

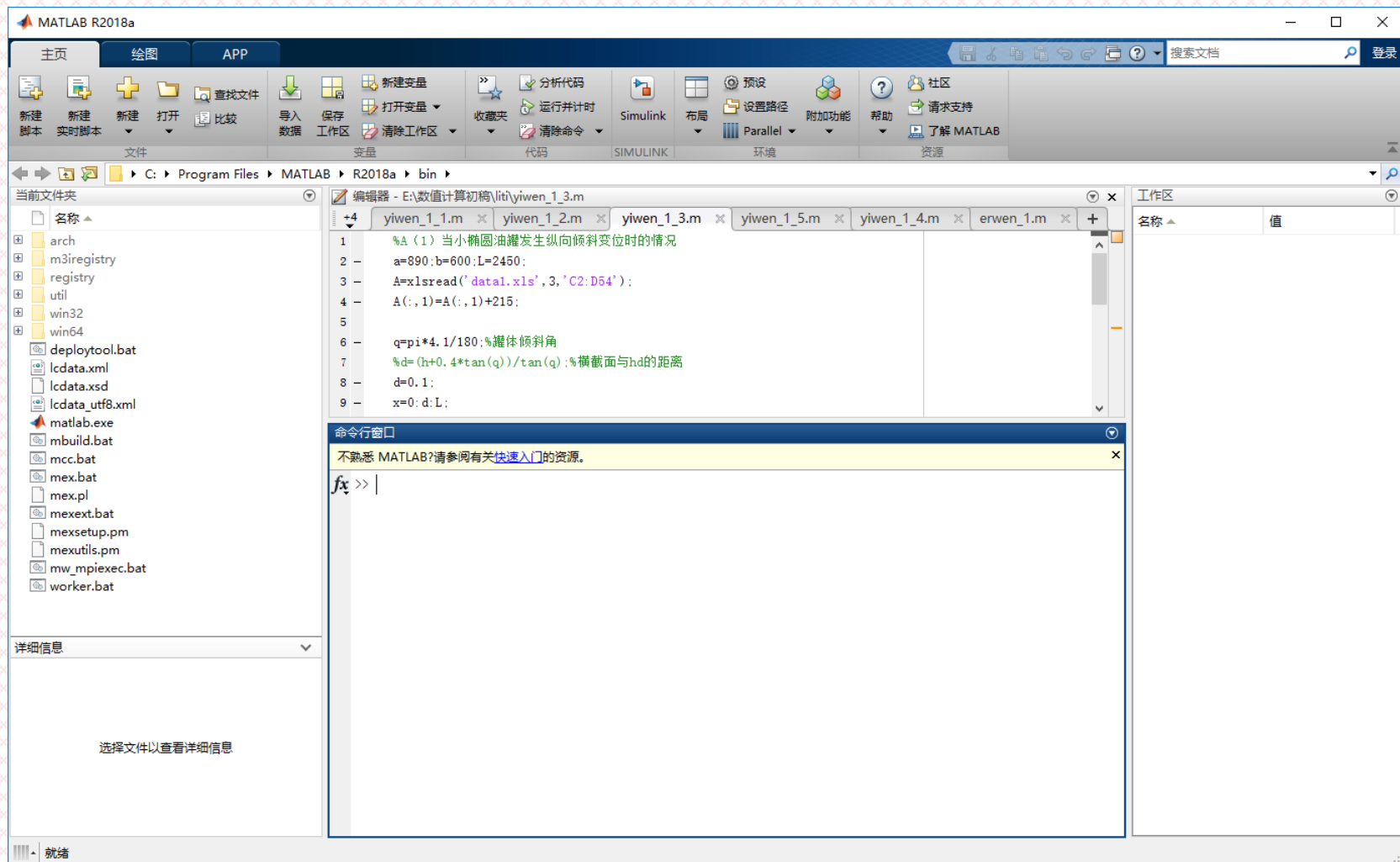


# 第一章 Matlab基本操作

## -----(2) 矩阵与数组



# Matlab系统界面



# 矩阵与数组

- 数据类型
- 矩阵（数组）的表示
- 数组基本运算
- 数组操作函数



## 1.2.1 数据类型

Matlab数据类型非常丰富，除数值型、字符型等基本数据类型外，还有结构体、单元等更为复杂的数据类型。各种数据类型都以矩阵形式存在，矩阵是Matlab最基本的数据对象，并且矩阵的运算是定义在复数域上的。

# 数值型



## 数值类型包含

- 整数;
- 浮点数;
- 复数;
- **Inf;**
- **NaN**

# (1) 整数类型

**MATLAB支持1、2、4和8字节的有符号整数和无符号整数。**  
**数据类型的名称、表示范围和转换函数如下表所示。**

名 称	表示范围	转换函数	名 称	表示范围	转换函数
有符号 1 字节整数	$-2^7 \sim 2^7 - 1$	int8()	无符号 1 字节整数	$0 \sim 2^8 - 1$	uint8()
有符号 2 字节整数	$-2^{15} \sim 2^{15} - 1$	int16()	无符号 2 字节整数	$0 \sim 2^{16} - 1$	uint16()
有符号 4 字节整数	$-2^{31} \sim 2^{31} - 1$	int32()	无符号 4 字节整数	$0 \sim 2^{32} - 1$	uint32()
有符号 8 字节整数	$-2^{63} \sim 2^{63} - 1$	int64()	无符号 8 字节整数	$0 \sim 2^{64} - 1$	uint64()

## (2) 浮点数类型

**MATLAB**有单精度和双精度两种浮点数。其名称、存储空间、表示范围和转换函数如下表所示。

名 称	存 储 空 间	表 示 范 围	转 换 函 数
单精度浮点数	4 字节	$-3.40282 \times 10^{38} \sim 3.40282 \times 10^{38}$	single()
双精度浮点数	8 字节	$-1.79769 \times 10^{308} \sim 1.79769 \times 10^{308}$	double()



### (3) 复数类型

复数包含实部和虚部。在**MATLAB**中可以用**i**或者**j**来表示虚部。

例 在命令窗口用赋值语句产生复数  $5+10i$ ，具体代码如下：

```
a=5+10i
```

例 在命令窗口用函数 `complex()`产生复数  $5+10i$ ，具体代码序列如下：

```
x=5;
```

```
y=10;
```

```
z=complex(x,y)
```





## (4) Inf和NaN

- **Inf**和**-Inf**分别表示正无穷大和负无穷大。除法运算中除数为**0**或者运算结果溢出都会导致**inf**或**-inf**的运行结果。
- 在**MATLAB**中用**NaN** (**Not a Number**) 来表示一个既不是实数也不是复数的数值。

# 逻辑类型



在**MATLAB**中逻辑类型包含**true**和**false**，分别由**1**和**0**表示。在**MATLAB**中用函数**logical()**将任何非零的数值转换为**true**（即**1**），将数值**0**转换为**false**（即**0**）。



# 字符和字符串类型

在**MATLAB**中，数据类型（**char**）表示一个字符。一个**char**类型的 $1 \times n$ 数组称为字符串**string**。

例 在命令窗口用“单引号对”表示字符串'I am a great person'，具体代码如下：

```
str='I am a great person'
```

