



Matlab语言及其应用

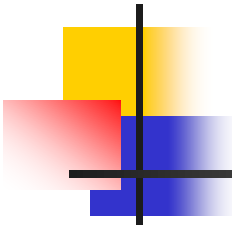
第2讲



第2章 Matlab数据及其运算

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。



2.1 数值表示、变量及表达式

■ 数值的记述

Matlab的数只采用习惯的十进制表示，可以带小数点和负号；其缺省的数据类型为**双精度浮点型**（double）。

例如：3 -10 0.001 1.3e10 1.256e-6

■ 变量命名规则

- 变量名、函数名对字母的大小写是敏感的。如 **myVar** 与 **myvar** 表示两个不同的变量。
- 变量名**第一个字母必须是英文字母**。
- 变量名可以包含**英文字母、下划线和数字**。
- 变量名**不能包含空格、标点**。
- 变量名最多可包含**63个字符**（6.5及以后的版本）。

2.1 数值表示、变量及表达式（续）

■ Matlab预定义的变量

变量名	意义
ans	最近的计算结果的变量名
eps	MATLAB定义的正的极小值=2.2204e-16
pi	圆周率 π
inf	∞ 值，无限大
i或j	虚数单元， sqrt(-1)
NaN	非数，0/0、 ∞/∞

[[说明]]

- 每当MATLAB启动完成，这些变量就被产生。
- MATLAB中，被0除不会引起程序中断，给出报警的同时用**inf**或**NaN**给出结果。
- 用户只能临时覆盖这些预定义变量的值，**Clear**或**重启MATLAB**可恢复其值。

2.1 数值表示、变量及表达式（续）

运算符和表达式

运算	数学表达式	MATLAB运算符	MATLAB表达式
加	$a+b$	+	$a+b$
减	$a-b$	-	$a-b$
乘	$a \times b$	*	$a*b$
除	a/b 或 $a \setminus b$	/或\	a/b 或 $a \setminus b$
幂	a^b	^	a^b

〔说明〕

- Matlab用“\”和“/”分别表示“左除”和“右除”。对标量而言，两者没有区别。对矩阵产生不同影响。
- MATLAB表达式的书写规则与“手写方式”几乎完全相同。
- 表达式按与常规相同的优先级自左至右执行运算。
- 优先级：指数运算级别最高，乘除次之，加减最低。
- 括号改变运算的次序。

2.1 数值表示、变量及表达式（续）

■ 复数及其运算

■ MATLAB中复数的表达： $z=a+bi$ ，其中a、b为实数。

■ MATLAB把复数作为一个整体，象计算实数一样计算复数。

【例2.3-1】复数 $z_1=3+4i$ ， $z_2=1+2i$ ， $z_3=2e^{\frac{\pi}{6}i}$

计算

$$z = \frac{z_1 z_2}{z_3}$$

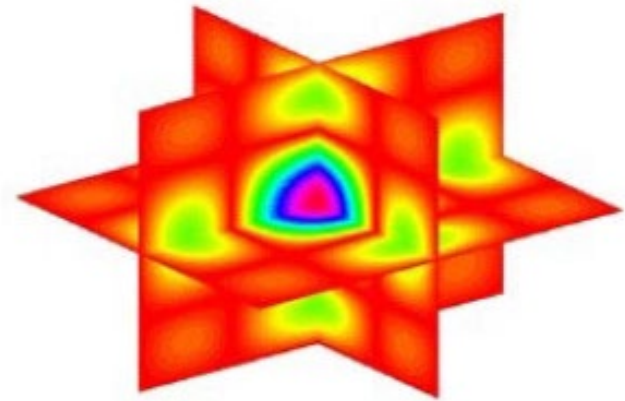
$z_1=3+4*i$, $z_2=1+2*i$, $z_3=\exp(i*pi/6)$, $z=z_1*z_2/z_3$

$z_real=\text{real}(z)$, $z_image=\text{imag}(z)$,

$z_angle=\text{angle}(z)$, $z_length=\text{abs}(z)$,

2.2 Matlab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
- 数组运算
- 多维数组



2.2.1. 数组(array)的概念

■ 数组定义:

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 1×1 的数组。

对m行、n列的2维数组a:

计为 $m \times n$ 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



2.2.1. 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。

2.2.1、数组(array)的概念 (续)

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

行向量

列向量

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

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2-1】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2-2】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
```

2.2.2、创建一维数组变量（续）

【例2-3】 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**。



2.2.2、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**元素的行向量。

【例2-4】 键入并执行**x=linspace(1,2,5)**

x=1.0000 1.2500 1.5000 1.7500 2.0000

同学们可以在实验时察看**x=linspace(1,2)**执行结果。



2.2.2、创建一维数组变量（续）

■ 第四种方法：利用函数logspace

通过实验认识该函数的功能。

■ 列向量的创建

■ 使用方括号 “[]” 操作符，使用分号 “;” 分割行。

【例2-5】键入并执行 `x= [1; 2; 3]`

