [zhynong@mail.sysu.edu.cn](mailto:zhynong@mail.sysu.edu.cn)



# Logical Arrays and Vectorization

- Relational Operators

```
>>a=[1 0  
      -2 1];
```

```
>> b=0;
```

```
>>d=(a>b)  
d=[1 0;  
   0 1]
```

```
>>e=(a>=-3)  
e= [1 0;  
    1 1]
```

created by relational and logic operators

named: **Logical array**

type: **logical**

How many bytes are necessary to store d or e?

# Which one is right?

>> whos

Name	Size	Bytes	Class
a	2x2	32	double array
d	2x2	4	logical array

Grand total is 8 elements using 36 bytes

Command Window

```
>> a=[1 0  
      -2 1];  
b=0;  
d=(a>b)
```

```
d =
```

```
     1     0  
     0     1
```

```
>> whos
```

Name	Size	Bytes	Class
a	2x2	32	double array
b	1x1	8	double array
d	2x2	32	double array (logical)

```
Grand total is 9 elements using 72 bytes
```

```
>>
```



# Additional Inspiration

---

## 《三字经●弟子规》之第四篇《信》第二段

见未真，勿轻言；

知未的，勿轻传；

事非宜，勿轻诺，

苟轻诺，进退错。



## Logical Arrays and Vectorization (Cont.)

- **logical** function

非0即有归为1

Values other than 0 or 1 converted to logical 1

```
>>c=logical(a)
```

```
c =
```

```
1    0
1    1
```

```
>> logical(a)
```

```
ans =
```

```
1    0
-2    1
```

```
>> help logical
```

**Which one is right?**



## Logical Arrays and Vectorization (Cont.)

---

>> help logical

**Which version?**

LOGICAL Convert numeric values to logical.

LOGICAL(X) returns an array that can be used for logical indexing or logical tests. Logical arrays are also created by the relational operators (`==`, `<`, `>`, `~`, etc.) and functions like ANY, ALL, ISNAN, ISINF, and ISFINITE.

A(B), where B is a logical array, returns the values of A at the indices where the real part of B is nonzero (B must be the same size as A). A(B) is the same as A(FIND(B)).

Most arithmetic operations remove the logicalness from an array. For example, adding zero to a logical array removes its logical characteristic. `A = +A` is the easiest way to convert a logical double array, A, to a strictly numeric double array.

# Logical Arrays and Vectorization (Cont.)



The image shows the MATLAB Help Navigator window and an 'About MATLAB' dialog box. The Help Navigator is on the left, displaying a search index for 'logical array [2]'. The 'About MATLAB' dialog box is in the foreground, showing the MATLAB logo and version information.

**Help Navigator**

Product filter: ☒ All ☐ Selected

Contents Index Search Favorites

Search index for:

logical array [2]

L	Product
logical array [1]	
converting numeric arr...	MATLAB - R
detecting	MATLAB - R
logical array [2]	MATLAB - E
logical arrays [1]	Image Proc
logical arrays [2]	Image Proc
...	MATLAB - R

**About MATLAB**

**MATLAB®**

The Language of Technical Computing

Version 6.1.0.450 Release 12.1

May 18, 2001

External Interfaces/API: Calling C and Fortran Programs from MATLAB

Find in page:

Logical Arrays

Any noncomplex numeric or sparse array can be flagged as logical. The storage for a logical array is the same as the storage for a nonlogical array.

Types

## Logical Arrays and Vectorization (Cont.)

```
>> whos
```

Name	Size	Bytes	Class
a	2x2	32	double array
b	2x2	4	logical array
c	2x2	4	logical array

Grand total is 12 elements using 40 bytes

```
>> c=logical(a)
```

```
c =  
     1     0  
    -2     1
```

```
>> whos
```

Name	Size	Bytes	Class
a	2x2	32	double array
ans	2x2	32	double array (logical)
b	1x1	8	double array
c	2x2	32	double array (logical)
d	2x2	32	double array (logical)