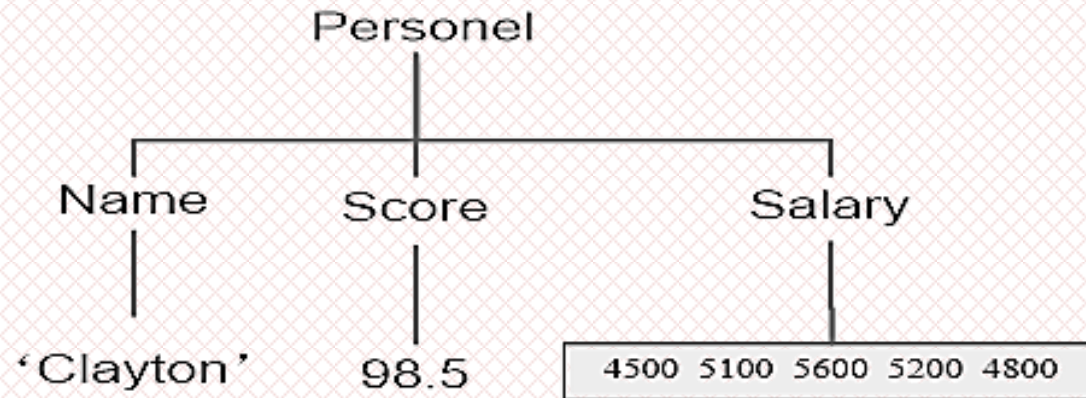
例 在命令窗口用函数 `char()`构造字符串'AB'，具体代码如下：

```
str=char([65 66])
```

# 结构体类型

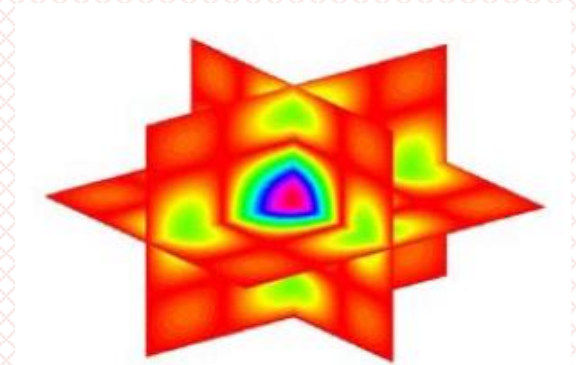
结构体类型是一种由若干属性（**field**）组成的**MATLAB**数组，其中的每个属性可以是任意数据类型。

下图表示了一个结构体（**Personel**），它包括**3**个属性（**Name**、**Score**和**Salary**），其中**Name**是一个字符串，**Score**是一个数值，**Salary**是一个**1×5**的向量。



## 1.2.2 矩阵(数组)的表示

- 数组的概念
- 一维数组变量的创建
- 二维数组变量的创建
- 数组元素的标识与寻访
- 多维数组
- 空数组
- 字符串



# (1) 数组(array)的概念

- 数组：按行(row)和列(column)顺序排列的实数或复数的有序集，被称为数组。
- 数组中的任何一个数都被称为这个数组的元素，由其所在的行和列标识，这个标识也称为数组元素的下标或索引。Matlab将标量视为 $1 \times 1$ 的数组。
- 对m行、n列的2维数组a：记为 $m \times n$ 的数组a；
- 行标识、列标识均从1开始；  
    行标识从上到下递增；  
    列标识从左到右递增。

a=

1	2	3	4	5
2	22	23	24	25
3	32	33	34	35
4	42	43	44	45

$a(3, 4)=34$  row is first

# (1) 数组(array)的概念

- 数组的分类
  - 一维数组，也称为向量(vector)。
    - 行向量(row vector)、列向量(column vector)。
  - 二维数组(矩阵matrix)。
  - 多维数组。
- 有效矩阵：每行元素的个数必须相同，每列元素的个数也必须相同。

# (1) 数组(array)的概念

数组 (array)	大小(size)
$a = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$	$3 \times 2$
$b = [1 \ 2 \ 3 \ 4]$	$1 \times 4$
$c = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$	$3 \times 1$

行向量

列向量

$$a(2,1)=3 \quad a(1,2)=2 \quad b(3)=3 \quad c(2)=2$$



## (2) 创建一维数组变量

- 第一种方法：使用方括号 “[ ]” 操作符

【例1】创建数组(行向量)  $a=[1\ 3\ \pi\ 3+5i]$

```
>>a=[1 3 pi 3+5*i] %or a=[1, 3, pi, 3+5*i]
```

```
a= 1.0000    3.0000    3.1416    3.0000 + 5.0000i
```

所有的向量元素必须在操作符 “[ ]” 之内；

向量元素间用空格或英文的逗点 “,” 分开。

- 第二种方法：使用冒号 “:” 操作符

【例2】创建以1~10顺序排列整数为元素的行向量b。

```
>>b=1:10
```

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

## (2) 创建一维数组变量 (续)

**【例3】** 键入并执行 $c=1:2:10$ 和 $d=1:2:9$

```
>> c=1:2:10
```

```
c=1 3 5 7 9
```

```
>>d=1:2:9
```

```
d=1 3 5 7 9
```

利用冒号“:”操作符创建行向量的基本语法格式:

$x=Start:Increment:End$

- Start表示新向量 $x$ 的第一个元素;
- 新向量 $x$ 的最后一个元素不能大于End ;
- Increment可正可负, 若负, 则必须 $Start>End$ ; 若正, 则必须 $Start<End$ , 否则创建的为空向量。
- 若 $Increment=1$ , 则可简写为:  $x=Start:End$ 。

## (2) 创建一维数组变量 (续)

- 第三种方法：利用函数 `linspace`

函数 `linspace` 的基本语法

`x = linspace(x1, x2, n)`

- 该函数生成一个由 `n` 个元素组成的行向量；
- `x1` 为其第一个元素；
- `x2` 为其最后一个元素；
- `x1`、`x2` 之间元素的间隔  $= (x2 - x1) / (n - 1)$ 。
- 如果忽略参数 `n`，则系统默认生成 100 个元素的行向

**【例4】** 键入并执行 `x = linspace(1,2,5)`

`x = 1.0000   1.2500   1.5000   1.7500   2.0000`

请查看 `x = linspace(1,2)` 执行结果。

## (2) 创建一维数组变量 (续)

- 列向量的创建

- 使用方括号 “[ ]” 操作符，使用分号 “;” 分割行。

**【例5】** 键入并执行  $x = [1; 2; 3]$

$x = 1$

2

3

- 使用冒号操作符

**【例6】** 键入并执行  $x = (1:3)'$     % 符号 “'” 表示矩阵的转置

## (2) 创建一维数组变量 (续)

- 创建数组变量的一般方法
  - 创建变量的赋值语句的一般格式  
**var=expression**
    - var为变量名
    - expression为MATLAB合法表达式
      - 可以是单独的常数值或数值数组;
      - 也可以由常数值、其他变量（部分或全部）、数值数组和运算符（+、-等）构成。

**【例7】** 键入并执行

```
a=[0 1+6]; b=[a 6 7]; c=[6 a 7]; d=[6 a 7 a];
```

## (2) 创建一维数组变量 (续)

- 一旦被创建，变量就被存储在工作空间，可以通过“**Workspace**”窗口或在“**Command Window**”执行“**whos**”命令查看。
  - 操作一：使用“**Workspace**”窗口查看变量
  - 操作二：使用“**whos**”命令查看变量

### (3) 创建二维数组变量

- 第一种方法：使用方括号 “[ ]” 操作符

使用规则

- 数组元素必须在 “[ ]” 内键入；
- 行与行之间须用分号 “;” 间隔，也可以在分行处用回车键间隔；
- 行内元素用空格或逗号 “,” 间隔。

**【例3-1】** 键入并执行 `a2=[1 2 3;4 5 6;7 8 9]`

`a2=`

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

### (3) 创建二维数组变量

**【例3-2】** 键入并执行`a2=[1:3;4:6;7:9]` %结果同上

**【例3-3】** 由向量构成二维数组。

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

```
>>c=[a;b];
```

```
>>c1=[a b];
```



