

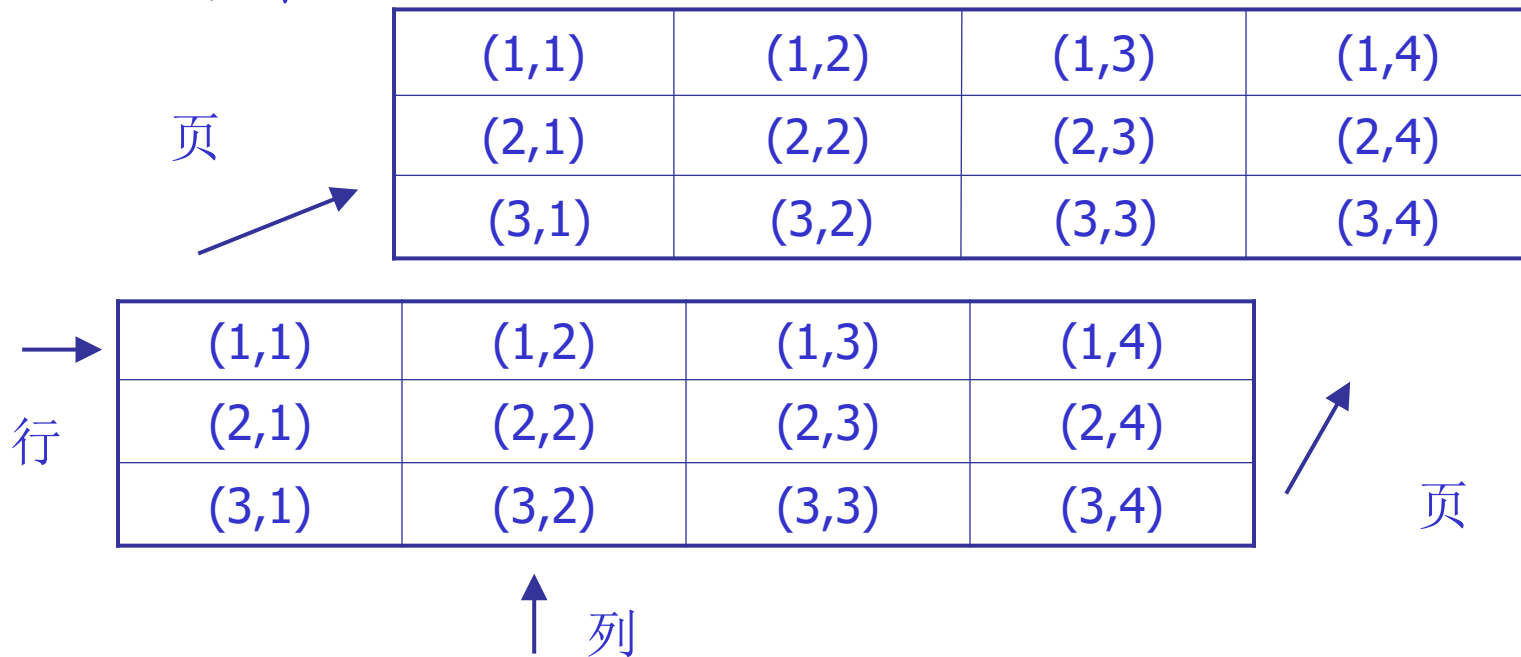


Matlab语言及其应用

第3讲

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）

■ 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：

Name	Size	Bytes	Class
A	3x4x2	192	double array

Grand total is 24 elements using 192 bytes



6、数组的算术运算

- 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



6、算术运算（续）

【例5-1】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

6、算术运算（续）

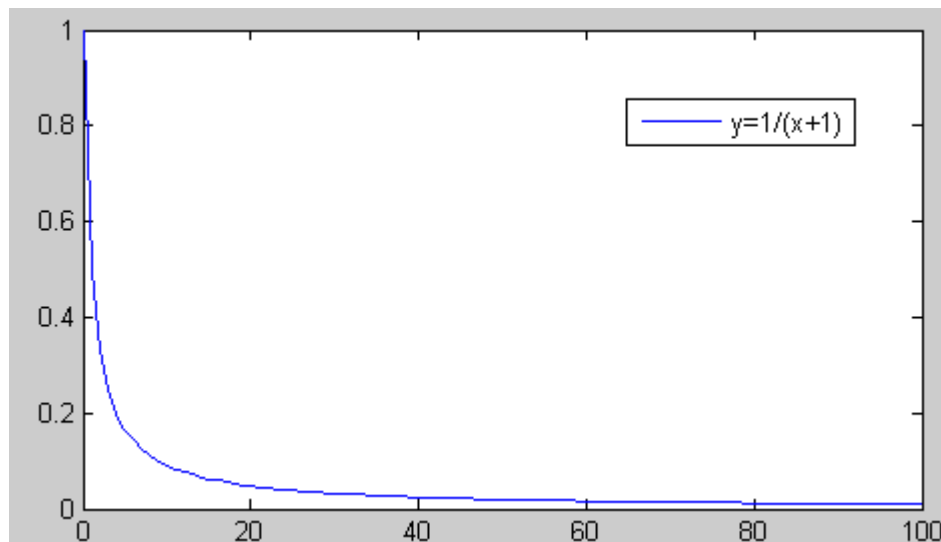
【例5-2】 画出 $y=1/(x+1)$ 的函数曲线， $x \in [0, 100]$ 。

