

第2章 MATLAB矩阵及其运算

2.1 变量和数据操作

2.2 MATLAB矩阵

2.3 MATLAB运算

2.4 矩阵分析

2.5 矩阵的超越函数

2.6 字符串

2.7 结构数据和单元数据

2.8 稀疏矩阵

2.1 变量和数据操作

2.1.1 变量与赋值

1. 变量命名

在MATLAB 6.5中，变量名是以字母开头，后接字母、数字或下划线的字符序列，最多63个字符。在MATLAB中，变量名区分字母的大小写。

2. 赋值语句

(1) 变量=表达式

(2) 表达式

其中表达式是用运算符将有关运算量连接起来的式子，其结果是一个矩阵。

例2-1 计算表达式的值，并显示计算结果。

在MATLAB命令窗口输入命令：

$x=1+2i$;

$y=3-\text{sqrt}(17)$;

$z=(\cos(\text{abs}(x+y))-\sin(78*\text{pi}/180))/(x+\text{abs}(y))$

**其中pi和i都是MATLAB预先定义的变量，
分别代表代表圆周率 π 和虚数单位。**

输出结果是：

$z =$

$-0.3488 + 0.3286i$

2.1.2 预定义变量

在MATLAB工作空间中，还驻留几个由系统本身定义的变量。例如，用pi表示圆周率 π 的近似值，用i，j表示虚数单位。预定义变量有特定的含义，在使用时，应尽量避免对这些变量重新赋值。

2.1.3 内存变量的管理

1. 内存变量的删除与修改

MATLAB工作空间窗口专门用于内存变量的管理。在工作空间窗口中可以显示所有内存变量的属性。当选中某些变量后，再单击Delete按钮，就能删除这些变量。当选中某些变量后，再单击Open按钮，将进入变量编辑器。通过变量编辑器可以直接观察变量中的具体元素，也可修改变量中的具体元素。

clear命令用于删除MATLAB工作空间中的变量。**who**和**whos**这两个命令用于显示在MATLAB工作空间中已经驻留的变量名清单。**who**命令只显示出驻留变量的名称，**whos**在给出变量名的同时，还给出它们的大小、所占字节数及数据类型等信息。

2. 内存变量文件

利用MAT文件可以把当前MATLAB工作空间中的一些有用变量长久地保留下来，扩展名是.mat。MAT文件的生成和装入由save和load命令来完成。常用格式为：

save 文件名 [变量名表] [-append][-ascii]

load 文件名 [变量名表] [-ascii]

其中，文件名可以带路径，但不需带扩展名.mat，命令隐含一定对.mat文件进行操作。变量名表中的变量个数不限，只要内存或文件中存在即可，变量名之间以空格分隔。当变量名表省略时，保存或装入全部变量。**-ascii**选项使文件以ASCII格式处理，省略该选项时文件将以二进制格式处理。**save**命令中的**-append**选项控制将变量追加到MAT文件中。

2.1.4 MATLAB常用数学函数

MATLAB提供了许多数学函数，函数的自变量规定为矩阵变量，运算法则则是将函数逐项作用于矩阵的元素上，因而运算的结果是一个与自变量同维数的矩阵。

函数使用说明：

- (1) 三角函数以弧度为单位计算。
- (2) `abs`函数可以求实数的绝对值、复数的模、字符串的ASCII码值。
- (3) 用于取整的函数有`fix`、`floor`、`ceil`、`round`，要注意它们的区别。
- (4) `rem`与`mod`函数的区别。`rem(x,y)`和`mod(x,y)`要求`x,y`必须为相同大小的实矩阵或为标量。

2.1.5 数据的输出格式

MATLAB用十进制数表示一个常数，具体可采用日常记数法和科学记数法两种表示方法。

在一般情况下，**MATLAB**内部每一个数据元素都是用双精度数来表示和存储的。数据输出时用户可以用**format**命令设置或改变数据输出格式。**format**命令的格式为：

format 格式符

其中格式符决定数据的输出格式

2.2 MATLAB矩阵

2.2.1 矩阵的建立

1. 直接输入法

最简单的建立矩阵的方法是从键盘直接输入矩阵的元素。具体方法如下：将矩阵的元素用方括号括起来，按矩阵行的顺序输入各元素，同一行的各元素之间用空格或逗号分隔，不同行的元素之间用分号分隔。

