

第 2 章



MATLAB 矩阵与数组的应用

MATLAB 名字来源于矩阵实验室，其所有代数计算都基于矩阵运算的处理工具，它把每个变量全部看成矩阵，即使是一个常数，也看作是一个 1×1 矩阵。数组是可存储任何类型的一维、二维或多维的数据结构，包括数字、字符串、逻辑值和结构体等。矩阵是特殊的数组，仅存储二维数值型数据，常用于数学中的线性代数运算，使用时无须预先指定维数，矩阵和数组在数学运算上有不同的结构。

2.1 常量和变量

变量是以占位符形式引用计算机内存地址的名字，常用于运行时用户输入的数据及存储的运算结果。MATLAB 计算过程的数值变量可以直接使用，但符号变量需要先定义再使用。

2.1.1 常量

常量是程序运行过程中不变化的量，MATLAB 的系统常量表示如表 2.1 所示。

表 2.1 常量表示

常 量	说 明
pi	圆周率 π 的双精度浮点
Inf	无穷大， ∞ 写成 Inf； $-\infty$ 为 -Inf
NaN	不定式，代表“非数值量”，通常由 0/0 或 Inf/Inf 运算得出
eps	正的极小值， $\text{eps}=2^{-32}$ ，大约是 2.22×10^{-16}
realmin	最小正实数；realmax 为最大正实数
i、j	若 i 和 j 不被定义，则它们表示纯虚数，即 $i/j=\text{sqrt}(-1)$

说明：若定义了与系统同名常量或变量，自定义则覆盖系统值。例如：若定义了 i、j 做循环变量，则原 i 或 j 纯虚数将不起作用。因此，定义时尽量避免与系统常量重名，若已经改变可通过 clear 命令清除变量名恢复其初始值，也可以通过重新启动 MATLAB 恢复原有值。

2.1.2 变量

变量工作区有导入数据、保存工作区、新建变量、打开变量和清除工作区菜单的功能，其中单击“新建变量”命令则打开一个二维表，类似 Excel 表，默认文件名是 unnamed1、unnamed2、…，如图 2.1 所示。

	1	2	3	4	5	6	7	8
1	100	200	567					
2	2010	2019	123					
3								
4								

图 2.1 新建变量

使用该方法适合成批导入变量到 MATLAB 中。

2.1.3 变量使用规则

MATLAB 变量名、函数名及文件名由字母、数字或下画线组成，字母大小写不同，如 MyVar 与 myvar 表示两个不同的变量，若未加变量名，系统使用 ans 作为变量名。基本规则包括：

- (1) 变量名要避免与系统预定义的变量名、函数名、保留字同名；
- (2) 变量名第一个字母必须是英文字母；
- (3) 变量名可以包含英文字母、下画线和数字；
- (4) 变量名不能包含空格、标点符号和特殊字符；
- (5) 变量名长度不能大于 63 个字符；
- (6) 若运算结果没有赋任何变量，则系统将其赋给 ans 特殊变量名，它只保留最新值。

2.1.4 全局变量

全局变量的作用域在整个 MATLAB 工作区内有效，所有函数都能对它进行存取和修改。若在函数文件中定义为局部变量，则只在本函数内有效，该函数返回后，这些变量会自动在工作区中清除，它与文本文件是不同的。

语法格式：

```
global<变量名> %定义一个全局变量
```

说明：

(1) 若一个函数内的变量没有特别声明，则该变量只在函数内部使用，即为局部变量，若只在某个内存空间使用一次，不建议使用全局变量；如果两个或多个函数共用一个变量，或子程序与主程序有同名变量（注意不是参数），可以用 global 声明为全局变量。全局变量的使

用可以减少参数传递，合理利用全局变量可以提高程序执行的效率。

(2) 由于各个函数之间、命令行窗口工作区、内存空间都是独立的，因此，在一个内存空间里声明 `global`，在另一个内存空间里使用时需要再次声明 `global`，各内存空间声明仅需一次。当全局变量影响内存空间变量时，可使用 `clear` 命令清除变量名。

(3) 若需要使用其他函数的变量，则需要在主程序与子程序中分别声明全局变量的方式实现变量传递，否则函数体内使用的都为局部变量。

(4) 函数较多时，全局变量将给程序调试和维护带来不便，一般不使用全局变量。如果必须要用全局变量，则原则上全部用大写字母表示，以免和其他变量混淆。

2.1.5 数据类型

MATLAB 中有 15 种基本数据类型，分别有：单精度浮点类型、双精度浮点类型、逻辑类型、字符串类型、单元数组类型、结构体类型、函数句柄类型和 8 种整型，共计 15 种基本数据类型，如图 2.2 所示。

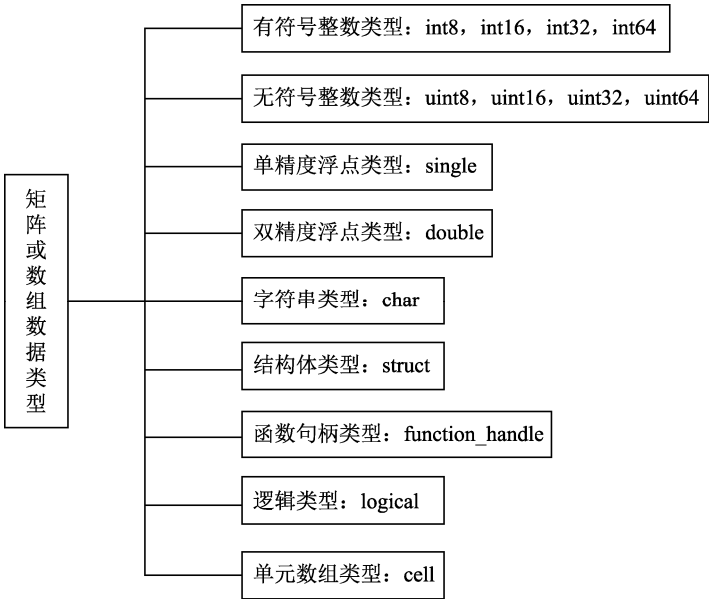


图 2.2 MATLAB 数据类型

其中：整数对含有小数的数据自动四舍五入处理，使用有符号和无符号整型变量可节约内存空间；浮点数将所有的数都看作是双精度变量，即：直接输入的变量属于 `double` 类型，若需要创建 `single` 类型变量，需要使用转换函数。其他类型数据可以利用转换函数转换为需要的类型，其表示方法如表 2.2 所示。

表 2.2 字符类型表示

表 示	说 明	表 示	说 明
uchar	无符号字符	uint16	16位无符号整数
schar	有符号字符	uint32	32位无符号整数
int8	8位有符号整数	uint64	64位无符号整数
int16	16位有符号整数	float32	32位浮点数
int32	32位有符号整数	float64	64位浮点数
int64	64位有符号整数	double	64位双精度数
uint8	8位无符号整数	single	32位浮点数

【实战练习 2-1】变量的应用

在命令行窗口建立多种类型变量，并运行、查看和保存。

编程代码如下：

```
a1=int8(10); a2=int16(-20); a3=int32(-30); a4=int64(40)
b1=uint8(50); b2=uint16(60); b3=uint32(70); b4=uint64(80)
c1=single(-90.99); d1=double(3.14159); f1='Hello'
g1.name='jiang'; h1=@sind; i1=true; j1 {2,1}=100;
```

输入 whos（查看内存变量）后的结果：

Name	Size	Bytes	Class	Attributes
a1	1x1	1	int8	
a2	1x1	2	int16	
a3	1x1	4	int32	
a4	1x1	8	int64	
ans	1x1	8	double	
b1	1x1	1	uint8	
b2	1x1	2	uint16	
b3	1x1	4	uint32	
b4	1x1	8	uint64	
c1	1x1	4	single	
d1	1x1	8	double	
f1	1x5	10	char	
g1	1x1	186	struct	
h1	1x1	32	function_handle	
i1	1x1	1	logical	
j1	2x1	128	cell	

说明：当双精度浮点数参与运算时，返回值的类型依赖于参与运算中的其他数据类型。当双精度浮点数与逻辑类型、字符串类型进行运算时，返回结果为双精度浮点类型，而与整数进行运算时返回结果为相应的整数类型，与单精度浮点类型运算时返回单精度浮点类型；单精度浮点类型与逻辑类型、字符串类型和任何浮点类型进行运算时，返回结果都是单精度浮点类型。

例如以下代码：

```
clc; b=int16(23);c=6.28;z=b+c
class(z)                                %判断 z 矩阵类型
```

运行结果：

```
z=29
ans = int16
```

说明：单精度浮点类型数据不能和整型数据进行算术运算，整数只能与相同类型的整数或双精度标量值组合使用。例如以下代码：

```
a=single(3.14); b=int16(23); c=a+b
```

运行结果：

显示“错误使用+整数只能与同类的整数或双精度标量值组合使用”的信息提示。

2.1.6 常用标点符号的使用

MATLAB 标点符号包括标识符、分隔符、运算符及结束符等，常用的标点符号如表 2.3 所示。

表 2.3 常用标点符号的功能

名 称	符 号	含 义
空格		输入变量之间的分隔符以及数组行元素之间的分隔符
逗号	,	输入变量之间的分隔符或矩阵行元素之间的分隔符，也可用于显示计算结果分隔符
点号	.	数值中的小数点
分号	;	用于矩阵或数组元素行之间的分隔符或不显示计算结果
冒号	:	生成一维数组，表示一维数组的全部元素或多维数组的某一维的全部元素
百分号	%	注释的前面，在它后面的命令不需要执行
单引号	' '	字符串变量需要加单引号
圆括号	()	引用矩阵或数组元素；用于函数输入变量列表；用于确定算术运算的先后次序
方括号	[]	构成向量和矩阵；用于函数输出列表
花括号	{ }	构成元胞数组
下画线	-	变量、函数或文件名中的连字符
续行号	...	将一行长命令分成多行时尾部的符号
At号	@	放在函数名前形成函数句柄；放在文件夹名前形成用户对象类目录

2.2 矩阵操作

MATLAB 矩阵操作是最基本的数据对象操作，包括构建矩阵、获取矩阵元素、通过下标（行、列索引）引用矩阵的元素等，构建矩阵使用方括号（[]）加入相应的元素，元素间用空

格或逗号分隔。

2.2.1 创建矩阵的方法

MATLAB 的矩阵包括实数、复数，可从键盘输入，也可通过计算和函数产生。

【实战练习 2-2】创建实数和复数矩阵

为指定数值创建矩阵 A 和矩阵 B 。

$$A = \begin{bmatrix} 10 & 20 & 30 \\ 4 & 5 & 6 \\ 7 & -1 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 1+2i & 2+5i \\ 3+7i & 5+9i \\ i & 8i \end{bmatrix}$$

编程代码如下：

```
A=[10 20 30;4 5 6;7 -1 0]
B= [1+2i, 2+5i; 3+7i,5+9i; i, 8i]
```

运行结果：

```
A = 10    20    30
     4     5     6
     7    -1     0
B = 1.0000 + 2.0000i    2.0000 + 5.0000i
     3.0000 + 7.0000i    5.0000 + 9.0000i
     0.0000 + 1.0000i    0.0000 + 8.0000i
```

2.2.2 创建向量的方法

MATLAB 将数组或向量看作 $1 \times n$ 或 $n \times 1$ 的矩阵，数组、向量和二维矩阵表示在本质上没有任何区别，它们的维数都是 2，均以矩阵形式保存。