```
X=1  
  2  
  3
```

■ 使用冒号操作符

【例2-6】键入并执行 `x= (1:3)'` % “'” 表示矩阵的转置

2.2.2、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2-7】 键入并执行

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



2.2.2、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”命令察看。
- 操作一：使用“Workspace”窗口察看变量
- 操作二：使用“whos”命令察看变量

2.2.3、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
```

【例3-2】键入并执行`a2=[1:3;4:6;7:9]` %结果同上

2.2.3、创建二维数组变量（续）

【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];
```

■ 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**

☞ “help elmat”获得基本的矩阵生成和操作函数列表

【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)
```

2.2.3、创建二维数组变量（续）

【例3-5】 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**。

4、数组元素的标识与寻访

数组元素的标识

“全下标 (index)” 标识

经典数学教科书采用“全下标”标识法：每一维对应一个下标。

- 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

所谓“单下标”标识就是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
```

```
>>a(3:end-1)
```

```
ans =
```

```
5.5000 7.7500
```

🔔 函数end作为参数使用，返回最后一个元素的下标

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



4、数组元素的标识与寻访（续）

■ 双下标到单下标的转换

【例4-5】 sub2ind函数-双下标转换为单下标

```
>>A = [17 24 1 8; 2 22 7 14; 4 6 13 20];
```

```
>>A(:, :, 2) = A - 10
```

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