2. 利用M文件建立矩阵

对于比较大且比较复杂的矩阵，可以为它专门建立一个M文件。下面通过一个简单例子来说明如何利用M文件创建矩阵。

例2-2 利用M文件建立MYMAT矩阵。

(1) 启动有关编辑程序或MATLAB文本编辑器，并输入待建矩阵：

(2) 把输入的内容以纯文本方式存盘(设文件名为mymatrix.m)。

(3) 在MATLAB命令窗口中输入mymatrix，即运行该M文件，就会自动建立一个名为MYMAT的矩阵，可供以后使用。

3. 利用冒号表达式建立一个向量

冒号表达式可以产生一个行向量，一般格式是：

$e1:e2:e3$

其中 $e1$ 为初始值， $e2$ 为步长， $e3$ 为终止值。

在MATLAB中，还可以用`linspace`函数产生行向量。其调用格式为：

`linspace(a,b,n)`

其中 a 和 b 是生成向量的第一个和最后一个元素， n 是元素总数。

显然，`linspace(a,b,n)`与 $a:(b-a)/(n-1):b$ 等价。

4. 建立大矩阵

大矩阵可由方括号中的小矩阵或向量建立起来。

2.2.2 矩阵的拆分

1. 矩阵元素

通过下标引用矩阵的元素，例如

A(3,2)=200

采用矩阵元素的序号来引用矩阵元素。矩阵元素的序号就是相应元素在内存中的排列顺序。在MATLAB中，矩阵元素按列存储，先第一列，再第二列，依次类推。例如

A=[1,2,3;4,5,6];

A(3)

ans =

2

显然，序号(Index)与下标(Subscript)是一一对应的，以 $m \times n$ 矩阵A为例，矩阵元素A(i,j)的序号为 $(j-1)*m+i$ 。其相互转换关系也可利用sub2ind和ind2sub函数求得。

2. 矩阵拆分

(1) 利用冒号表达式获得子矩阵

① $A(:,j)$ 表示取A矩阵的第j列全部元素； $A(i,:)$ 表示A矩阵第i行的全部元素； $A(i,j)$ 表示取A矩阵第i行、第j列的元素。

② $A(i:i+m,:)$ 表示取A矩阵第i~i+m行的全部元素； $A(:,k:k+m)$ 表示取A矩阵第k~k+m列的全部元素， $A(i:i+m,k:k+m)$ 表示取A矩阵第i~i+m行内，并在第k~k+m列中的所有元素。

此外，还可利用一般向量和end运算符来表示矩阵下标，从而获得子矩阵。end表示某一维的末尾元素下标。

(2) 利用空矩阵删除矩阵的元素

在MATLAB中，定义[]为空矩阵。给变量X赋空矩阵的语句为X=[]。注意，X=[]与clear X不同，clear是将X从工作空间中删除，而空矩阵则存在于工作空间中，只是维数为0。

2.2.3 特殊矩阵

1. 通用的特殊矩阵

常用的产生通用特殊矩阵的函数有：

zeros：产生全0矩阵(零矩阵)。

ones：产生全1矩阵(幺矩阵)。

eye：产生单位矩阵。

rand：产生0~1间均匀分布的随机矩阵。

randn：产生均值为0，方差为1的标准正态分布随机矩阵。

例2-3 分别建立 3×3 、 3×2 和与矩阵A同样大小的零矩阵。

(1) 建立一个 3×3 零矩阵。

`zeros(3)`

(2) 建立一个 3×2 零矩阵。

`zeros(3,2)`

(3) 设A为 2×3 矩阵，则可以用`zeros(size(A))`建立一个与矩阵A同样大小零矩阵。

`A=[1 2 3;4 5 6];` %产生一个 2×3 阶矩阵A

`zeros(size(A))` %产生一个与矩阵A同样大小的零矩阵

例2-4 建立随机矩阵：

- (1) 在区间[20,50]内均匀分布的5阶随机矩阵。
- (2) 均值为0.6、方差为0.1的5阶正态分布随机矩阵。

命令如下：

```
x=20+(50-20)*rand(5)
```

```
y=0.6+sqrt(0.1)*randn(5)
```

此外，常用的函数还有`reshape(A,m,n)`，它在矩阵总元素保持不变的前提下，将矩阵A重新排成 $m \times n$ 的二维矩阵。

2. 用于专门学科的特殊矩阵

(1) 魔方矩阵

魔方矩阵有一个有趣的性质，其每行、每列及两条对角线上的元素和都相等。对于 n 阶魔方阵，其元素由 $1, 2, 3, \dots, n^2$ 共 n^2 个整数组成。MATLAB提供了求魔方矩阵的函数 `magic(n)`，其功能是生成一个 n 阶魔方阵。