1. 用线性等间距生成向量矩阵

语法如下：

```
(start:step:end) %start 为起始值，step 为步长，end 为终值
```

例如：a=[1:3:15]

```
结果：a = 1    4    7    10    13
```

2. 线性向量

语法如下：

```
linspace(n1,n2,k) %产生线性向量，其中 n1 为初始值，n2 为终值，k 为个数
```

例如：b=linspace(3,18,4)

```
结果：b = 3    8    13    18
```

3. 对数向量

语法如下：

```
logspace(n1,n2,n)           %产生对数向量，其中行向量的值为  $10^{n1} \sim 10^{n2}$ ，数据个数为
                             %n，默认 n 为 50
```

例如： `c=logspace(1,3,3)` %对数向量常用于建立对数频域坐标

结果： `c= 10 100 1000`

2.2.3 常用特色矩阵

MATLAB 的常用特色矩阵是由系统产生的，包括全零矩阵、全 1 矩阵、对角矩阵、单位矩阵、杨辉三角矩阵及魔方矩阵等，例如：

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

全零矩阵输入 `zeros(3,4)`，其中，`zeros(m,n)`为 $m \times n$ 矩阵，若只有一个下标值，则表示行、列相同的方阵。

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

全 1 矩阵输入 `ones(3,4)`，其中，`ones(m,n)`为 $m \times n$ 矩阵，若只有一个下标值，则表示为方阵。

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

单位矩阵矩阵输入 `eye(3,4)`，其中，`eye(m,n)`为 $m \times n$ 矩阵，若只有一个下标值，则表示为方阵。

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

对角矩阵输入 `V=[1 2 3]; diag(v)`；其中，`v` 为对角元素值。

特色矩阵函数如表 2.4 所示。

说明：

- (1) `magic(n)`表示为魔方矩阵，是指行、列、对角线元素的和相等，它必须是 n 阶方阵。
- (2) `pascal(n)`表示杨辉三角矩阵（也称为帕斯卡三角矩阵），是 $(x+y)^n$ 的系数随 n 增大的三角形表。
- (3) `toeplitz(m,n)`表示托普利兹矩阵，是除第 1 行第 1 列元素外，每个元素与它的左上角元素相等。

(4) `triu(A)`表示上三角矩阵,是保存矩阵 A 上三角矩阵为原值,下三角为 0 的矩阵。

(5) `triu(A,k)`表示将矩阵 A 的第 k 条对角线以上的元素变成上三角矩阵。

表 2.4 特色矩阵函数

函 数 名	含 义	函 数 名	含 义
<code>zeros(m,n)</code>	m×n零矩阵	<code>company(m,n)</code>	m×n伴随矩阵
<code>zeros(m)</code>	m×m零矩阵	<code>pascal(n)</code>	n×n帕斯卡三角矩阵
<code>eye(m,n)</code>	m×n单位矩阵	<code>magic(n)</code>	n×n魔方矩阵
<code>eye(m)</code>	m×m单位矩阵	<code>diag(V)</code>	以V为对角元素对角矩阵
<code>ones(m,n)</code>	m×n全1矩阵	<code>tril(A)</code>	矩阵A的下三角矩阵
<code>ones(m)</code>	m×m全1矩阵	<code>triu(A)</code>	矩阵A的上三角矩阵
<code>rand(m,n)</code>	m×n的均匀分布的随机矩阵	<code>rot90(A)</code>	旋转90度A矩阵
<code>rand(size(A))</code>	[0,1]区间A维均匀随机矩阵	<code>flipud(A)</code>	矩阵A的上下翻转
<code>hilb(n)</code>	n阶希尔伯特矩阵	<code>fliplr(A)</code>	矩阵A的左右翻转
<code>toeplitz(m,n)</code>	托普利兹矩阵	<code>Sparse(A)</code>	稀疏矩阵

【实战练习 2-3】创建特色矩阵

求 3×4 的全 1 矩阵、4×5 均匀分布的随机矩阵、上三角矩阵、魔方矩阵、杨辉三角矩阵、方程 $x^4+3x^3+7x^2+5x-9=0$ 的伴随矩阵,并求 4 行、4 列的托普利兹矩阵。

编程代码如下:

```
Y=ones(3,4)           %3×4 的全 1 矩阵
Z=rand(4,5)           %4×5 的均匀分布的随机矩阵
W=triu(Z)             %上三角矩阵
K=magic(4)            %魔方矩阵必须是方阵
L=pascal(4)           %杨辉三角矩阵必须是方阵
A=[1 3 7 5 -9];       %方程系数
B=company(A)          %求伴随矩阵
M=toeplitz(1:4)       %托普利兹矩阵
```

运行结果:

```
Y=  1      1      1      1
    1      1      1      1
    1      1      1      1
Z=  0.8147    0.6324    0.9575    0.9572    0.4218
    0.9058    0.0975    0.9649    0.4854    0.9157
    0.1270    0.2785    0.1576    0.8003    0.7922
    0.9134    0.5469    0.9706    0.1419    0.9595
W=  0.8147    0.6324    0.9575    0.9572    0.4218
    0          0.0975    0.9649    0.4854    0.9157
    0          0          0.1576    0.8003    0.7922
```



```

      0      0      0      0.1419      0.9595
K=  16      2      3      13
    5     11     10      8
    9      7      6     12
    4     14     15      1
L=   1      1      1      1
    1      2      3      4
    1      3      6     10
    1      4     10     20
B=  -3     -7     -5      9
    1      0      0      0
    0      1      0      0
    0      0      1      0
M=   1      2      3      4
    2      1      2      3
    3      2      1      2
    4      3      2      1

```

2.2.4 稀疏矩阵

若矩阵中非零元素的个数远远小于矩阵元素的总数，且非零元素的分布没有规律，则定义矩阵为稀疏矩阵(sparse matrix)。

1. 创建稀疏矩阵

创建稀疏矩阵语法格式：

```

S= sparse(A)           %将矩阵 A 中任何零元素去除，非零元素及其列组成矩阵 S
S= sparse(i,j,s,m,n,maxn)

```

其中：由向量 i, j, s 生成一个 $m \times n$ 含有 $\max n$ 个非零元素的稀疏矩阵 S ，并且有 $S(i(k), j(k)) = s(k)$ 。向量 i, j 和 s 有相同的长度。对应向量 i 和 j 的值 s 中任何零元素将被忽略。 s 中在 i 和 j 处的重复值将被叠加。稀疏矩阵存储特点是所占内存少，运算速度快。

说明：

- (1) 创建的稀疏矩阵只显示非零元素行、列值，可用命令 `full(S)` 显示所有矩阵元素。
- (2) 如果 i 或 j 任意一个大于最大整数值范围 ($2^{31}-1$)，则稀疏矩阵不能被创建。
- (3) 若 $S = \text{sparse}(i, j, s)$ ，则使 $m = \max(i)$ 和 $n = \max(j)$ ，在 s 中零元素被移除前计算最大值， $[i \ j \ s]$ 中其中一行可能为 $[m \ n \ 0]$ 。
- (4) `sparse([], [], [], m, n, 0)` 表示生成 $m \times n$ 所有元素都是零的稀疏矩阵。
- (5) 当构造矩阵比较大，而非零元素位置又比较有规律时，可以用 `sparse()` 函数构建稀疏矩阵。

【实战练习 2-4】创建稀疏矩阵

已知 $i=[2 \ 2 \ 3 \ 3 \ 3 \ 4]$ ， $j=[2 \ 4 \ 3 \ 2 \ 1 \ 4]$ ， $A=[2 \ 3 \ 7 \ 1 \ 4 \ 6]$ ，创建稀疏矩阵。

编程代码如下：

```
i=[2 2 3 3 3 4]; j=[2 4 3 2 1 4];
A=[2 3 7 1 4 6];
S= sparse(i,j,A,4,4)
A=full(S)
```

结果:

```
S= (3,1)      4
    (2,2)      2
    (3,2)      1
    (3,3)      7
    (2,4)      3
    (4,4)      6
A=  0      0      0      0
    0      2      0      3
    4      1      7      0
    0      0      0      6
```

2. 创建带状稀疏矩阵

带状稀疏矩阵语法格式:

```
S=spdiags(A,d,m,n) %生成 m×n 所有元素，非零元素均在对角线上，且对角线元
                  %素有规律的稀疏矩阵
```

其中: A 表示全元素矩阵, d 是长度为 p 的整数向量, 指定 S 矩阵对角线位置, m, n 表示构造的系数矩阵的行列数。

【实战练习 2-5】创建对角稀疏矩阵

根据产生的随机矩阵创建一个 5×5 的对角稀疏矩阵。

编程代码如下:

```
A=rand(5);
A= floor(100*A);
S=spdiags(A,[0 1],5,5);
S1=full(S)
```

结果:

```
S1=34      83      0      0      0
    0      19      58      0      0
    0      0      25      54      0
    0      0      0      61      91
    0      0      0      0      47
```

3. 稀疏矩阵操作函数

稀疏矩阵操作函数语法格式:

```
nnz(S): %查看非零元素的个数
nonzeros(S) %获取非零元素的值
nzmax(S) %获取存储非零元素的空间长度
spy(S): %稀疏矩阵非零元素的图形表示
```

【实战练习 2-6】创建带状稀疏矩阵

产生一个 5×5 带状稀疏矩阵, 获取非零元素的个数 a 、非零元素的值 B 及非零元素的空间长度 n , 并进行图形化显示该稀疏矩阵。

编程代码如下:

```
A=rand(5);
B= floor(100*A);
S=spdiags(B,[0 1],5,5)
S1=full(S)
a=nnz(S1)
b=nonzeros(S1) '
c=nzmax(S1)
spy(S)
```

结果:

```
S1 =  96    67     0     0     0
      0    54    39     0     0
      0     0    52    36     0
      0     0     0    23    98
      0     0     0     0    48

a =      9
b =      96
      67
      54
      39
      52
      36
      23
      98
      48
c =      25
```

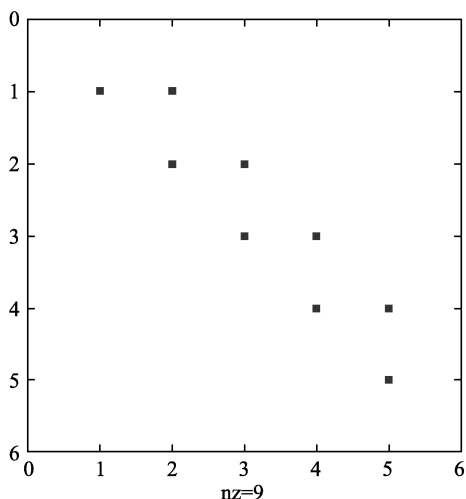


图 2.3 稀疏矩阵非零元素图形表示

2.2.5 矩阵拆分 (分解)

矩阵拆分能够将一个复杂矩阵分解成多个较为简单的矩阵, 简单矩阵在计算上更为高效, 因此矩阵拆分方法广泛应用于数据分析。矩阵拆分是按照给定的行、列索引提取元素, 主要包括:

- (1) $A(m,n)$: 提取矩阵 A 第 m 行, 第 n 列元素;
- (2) $A(:,n)$: 提取矩阵 A 第 n 列元素;
- (3) $A(m,:)$: 提取矩阵 A 第 m 行元素;
- (4) $A(m1:m2,n1:n2)$: 提取矩阵 A 第 $m1$ 行到第 $m2$ 行和第 $n1$ 列到第 $n2$ 列的所有元素 (提取子块);
- (5) $A(:)$: 矩阵 A 按列元素输出;