## (3) 创建二维数组变量 (续)

- 第二种方法：函数方法

函数**ones**(生成全1矩阵)、**zeros** (生成全0矩阵)、**reshape**

【例3-4】创建全1的3x3数组。

```
>>ones(3)
```

【例3-5】创建全1的3x4数组。

```
>>ones(3,4)
```

### (3) 创建二维数组变量 (续)

**【例3-6】** reshape的使用演示

```
>>a=-4:4
```

```
a=
```

```
-4 -3 -2 -1 0 1 2 3 4
```

```
>>b=reshape(a, 3, 3)
```

```
b=
```

```
-4 -1 2
```

```
-3 0 3
```

```
-2 1 4
```

- ☞ 数组元素的排列顺序，从上到下按列排列，先排第一列，然后第二列，...
- ☞ 要求数组的元素总数不变。

## 2.2.3、创建二维数组变量（续）

<code>zeros(m,n)</code>	生成一个 $m$ 行 $n$ 列的零矩阵, $m=n$ 时可简写为 <code>zeros(n)</code>
<code>ones(m,n)</code>	生成一个 $m$ 行 $n$ 列的元素全为 1 的矩阵, $m=n$ 时可写为 <code>ones(n)</code>
<code>eye(m,n)</code>	生成一个主对角线全为 1 的 $m$ 行 $n$ 列矩阵, $m=n$ 时可简写为 <code>eye(n)</code> , 即为 $n$ 维单位矩阵
<code>diag(A)</code> <code>diag(x)</code>	$A$ 是矩阵, 则 <code>diag(A)</code> 为 $A$ 的主对角线向量 $x$ 是向量, <code>diag(x)</code> 产生以 $x$ 为主对角线的对角矩阵
<code>rand(m,n)</code>	产生 0~1 间均匀分布的随机矩阵 $m=n$ 时简写为 <code>rand(n)</code>
<code>randn(m,n)</code>	产生均值为0, 方差为1的标准正态分布随机矩阵 $m=n$ 时简写为 <code>randn(n)</code>

## (4) 数组元素的标识与寻访

- 数组元素的标识

- “全下标 (index)” 标识

如对于二维数组，用“行下标和列下标”标识数组的元素， $a(2,3)$ 就表示二维数组 $a$ 的“第2行第3列”的元素。

对于一维数组，用一个下标即可， $b(2)$ 表示一维数组 $b$ 的第2个元素，无论 $b$ 是行向量还是列向量。

- “单下标” (linear index) 标识

用一个下标来表明元素在数组的位置。

对于二维数组，“单下标”编号：设想把二维数组的所有列，按先后顺序首尾相接排成“一维长列”，然后自上往下对元素位置执行编号。

- 两种“下标”标识的变换：sub2ind、ind2sub

## (4) 数组元素的标识与寻访 (续)

### 【例4-1】单下标的使用

```
>>a=zeros(2,5);
```

```
>>a(:)=-4:5
```

a =

-4   -2   0   2   4

-3   -1   1   3   5

🔔 注意数组的排列顺序。

## (4) 数组元素的标识与寻访 (续)

- 元素与子数组的寻访与赋值

**【例4-3】** 一维数组元素与子数组的寻访与赋值

```
>>a=linspace(1,10,5)
```

```
a =
```

```
1.0000 3.2500 5.5000 7.7500 10.0000
```

```
>>a(3) %寻访a的第3个元素
```

```
ans =
```

```
5.5000
```

```
>>a([1 2 5]) %寻访a的第1、2、5个元素组成的子数组
```

```
ans =
```

```
1.0000 3.2500 10.0000
```

## (4) 数组元素的标识与寻访 (续)

>>a(1:3) %寻访前3个元素组成的子数组

ans =

1.0000 3.2500 5.5000

>>a(3:-1:1) %由前3个元素倒序构成的子数组

ans =

5.5000 3.2500 1.0000

>>a(3:end)

🔔 %第3个及其后所有元素构成的子数组

ans =

5.5000 7.7500 10.0000

🔔 函数end作为参数使用，返回最后一个元素的下标

>>a(3:end-1)

ans =

5.5000 7.7500

## (4) 数组元素的标识与寻访 (续)

```
>>a([1 2 3 5 5 3 2 1])
```

```
ans =
```

```
1.0000 3.2500 5.5000 10.0000 10.0000 5.5000  
3.2500 1.0000
```

🔔 数组元素可以被任意重复访问，构成长度大于原数组的新数组。

```
>>a(6)
```

```
??? Index exceeds matrix dimensions.
```

☛ 下标值超出了数组的维数，导致错误

```
>>a(2.1)
```

```
??? Subscript indices must either be real positive  
integers or logicals.
```

☛ 下标值只能取正整数或逻辑值



## (4) 数组元素的标识与寻访 (续)

>>a(3)=0            %修改数组a的第3元素值为0

a =

1.0000   3.2500   0   7.7500   10.0000

>>a([2 5])=[1 1]

a =

1.0000   1.0000   0   7.7500   1.0000

- 可以修改指定数组元素的值
- 一次可以修改多个数组元素的值
- 要修改的数组元素的个数应与送入数组的元素个数相同

## (4) 数组元素的标识与寻访 (续)

【例4-3】 二维数组元素与子数组的寻访与赋值

>>a\_2=zeros(2,4) %创建2x4的全0数组

a\_2 =

0 0 0 0

0 0 0 0

>>a\_2(:)=1:8

a\_2 =

1 3 5 7

2 4 6 8

🔔 注意元素的排列顺序

>>a\_2([2 5 8]) %单下标方式寻访多个元素

ans =

2 5 8

## (4) 数组元素的标识与寻访 (续)

```
>> a_2([2 5 8])=[10 20 30]
```

```
a_2 =
```

```
1    3    20    7  
10   4    6   30
```

```
>>a_2(:,[2 3])=ones(2) %双下标方式寻访并修改
```

```
a_2 =
```

```
1    1    1    7  
10   1    1   30
```

- 🔔 二维数组可以“单下标”方式或“全下标”方式访问、赋值；
- 🔔 “单下标”方式赋值时，等号两边涉及的元素个数必须相等；
- 🔔 “全下标”方式赋值时，等号右边数组的大小必须等于原数组中涉及元素构成的子数组的大小。

## (4) 数组元素的标识与寻访 (续)

```
>>a_2(:,end)
```

```
ans =
```

```
7
```

```
30
```

```
>>a_2(:,end-1)
```

```
ans =
```

```
1
```

```
1
```

```
>>a_2(:, end:-1:3)
```

```
ans =
```

```
7 1
```

```
30 1
```

```
>>a_2(end,:)
```

```
ans =
```

```
10 1 1 30
```

```
>>a_2(end,[2:4])
```

```
ans =
```

```
1 1 30
```

```
>>a_2 ([4 6])=6:7
```

```
a_2 =
```

```
1 1 1 7
```

```
10 6 7 30
```

```
>>a_2(end,[2:end-1])
```

**What is the result?**

## (4) 数组元素的标识与寻访 (续)

### 【例4-4】 size、length函数

```
>>a=ones(4,6)*6
```

```
>>m=size(a)
```

```
>>len=length(a)
```

```
>>b=1:5;
```

```
>>length(b)
```

```
>>c=b'
```

```
>>length(c)
```

➤ size函数返回变量的大小，即变量数组的行列数

➤ length函数返回变量数组的最大维数

# 课堂练习

`A=10*rand(15);`

`A(2,3)` % 查询第2行第3列所对应的元素

`A(12)` % 查询第12个元素

`A([9 14],[10 15])` % 思考之

`A(:,4)` % 查询第4列元素

`A(4,:)` % 查询第4行元素

`A(end,:)` % 最后一行元素

`A(find(A>5))` % 查询大于5的元素

`A(2,3) = 5` % 重新赋值第2行3列元素为5

## (5) 多维数组

- 多维数组的定义

在 MATLAB 的数据类型中，向量可视为一维数组，矩阵可视为二维数组，对于维数(Dimensions)超过2的数组均可视为「多维数组」(Multidimensional Arrays，简称N-D Arrays)。