例2-5 将101~125等25个数填入一个5行5列的表格中，使其每行每列及对角线的和均为565。

$M=100+\text{magic}(5)$

(2) 范得蒙矩阵

范得蒙(Vandermonde)矩阵最后一列全为1, 倒数第二列为一个指定的向量, 其他各列是其后续列与倒数第二列的点乘积。可以用一个指定向量生成一个范得蒙矩阵。在MATLAB中, 函数`vander(V)`生成以向量`V`为基础向量的范得蒙矩阵。例如, `A=vander([1;2;3;5])`即可得到上述范得蒙矩阵。

(3) 希尔伯特矩阵

在MATLAB中，生成希尔伯特矩阵的函数是`hilb(n)`。

使用一般方法求逆会因为原始数据的微小扰动而产生不可靠的计算结果。MATLAB中，有一个专门求希尔伯特矩阵的逆的函数`invhilb(n)`，其功能是求 n 阶的希尔伯特矩阵的逆矩阵。

例2-6 求4阶希尔伯特矩阵及其逆矩阵。

命令如下：

format rat %以有理形式输出

H=hilb(4)

H=invhilb(4)

(4) 托普利兹矩阵

托普利兹(Toeplitz)矩阵除第一行第一列外，其他每个元素都与左上角的元素相同。生成托普利兹矩阵的函数是`toeplitz(x,y)`，它生成一个以`x`为第一列，`y`为第一行的托普利兹矩阵。这里`x,y`均为向量，两者不必等长。`toeplitz(x)`用向量`x`生成一个对称的托普利兹矩阵。例如

```
T=toeplitz(1:6)
```

(5) 伴随矩阵

MATLAB生成伴随矩阵的函数是 `compan(p)`，其中 `p` 是一个多项式的系数向量，高次幂系数排在前，低次幂排在后。例如，为了求多项式的 x^3-7x+6 的伴随矩阵，可使用命令：

```
p=[1,0,-7,6];  
compan(p)
```

(6) 帕斯卡矩阵

我们知道，二次项 $(x+y)^n$ 展开后的系数随 n 的增大组成一个三角形表，称为杨辉三角形。由杨辉三角形表组成的矩阵称为帕斯卡(Pascal)矩阵。函数`pascal(n)`生成一个 n 阶帕斯卡矩阵。

例2-7 求 $(x+y)^5$ 的展开式。

在MATLAB命令窗口，输入命令：

pascal(6)

矩阵次对角线上的元素1,5,10,10,5,1即为展开式的系数。

2.3 MATLAB运算

2.3.1 算术运算

1. 基本算术运算

MATLAB的基本算术运算有： $+$ (加)、 $-$ (减)、 $*$ (乘)、 $/$ (右除)、 \backslash (左除)、 $^$ (乘方)。注意，运算是在矩阵意义下进行的，单个数据的算术运算只是一种特例。

(1) 矩阵加减运算

假定有两个矩阵A和B，则可以由A+B和A-B实现矩阵的加减运算。运算规则是：若A和B矩阵的维数相同，则可以执行矩阵的加减运算，A和B矩阵的相应元素相加减。如果A与B的维数不相同，则MATLAB将给出错误信息，提示用户两个矩阵的维数不匹配。

(2) 矩阵乘法

假定有两个矩阵A和B，若A为 $m \times n$ 矩阵，B为 $n \times p$ 矩阵，则 $C=A*B$ 为 $m \times p$ 矩阵。

(3) 矩阵除法

在MATLAB中，有两种矩阵除法运算： \backslash 和 $/$ ，分别表示左除和右除。如果A矩阵是非奇异方阵，则 $A \backslash B$ 和 B/A 运算可以实现。 $A \backslash B$ 等效于A的逆左乘B矩阵，也就是 $\text{inv}(A)*B$ ，而 B/A 等效于A矩阵的逆右乘B矩阵，也就是 $B*\text{inv}(A)$ 。