(6) 矩阵扩展：如果在矩阵中赋值给一个不存在的地址，则该矩阵会自动扩展行、列数，并在该位置上添加这个数，且在其他没有指定的位置补零。

【实战练习 2-7】矩阵拆分的应用

根据给定的矩阵 A ，输出矩阵拆分的行和列元素。

编程代码如下：

```
A=[1 2 3 4 5;6 7 8 9 10;11 12 13 14 15]
B=A(2,:)           %取第 2 行元素
C=A(1:2,3:4)       %取第 1~2 行的第 3~4 列元素
```

结果：

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

【实战练习 2-8】矩阵扩展的应用

根据给定的 2×3 矩阵 A ，扩展为 3×4 矩阵。

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

编程代码如下：

```
A=[1 2 3;4 5 6];           %建立矩阵
A(3,4)=20                  %赋值给一个不存在的地址
```

结果：

```
A =   1     2     3     0
      4     5     6     0
      0     0     0    20
```

2.3 矩阵基本运算

矩阵的基本运算有加、减、乘、点乘、乘方、左除、右除、求逆。其中，加、减、乘与线性代数中一致，除法分为左除、点左除、右除和点右除。此外，MATLAB 系统还提供了特色矩阵运算函数，包括求矩阵元素和与积、元素累加和与累乘积、平均值、中值、最大值、最小值，还有求矩阵的秩、逆、迹、条件数及特征值等。

2.3.1 矩阵常用运算

矩阵常用运算包括算术运算、关系运算及逻辑运算。

1. 算术运算

常用的算术运算符，如表 2.5 所示。

表 2.5 常用的算术运算符

运算符	说明	运算符	说明
+	矩阵相加	\	矩阵左除
-	矩阵相减	.\	点左除
*	矩阵相乘	./	点右除
.*	点乘	^	矩阵乘方
/	矩阵右除	.^	点乘方

说明：

(1) 矩阵相加、减为对应元素的加、减，行、列数必须有一个相同；当矩阵与标量相加、减时，矩阵的各元素都将与该标量进行运算。

(2) 点运算是一种特殊的运算，需要在算术运算符前面加点，其“.*”、“./”、“.\”和“.^” ，分别表示点乘、点右除、点左除和点乘方。两矩阵进行点运算是指对应元素进行相关运算，要求两矩阵的维数必须相同。

(3) 左除：当方程为 $A \times X = B$ 时，其解为 $X = A \backslash B (A \setminus B)$ ，称 A 左除 B ，表示矩阵 A 的逆乘以矩阵 B ，即： $\text{inv}(A) * B$ 。

(4) 右除：当方程为 $X \times A = B$ 时，其解为 $X = B / A (B / A)$ ，称 B 右除 A ，表示矩阵 A 乘以矩阵 B 的逆，即： $A \times \text{inv}(B)$ ，当 A 为非奇异矩阵时，则： $X = A \setminus B$ ，即：

$A \setminus B$ 表示矩阵 B 中每个元素除以矩阵 A 的对应元素。

$A ./ B$ 表示矩阵 A 中每个元素除以矩阵 B 的对应元素。

(5) 一个矩阵的乘方运算可以表示成 A^x ，要求 A 为方阵， x 为标量。

【实战练习 2-9】矩阵算术运算的应用

已知矩阵 A 和 B ，求两个矩阵加、减、乘、除、点乘、点除及 2 次乘方。

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

$$B = \begin{bmatrix} 1 & 0 & 3 \\ 5 & 9 & 13 \\ 7 & 12 & 11 \end{bmatrix}$$

说明：当有多条执行语句时，可单击主菜单栏“新建脚本”命令或在命令行窗口输入 `edit`，建立 `.m` 程序文件后，在脚本编辑器中输入多条命令，单击“Run/运行”按钮即可查看结果。因此，使用脚本编写程序方式，在脚本编辑器中不需要“>>”提示符，按 `Enter` 键不会运行，且可保存修改。

编程代码如下：

```
A=[1,2,3;4,5,6;7,8,9];  
B=[1 0 3;5 9 13;7 12 11];  
C=A+B
```

```
%构建矩阵 A  
%构建矩阵 B  
%矩阵相加
```

```

D=A-B           %矩阵相减
E=A*B           %矩阵相乘
F=A.*B          %矩阵点乘
G=A/B           %矩阵右除
G1=A*inv(B)     %与 A/B 等价
H=A\B           %矩阵左除
H1=inv(A)*B     %与 H 等价
I=A./B          %点右除
J=A.\B          %点左除
K=A^2           %A 的 2 次乘方, 相当于 A*A

```

结果:

```

C =      2      2      6
      9     14     19
     14     20     20
E =     32     54     62
     71     117    143
    110     180    224
G =  -0.0909    0.3030   -0.0606
     1.0000   -0.3333    0.6667
     2.0909   -0.9697    1.3939
H =  1.0e+16 *
    -0.6305   -1.8915   -3.7830
     1.2610    3.7830    7.5660
    -0.6305   -1.8915   -3.7830
I =      1.0000      Inf      1.0000
     0.8000     0.5556     0.4615
     1.0000     0.6667     0.8182
K =      30      36      42
      66      81      96
     102     126     150

D =      0      2      0
     -1     -4     -7
      0     -4     -2
F =      1      0      9
     20     45     78
     49     96     99
G1 =  -0.0909    0.3030   -0.0606
     1.0000   -0.3333    0.6667
     2.0909   -0.9697    1.3939
H1 =  1.0e+16 *
    -0.6305   -1.8915   -3.7830
     1.2610    3.7830    7.5660
    -0.6305   -1.8915   -3.7830
J =      1.0000      0      1.0000
     1.2500     1.8000     2.1667
     1.0000     1.5000     1.2222

```

【实战练习 2-10】矩阵乘法运算的应用

根据 3×3 魔方矩阵 A 的值, 计算 $A \times A$ 、 A^2 与 $A.^2$ 。

编程代码如下:

```

A=magic(3)      %构建 3 阶魔方矩阵
A1=A*A
B1=A^2           %A 的 2 次方
L=A.^2          %A 的 2 次乘方, 加点表示对应元素平方

```

结果:

```

A =      8      1      6
      3      5      7
      4      9      2
A1 =     91     67     67
      67     91     67
      67     67     91

```

```

B1 = 91    67    67
      67    91    67
      67    67    91
L = 64     1    36
      9    25    49
      16    81     4

```

结论： $A \times A$ 等价于 A^2 ，但不等价于 $A.^2$ 。

【实战练习 2-11】矩阵左除运算的使用

利用左除运算求解方程组

$$\begin{cases} 6x_1 + 3x_2 + 4x_3 = 3 \\ -2x_1 + 5x_2 + 7x_3 = -4 \\ 8x_1 - 4x_2 - 3x_3 = -7 \end{cases}$$

编程代码如下：

```

A=[6 3 4;-2 5 7;8 -4 -3];           %方程组系数
B=[3;-4;-7];
x=A\B                                %左除求解 x

```

结果：

```

x=  0.6000
    7.0000
   -5.4000

```

2. 复数运算

MATLAB 把复数作为一个整体，如同计算实数一样计算复数，复数表达式为： $z=a+bi$ ；

其中： a 、 b 为实数， i 表示虚数。

【实战练习 2-12】矩阵复数运算的应用

已知复数 $z_1=3+4i$ ， $z_2=1+2i$ ， $z_3=2e^{i\pi/6}$ ，计算 $z=z_1*z_2/z_3$ 。

编程代码如下：

```

z1=3+4*i;
z2=1+2*i;
z3=2*exp(i*pi/6); z=z1*z2/z3

```

结果：

```

z =  0.3349 + 5.5801i

```

对于复数矩阵，常用两种输入方法，其结果相同，例如：

```

A=[1,2;3,4]+i*[5,6;7,8]
B=[1+5i 2+6i;3+7i 4+8i]

```

结果：

```

A = 1.0000 + 5.0000i    2.0000 + 6.0000i
    3.0000 + 7.0000i    4.0000 + 8.0000i
B = 1.0000 + 5.0000i    2.0000 + 6.0000i
    3.0000 + 7.0000i    4.0000 + 8.0000i

```

说明：A 与 B 的结果是相同的。

3. 关系运算

关系运算符如表 2.6 所示。

表 2.6 关系运算符

运 算 符	说 明	运 算 符	说 明
>	大于	<=	小于或等于
>=	大于或等于	==	等于
<	小于	~=	不等于

【实战练习 2-13】关系运算的应用

根据给定的向量[9,2,4]与产生的随机 3×3 矩阵进行关系运算。

编程代码如下：

```
clc                                %清除屏幕
y=[9,2,4]>5
A=rand(3)
B=A<0.5
```

结果：

```
y =      1      0      0
A = 0.3922    0.7060    0.0462
    0.6555    0.0318    0.0971
    0.1712    0.2769    0.8235
B = 1      0      1
    0      1      1
    1      1      0
```

4. 逻辑运算

逻辑运算符如表 2.7 所示。

表 2.7 逻辑运算符

运 算 符	说 明	运 算 符	说 明	运 算 符	说 明
&	与运算		或运算	~	非运算

说明：关系运算和逻辑运算结果都是逻辑值，结果元素为真用 1 表示；为假用 0 表示。“&”和“|”操作符可比较两个标量或两个同阶次矩阵。如果 A 和 B 都是 0-1 矩阵，则 A&B 或 A|B 也都是 0-1 矩阵，且 0-1 矩阵是 A 和 B 对应元素的逻辑值。一般逻辑数在条件语句和数组索引中使用。

【实战练习 2-14】逻辑运算的应用

根据给定的逻辑向量 X 和产生的随机 3×3 矩阵 K、L，进行逻辑运算。

编程代码如下：


```
X=[true,false,true]
K=rand(3)
L=rand(3)
Y1=K|L
Y2=K&~K
```

结果:

```
X= 1      0      1
K= 0.0759    0.7792    0.5688      L = 0.3371    0.3112    0.6020
    0.0540    0.9340    0.4694      0.1622    0.5285    0.2630
    0.5308    0.1299    0.0119      0.7943    0.1656    0.6541
Y1 = 1      1      1      Y2= 0      0      0
    1      1      1      0      0      0
    1      1      1      0      0      0
```

2.3.2 矩阵、向量元素的和与积运算

向量或矩阵求和与求积的函数是 `sum()` 和 `prod()`，语法格式：

```
S=sum(X)           %若 x 为向量，则求 x 的元素和；若 x 为矩阵，则返回列向量的和
S1=prod(X)         %若 x 为向量，则求 x 的元素积；若 x 为矩阵，则返回列向量的积
```

【实战练习 2-15】求矩阵元素的和与积

求一维向量 $[1,2,\dots,9]$ 与 3×3 矩阵元素 $[1\ 2\ 3;4\ 5\ 6;7\ 8\ 9]$ 和与积。

编程代码如下：

```
X1=[1 2 3 4 5 6 7 8 9]; X2=[1 2 3;4 5 6;7 8 9];
S1=sum(X1)
S2=prod(X1)
S3=sum(X2)
S4=prod(X2)
```

结果：

```
S1 =    45
S2 =   362880
S3 =    12    15    18
S4 =    28    80   162
```

2.3.3 矩阵、向量累加和与累乘积

矩阵、向量的累加和与累乘积运算分别使用函数 `cumsum()` 和 `cumprod()` 计算，语法格式：

```
A=cumsum(X)        %当 x 是向量时，返回 x 的元素累加和；若 x 为矩阵时，返回一个与 x
                   %大小相同的列累加和矩阵
A=cumprod(X)       %当 x 是向量时，返回 x 中相应元素与其之前的所有元素累乘积；若 x
                   %为矩阵，返回一个与 x 大小相同的累乘积矩阵
```