## (5) 多维数组 (续)

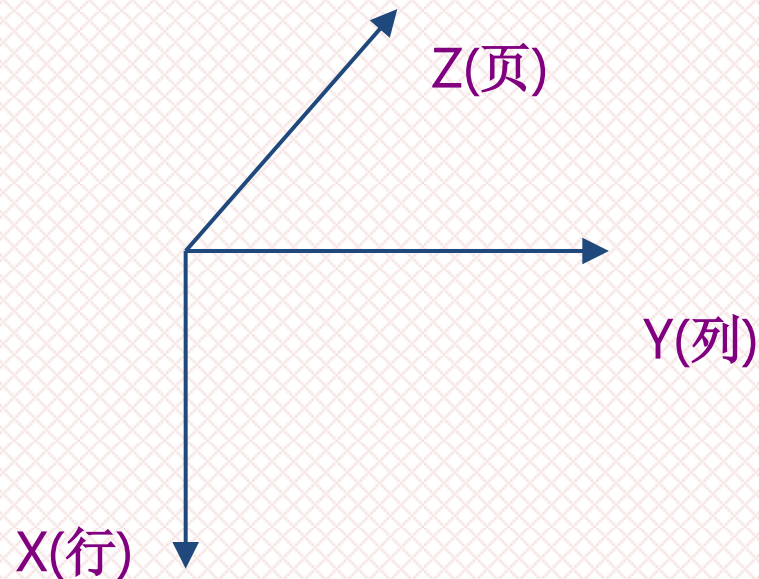
- 将两个二维 (平面) 数组叠在一起, 就构成三维数组, 第三维称为「页」(Page), 如下图所示:





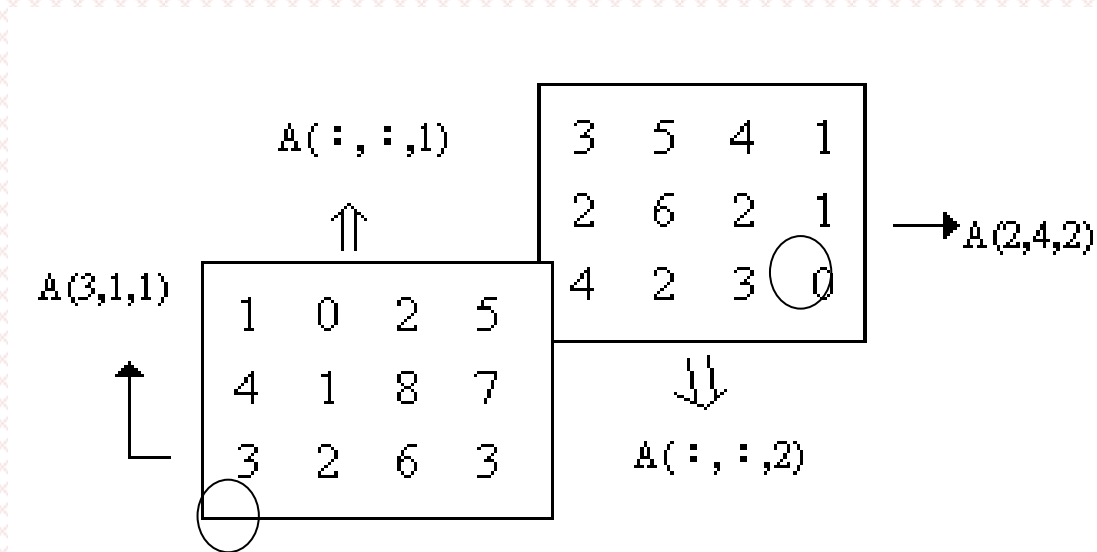
## (5) 多维数组 (续)

- 三维数组，可对应至一个  $X - Y - Z$  三维立体坐标，如下图所示：



## (5) 多维数组 (续)

- 三维数组元素的寻址：可以(行、列、页)来确定。
- 以维数为  $3 \times 4 \times 2$  的三维数组为例，其寻址方式如下图所示：



数组  $A$  是三维数组，其中  $A(:,:,1)$  代表第一页的二维数组， $A(:,:,2)$  代表第二页的二维数组。

## (5) 多维数组 (续)

- 多维数组的建立

- 建立一个简单的多维数组，可直接由 MATLAB 命令视窗内输入（使用 “[ ]”操作符）
- 例：由两个相同大小二维数组创建三维数组

```
A(:, :, 1) = [1 0 2 5; 4 1 8 7; 3 2 6 3];
```

```
A(:, :, 2) = [3 5 4 1; 2 6 2 1; 4 2 3 0]
```

$A(:, :, 1) =$

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

$A(:, :, 2) =$

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

## (5) 多维数组 (续)

执行命令: **whos A**, 得到如下结果:

<b>Name</b>	<b>Size</b>	<b>Bytes</b>	<b>Class</b>
<b>A</b>	<b>3x4x2</b>	<b>192</b>	<b>double array</b>

**Grand total is 24 elements using 192 bytes**

## (6) 空数组 (empty array)

- 有一维是0的数组即为空数组
- 空数组不占据存储空间
- 最简单的空数组：0 x 0的矩阵
- 复杂的空数组：0 x 5 or 10 x 0

例如：>>a=[]; b=ones(0,5);

察看空数组：>>a, b, c    % or whos a b c

\* 空数组并非全0数组

## (6) 空数组 (续)

- 数组维数的减小

- 删除数组的某列和行

- ```
>>a = magic(4), a(:,2)=[]
```

- 删除(2-D、3-D)数组的单个元素

- 使用“全下标”方式，不能删除单个元素

- ```
>>a(1,2)=[] %系统会警告信息
```

- 使用“单下标”可以删除单个元素

- ```
>>a(2:4)=[] %数组a将变为向量
```

- 使用“[]”同样可以减小字符数组的维数

## (7) 字符串

- 字符（Characters）可以构成一个字符串（Strings），或字符数组(character array)。
- 一个字符串是被视为一个行向量（row vector）。
- 字符串中的每一个字符（含空格），以其 ASCII 码的形式存放于行向量中，是该字符串变量的一个元素（element）。

## (7) 字符串 (续)

- Matlab 用「单引号」来界定一个字符串。
- 可以使用**方括号**“**[ ]**”直接连接多个字符串变量，得到一个新字符串变量。

### 【例1】命名字符串变量

```
str1 = 'I like MATLAB,';      % 建立字符串变量 str1  
str2 = ' JavaScript, and Perl!'; % 建立字符串变量str2  
str3 = [str1 str2]           % 直接连接str1及str2，以建立str3
```

str3 =

I like MATLAB, JavaScript, and Perl!