对于含有标量的运算，两种除法运算的结果相同，如 $3/4$ 和 $4 \backslash 3$ 有相同的值，都等于0.75。又如，设 $a=[10.5,25]$ ，则 $a/5=5 \backslash a=[2.1000 \ 5.0000]$ 。对于矩阵来说，左除和右除表示两种不同的除数矩阵和被除数矩阵的关系。对于矩阵运算，一般 $A \backslash B \neq B/A$ 。

(4) 矩阵的乘方

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。

2. 点运算

在MATLAB中，有一种特殊的运算，因为其运算符是在有关算术运算符前面加点，所以叫点运算。点运算符有 $.*$ 、 $./$ 、 $.\$ 和 $.^$ 。两矩阵进行点运算是指它们的对应元素进行相关运算，要求两矩阵的维参数相同。

2.3.2 关系运算

MATLAB提供了6种关系运算符： $<$ (小于)、 \leq (小于或等于)、 $>$ (大于)、 \geq (大于或等于)、 $==$ (等于)、 \sim (不等于)。它们的含义不难理解，但要注意其书写方法与数学中的不等式符号不尽相同。

关系运算符的运算法则为：

(1) 当两个比较量是标量时，直接比较两数的大小。若关系成立，关系表达式结果为1，否则为0。

(2) 当参与比较的量是两个维数相同的矩阵时，比较是对两矩阵相同位置的元素按标量关系运算规则逐个进行，并给出元素比较结果。最终的关系运算的结果是一个维数与原矩阵相同的矩阵，它的元素由0或1组成。

(3) 当参与比较的一个是标量，而另一个是矩阵时，则把标量与矩阵的每一个元素按标量关系运算规则逐个比较，并给出元素比较结果。最终的关系运算的结果是一个维数与原矩阵相同的矩阵，它的元素由0或1组成。

例2-8 产生5阶随机方阵A，其元素为[10,90]区间的随机整数，然后判断A的元素是否能被3整除。

(1) 生成5阶随机方阵A。

```
A=fix((90-10+1)*rand(5)+10)
```

(2) 判断A的元素是否可以被3整除。

```
P=rem(A,3)==0
```

其中，`rem(A,3)`是矩阵A的每个元素除以3的余数矩阵。此时，0被扩展为与A同维数的零矩阵，P是进行等于(==)比较的结果矩阵。

2.3.3 逻辑运算

MATLAB提供了3种逻辑运算符： $\&$ (与)、 $|$ (或)和 \sim (非)。逻辑运算的运算法则为：

(1) 在逻辑运算中，确认非零元素为真，用1表示，零元素为假，用0表示。

(2) 设参与逻辑运算的是两个标量a和b，那么，

$a\&b$ a,b全为非零时，运算结果为1，否则为0。

$a|b$ a,b中只要有一个非零，运算结果为1。

$\sim a$ 当a是零时，运算结果为1；当a非零时，运算结果为0。

(3) 若参与逻辑运算的是两个同维矩阵，那么运算将对矩阵相同位置上的元素按标量规则逐个进行。最终运算结果是一个与原矩阵同维的矩阵，其元素由1或0组成。

(4) 若参与逻辑运算的一个是标量，一个是矩阵，那么运算将在标量与矩阵中的每个元素之间按标量规则逐个进行。最终运算结果是一个与矩阵同维的矩阵，其元素由1或0组成。