【实战练习 2-16】求矩阵列元素的累加和与累乘积

求一维向量 $X1=[1,2,\dots,9]$ 与 3×3 矩阵 $X2=[1\ 2\ 3;4\ 5\ 6;7\ 8\ 9]$ 累乘积。

编程代码如下：

```
X1=[1 2 3 4 5 6 7 8 9]; X2=[1 2 3;4 5 6;7 8 9];
A1=cumsum(X1)           %X1 求累加和
A2=cumsum(X2)           %X2 求累加和
A3=cumprod(X1)          %X1 求累乘积
A4=cumprod(X2)          %X2 求累乘积
```

结果：

```
A1 = 1    3    6   10   15   21   28   36   45
A2 = 1     2     3
      5     7     9
      12    15    18
A3 = 1    2    6   24  120  720  5040  40320  362880
A4 = 1     2     3
      4    10    18
      28    80   162
```

2.3.4 矩阵平均值和中值

求序列数据平均值和中值的函数是 `mean()` 和 `median()`，其语法格式：

```
mean(X):           %返回向量 X 的算术平均值；若 X 为矩阵，则返回列向量平均值
median(X):         %返回向量 X 的中间值；若 X 为矩阵，则返回列向量中间值
mean(X,dim):       %当 dim 为 1 时，返回 X 列向量平均值；当 dim 为 2 时，返回行向量平均值
```

【实战练习 2-17】求矩阵的平均值和中值

使用随机函数生成 100 以内 3×4 随机整数矩阵，求其平均值和中值。

编程代码如下：

```
A=floor(rand(3,4)*100) %产生一个 100 以内的  $3 \times 4$  随机整数矩阵 A
M1=mean(X)             %计算平均值
M2=median(X)           %计算列向量平均值
M3=mean(X,2)           %计算行向量平均值
```

结果：

```
A = 91    75     7    77
      28    38     5    93
      75    56    53    12
M1 = 2.3333    6.0000    4.3333    5.0000
M2 = 2     5     5     4
M3 = 2.2500
      7.0000
      4.0000
```

2.3.5 矩阵最大值、最小值与排序

MATLAB 提供了求矩阵元素的最大值、最小值和排序函数。

1. 求向量的最大值和最小值

语法格式：

```
Y=max(X)           %Y 为向量 X 的最大值，若 X 包含复数，则按模数取最大值；若 X 是矩
                   %阵，则返回矩阵每列的最大值
[Y,I]=max(X)       %Y 为最大值，I 为最大值序号。若 X 包含复数，则按其模取最大值
Y=max(X,[],dim)    %Y 为最大值，dim 表示维数，dim=1 时，同 max(X)；dim=2 时，若 X
                   %为向量直接取最大值，若 X 是矩阵，则返回矩阵每行的最大值
```

说明：求向量 X 的最小值函数是 min(X)，用法和 max(X) 完全相同。

【实战练习 2-18】求向量的最大值、最小值

求给定 X 向量 [12,56,4,0,19,100,-1,20,30] 的最大值、最小值。

编程代码如下：

```
X=[12,56,4,0,19,100,-1,20,30]    %构建向量 X
A=max(X)                         %返回向量 X 最大值
B=min(X)                         %返回向量 X 最小值
[M1,I]=max(X)                    %返回向量 X 最大值及位置
[M2,I]=min(X)                    %返回向量 X 最小值及位置
C=max(X,[],2)                   %返回向量 X 的最大值
```

结果：

```
X =    12    56     4     0    19   100    -1    20    30
A =    100
B =     -1
M1 =    100
I =     6
M2 =     -1
I =     7
C =    100
```

2. 求矩阵的最大值和最小值

【实战练习 2-19】求矩阵最大值、最小值

给定 X 矩阵，求 X 的最大值、最小值。

$$X = \begin{bmatrix} 12 & 56 & 4 \\ 0 & 19 & 100 \\ -1 & 20 & 30 \end{bmatrix}$$

编程代码如下：

```
X=[12,56,4,0,19,100,-1,20,30]    %构建矩阵 X
A=max(X)                         %返回矩阵 X 列向量最大值
B=min(X)                         %返回矩阵 X 列向量最小值
[M1,I]=max(X)                    %返回矩阵 X 列向量最大值及位置
[M2,I]=min(X)                    %返回矩阵 X 列向量最小值及位置
C=max(X,[],2)                   %返回矩阵 X 每行的最大值
```

结果:

```
X = 12    56    4
      0    19   100
     -1    20    30
A = 12    56   100
B = -1    19     4
M1 = 12    56   100
I = 1     1     2
M2 = -1    19     4
I = 3     2     1
C = 56
      100
      30
```

3. 矩阵的排序

(1) sort()函数。

排序函数 sort()可以对向量、数组或数组等元素进行升序或降序排列,当 X 是矩阵时,对 X 的每一列进行升序或降序排列;当 X 为向量或数组时,按照行进行升序或降序排列。语法格式:

```
Y1=sort(A)
Y=sort(A,dim,mode)
```

其中:若 A 是二维矩阵,当 $\text{dim}=1$ (默认)时,表示对 A 的每一列进行排序;当 $\text{dim}=2$ 时表示对 A 的每一行进行排序。Mode 表示排列方式,mode= ascend 时进行升序排列;当 mode= descend 时,进行降序排列,默认为升序排列。

【实战练习 2-20】矩阵升序和降序的应用

对给定的矩阵 $A=[74\ 22\ 82;7\ 45\ 91;53\ 44\ 8]$ 进行升序和降序排列。

编程代码如下:

```
A=[74 22 82;7 45 91; 53 44 8]
B=sort(A) %按矩阵列向量升序排列
C=sort(A,2) %按矩阵行向量升序排列
D=sort(A,2,'descend') %按矩阵行向量降序排列
```

结果:

```
A = 74    22    82
      7    45    91
      53    44     8
B = 7     22     8
      53    44    82
      74    45    91
C = 22     74     82
      7     45     91
      8     44     53
D = 82     74     22
      91     45     7
```

53 44 8

(2) `sortrows()` 函数。

`sortrows()` 函数可以使用选定的列值对矩阵行进行排序，语法格式：

```
[Y,I]=sortrows(A,Column)
```

其中： A 是待排序的矩阵； $Column$ 是列的序号，指定按照第几列进行排序，正数表示按照升序进行排列，负数表示按照降序进行排列； Y 是排列后的矩阵； I 是排序后的行在之前矩阵中的行标值。

例如：

```
E=sortrows(A,-2)                      %对矩阵 A 第 2 列进行降序排列
```

结果：

```
E =      7      45      91  
     53      44      8  
     74      22      82
```

2.3.6 矩阵的秩、迹和条件数

1. 矩阵的秩

将矩阵做初等行变换后，非零行的个数叫行秩；矩阵做初等列变换后，非零列的个数称为列秩；当矩阵的列秩和行秩相等时（方阵）才称矩阵的秩。当矩阵的秩等于行数时，则称矩阵满秩，语法格式：

```
rank(A)                                  %求矩阵 A 的秩，A 必须是方阵才能求秩
```

2. 矩阵的迹

在线性代数中，把矩阵的对角线之和称为矩阵的迹，只有方阵才可以求迹，语法格式：

```
trace(A)                                 %求矩阵 A 的迹，A 必须是方阵才能求迹
```

3. 矩阵的条件数

矩阵 A 的条件数等于 A 的范数与 A 逆的范数乘积，表示矩阵运算对于误差的敏感性。对于线性方程组 $Ax=b$ ，如果 A 的条件数大， b 的微小改变就能引起解 x 较大的改变，数值稳定性差；如果 A 的条件数小， b 有微小的改变， x 的改变也很微小，数值稳定性好。它也可以表示 b 不变、 A 有微小改变时， x 的变化情况。语法格式：

```
cond(A)                                  %求 A 矩阵的条件数
```

【实战练习 2-21】求矩阵的秩、迹和条件数

若给定矩阵 A ，求 A 的秩、迹和条件数。

$$A = \begin{bmatrix} 1 & 3 & 9 \\ 0 & 5 & 7 \\ 11 & 13 & 10 \end{bmatrix}$$

编程代码如下：

```
a=rank(A)
b=trace(A)
c=cond(A)
```

结果:

```
a=      3
b=     16
c=    11.6364
```

2.3.7 矩阵的逆

逆矩阵与原矩阵有着非常重要的关系，它可以用来解决许多实际求解问题，例如控制系统的可控、可观性、线性方程组及最小二乘法等。

1. 定义

定义：若 $A \cdot B = B \cdot A = E$ ，则称 A 和 B 互为逆矩阵。

2. 运算函数

当 A 矩阵满秩时，才存在逆矩阵，语法格式：

```
inv(A)
```

【实战练习 2-22】利用逆矩阵求方程组的解

使用逆矩阵求解下列线性方程组的解。

$$\begin{cases} 2x + 3y + 5z = 5 \\ x + 4y + 8z = -1 \\ x + 3y + 27z = 6 \end{cases}$$

根据方程组的等式： $Ax=B$ ， $x=A^{-1}B$ ，编程代码如下：

```
A=[2 3 5;1 4 8;1 3 27]
B=[5 -1 6]'
```

```
x=inv(A)*B
```

结果：

```
x=      4.8113
     -1.9811
      0.2642
```

2.3.8 矩阵的特征值和特征向量

矩阵的特征值与特征向量是线性代数中的重要概念。对于一个 n 阶方阵 A ，若 n 维向量 x 与常数 λ ，使得 $\lambda x = Ax$ ，则称 λ 是 A 的一个特征值， x 是属于特征值 λ 的特征向量。可以使用 $|\lambda E - A| = 0$ 先求解出 A 的特征值，再代入等量关系求解特征向量（不唯一）。

语法格式：

```
E=eig(A)           %求矩阵 A 的全部特征值，构成向量 E
[v d]=eig(A)       %求矩阵 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
```

【实战练习 2-23】求矩阵的特征向量

利用 $A = [1\ 2\ 3\ 4; 6\ 7\ 8\ 9; 11\ 12\ 13\ 14; 0\ 12\ 17\ 13]$ 矩阵构成对角阵 D ，并求 A 的特征向量构成的列向量 V 。

编程代码如下：

```
A = [1 2 3 4; 6 7 8 9; 11 12 13 14; 0 12 17 13];
[V, D] = eig(A)
```

结果：

```
V = -0.1465 + 0.0000i 0.1107 + 0.2881i 0.1107 - 0.2881i 0.2512 + 0.0000i
      -0.3923 + 0.0000i 0.2826 + 0.0574i 0.2826 - 0.0574i -0.6977 + 0.0000i
      -0.6381 + 0.0000i 0.4546 - 0.1732i 0.4546 + 0.1732i 0.6419 + 0.0000i
      -0.6461 + 0.0000i -0.7648 + 0.0000i -0.7648 + 0.0000i -0.1954 + 0.0000i
D = 37.0763 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i
      0.0000 + 0.0000i -1.5382 + 2.9483i 0.0000 + 0.0000i 0.0000 + 0.0000i
      0.0000 + 0.0000i 0.0000 + 0.0000i -1.5382 - 2.9483i 0.0000 + 0.0000i
      0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i 0.0000 + 0.0000i
```

2.3.9 矩阵的海森伯格变换

海森伯格矩阵是一种和三角矩阵相似的特殊方阵，是指一个方阵次对角元以下的所有元素都为 0 的矩阵，如图 2.4 所示。海森伯格矩阵以卡尔·阿道夫·海森伯格名字命名。