## (7) 字符串 (续)

- 如要输入的字符串中有单引号，则由两个连续的单引号来表示。
- 若要计算字符串变量的长度（即组成字符串的个数），可用 `length` 指令。
- **【例2】** 含单引号字符串的输入

```
sentence = 'I've got a date!';  
length(sentence) % 计算字符串sentence的长度
```

```
ans = 16
```

## (7) 字符串 (续)

- **double** 指令: 查看字符串变量的存储内容 (即 ASCII 内码)
- **char** 指令: 将 ASCII 内码转换为字符串形式

### 【例3】: 字符串与ASCII码的相互转换

```
sentence = 'I've got a date!';  
sentenceAscii = double(sentence) %查看 sentence 的 ASCII 码  
sentence2 = char(sentenceAscii) % 将 ASCII 码恢复成字符串形式
```

```
sentenceAscii =  
    73    39   118   101    32   103   111   116    32    97    32   100    97   116   101    33  
sentence2 =  
    I've got a date!
```

## (7) 字符串 (续)

- **class** 或 **ischar** 指令: 可以用来判断一个变量的类型或它是否为字符串变量。

【例4】：判断一个变量是否为字符串变量。

```
chinese = '今日事，今日毕';  
out1 = class(chinese)    % out1 的值是 “char”  
x = chinese+1;  
out2 = ischar(x)    % out2 的值是 0，代表 x 不是字符串变量
```

## (7) 字符串 (续)

- **strcmp** 指令: 用于比较字符串的内容的异同

### 【例5】：字符串比较

```
str1 = 'today';
```

```
str2 = 'tomorrow';
```

```
str3 = 'today';
```

```
out1 = strcmp(str1, str2)
```

```
out1 = 0
```

```
out2 = strcmp(str1, str3)
```

```
out2 = 1
```

% 比较字符串 str1 和 str2  
%表示字符串 str1 和 str2不同  
% 比较字符串 str1 和 str3  
%表示字符串 str1 和 str2相同

- 不相等返回0, 相等返回1

## 1.2.3、数组的基本运算

- MATLAB数组支持线性代数中所有的矩阵运算。
- 建立特有的数组运算符，如：“.\*”、“./”等。

### MATLAB数组运算符列表

| 运算      | 运算符   | 含义说明         |
|---------|-------|--------------|
| 加       | +     | 相应元素相加       |
| 减       | -     | 相应元素相减       |
| 乘       | *     | 矩阵乘法         |
| 点乘      | .*    | 相应元素相乘       |
| 幂       | ^     | 矩阵幂运算        |
| 点幂      | .^    | 相应元素进行幂运算    |
| 左除或右除   | \或/   | 矩阵左除或右除      |
| 左点除或右点除 | .\或./ | A的元素被B的对应元素除 |

# (1) 算术运算

## 【例1】数组加减法

```
>>a=zeros(2,3);
```

```
>>a(:)=1:6;
```

```
>>b=a+2.5
```

```
b =
```

|        |        |        |
|--------|--------|--------|
| 3.5000 | 5.5000 | 7.5000 |
| 4.5000 | 6.5000 | 8.5000 |

```
>>c=b-a
```

```
c =
```

|        |        |        |
|--------|--------|--------|
| 2.5000 | 2.5000 | 2.5000 |
| 2.5000 | 2.5000 | 2.5000 |

# (1) 算术运算 (续)

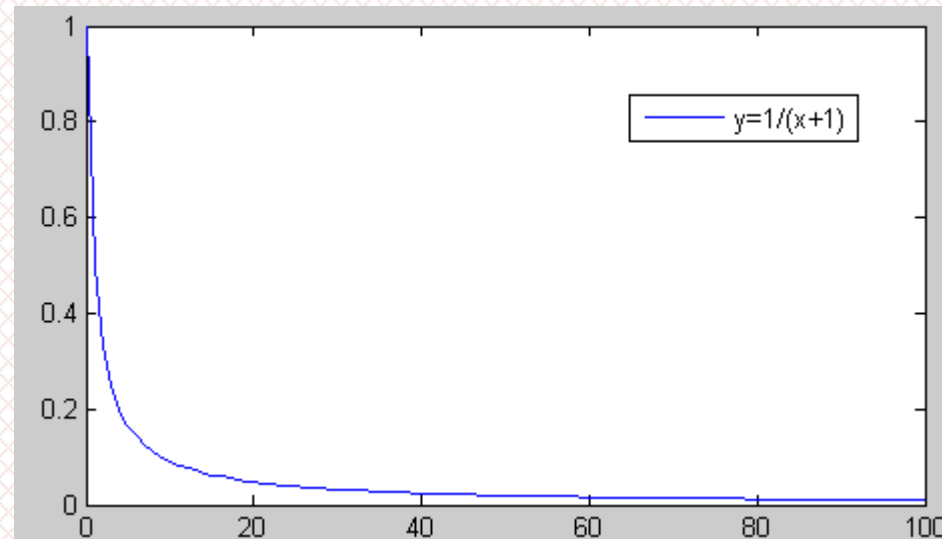
**【例2】** 画出 $y=1/(x+1)$ 的函数曲线,  $x \in [0, 100]$ 。

```
x=0:100;
```

```
y=1./(x+1);
```

```
plot(x, y);
```

```
legend('y=1/(x+1)');
```



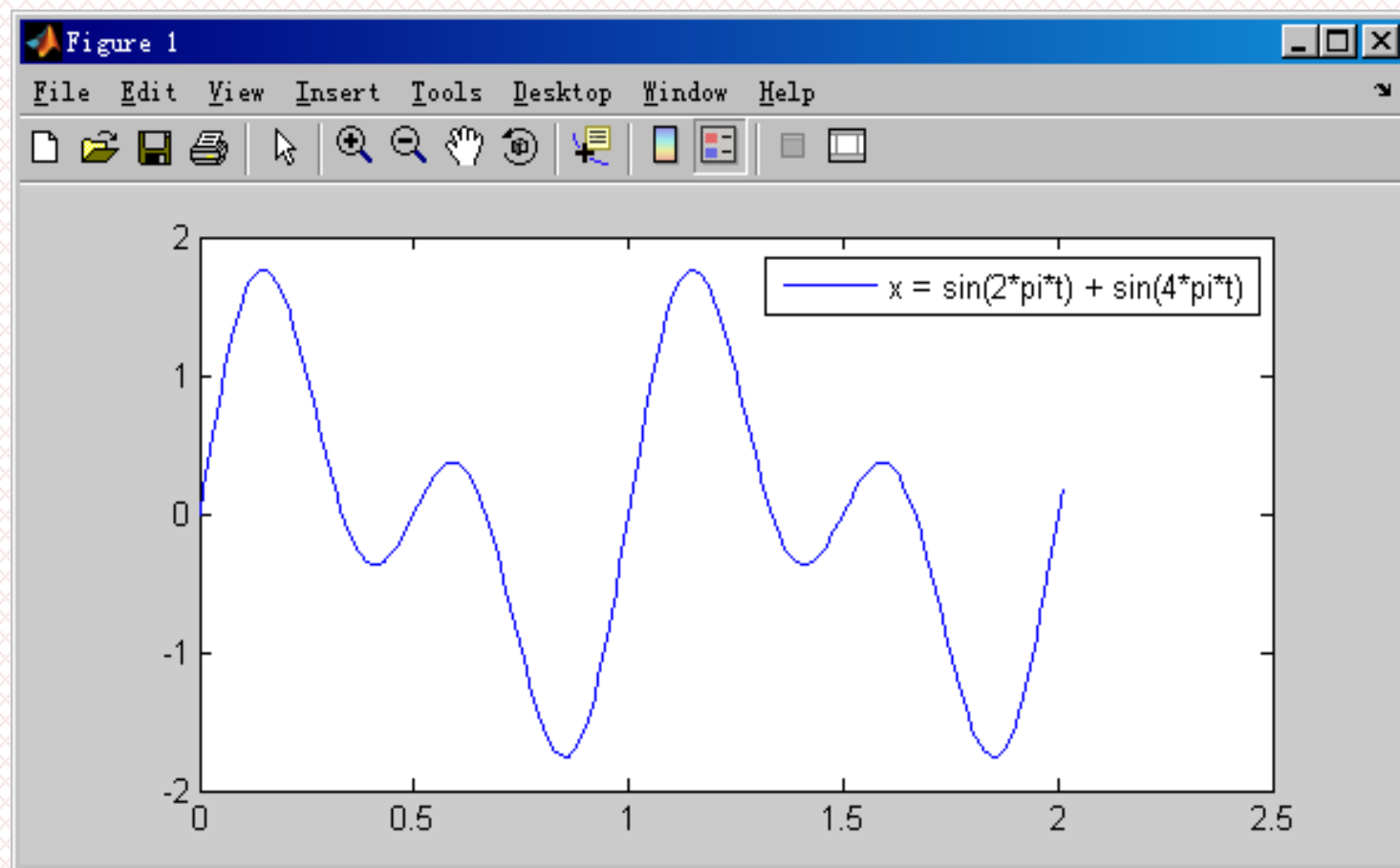
# (1) 算术运算 (续)