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

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

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

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



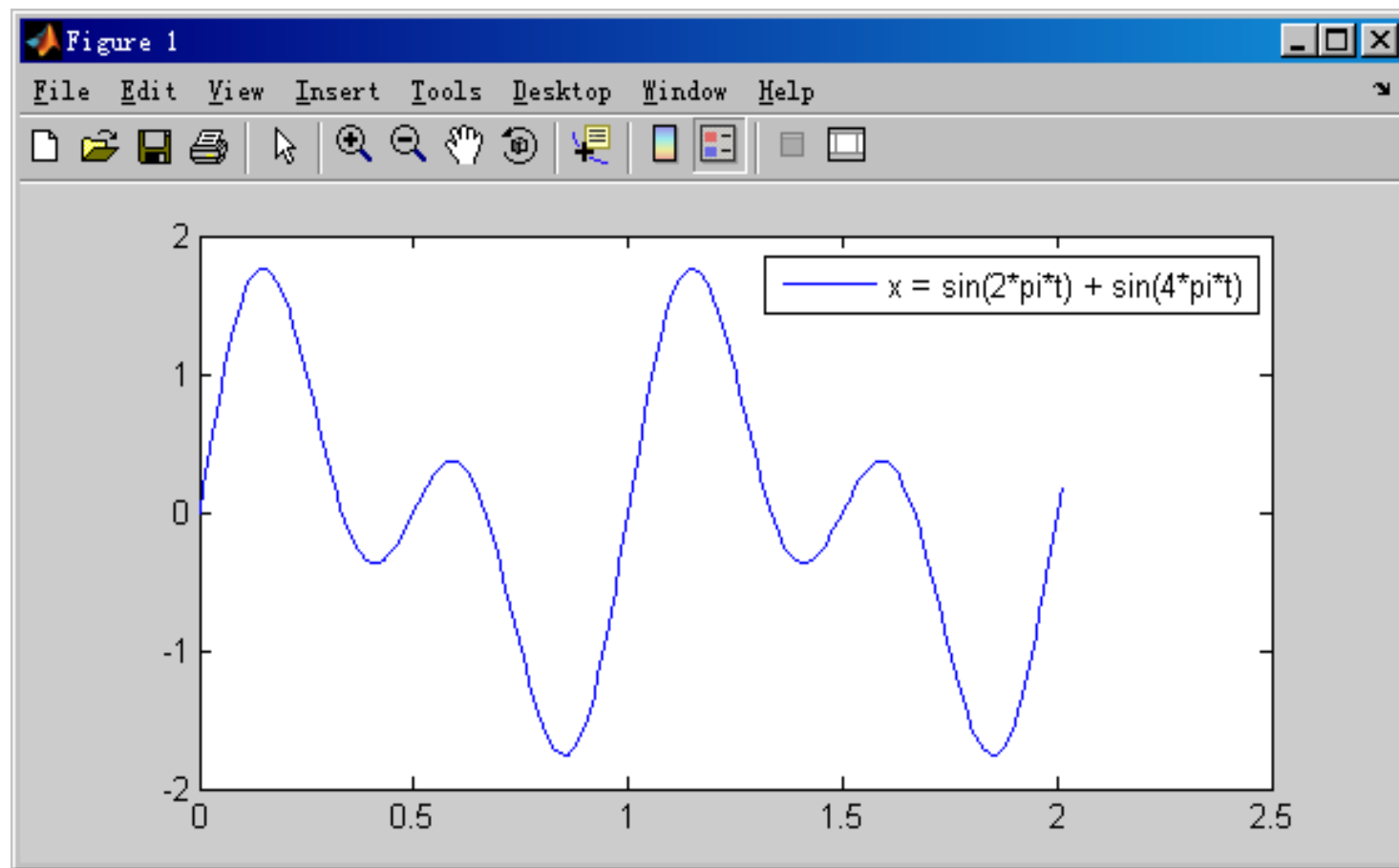


6、算术运算（续）

【例5-2】 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)');
```


6、算术运算（续）





6、算术运算（续）

【例5-2】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
```