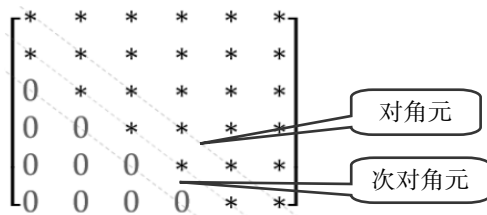


图 2.4 矩阵次对角元的表示

语法格式：

```
H=hess(A) %求矩阵 A 的海森伯格转换矩阵
[P, H]=hess(A) %H 为矩阵 A 的海森伯格矩阵，P 为满足 PHP' 且 P'P=I 矩阵
```

【实战练习 2-24】求海森伯格变换矩阵

给定下列矩阵 A ，求 A 的海森伯格转换矩阵。

$$A = \begin{bmatrix} -1 & 2 & 3 & 0 \\ 0 & -2 & 3 & 4 \\ 1 & 0 & 4 & 5 \\ 1 & 2 & 9 & -3 \end{bmatrix}$$

编程代码如下：

```
A = [-1 2 3 0; 0 -2 3 4; 1 0 4 5; 1 2 9 -3]
H1=hess(A)
[P, H]=hess(A)
```

结果:

```
H1 =  -1.0000   -2.1213    2.5293   -1.4501
      -1.4142    7.5000   -2.9485    4.8535
           0    -5.1720   -2.9673    1.7777
           0         0    2.4848   -5.5327

P =  1.0000    0         0         0
      0         0    0.9570    0.2900
      0   -0.7071    0.2051   -0.6767
      0   -0.7071   -0.2051    0.6767

H =  -1.0000   -2.1213    2.5293   -1.4501
      -1.4142    7.5000   -2.9485    4.8535
           0    -5.1720   -2.9673    1.7777
           0         0    2.4848   -5.5327
```

2.4 MATLAB 常用函数及应用

本节主要介绍矩阵常用函数操作,包括数学函数、转换函数、字符串处理函数、测试函数、日期时间函数等。

2.4.1 常用数学函数

常用数学函数如表 2.8 所示。

表 2.8 常用数学函数

函 数 名	含 义	函 数 名	含 义
abs(x)	绝对值(复数的模)	real(x)	复数x的实部
rem(x,y)	余数,符号不同取决x	imag(z)	复数z的虚部
size(x)	矩阵最大元素数	conj(z)	复数z的共轭复数
sqrt(x)	平方根	lcm(x,y)	x和y最小公倍数
log(x)	自然对数	max(x)	每列最大值
log2(x)	以2为底的对数	min(x)	每列最小值
power(x,y)	x为底y的幂	sum(x)	元素的总和
pow2(x)	2为底x的幂	mean(x)	各元素的平均值
sort(x)	对矩阵x按列排序	exp(x)	以e为底的指数
rank(x)	矩阵的秩	log10(x)	以10为底的对数
dot(x,y)	向量x、y的点积	nchoosek(x,m)	向量x中选m个元素组合
det(x)	行列式值	factorial(x)	x的阶乘
complex(a,b)	生成a+bj的复数	perms(x)	向量x的全排列
round(x)	最接近x的整数	prod(x)	矩阵x的列乘积
angle(z)	复数z的相角	inv(x)	矩阵的逆

续表

函 数 名	含 义	函 数 名	含 义
length(x)	向量阵长度(矩阵最大行或列数)	gcd(x,y)	x和y的最大公约数
std(x)	返回向量x的标准方差	ndims(x)	矩阵的维数
nnz(x)	非零元素个数	pinv(x)	伪逆矩阵
trace(x)	矩阵对角元素的和	sign(x)	符号函数

例如：符号函数的用法

当 $x < 0$ 时， $\text{sign}(x) = -1$

当 $x = 0$ 时， $\text{sign}(x) = 0$

当 $x > 0$ 时， $\text{sign}(x) = 1$

设 $x = -5$

则

```
sign(x)
ans =      -1
```

【实战练习 2-25】求指数、余数及最小公倍数

求 2^3 、 e^1 、 $100/6$ 的余数、及 76 和 24 的最小公倍数。

编程代码如下：

```
power(2,3) =      8           %表示求  $2^3$ 
exp(1) =      2.7183         %表示  $e^1$  自然指数
rem(100,6) =      4          %表示 100 除以 6 的余数
lcm(76,24) =      456        %表示 76 和 24 的最小公倍数
```

【实战练习 2-26】求复数的模、相角和共轭复数

创建一维复数矩阵 $[3.5+6.78j]$ ，求模、相角和共轭复数。

编程代码如下：

```
z=complex(3.5,6.78)          %构造复数矩阵
A=abs(A)                     %求模
AM=angle(z)                  %求相角
Zi=coni(z)                   %求共轭复数矩阵
```

结果：

```
z =      3.5000 + 6.7800i
A =      7.6301
AM =      1.0943
Zi =      3.5000 - 6.7800i
```

【实战练习 2-27】求阶乘及组合数

求数值 5 的阶乘及 $m=5,n=3$ 的组合数。

编程代码如下：

```
F1=factorial(5)              %求阶乘 5!
K=nchoosek(5,3)              %求组合数  $5!/(5-3)!/3!$ 
```

```
F1= 120
K= 10
```

【实战练习 2-28】求矩阵的点乘

给定矩阵 A 和 B ，求 A 、 B 的点乘（注意矩阵 A 和 B 必须具有相同的行和列 $m \times n$ ）。

$$A = \begin{bmatrix} 1 & 2 \\ 4 & 5 \end{bmatrix} \quad B = \begin{bmatrix} 10 & 20 \\ 3 & 7 \end{bmatrix}$$

编程代码如下：

```
A=[1, 2;4, 5];
B=[10,20;3 7]
C=dot(A,B)
```

结果：

```
C = 22 75
```

说明：点乘相当于 $[1 \times 10 + 4 \times 3, 2 \times 20 + 5 \times 7] = [22 \ 75]$ 。

【实战练习 2-29】求矩阵的秩、逆及行列式的值

已知矩阵 A ，判断矩阵 A 是否满秩，若满秩，求其逆矩阵，并计算 A 的行列式值。

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 2 & 3 & 5 \end{bmatrix}$$

编程代码如下：

```
A=[1 2 3; 4 5 6; 2 3 5];
B=rank(A)           %求 A 的秩
C=inv(A)             %求 A 的逆矩阵
D=det(A)             %求 A 行列式的值
```

结果：

```
B = 3
C = -2.3333    0.3333    1.0000
     2.6667    0.3333   -2.0000
    -0.6667   -0.3333    1.0000
D = -3
```

【实战练习 2-30】求矩阵维数及非零元素个数

建立脚本程序，生成一个随机矩阵，放大 10 倍并取整数赋给矩阵 A 。输出矩阵 A 的维数 a 、行列数 m 和 n 、矩阵所有维的最大长度 c ，并计算 A 中非 0 元素的个数 e 。

编程代码如下：

```
A=floor(rand(5,4)*10)      %生成 10 以内 5×4 整数矩阵
a=ndims(A)                 %返回 A 的维数。m×n 矩阵为 2 维
[m,n]=size(A)              %如果 A 是二维数组，返回行数和列数
c=length(A)                %返回行、列中的最大长度
e=nnz(A)                   %返回 A 中非 0 元素的个数
```

结果：

```
A = 3      2      0      1
     8      7      0      5
     5      7      5      4
     5      3      7      0
     9      5      9      3

a = 2
m = 5
n = 4
c = 5
e = 17
```

【实战练习 2-31】求向量元素的全排列及排列个数
求向量 $v=[1,2,3,4,5]$ 中的 4 个元素全排列及排列个数。
编程代码如下：

```
v=[1,2,3,4,5];
y=nchoosek(v,4)
n=length(v)
```

结果：

```
y = 1      2      3      4
     1      2      3      5
     1      2      4      5
     1      3      4      5
     0      3      4      5

n=5
```

2.4.2 常用三角函数

三角函数是角的集合与比值集合变量之间的映射，一般三角函数是在平面直角坐标系中定义的，常用于计算直角三角形中未知长度的边和未知的角度，它对研究三角形、圆形等几何形状的性质有着重要的作用。常用三角函数如表 2.9 所示。

表 2.9 常用三角函数

函 数 名	含 义	函 数 名	含 义
sin(x)	正弦函数	asin(x)	反正弦函数
cos(x)	余弦函数	acos(x)	反余弦函数
tan(x)	正切函数	atan(x)	反正切函数
cot(x)	余切函数	acot(x)	反余切函数
sinh(x)	双曲正弦函数	asinh(x)	反双曲正弦函数
cosh(x)	双曲余弦函数	acosh(x)	反双曲余弦函数
tanh(x)	双曲正切函数	atanh(x)	反双曲正切函数
sec(x)	正割函数	asec(x)	反正割函数
sech(x)/asech(x)	双曲正割/反双曲正割函数	csch(x)/acsch(x)	双曲余割/反双曲余割函数

【实战练习 2-32】求给定角度的三角函数值

按照下列要求完成计算：

- (1) 将 30 度转换成弧度，并求其正弦、正切、余切、双曲余弦和正割函数值；
- (2) 将 1 弧度转换成度，求其反正弦、反正切、反双曲正弦函数和反余割函数值。

编程代码如下：

```
alpha=30*pi/180           %30 度需要乘以  $\pi/180$  变成弧度
a=sin(alpha)              %求正弦函数值
b=tan(alpha)              %求正切函数值
c=cot(alpha)              %求余切函数值
d=cosh(alpha)             %求双曲余弦函数值
e=sec(alpha)              %求正割函数值
val=1*180/pi              %1 弧度需要乘以  $180/\pi$  变成度
aa=asin(1)                %求反正弦函数值
ab=atan(1)                %求反正切函数值
ac= asinh(1)              %求反双曲正弦函数值
ad= acsc(1)               %求反余割函数值
```

结果：

```
alpha =    0.7854
a =    0.7071
b =    1.0000
c =    1.0000
d =    1.3246
e =    1.4142
val =   57.2958
aa =    1.5708
ab =    0.7854
ac =    0.8814
ad =    1.5708
```

2.4.3 常用取整函数

MATLAB 提供了 4 种取整（数）函数，取最接近整数，分别按照：向上（取最大整数）、向下（取最小整数）进行取值。常用取整函数如表 2.10 所示。

表 2.10 常用取整函数

函 数 名	含 义	函 数 名	含 义
round(x)	四舍五入至最近整数	floor(x)	舍去正小数至最近整数
fix(x)	舍去小数至最近整数	ceil(x)	加入正小数至最近整数

【实战练习 2-33】不同取整函数的应用

给定不同的正、负小数，按照向上、向下取整函数进行取值。

编程代码如下：

```
a=fix(-1.3)           %取-1.3 的最近整数
b=fix(1.3);          %取 1.3 的最近整数
c=floor(-1.3)         %向下取-1.3 的最近整数
d=floor(1.3)          %向下取 1.3 的最近整数
e=ceil(-1.3)          %向上取-1.3 的最近整数
f=ceil(1.3)           %向上取 1.3 的最近整数
g=round(-1.6534)      %四舍五入取-1.6543 的最近整数
h=round(1.6534)       %四舍五入取 1.6543 的最近整数
i=round(1.6543,2)     %四舍五入取 1.6543 的最近整数并保留 2 位小数
j=round(-1.6543,2)    %四舍五入取-1.6543 的最近整数并保留 2 位小数
```

结果：

```
a =    -1
b =     1
c =    -2
d =     1
e =    -1
f =     2
g =    -2
h =     2
i =    1.6500
j =   -1.6500
```