```
>>sub2ind(size(A),2,1,2)
```

```
>>A(14)
```



4、数组元素的标识与寻访（续）

■ 单下标到双下标的转换

【例4-6】 ind2sub函数-双下标转换为单下标

```
>>b = zeros(3);
```

```
>>b(:) = 1:9
```

```
>>IND = [3 4 5 6]
```

```
>>[I,J] = ind2sub(size(b),IND)
```



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）

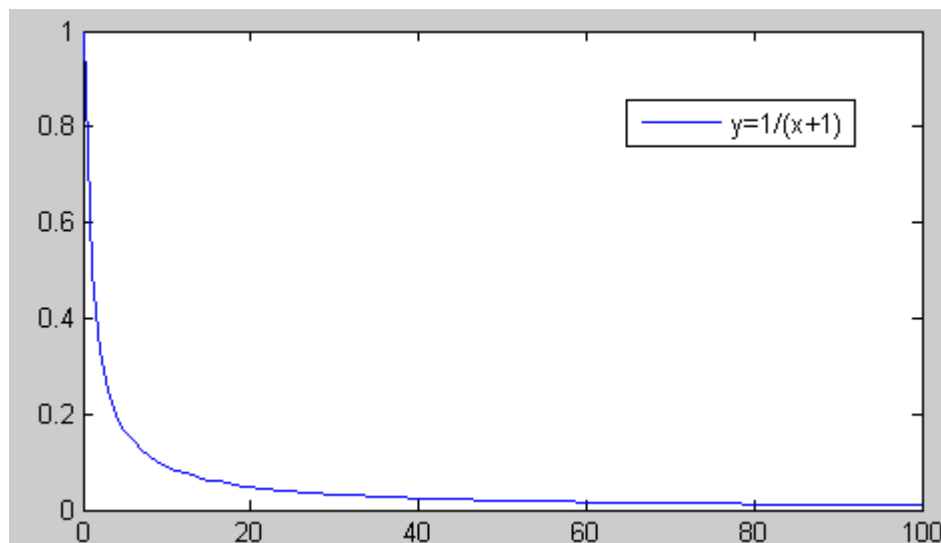
【例5-2】 画出 $y=1/(x+1)$ 的函数曲线， $x \in [0, 100]$ 。

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

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

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

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



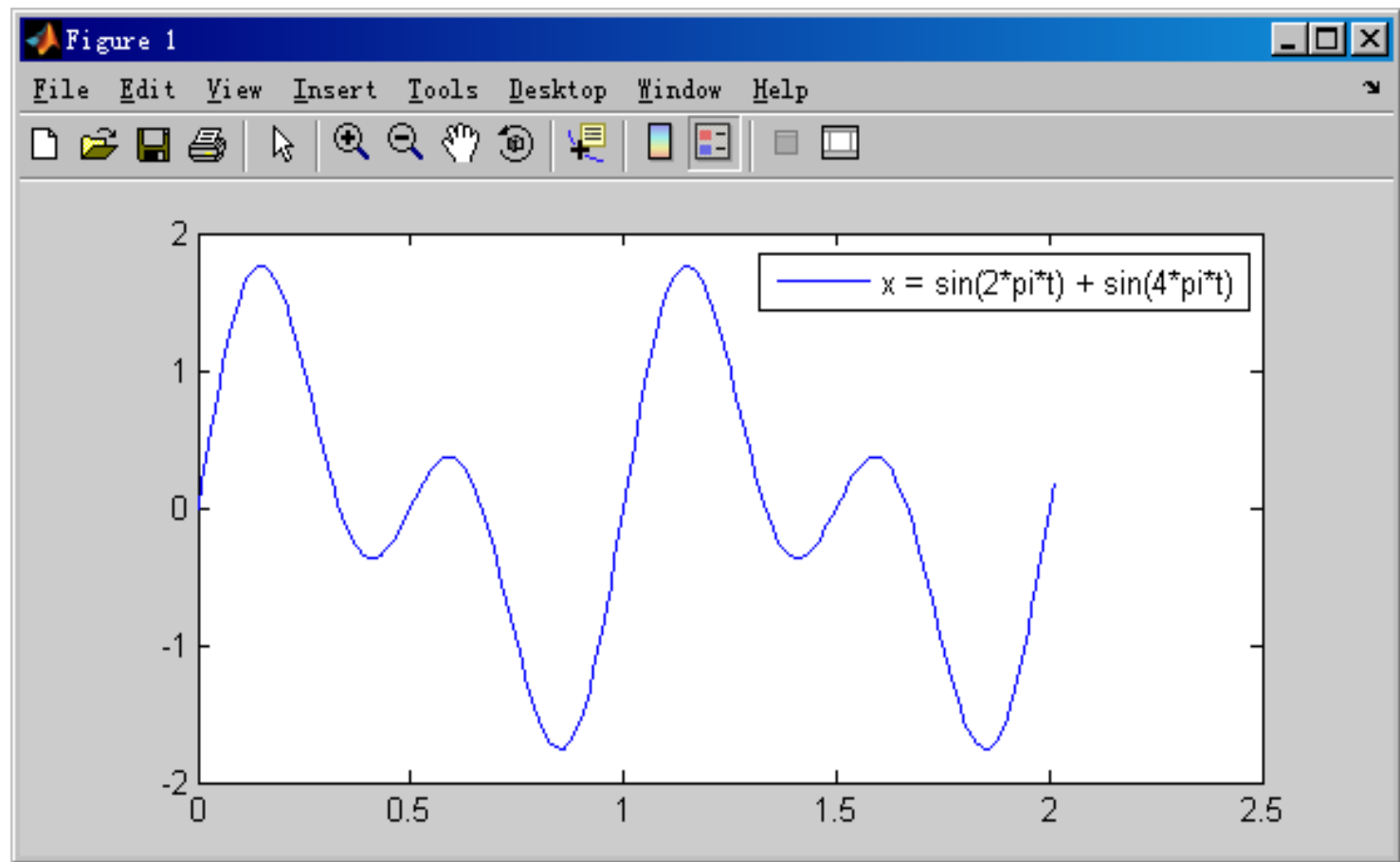


6、算术运算（续）

【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



8、逻辑运算

Matlab提供了3种逻辑运算符：

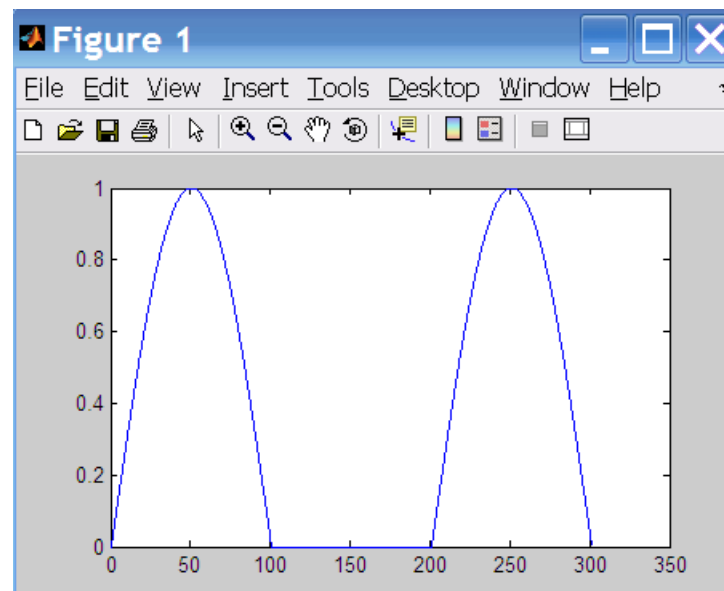
&（与）、|（或）、~（非）

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】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
```





【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。