**【例3】** 生成一个信号:  $x = \sin(2\pi t) + \sin(4\pi t)$

```
t = [0:199]./100;    %采样时间点  
% 生成信号  
x = sin(2*pi*t) + sin(4*pi*t);  
plot(t,x);  
legend('x = sin(2*pi*t) + sin(4*pi*t)');
```



# (1) 算术运算 (续)



# (1) 算术运算 (续)

## 【例4】点幂 “.^”举例

```
>>a=1:6
```

```
a =
```

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

```
>>b=reshape(a,2,3)
```

```
b =
```

```
1 3 5
```

```
2 4 6
```

```
>>a=a.^2
```

```
a =
```

```
1 4 9 16 25 36
```

```
>>b=b.^2
```

```
b =
```

```
1 9 25
```

```
4 16 36
```

## (2) 关系运算

Matlab提供了6种关系运算符:

$<$ 、 $>$ 、 $\leq$ 、 $\geq$ 、 $==$ 、 $\neq$  (不等于)

关系运算符的运算法则:

- 1、当两个标量进行比较时, 直接比较两数大小。若关系成立, 结果为1, 否则为0。
- 2、当两个维数相等的矩阵进行比较时, 其相应位置的元素按标量关系进行比较, 并给出结果, 形成一个维数与原来相同的0、1矩阵。
- 3、当一个标量与一个矩阵比较时, 该标量与矩阵的各元素进行比较, 结果形成一个与矩阵维数相等的0、1矩阵。

## (2) 关系运算 (续)

【例5】建立5阶方阵A，判断其元素能否被3整除。

A = [24, 35, 13, 22, 63; 23, 39, 47, 80, 80; ...  
90, 41, 80, 29, 10; 45, 57, 85, 62, 21; 37, 19,  
31, 88, 76]

A =

|    |    |    |    |    |
|----|----|----|----|----|
| 24 | 35 | 13 | 22 | 63 |
| 23 | 39 | 47 | 80 | 80 |
| 90 | 41 | 80 | 29 | 10 |
| 45 | 57 | 85 | 62 | 21 |
| 37 | 19 | 31 | 88 | 76 |

P = rem(A,3)==0 %被3除，求余

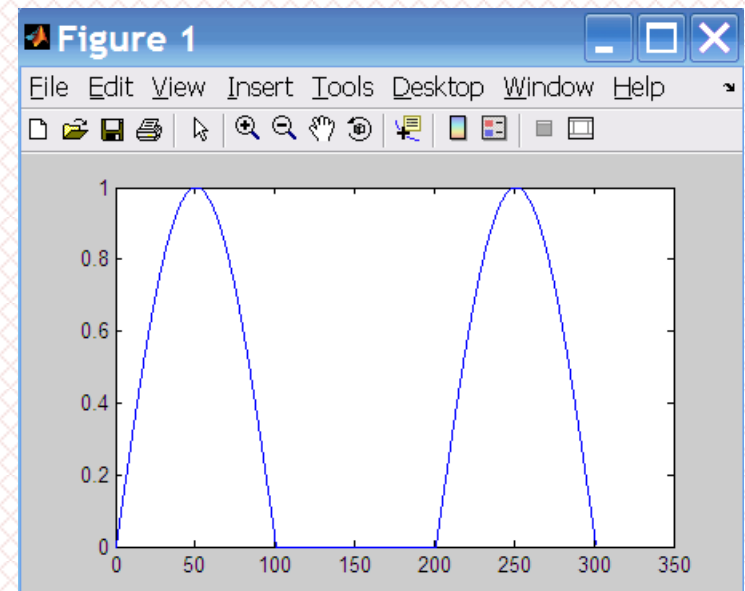
P =

|   |   |   |   |   |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 |

## (2) 关系运算 (续)

【例6】在 $[0, 3\pi]$ 区间，求  $y = \sin x$  的值。要求  
消去负半波，即 $(\pi, 2\pi)$ 区间内的函数值置零。

```
x = 0:pi/100:3*pi;  
y = sin(x);  
y1 = (y>=0).*y; %消去负半波
```



## (2) 关系运算 (续)

【例7】 建立矩阵A，找出在[10, 20]区间的元素的位置。

```
A = [4,15,-45,10,6;56,0,17,-45,0];  
find(A>=10 & A<=20) %找到非零元素的位置
```

A =

```
4   15  -45   10    6  
56    0   17  -45    0
```

ans =

3

6

7

## (3) 逻辑运算

Matlab提供了3种逻辑运算符:

**& (与)、| (或)、~ (非)**

逻辑运算符的运算法则:

- 1、在逻辑运算中，确认非零元素为真（1），零元素为假（0）。
- 2、当两个维数相等的矩阵进行比较时，其相应位置的元素按标量关系进行比较，并给出结果，形成一个维数与原来相同的0、1矩阵；
- 3、当一个标量与一个矩阵比较时，该标量与矩阵的各元素进行比较，结果形成一个与矩阵维数相等的0、1矩阵；
- 4、算术运算优先级最高，逻辑运算优先级最低。

## (4) 向量运算

- 最大值和最小值

MATLAB提供的求数据序列的最大值和最小值的函数分别为**max**和**min**，两个函数的调用格式和操作过程类似。

### 1、求向量的最大值和最小值

求一个向量X的最大值的函数有两种调用格式，分别是：

(1) **y=max(X)**：返回向量X的最大值存入y，如果X中包含复数元素，则按模取最大值；

(2) **[y,I]=max(X)**：返回向量X的最大值存入y，最大值的序号存入I，如果X中包含复数元素，则按模取最大值。

求向量X的最小值的函数是**min(X)**，用法和**max(X)**完全相同。



## (4) 向量运算 (续)

**【例8】** 求向量的最大值

```
>>x=[-43,72,9,16,23,47];
```

```
>>y=max(x)    %求向量x中的最大值
```

```
y =  
    72
```

```
>>[y,l]=max(x) %求向量x中的最大值及其该元素的位置
```

```
y =  
    72
```

```
l =  
    2
```

## (4) 向量运算 (续)

### 2. 求矩阵的最大值和最小值

求矩阵A的最大值的函数有3种调用格式，分别是：

- (1) **max(A)**: 返回一个行向量，向量的第i个元素是矩阵A的第i列上的最大值；
- (2) **[Y,U]=max(A)**: 返回行向量Y和U，Y向量记录A的每列的最大值，U向量记录每列最大值的行号；
- (3) **max(A,[],dim)**: dim取1或2。dim取1时，该函数和max(A)完全相同；dim取2时，该函数返回一个列向量，其第i个元素是A矩阵的第i行上的最大值。

求最小值的函数是min，其用法和max完全相同。

## (4) 向量运算 (续)

### 【例9】求矩阵的最大值

```
>>x=[-43,72,9; 16,23,47];
```

```
>>y=max(x) %求矩阵x中每列的最大值
```

```
y =  
    16    72    47
```

```
>>[y,l]=max(x) %求矩阵x中每列的最大值及其该元素的位置
```

```
y =  
    16    72    47
```

```
l =  
     2     1     2
```

```
>>max(x, [],1), max(x, [],2) %求矩阵中每行的最大值
```