2.4.4 随机函数

随机变量表示对应于某个实验所有样本中，都有一个相对应的数值，MATLAB 提供了产生随机变量的多个函数，常用的随机函数如表 2.11 所示。

表 2.11 常用随机函数

函 数 名	含 义
rand	产生均值为0.5、幅度在0~1的伪随机数
rand(n)	生成0~1的n阶随机数方阵
rand(m,n)	生成0~1的m×n的随机数矩阵
randn	生成均值为0、方差为1的高斯白噪声
randn(n)	生成均值为0、方差为1的高斯白噪声方阵
randi(n)	生成一个1~n的随机整数
randn(m,n)	产生0-1均匀分布，均值为0、方差为1的正态分布矩阵
randperm(n)	产生1~n的均匀分布随机序列
normrnd(a,b,c,d)	产生均值为a、方差为b大小为c×d的随机矩阵

【实战练习 2-34】 随机函数的多种应用

按照下列要求生成随机数：

- (1) 生成一个 0~1 的随机数 **A1**；
- (2) 生成一个 2 行 3 列 0~1 的矩阵 **A2**；
- (3) 生成一个由 1~10 的随机整数组成的 5×5 矩阵 **A3**；
- (4) 生成一个 10~20 均匀分布 2 行 4 列的随机整数数组 **A4**；
- (5) 生成一个均值为 0、方差为 1 的高斯白噪声 3 阶方阵 **A5**；
- (6) 生成一个 1~5 的均匀分布随机序列 **A6**；
- (7) 生成一个均值为 1、方差为 3 大小为 2×4 的随机矩阵 **A7**。

编程代码如下：

```
A1=rand %生成一个 0~1 的随机数
A2=rand(2,3) %生成一个 2 行 3 列 0~1 的矩阵
A3=randi(10,5) %生成一个由 1~10 的随机整数组成的 5×5 矩阵
A4=randi([10,20],2,4) %生成一个 10~20 均匀分布 2 行 4 列的随机整数数组
A5= randn(3) %生成一个均值为 0、方差为 1 的高斯白噪声 3 阶方阵
A6=randperm(5) %生成一个 1~5 的均匀分布随机序列
A7= normrnd(1,3,2,4) %生成一个均值为 1、方差为 3 大小为 2×4 的随机矩阵
```

结果：

```
A1 = 0.8147
A2 = 0.9058 0.9134 0.0975
      0.1270 0.6324 0.2785
A3 = 6 10 10 9 4
      10 5 8 10 7
      10 9 10 7 2
      2 2 7 8 8
      10 5 1 8 1
A4 = 18 16 10 18
      14 10 15 20
A5 = -1.0689 1.4384 1.3703
      -0.8095 0.3252 -1.7115
      -2.9443 -0.7549 -0.1022
A6 = 4 1 5 2 3
A7 = 0.5054 4.2798 -1.5910 -2.6424
      2.8831 4.3278 1.2321 -2.3405
```

2.4.5 转换函数

转换函数主要是数据类型及进制的转换，用于将一个数据类型转换为另一个数据类型或转换不同进制，转换函数是必备的数据处理和分析函数。MATLAB 常用转换函数如表 2.12 所示。

表 2.12 常用转换函数

函 数 名	含 义	函 数 名	含 义
str2num('str')	字符串转换为数值	str2double('num')	字符串转换为双精度
num2str(num)	数值转换为字符串	int2str(num)	整数转换为字符串
str2mat('s1','s2'...)	字符串转换为矩阵	setstr(ascii)	ASCII转换为字符串
dec2bin(num)	十进制转换为二进制	dec2hex(num)	十进制转换为十六进制
dec2base(num)	十进制转换为X进制	base2dec(num)	X进制转换为十进制
bin2dec(num)	二进制转换为十进制	sprintf('%x ',num)	输出格式转换
lower('str')	字符串转换成小写	upper('str')	字符串转换成大写

【实战练习 2-35】不同进制转换函数的应用

将下列二进制转换成十进制，再将十进制分别转换成二、十六和八进制。

编程代码如下：

```
x=bin2dec('111101')    %二进制转换成十进制
y=dec2bin(61)           %十进制转换成二进制
z=dec2hex(61)           %十进制转换成十六进制
w=dec2base(61,8)        %十进制转换成八进制
q=sprintf('%5d',23)     %将数字转换为字符串，5 表示 5 位数，不足 5 位前面补零
```

结果：

```
x =      61
y =     111101
z =       3D
w =       75
q= '00023'
```

2.4.6 字符串处理函数

MATLAB 将字符串作为字符数组处理，一个字符串由多个字符组成，用单引号（' '）界定，字符串是按行向量存储的，每个字符（包括空格）是以 ASCII 的形式存放。常用字符串操作函数如表 2.13 所示。

表 2.13 常用字符串操作函数

函 数 名	含 义	函 数 名	含 义
deblank('str')	去掉字符串末尾的空格	blanks(n)	创建由n个空格组成字符串
findstr(s1,s2)	字符串s1是否存在字符串s2中	strcat(s1,s2)	字符串s1、s2横向连接组合
strrep(s1,s2,s3)	从字符串s1中找到s2，并用s3替代	strvcat(s1,s2)	字符串s1、s2竖向连接组合

续表

函 数 名	含 义	函 数 名	含 义
strcmp(s1,s2)	判断字符串s1和s2是否相等,相等返回1; 否则返回0	strmatch()	寻找符合条件的行
strcmpi(s1,s2)	同strcmp(), 但忽略大小写	strtok(s1,s2...)	查找字符串s1中第一个给定的分隔符之前和之后的字符串
strncmp(s1,s2)	比较字符串s1和s2的前n个字符	strjust(s1,对齐)	字符串s1对齐方式('left'/'center'/'right')

【实战练习 2-36】字符串的查找及连接

按照给定字符串 char1 和 char2，完成字符串的查找和连接。

编程代码如下：

```
char1='我们在学习计算机语言 MATLAB';
char2=MATLAB';
A=findstr(char1,char2)           %返回字符串 char2 在字符串 char1 中位置,若不
                                %存在返回空矩阵
B=strcat(char1,char2)           %连接字符串 char1 和字符串 char2
C=strrep(char1,char2,'Simulink') %若字符串 char1 中有字符串 char2,则使用字符
                                %串 Simulink 替换; 否则不替换
D=strtok(char1,char2)           %若字符串 char1 中有字符串 char2,则去掉相同
                                %部分, 否则不变
```

结果：

```
A =      11
B =      '我们在学习计算机语言 MATLABMATLAB'
C =      '我们在学习计算机语言 Simulink'
D =      '我们在学习计算机语言'
```

2.4.7 判断函数

判断函数常用于判断数据类型、矩阵之间是否包含，以及元素是否非零、是否为空、是否是实数等，它们对错误处理很受用。MATLAB 常用的判断函数如表 2.14 所示。

表 2.14 常用的判断函数

函 数 名	含 义
isnumeric(x)	判断x是否为数值类型
exist(x)	判断参数变量是否存在
isa(x, 'integer')	判断x是否为引号中指定的数据类型（包括其他数据类型）
isreal(x)	判断x是否为实数
isprime(x)	判断x是否为质数

续表

函 数 名	含 义
isinf(x)	判断x是否为无穷大数
isfinite(x)	判断x是否为有限数
ismember(a,b)	判断矩阵（向量）a是否包含b的元素
all	判断向量或矩阵的列向量是否都为非零元素
any	判断向量或矩阵的列向量是否都为零元素

说明：判断函数的结果是逻辑值，当判断为真时用 1 表示；否则用 0 表示。

【实战练习 2-37】判断矩阵是否包含元素

给定矩阵 *A* 和矩阵 *B*，判定矩阵 *A* 是否包含矩阵 *B* 的元素。

编程代码如下：

```
A=[1 2 3;4 5 6;7 8 9] ; B=[1 10 20;9 11 8]
C=ismember(a,b)
```

结果：

```
C = 1     0     0
     0     0     0
     0     1     1
```

说明：为 1 表示包含；否则为不包含。

【实战练习 2-38】判断矩阵数据类型

给定实数、复数、数值及字符串，判定其数据类型。

编程代码如下：

```
p=[1 2 1 5] ;n=isreal(p)           %p 都是实数
p1 =[1+5i 2+6i;3+7i 4+8i];n1=isreal(p1) %p1 有非实数
x=2.34; n2= isnumeric(x)           %x 为数值型
x1= 'China123'; n3=isnumeric(x1)   %x1 为非数值型
```

结果：

```
n =      1
n1 =      0
n2 =      1
n3 =      0
```

【实战练习 2-39】判断矩阵中的质数

按照要求编写程序，完成以下功能：

- (1) 找出 10~20 的所有质数，将这些质数存放在一个行数组里；
- (2) 求出这些质数之和；
- (3) 求出 10~20 的所有非质数之和（包括 10 和 20）。

编程代码如下：

```
X=10:20;
p1=X(isprime(X))
s1=sum(p1)
p2=(X(~isprime(X)))
s2=sum(p2)
```

结果:

```
p1 =    11    13    17    19
s1 =     60
p2 =    10    12    14    15    16    18    20
s2 =   105
```

【实战练习 2-40】判断矩阵中的非零元素

判断矩阵 **A** 所有的非零元素，并检测矩阵 **B** 每一列是否全为非零元素。

编程代码如下:

```
A=[1 0 1;2 3 5;9 10 0]; B=[0 0 0;4 5 0;7 8 0];
C=all(A)                %某列含有 0 元素结果为 0
D=any(B)                %某列都是 0 元素结果为 0
```

结果:

```
C =     1     0     0
D =     1     1     0
```

2.4.8 查找函数

查找函数的作用是在一定范围内查找某个特定的值或字符串，并返回其在该范围内的位置或索引值，语法格式:

```
find(A)                %A 是一个矩阵, 查询非零元素的位置。如果 A 是一个行向量, 则返回
                        %一个行向量; 否则返回一个列向量。若 A 全是零元素或是空数组, 则
                        %返回一个空数组
[m,n]=find(A)          %返回矩阵 A 中非零项的坐标, m 为行数, n 为列数
[m,n]=find(A>2)        %返回矩阵 A 中大于 2 的元素的坐标, m 为行数, n 为列数
[m,n,v]=find(A)        %返回矩阵 A 中非零项的坐标, 并将数值按列放在 v 中
```

【实战练习 2-41】查找矩阵元素坐标位置

按照要求建立 3×3 魔方矩阵 **A**，完成下列操作:

- (1) 返回的是矩阵 **A** 中大于 5 的元素的坐标;
- (2) 查找第 2 列中等于 5 的元素;
- (3) 查找矩阵 **A** 中等于 9 的元素的坐标。

编程代码如下:

```
A = magic(3)
[m,n]=find(A>5)        %查找大于 5 的坐标
find(A(:,2) ==5)       %查找第 2 列中等于 5 的元素
```

```
[m1,n1]=find(A==9)           %查找等于 9 的元素的坐标
```

结果：

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

A 中大于 5 的元素的坐标：

```
m =      n=
     1         1
     3         2
     1         3
     2         3
```

A 中第 2 列中等于 5 的元素的坐标：

```
ans =     2
```

A 中等于 9 的元素的坐标：

```
m1 =     3      n1 =     2
```

2.4.9 测试向量（矩阵）零元素函数

1. 测试向量零元素

语法格式：

```
all(A)           %A 为向量，若 A 的所有元素都不等于零，all(A) 返回 1；否则返回零
```

2. 测试向量非零元素

语法格式：

```
any(A)           %测试向量或矩阵 A 中是否存在非零元素，按列数返回值，若一列中存在非零
                  %元素，返回 1；否则返回 0
```

【实战练习 2-42】测试向量和矩阵是否存在非零元素

```
A=[11 3 2 0 16 0 19]
```

```
all(A)           %A 中存在零元素，返回 0
```

```
B=[2 0 3;5 0 1;7 0 0]
```

```
any(B)           %B 中存在都是零的列，返回 0；其他列返回 1
```