Grand total is 17 elements using 136 bytes

```
>> ver
```

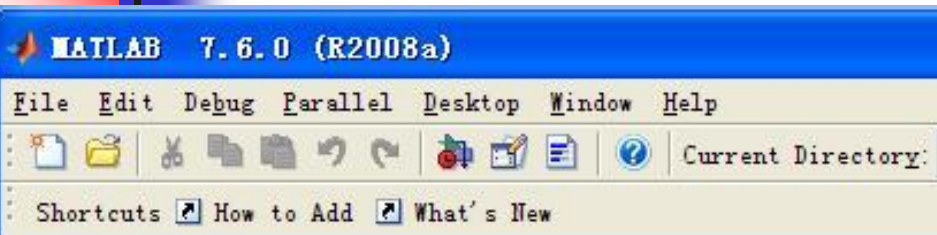
---

MATLAB Version 6.1.0.450 (R12.1) on PCWIN

**Which one is right?**  
**Which is earlier?**



# MATLAB 7.6 version:



```
>> b=0; d=(a>b)
```

```
d =
```

```
    1    0
    0    1
```

```
>> whos
```

Name	Size	Bytes	Class	Attributes
a	2x2	32	double	
b	1x1	8	double	
d	2x2	4	logical	
e	2x2	4	logical	

```
>> c=logical(a)
```

```
c =
```

```
    1    0
    1    1
```

```
>> whos
```

Name	Size	Bytes	Class	Attributes
a	2x2	32	double	
b	1x1	8	double	
c	2x2	4	logical	
d	2x2	4	logical	
e	2x2	4	logical	

```
>> e=(a>=-3)
```

```
e =
```

```
    1    1
    1    1
```

# Significance of Logical Arrays

- Select a portion of a matrix

Array =

1	2	3
4	5	6

Array(:,1)

Array =

1	2	3
4	5	6

Array(1,:)


- How to select elements whose values are  $\geq a0$ ?



## Significance of Logical Arrays (Cont.)

---

**Example (i):** Assume that  $a$  is an  $m \times n$  matrix.

```
a0=4;
```

```
for i=1:m
```

```
    for j=1:n
```

```
        if (a(i,j))>=a0)
```

```
            a(i,j)=a(i,j)^2;
```

```
        end
```

```
    end
```

```
end
```



## Significance of Logical Arrays (Cont.)

---

```
b=(a>=a0);
```

```
a(b) = a(b).^2;
```

**Function:** Select all elements for which the logical array *b* is nonzero and leave all the other elements in the array unchanged.



# Significance of Logical Arrays (Cont.)

## ■ Example (ii)

```
a=[1 2 3;4 5 6;7 8 10];
for i=1:size(a,1)
    for j=1:size(a,2)
        if a(i,j)>5
            a(i,j)=sqrt(a(i,j));
        else
            a(i,j)=a(i,j)^2;
        end
    end
end
end
```

**>> help size**

SIZE Size of matrix.

D = SIZE(X), for M-by-N matrix X, returns the two-element row vector D = [M, N] containing the numbers of rows and columns in the matrix. For N-D arrays, SIZE(X) returns a 1-by-N vector of dimension lengths.

[M,N] = SIZE(X) returns the numbers of rows and columns in separate output variables. [M1,M2,M3,...,MN] = SIZE(X) returns the lengths of the first N dimensions of X.

**M = SIZE(X,DIM) returns the length of the dimension specified by the scalar DIM. For example, SIZE(X,1) returns the number of rows.**



# Significance of Logical Arrays (Cont.)

---

a =

1.0000	4.0000	9.0000
16.0000	25.0000	2.4495
2.6458	2.8284	3.1623



## Significance of Logical Arrays (Cont.)

```
b=(a>5)
```

```
a(b)=sqrt(a(b));
```

```
a(~b)=a(~b).^2;
```

Wonderful! Brilliant! Marvelous! Ingenious!

b =

0	0	0
0	0	1
1	1	1

a =

1.0000	4.0000	9.0000
16.0000	25.0000	2.4495
2.6458	2.8284	3.1623

It is much **faster** than the way of using for-loop-branch.



# Sincere Thanks!

---

- Using this group of PPTs, please read
- [1] Yunong Zhang, Weimu Ma, Xiao-Dong Li, Hong-Zhou Tan, Ke Chen, MATLAB Simulink modeling and simulation of LVI-based primal-dual neural network for solving linear and quadratic programs, Neurocomputing 72 (2009) 1679-1687
- [2] Yunong Zhang, Chenfu Yi, Weimu Ma, Simulation and verification of Zhang neural network for online time-varying matrix inversion, Simulation Modelling Practice and Theory 17 (2009) 1603-1617