## 1.2.4 数组操作函数

矩阵操作函数、三角函数、指数/对数函数、复数函数、  
截断/求余函数、坐标转换函数、其它函数

## 1.2.4 数组操作函数

|                                   |                                                                 |
|-----------------------------------|-----------------------------------------------------------------|
| <b>diag(A)</b>                    | 用于提取矩阵A主对角线元素,产生一个具有 $\min(m, n)$ 个元素的列向量(其中A为 $m \times n$ 矩阵) |
| <b>diag(A, k)</b>                 | 提取矩阵A的第k条对角线的元素                                                 |
| <b>triu(A)</b>                    | 求矩阵A的上三角阵                                                       |
| <b>triu(A, k)</b>                 | 求矩阵A的第k条对角线以上的元素                                                |
| <b>tril(A)</b>                    | 求矩阵A的下三角阵                                                       |
| <b>tril(A, k)</b>                 | 求矩阵A的第k条对角线以下的元素                                                |
| <b>rot90(A, k)</b>                | 将矩阵A旋转90°的k倍                                                    |
| <b>fliplr(A)</b>                  | 对矩阵A实施左右翻转                                                      |
| <b>flipud(A)</b>                  | 对矩阵A实施上下翻转                                                      |
| <b>inv(A)</b>                     | 求A矩阵的逆矩阵                                                        |
| <b>pinv(A)</b>                    | 求A矩阵的伪逆(也称为广义逆矩阵)                                               |
| <b>det(A)</b>                     | 求方阵A所对应的行列式的值<br>(把一个方阵看作一个行列式,并对其按行列式规则求得值)                    |
| <b>rank(A)</b>                    | 求矩阵A的秩(矩阵线性无关的行数与列数称为矩阵的秩)                                      |
| <b>trace(A)</b>                   | 求矩阵A的迹<br>(矩阵的迹等于矩阵的对角线元素之和,也等于矩阵的特征值之和)                        |
| <b>norm(V)或norm(V, 2)</b>         | 计算向量V的2—范数                                                      |
| <b>norm(V, 1)</b>                 | 计算向量V的1—范数                                                      |
| <b>norm(V, inf)</b>               | 计算向量V的 $\infty$ —范数                                             |
| <b>cond(A, 1)</b>                 | 计算A的1—范数下的条件数                                                   |
| <b>cond(A)或cond(A, 2)</b>         | 计算A的2—范数下的条件数                                                   |
| <b>cond(A, inf)</b>               | 计算A的 $\infty$ —范数下的条件数                                          |
| <b>E=eig(A)</b>                   | 求A的全部特征值,构成向量E                                                  |
| <b>[V, D]=eig(A)</b>              | 求A的全部特征值,构成对角阵D;并求A的特征向量构成V的列向量。                                |
| <b>[V, D]=eig(A, 'nobalance')</b> | 直接求矩阵A的特征值和特征向量。                                                |
| <b>sqrtm(A)</b>                   | 计算矩阵A的平方根                                                       |
| <b>logm(A)</b>                    | 计算矩阵A的自然对数                                                      |

## 1.2.4 数组操作函数

### 三角函数

| 函 数 名 | 功 能 描 述     | 函 数 名 | 功 能 描 述     |
|-------|-------------|-------|-------------|
| sin   | 正弦          | sec   | 正割          |
| sind  | 正弦，输入以度为单位  | secd  | 正割，输入以度为单位  |
| sinh  | 双曲正弦        | sech  | 双曲正割        |
| asin  | 反正弦         | asec  | 反正割         |
| asind | 反正弦，输出以度为单位 | asecd | 反正割，输出以度为单位 |
| asinh | 反双曲正弦       | asech | 反双曲正割       |
| cos   | 余弦          | csc   | 余割          |

## 1.2.4 数组操作函数

| 函 数 名 | 功 能 描 述     | 函 数 名 | 功 能 描 述     |
|-------|-------------|-------|-------------|
| cosd  | 余弦，输入以度为单位  | cscd  | 余割，输入以度为单位  |
| cosh  | 双曲余弦        | csch  | 双曲余割        |
| acos  | 反余弦         | acsc  | 反余割         |
| acosd | 反余弦，输出以度为单位 | acscd | 反余割，输出以度为单位 |
| acosh | 反双曲余弦       | acsch | 反双曲余割       |
| tan   | 正切          | cot   | 余切          |
| tand  | 正切，输入以度为单位  | cotd  | 余切，输入以度为单位  |
| tanh  | 双曲正切        | coth  | 双曲余切        |
| atan  | 反正切         | acot  | 反余切         |
| atand | 反正切，输出以度为单位 | acotd | 反余切，输出以度为单位 |
| atan2 | 四象限反正切      | acoth | 反双曲余切       |
| atanh | 反双曲正切       |       |             |

## 1.2.4 数组操作函数

### 指数和对数函数

| 函 数<br>名 | 功 能 描 述             | 函 数<br>名 | 功 能 描 述                                         |
|----------|---------------------|----------|-------------------------------------------------|
| exp      | 指数                  | realpow  | 对数，若结果是复数则报错                                    |
| expm1    | 准确计算 $\exp(x)-1$ 的值 | reallog  | 自然对数，若输入不是正数则报错                                 |
| log      | 自然对数（以 e 为底）        | realsqrt | 开平方根，若输入不是正数则报错                                 |
| log1p    | 准确计算 $\log(1+x)$ 的值 | sqrt     | 开平方根                                            |
| log10    | 常用对数（以 10 为底）       | nthroot  | 求 x 的 n 次方根                                     |
| log2     | 以 2 为底的对数           | nextpow2 | 返回满足 $2^P \geq \text{abs}(N)$ 的最小正整数 P，其中 N 为输入 |
| pow2     | 以 2 为底的指数           |          |                                                 |



## 1.2.4 数组操作函数

### 指数和对数函数

例 计算矩阵  $A = [6 \ 12 \ 19; -9 \ -20 \ -33; 4 \ 9 \ 15]$  每个元素的指数,具体

代码序列如下:

```
A = [6 12 19; -9 -20 -33; 4 9 15];  
Y=exp(A)
```

运行结果如下:

```
Y =  
    403.43    1.6275e+005    1.7848e+008  
    0.00012341    2.0612e-009    4.6589e-015  
    54.598         8103.1    3.269e+006
```

## 1.2.4 数组操作函数

### 复数函数

| 函 数 名   | 功 能 描 述      | 函 数 名    | 功 能 描 述       |
|---------|--------------|----------|---------------|
| abs     | 模            | real     | 复数的实部         |
| angle   | 复数的相角        | unwrap   | 调整矩阵元素的相位     |
| complex | 用实部和虚部构造一个复数 | isreal   | 是否为实数矩阵       |
| conj    | 复数的共轭        | cplxpair | 把复数矩阵排列成为复共轭对 |
| imag    | 复数的虚部        |          |               |

## 1.2.4 数组操作函数

### 截断和求余函数

| 函 数 名 | 功 能 描 述  | 函 数 名 | 功 能 描 述      |
|-------|----------|-------|--------------|
| fix   | 向零取整     | mod   | 除法求余（与除数同号）  |
| floor | 向负无穷方向取整 | rem   | 除法求余（与被除数同号） |
| ceil  | 向正无穷方向取整 | sign  | 符号函数         |
| round | 四舍五入     |       |              |

## 1.2.4 数组操作函数

例 分别使用函数 `fix()`、`floor()`、`ceil()`和 `round()`，对向量  $A=[-1.55$   
 $-1.45 \ 1.45 \ 1.55]$ 的每个元素进行截断运算，具体代码序列如下：

```
A=[-1.55 -1.45 1.45 1.55];  
A_fix=fix(A);  
A_floor=floor(A);  
A_ceil=ceil(A);  
A_round=round(A);  
Y=[A_fix;A_floor;A_ceil;A_round]
```