结果：

```
A =     11     3     2     0    16     0    19
ans= logical
      0
B =     2     0     3
      5     0     1
      7     0     0
```

```
ans = 1×3 logical 数组
      1    0    1
```

2.4.10 日期和时间函数

日期和时间函数可计算时间间隔，及获得给定时间、日期、月份、星期等，常用的日期和时间函数如表 2.15 所示。

表 2.15 常用的日期和时间函数

函 数 名	含 义
tic()	用来记录MATLAB命令执行的时间并保存当前时间
now()	获取当前时间至0000年的天数，以浮点型常量表示
datetime()	获取当前日期、时间，并显示字符datetime
year(日期)	获取指定日期的年份
month(日期)	获取指定日期的月份
day(日期)	获取指定日期
date()	得到当前的日-月-年
today()	获取当前时间至0000年的天数，以整型常量表示
datenum(日期)	获取0000年到给定时间的天数
weekday(日期)	获得指定日期的星期数+1
yeardays(年份)	获得某一年有多少天
eomday(年,月)	给出指定年月最后一天日期
etime(t1,t2)	估算t2到t1命令的时间间隔
calendar(年,月)	获取当前月的日历，包括日期和星期
toc()	记录程序完成时间，与tic()联用记录MATLAB命令执行的时间

其中：函数 year()、month()、day()、today()、datetime()需要载入 Financial Toolbox 中。

【实战练习 2-43】利用函数输出指定时间

输出当前日期、时间，完成计算并输出指定月的日历。

编程代码如下：

```
tic;                                %开始计时
format short g                      %指定显示格式
t1=clock                            %显示日期、时间
d1=now()
datetime()                          %获得当前日期时间
y=year(now)                         %获取当前年份
m=month(now)                        %获取当前月份
d=day(now)                          %获取 0000 年到今天的天数
todaydate=date()                   %获取当前日期
```

```

T=today()
datenum1=datenum('12-31-2025') %给出 0000 年到给定时间的天数
[a,b]=weekday('2024-8-15')      %b 为指定日期的星期数；a 为指定日期第 2 天星期数
toyears=yeardays(2024)         %某一年有多少天
dd=eomday(2024,2)               %给出 2024 年 2 月最后日期
t2=clock                        %当前日期时间
calendar                        %获取当前一个月的日历
timecal=etime(t2,t1)            %计算执行 t2-t1 的所用时间
toc

```

结果：

```

t1 = 2023      3      20      17      34      14.015
d1 = 7.3897e+05
ans =
    datetime
    2023-03-20 17:34:14
y = 2023
m = 3
d = 20
todaydate = '20-Mar-2023'
T = 738965
datenum1 = 739982
a = 5
b = 'Thu'
toyears = 366
dd = 29
t2 = 2023      3      20      17      34      14.062

      Mar 2023
      S      M      Tu      W      Th      F      S
      0      0      0      1      2      3      4
      5      6      7      8      9     10     11
     12     13     14     15     16     17     18
     19     20     21     22     23     24     25
     26     27     28     29     30     31      0
      0      0      0      0      0      0      0
timecal = 0.047

```

2.4.11 标准差函数

标准差是统计学中常用的一种测量数据分散程度的方法，它是指一组数据的各个数据与其平均数之差的平方和的平均数的平方根。在 MATLAB 中标准差函数是 `std()`，它可以快速计算一组数据的标准差。若矩阵为 **A**，语法格式：

```
y=std(A,flag,dim)      %dim 为维数，取 1 或 2；flag 取 0 或 1
```

其中：

当 `dim=1` 时，求各列元素的标准差，其第 *i* 个元素是 **A** 的第 *i* 列元素的平均值；

当 $\text{dim}=2$ 时, 求各行元素的标准差, 其第 i 个元素是 A 的第 i 行元素的平均值。

当 $\text{flag}=0$ 时, 计算标准差;

当 $\text{flag}=1$ 时, 计算方差。

默认: $\text{flag}=0$, $\text{dim}=1$ 。

【实战练习 2-44】求矩阵的标准差

求一个 4 阶魔方矩阵的行、列标准差。

编程代码如下:

```
A=magic(4)
s1=std(A)
S2=std(A,0,2)
```

结果:

```
A =
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1

s1 =
    5.4467    5.1962    5.1962    5.4467

S2 =
    7.0475
    2.6458
    2.6458
    7.0475
```

2.4.12 函数句柄

MATLAB 提供的访问函数方式既可用函数名调用, 也可用句柄 (handle) 调用。若在已有函数名前加符号 $@$, 即可创建函数句柄 (function handle), 它是 MATLAB 中的一类特殊的数据结构, 类似 C++ 语言的函数指针, 作用是将一个函数封装成一个函数句柄 (指针), 再使用该句柄访问函数。

创建和调用句柄格式:

```
handle = @functionname 或 fun1=@functionname
```

调用格式:

```
fun1(arg1,arg2,...,argn)
```

例如:

```
fun=@(x,y)x^2+y^2; %表示 fun=f(x,y)=x^2+y^2, fun 表示(匿名函数)句柄。
```

【实战练习 2-45】利用句柄实现函数计算

定义句柄 $f1(n)=x^2$, $f2(n)=(x+y)^2+(x-y)^2$, $f3(n)=n+10\log(n^2+8)$, 使用句柄分别计算:

$y1=5^2$; $y2=(3+5)^2+(3-5)^2$; $y3=f3(30)/(f3(20)+f3(10))$ 。

编程代码如下：

```
f1=@(x)x.^2;
f2=@(x,y)(x+y).^2+(x-y).^2;
f3=@(n)n+10*log(n^2+8);
y1=f1(5)
y2=f2(3,5)
y3=f3(30)/(f3(20)+f3(10))
```

结果：

```
y1 =    25
y2 =    68
y3 =    0.7165
```

【实战练习 2-46】利用句柄实现求导数

已知函数 $y=x^5+7x^4+5x^2-12x+3$ ，利用句柄求导数 y' ，并计算 $y'+y$

编程代码如下：

```
syms x %定义 x 为符号
y=@(x)x.^5+7.*x.^4+5.*x^2-12.*x+3 %构造句柄
y1=y(x) %调用句柄
y2=diff(y1) %利用句柄求导数
f=y1+y2 %计算 y'+y
```

结果：

```
y= 包含以下值的 function_handle:
    @(x)x.^5+7.*x.^4+5.*x^2-12.*x+3
y1=x^5+7*x^4+5*x^2-12*x+3
y2=5*x^4+28*x^3+10*x-12
f=x^5+12*x^4+28*x^3+5*x^2-2*x-9
```

【实战练习 2-47】利用句柄实现矩阵运算

定义句柄 A，通过调用句柄完成矩阵计算。

编程代码如下：

```
A=@(x,y)[x,y,x+y;y,x,x-y;x+y,x-y,x*y]; %定义句柄 A
B=A(5,4) %调用句柄 x=5,y=4，获得矩阵 B
C=A(12,23) %调用句柄 x=12,y=23，获得矩阵 C
D=B.*C %计算 B.*C
```

结果：

```
B =    5     4     9
      4     5     1
      9     1    20
C =   12    23    35
      23    12   -11
      35   -11   276
```

```
D =      60      92      315
      92      60      -11
     315     -11     5520
```

2.5 MATLAB 数组表示

数组运算是计算的基础，矩阵是数学上的表示，有着明确而严格的数学规则。MATLAB 的矩阵是以数组形式存在的，它将一维数组视为向量，二维数组为矩阵，所以矩阵是数组的子集。MATLAB 常以二维数组运算为例，再推广到多维数组和多维矩阵的运算。结构数组和元胞数组相当于将多个矩阵或数组联合起来，使用结构框架进行管理。

2.5.1 结构数组

1. 定义结构体数组

结构数组是指根据字段组合起来的不同类型的数据集合。结构体通过字段（fields）对元素进行索引，在访问时只需通过点号访问数据变量。结构体数组可以通过两种方法进行创建，即通过直接赋值方式创建或通过结构函数 `struct()` 创建结构，其调用格式为：

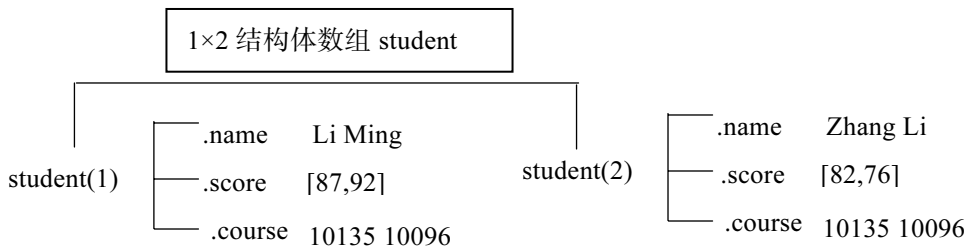
```
strArray=struct('field1',val1, 'field2',val2, ...)
```

其中：field 和 val 为字段和对应值。

字段可以是单一值或单元数组，但是必须保证它们具有相同的大小。

【实战练习 2-48】建立结构体数组

使用结构数组定义 1×2 结构体数组 `student`，表示 2 个学生成绩。



编程代码如下：

```
student(1).name='Li Ming'; student(1).course=[10135 10096]; student(1).
score=[87 92];
student(2).name='Zhang Li';
student(2).course=[10135 10096];
student(2). score=[82 76];
n1=student(1)
n2=student(2)
student(2).name
```

结果：


```
n1= name: 'Li Ming'
    course: [10135 10096]
    score: [87 92]
n2 = name: 'Zhang Li'
    course: [10135 10096]
    score: [82 76]
ans= Zhang Li
```

若输入：

```
stu=struct('name','WangFang','course',[10568 10063],'score',[76 82])
```

结果：

```
stu = name: 'Wang Fang'
    course: [10568 10063]
    score: [76 82]
```

也可以直接输入：

```
student.name='Li Ming';
student.score=[87 92];
student.course=[10568 10063]
.....
```

2. 使用结构体数组

常用的结构数组操作函数如表 2.16 所示。

表 2.16 常用的结构数组操作函数

函 数 名	含 义	函 数 名	含 义
deal(X)	把输入变量X处理成输出	fieldnames(stu)	获取结构的字段名
getfield(field)	获取结构中指定字段的值	rmfield(field)	删除结构field字段
setfield(field)	设置结构数组中field字段的值	struct(数组值)	创建结构数组内容
struct2cell(stu)	结构数组转换成元胞数组	isfield(field)	判断是否存在field字段
isstruct(X)	判断变量X是否是结构类型	orderfields(str)	对字段按照字符串进行排序

【 实战练习 2-49 】 结构数组操作

根据结构数组 student 的定义，完成结构数组操作。

编程代码如下：

```
isstruct(student)           %判断是否为结构数组
isfield(student,{ 'name','score','weight'}) %判断结构字段是否存在
fieldnames(student)         %显示结构字段名
setfield(student(1,1), 'name','wang Hong') %赋的值多一个参数并影响原字段值
getfield(student,{1,1})     %显示结构数据
student(1,1)                %显示结构第一个数据
[name1,order1]=orderfields(student) %显示排序后字段名和排序前序号
```

结果:

```
ans =      1
ans =      1      1      0
ans =      'name'
           'course'
           'score'
ans =      name: 'wang Hong'
           course: [10135 10096]
           score: [87 92]
ans =      name: 'Li Ming'
           course: [10135 10096]
           score: [87 92]
ans =      name: 'Li Ming'
           course: [10135 10096]
           score: [87 92]
name1 =      1x2 struct array with fields:
            course
            name
            score
order1 =      2
           1
           3
```