7、关系运算

Matlab提供了6种关系运算符：

<、>、<=、>=、==、~=（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。

7、关系运算（续）

【例】 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

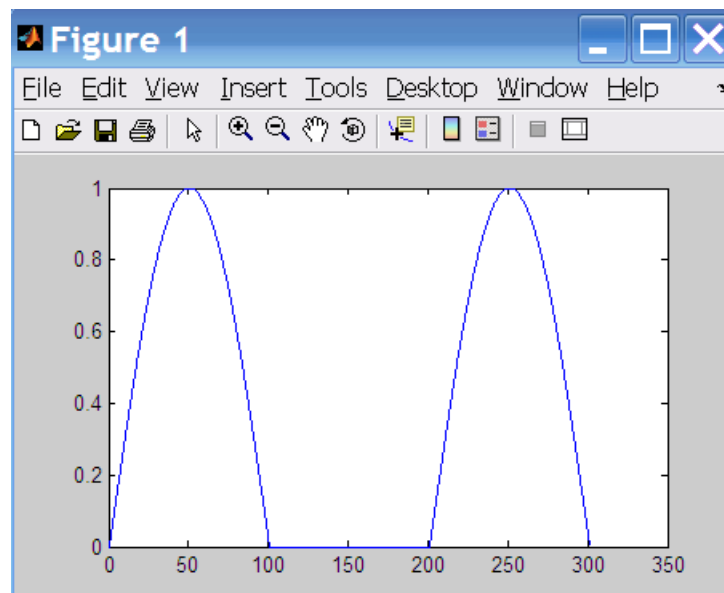
1 0 0 0 0

1 1 0 0 1

0 0 0 0 0

【例】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
```





8、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。



【例】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
```



7. 数据分析与统计

■ 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。



【例7-1】 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
```



9. 数据分析与统计

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的每一列的最大值，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。



【例7-2】 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
```



9. 数据分析与统计

■ 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。



9. 数据分析与统计

■ 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。



2.3 字符串 (string、array of characters)

■ 关于字符串

- MATLAB处理字符(Characters)与字符串(Strings)的相关指令大部分都放在下列目录之中：

{MATLAB根目录}\toolbox\matlab\strfun

- 其中的「strfun」就是代表「String Functions」。若要查询与字符和字符串相关的指令，可在 MATLAB 下输入：

help strfun 或是 help strings



2.3 字符串的概念

- 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）。

2.3 字符串

- 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3-2】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!



2.3 字符串

- 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3-2】 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
```

2.3 字符串

- **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3-2】: 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!
```



2. 字符串的判断

- **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3-4】：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
```

3. 一个字符数组变量存储多行字符串

- 第一种方法是使用二维字符数组 (Two Dimensional Character Arrays)
- 必须先确认每个字符串 (即每一行) 的长度一样, 否则就必须在短字符串结尾补齐空格

【例3-5】：多行字符串变量

```
departments = ['ee '; 'cs '; 'econ']
```

* 注意上述语句中空格字符的使用。

```
departments =
```

```
ee
```

```
cs
```

```
econ
```

3、一个变量存储多个字符串（续）

■ 用char指令存储多字符串

```
departments = char('ee', 'cs', 'econ')    % 注意空格及「,」的使用
```

得到结果和上例的一样;

■ 从二维字符数组访问字符串时，切记要使用 **deblank** 指令来清除字符串尾部的空格字符

【例3-5】：使用deblank命令清除字符串尾部空格

```
departments = char('ee', 'cs', 'econ');  
dept1 = departments(1,:);           % (1,:)代表第一行的全部元素  
dept2 = deblank(dept1);              % 使用 deblank 指令清除尾部的空格字符  
len1 = length(dept1)                 % 显示变量 dept1 的长度=4  
len2 = length(dept2)                 % 显示变量 dept2 的长度=2
```



4.字符串的操作

- **strcmp** 指令: 用于比较字符串的内容的异同

【例3-6】：字符串比较

```
str1 = 'today';  
str2 = 'tomorrow';  
str3 = 'today';  
out1 = strcmp(str1, str2)    % 比较字符串 str1 和 str2  
out1 = 0                    %表示字符串 str1 和 str2不同  
out2 = strcmp(str1, str3)    % 比较字符串 str1 和 str3  
out2 = 1                    %表示字符串 str1 和 str2相同
```

- 不相等返回0,相等返回1



5. 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



5. 空数组（续）

■ 数组维数的减小

■ 删除数组的某列和行

```
>>a = ones(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