运行结果如下：  $Y =$

|    |    |   |          |
|----|----|---|----------|
| -1 | -1 | 1 | <u>1</u> |
| -2 | -2 | 1 | <u>1</u> |
| -1 | -1 | 2 | <u>2</u> |
| -2 | -1 | 1 | 2        |

不难看出，4 种函数的运算结果不同，从而比较出它们的差异。

## 1.2.4 数组操作函数

例 分别使用函数 `mod()` 和 `rem()`，对标量除法  $-5/2$  进行求余，具体代码序列如下：

```
c_rem=rem(-5,2);  
c_mod=mod(-5,2);  
c=[c_rem c_mod]
```

运行结果如下：

|           |
|-----------|
| c =       |
| -1      1 |

不难看出，两种函数的运算结果不同，从而比较出它们的差异。

## 1.2.4 数组操作函数

### 坐标转换函数

| 函 数<br>名 | 功 能 描 述       | 函 数<br>名 | 功 能 描 述               |
|----------|---------------|----------|-----------------------|
| cart2sph | 笛卡尔坐标系转换为球坐标系 | sph2cart | 球坐标系转换为笛卡尔坐标系         |
| cart2pol | 笛卡尔坐标系转换为极坐标系 | hsv2rgb  | 灰度饱和度颜色空间转换为 RGB 颜色空间 |
| pol2cart | 极坐标系转换为笛卡尔坐标系 | rgb2hsv  | RGB 颜色空间转换为灰度饱和度颜色空间  |

## 1.2.4 数组操作函数

### 坐标转换函数

例 将笛卡尔坐标系中的点 ( 1,1,1 ) 分别转换到球坐标系和极坐标系中，具体代码序列如下：

```
[THETA,PHI,R] = cart2sph(1,1,1);  
P=[ THETA,PHI,R];  
[THETA,RHO,Z] = cart2pol(1,1,1);  
Q=[ THETA,RHO,Z];  
R=[P;Q]
```

运行结果如下：

```
R =  
  
    0.7854    0.61548    1.7321  
    0.7854    1.4142         1
```

## 1.2.4 数组操作函数

### 其它函数—多项式函数

| 函 数 名   | 功 能 描 述   | 函 数 名    | 功 能 描 述  |
|---------|-----------|----------|----------|
| conv    | 多项式乘法     | polyint  | 求多项式的积分  |
| deconv  | 多项式除法     | polyvar  | 求多项式的值   |
| poly    | 求多项式的系数   | polywarm | 求矩阵多项式的值 |
| polyfit | 多项式曲线拟合   | residue  | 部分分式展开   |
| polyder | 求多项式的一阶导数 | roots    | 求多项式的根   |



## 1.2.4 数组操作函数

其它函数—数据分析函数

- 求和与求积

**sum(X)**: 返回向量X各元素的和。

**prod(X)**: 返回向量X各元素的乘积。

**sum(A)**: 返回一个行向量，其第i个元素是A的第i列的元素和。

**prod(A)**: 返回一个行向量，其第i个元素是A的第i列的元素乘积。

**sum(A,dim)**: 当dim为1时，该函数等同于sum(A)；当dim为2时，返回一个列向量，其第i个元素是A的第i行的各元素之和。

**prod(A,dim)**: 当dim为1时，该函数等同于prod(A)；当dim为2时，返回一个列向量，其第i个元素是A的第i行的各元素乘积。

## 1.2.4 数组操作函数

其它函数—数据分析函数

- 平均值与中值

求数据序列平均值的函数是**mean**，求数据序列中值的函数是**median**。两个函数的调用格式为：

**mean(X)**：返回向量X的算术平均值。

**median(X)**：返回向量X的中值。

**mean(A)**：返回一个行向量，其第i个元素是A的第i列的算术平均值。

**median(A)**：返回一个行向量，其第i个元素是A的第i列的中值。

**mean(A,dim)**：当dim为1时，该函数等同于mean(A)；当dim为2时，返回一个列向量，其第i个元素是A的第i行的算术平均值。

**median(A,dim)**：当dim为1时，该函数等同于median(A)；当dim为2时，返回一个列向量，其第i个元素是A的第i行的中值。82

# 实验题

1. 采用向量构造法得到向量 **$[1,5,9,\dots,41]$** ;

2. 已知  $\mathbf{x}=[1 \ 2 \ 3]$ ,  $\mathbf{y}=[4 \ 5 \ 6]$ , 计算:

$$\mathbf{x}.\wedge\mathbf{y}, \quad \mathbf{x}.\wedge 2, \quad 2.\wedge\mathbf{x}$$

3. 已知  $\mathbf{A}=[1 \ 2 \ 3; \ 4 \ 5 \ 6];$

$$\mathbf{B}=[3 \ 2 \ 1; \ 6 \ 5 \ 4];$$

计算:  $\mathbf{C}=\mathbf{A}.*\mathbf{B}; \ \mathbf{D}=\mathbf{A}./\mathbf{B}; \ \mathbf{E}=\mathbf{A}.\backslash\mathbf{B}; \ \mathbf{F}=\mathbf{A}.\wedge\mathbf{B}$

## 实验题

4. 已知  $a = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$   $b = \begin{bmatrix} 2 & 4 & -1 \\ 1 & 3 & 5 \end{bmatrix}$   $c = \begin{bmatrix} 1 \\ 0 \\ -2 \end{bmatrix}$   $d = \begin{bmatrix} 1 & 4 & 7 \\ 8 & 5 & 2 \\ 3 & 6 & 0 \end{bmatrix}$

下列运算是否合法？如合法，请将结果写在表达式后面，给出注释说明；如不合法，请写明原因。

(1) **result1 = a'**

(2) **result2 = a \* b**

(3) **result3 = a + b**

(4) **result4 = b \* d**

(5) **result5 = [b ; c'] \* d**

(6) **result6 = a .\* b**

(7) **result7 = a ./ b**

(8) **result8 = a .\* c**

(9) **result9 = a . \ b**

(10) **result10 = a . ^2**

(11) **result11 = a ^2**

(12) **result12 = 2 . ^ a**

# 实验题

**5.** 按水平和竖直方向分别合并下述两个矩阵：

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 2 & 3 & 4 \\ 5 & 6 & 7 \\ 8 & 9 & 10 \end{bmatrix}$$

- 6.** 分别删除第**5**题两个结果的第**2**行。
- 7.** 分别将第**5**题两个结果的第**2**行最后**3**列的数值改为  
**[11 12 13]**。
- 8.** 分别查看第**5**题两个结果的各方向长度。

# 实验题

9. 分别将第5题两个结果均转换为 $2 \times 9$ 的矩阵。
10. 分别计算第5题矩阵 $A$ 和 $B$ 的 $A+B$ 、 $A.* B$ 和 $A \setminus B$ 。
11. 判断第5题矩阵 $A$ 和 $B$ 中哪些元素值不小于4。

12. 已知

$$A = \begin{bmatrix} 7 & 2 & 1 & -2 \\ 9 & 15 & 3 & -2 \\ -2 & -2 & 11 & 5 \\ 1 & 3 & 2 & 13 \end{bmatrix}$$

写出满足下列计算的命令语句，并给出计算结果：

- (1) 求矩阵 $A$ 的秩(rank); (2) 求矩阵 $A$ 的行列式(det);
- (2) (3) 求矩阵 $A$ 的逆(inv); (4) 求矩阵 $A$ 的最大值(max)。



# Q & A

- 有什么问题吗？