2.5.2 元胞（单元）数组

元胞数组是 MATLAB 特有的一种数据类型，组成它的元素称元胞，可视它为无所不包的通用矩阵。元胞是用来存储不同类型数据的单元，元胞数组中每个元胞存储一种类型的数组，此数组中的数据可以是任何一种 MATLAB 数据类型或用户自定义的类型，其大小也可以是任意的。相同元胞数组中第二个元胞类型、大小可与第一个元胞完全不同。

例如：2×2 元胞数组结构如图 2.5 所示。

说明：元胞数组可以将不同类型或不同尺寸的数据存储到同一个数组中。访问元胞数组的方法与矩阵索引方法基本相同，区别在于元胞数组索引时，需要用 {} 将下标置于其中。

创建元胞数组与创建矩阵基本相同，区别在于矩阵用 []；元胞数组用 {}。

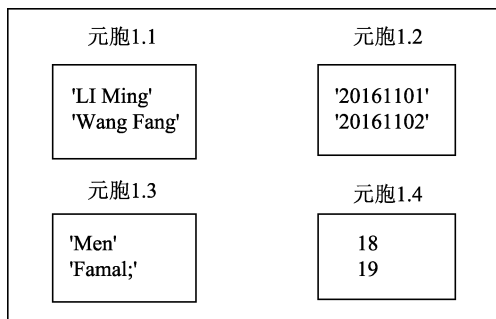


图 2.5 元胞数组结构图

1. 创建元胞数组

语法格式：

`cell(m,n)` %创建规格为 $m \times n$ 的空元胞数组或用大括号 "{}" 创建元胞数组并赋值

例如：

```
a=cell(2,3); b={'s1',[1,2,3];88,'name'}
```

结果：

```
a=      []      []      []  
      []      []      []  
b =      's1'      [1x3 double]      [88]      'name'
```

用函数 cellstr()将字符串数组转换成元胞数组。

例如：

```
B=char('姓名','住址','联系方式');  
C=cellstr(B)
```

结果：

```
C =      '姓名'  
      '住址'  
      '联系方式'
```

2. 元胞数组操作

元胞数组操作函数如表 2.17 所示。

表 2.17 元胞数组操作函数

函 数 名	含 义	函 数 名	含 义
celldisp(A)	显示元胞数组的内容	cellstr(A)	创建字符串数组A为元胞数组
cellplot(A)	元胞数组结构的图形描述	iscell(A)	判断A是否是元胞数组

- (1) 获取指定元胞的大小，用小括号；
- (2) 获取元胞的内容，用大括号；
- (3) 获取元胞数组指定元素，用大括号和小括号。

例如：

```
a=cell(2,3); b={'s1',[10,20,30],88,'name'}  
c=b(1,3)  
d=b{1,3}  
e=b{1,2}(1,3)
```

结果：

```
c=      [88]  
d =      88  
e =      30
```

例如：创建元胞数组（一维）

```
a={ [2 4 7;3 9 6;1 8 5], 'Li Ming',2+3i,1:2:10}  
A1=a{1,1}  
A2=a{1,2}
```

结果：

```
a =      [3x3 double]      'Li Ming'      [2.0000 + 3.0000i]      [1x5 double]  
A1 =      2      4      7      %取第 1 个元胞  
      3      9      6  
      1      8      5  
A2 =      'Li Ming'      %取第 2 个元胞
```

3. 元胞数组的删除

给元胞数组向量下标赋空值相当于删除元胞数组的行或列。

例如：删除元胞数组的列。

```
a(:,2)=[ ]
```

结果：

```
a = [3x3 double] [2.0000 + 3.0000i] [1x5 double]
```

说明：直接在命令行窗口输入元胞数组名，可显示元胞数组的构成元胞，可使用 `celldisp()` 函数显示元胞的元素，利用索引可以对元胞数组进行运算操作。

【实战练习 2-50】元胞数组的应用

建立元胞数组，添加魔方矩阵和随机矩阵元素，并完成相应的元胞数组的和与积运算。

编程代码如下：

```
cell11={3,3}; %建立空元胞数组
cell11{1,1}=magic(3); %构建元胞 1.1 为魔方矩阵
cell11{1,2}= {[2,3,4];[5,6,7];[10,11,12]}; %构建元胞 1.2 为 3×3 矩阵
cell11{1,3}=floor(rand(3,3)*100); %构建元胞 1.3 为随机矩阵
celldisp(cell11)
cell12=cell11{1,1}+cell11{1,3} %计算元胞数组的和
cell13=cell11{1,1}*cell11{1,3} %计算元胞数组的积
```

结果：

```
cell11{1} =      8      1      6
              3      5      7
              4      9      2
cell11{2}{1} =      2      3      4
cell11{2}{2} =      5      6      7
cell11{2}{3} =     10     11     12
cell11{3} =     41     58     51
              60     55      8
              75     58     71
cell12 =      49     59     57
              63     60     15
              79     67     73
cell12 =     838     867     842
              948     855     690
              854     843     418
```

【实战练习 2-51】元胞数组操作及绘图

创建元胞数组，按照下列要求完成操作：

- (1) 求元胞数组列的和；
- (2) 判断元胞数组元素；
- (3) 使用元胞数组绘图。

编程代码如下：

```
A{1,1}=[2 5;7 3];
A{1,2}=rand(3,3);
celldisp(A) %显示元胞数组
B=sum(A{1,1}) %求 A{1,1}列的和
a=iscell(A) %判断 A 是否是元胞数组
C={'身高','体重','年龄';176,70,30};
cellplot(C,'legend') %绘制元胞数组图
```

结果：

```
A{1} =      2      5
        7      3
A{2}=  0.4447    0.9218    0.4057
        0.6154    0.7382    0.9355
        0.7919    0.1763    0.9169
B=      9      8
a=      1
```

不同数据类型的元胞数组元素用不同的颜色表示，如图 2.6 所示。

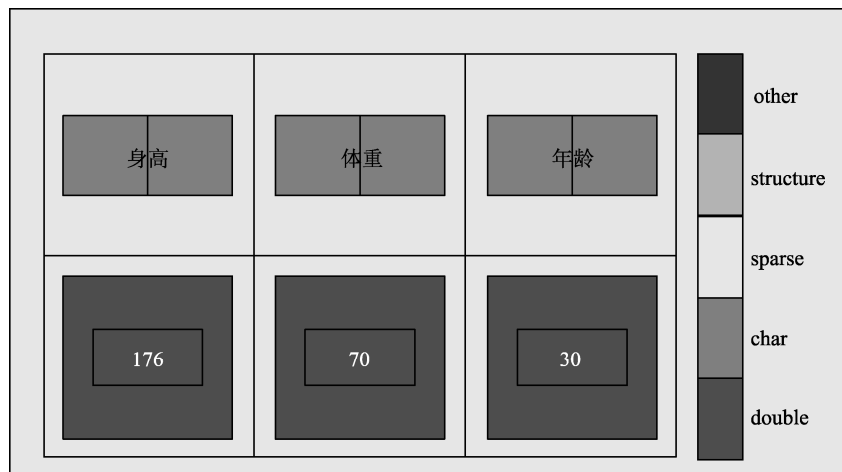


图 2.6 元胞数组图形表示

【实战练习 2-52】函数句柄的元胞数组应用

定义句柄为元胞数组，调用句柄完成计算。

编程代码如下：

```
funcs={@sin,@cos,@tan}
y1=funcs{1}(pi/2)
y2=funcs{2}(pi/4)
y3=funcs{3}(0)
```

结果：

```

funcs = 1×3 cell 数组
    {@sin}    {@cos}    {@tan}
y1 =      1
y2 =    0.7071
y3 =      0

```

2.6 数组集合运算

集合是一种不允许值重复的顺序数据结构,由数字、日期、时间或文本数据组成,它属于一种集合数组。数组的运算主要有算术运算、关系运算、逻辑运算和集合运算等,集合运算包括交、差、并、异或、唯一性、集合判断及集合连接等运算。

2.6.1 交运算

交运算的结果是元素属于矩阵 A 也属于矩阵 B ,若矩阵 A 中的元素在矩阵 B 中都不存在,则结果为空矩阵 0×1 ,语法格式:

```
intersect(A,B) %A 与 B 的交集矩阵,结果显示为列
```

2.6.2 差运算

矩阵 A 减矩阵 B 的差,称为矩阵差运算。若矩阵 A 中的元素都在矩阵 B 中存在,则结果为空矩阵 0×1 ,语法格式:

```
setdiff(A,B) %A 与 B 的差集矩阵
```

2.6.3 并运算

矩阵的并运算可以将多个矩阵合并成一个列的序列集。若矩阵 A 中的元素都在矩阵 B 中存在,则结果按照矩阵 A 或矩阵 B 的值大小,依次按照列排序,语法格式:

```
union(A,B) %A 与 B 的并运算
```

2.6.4 异或运算

矩阵的异或运算是属于 A 或属于 B ,但不同时属于 A 和 B 的元素的集合称为 A 和 B 的对称差,即 A 和 B 的异或运算。若矩阵 A 中的元素都在 B 中存在,则结果为空矩阵 0×1 ,语法格式:

```
setxor(A,B); %异或运算
```

2.6.5 唯一性运算

唯一性是指在数组/矩阵中不重复的元素,保持唯一性,语法格式:

```
unique(A); %使 A 中的元素保持互异性,将结果存到 C 中, A 不变
```

2.6.6 集合连接运算

把两个数组集合连接在一起称集合连接，语法格式：

```
cat(n,A,B)           %把数组 A 和 B 按指定的维数 n 连接起来：若 n=1 按照行连接，此时数
                      %组 A、B 的列相同；若 n=2 按照列连接，此时数组 A、B 的行相同；
```

2.6.7 集合判断运算

判断 b 是否为 A 的元素，若是，相应位置为 1；否则置 0，语法格式：

```
ismember(A,b);
```

判断集合是否排序，若是则为 1；否则为 0，语法格式：

```
issorted(A);          %也可用 sort() 排序
```

【实战练习 2-53】数组集合的应用

已知矩阵 A 和 B ，求它们的交、差、并及异或运算。

编程代码如下：

```
A=[5,12,18;4, 5, 6;7,18,21]
B= [0 1 2;30 6 9;21 0,3]
a=intersect(A,B) '           %输出交运算 a 的转置
b=setdiff (A,B) '           %输出差运算 b 的转置
c=union(A,B) '              %输出并运算 c 的转置
d=setxor(A,B) '             %输出异或运算 d 的转置
e=unique(A) '               %输出矩阵 A 的互异元素并进行转置
f=cat(2,A,B)                %输出 A 与 B 矩阵按行连接结果
g=ismember(A,5)              %判断矩阵 A 出现元素 5 的位置
h=issorted(A)                %判断矩阵 A 是否为排序集合
C=sort(A)                    %输出矩阵 A 排序的矩阵 C
h=issorted(C)                %判断矩阵 C 是否为排序集合
```

结果：

```
A =      5      12      18
      4       5       6
      7      18      21
B =      0       1       2
      30       6       9
      21       0       3
a =      6      21
b =      4       5       7      12      18
c =      0       1       2       3       4       5       6       7       9      12      18      21      30
d =      0       1       2       3       4       5       7       9      12      18      30
e =      4       5       6       7      12      18      21
f =      5      12      18       0       1       2
      4       5       6      30       6       9
```

```
      7    18    21    21    0    3
g = 3×3 logical 数组
      1    0    0
      0    1    0
      0    0    0
h = logical
      0
C =      4      5      6
      5     12     18
      7     18     21
h = logical
      1
```