- (5) 逻辑非是单目运算符，也服从矩阵运算规则。
- (6) 在算术、关系、逻辑运算中，算术运算优先级最高，逻辑运算优先级最低。

例2-9 建立矩阵A，然后找出大于4的元素的位置。

(1) 建立矩阵A。

$A=[4,-65,-54,0,6;56,0,67,-45,0]$

(2) 找出大于4的元素的位置。

`find(A>4)`

2.4 矩阵分析

2.4.1 对角阵与三角阵

1. 对角阵

只有对角线上有非0元素的矩阵称为对角矩阵，对角线上的元素相等的对角矩阵称为数量矩阵，对角线上的元素都为1的对角矩阵称为单位矩阵。

(1) 提取矩阵的对角线元素

设 A 为 $m \times n$ 矩阵，`diag(A)`函数用于提取矩阵 A 主对角线元素，产生一个具有 $\min(m,n)$ 个元素的列向量。

`diag(A)`函数还有一种形式`diag(A,k)`，其功能是提取第 k 条对角线的元素。

(2) 构造对角矩阵

设 V 为具有 m 个元素的向量，`diag(V)`将产生一个 $m \times m$ 对角矩阵，其主对角线元素即为向量 V 的元素。

`diag(V)`函数也有另一种形式`diag(V,k)`，其功能是产生一个 $n \times n$ ($n=m+1$) 对角阵，其第 k 条对角线的元素即为向量 V 的元素。

例2-10 先建立 5×5 矩阵A，然后将A的第一行元素乘以1，第二行乘以2，...，第五行乘以5。

```
A=[17,0,1,0,15;23,5,7,14,16;4,0,13,0,22;10,12,  
19,21,3;...
```

```
11,18,25,2,19];
```

```
D=diag(1:5);
```

```
D*A           %用D左乘A，对A的每行乘  
以一个指定常数
```

2. 三角阵

三角阵又进一步分为上三角阵和下三角阵，所谓上三角阵，即矩阵的对角线以下的元素全为0的一种矩阵，而下三角阵则是对角线以上的元素全为0的一种矩阵。

(1) 上三角矩阵

求矩阵A的上三角阵的MATLAB函数是triu(A)。
triu(A)函数也有另一种形式triu(A,k)，其功能是求矩阵A的第k条对角线以上的元素。例如，提取矩阵A的第2条对角线以上的元素，形成新的矩阵B。

(2) 下三角矩阵

在MATLAB中，提取矩阵A的下三角矩阵的函数是tril(A)和tril(A,k)，其用法与提取上三角矩阵的函数triu(A)和triu(A,k)完全相同。

2.4.2 矩阵的转置与旋转

1. 矩阵的转置

转置运算符是单撇号(')。

2. 矩阵的旋转

利用函数`rot90(A,k)`将矩阵A旋转 90° 的k倍，当k为1时可省略。

3. 矩阵的左右翻转

对矩阵实施左右翻转是将原矩阵的第一列和最后一列调换，第二列和倒数第二列调换，...，依次类推。MATLAB对矩阵A实施左右翻转的函数是`fliplr(A)`。

4. 矩阵的上下翻转

MATLAB对矩阵A实施上下翻转的函数是`flipud(A)`。

2.4.3 矩阵的逆与伪逆

1. 矩阵的逆

对于一个方阵A，如果存在一个与其同阶的方阵B，使得：
 $A \cdot B = B \cdot A = I$ (I为单位矩阵)

则称B为A的逆矩阵，当然，A也是B的逆矩阵。

求一个矩阵的逆是一件非常烦琐的工作，容易出错，但在MATLAB中，求一个矩阵的逆非常容易。求方阵A的逆矩阵可调用函数inv(A)。

例2-11 用求逆矩阵的方法解线性方程组。

$$Ax=b$$

其解为：

$$x=A^{-1}b$$

2. 矩阵的伪逆

如果矩阵A不是一个方阵，或者A是一个非满秩的方阵时，矩阵A没有逆矩阵，但可以找到一个与A的转置矩阵A'同型的矩阵B，使得：

$$A \cdot B \cdot A = A$$

$$B \cdot A \cdot B = B$$

此时称矩阵B为矩阵A的伪逆，也称为广义逆矩阵。在MATLAB中，求一个矩阵伪逆的函数是 `pinv(A)`。

2.4.4 方阵的行列式

把一个方阵看作一个行列式，并对其按行列式的规则求值，这个值就称为矩阵所对应的行列式的值。在MATLAB中，求方阵A所对应的行列式的值的函数是 $\det(A)$ 。

2.4.5 矩阵的秩与迹

1. 矩阵的秩

矩阵线性无关的行数与列数称为矩阵的秩。
在MATLAB中，求矩阵秩的函数是`rank(A)`。

2. 矩阵的迹

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。在MATLAB中，求矩阵的迹的函数是`trace(A)`。

2.4.6 向量和矩阵的范数

矩阵或向量的范数用来度量矩阵或向量在某种意义下的长度。范数有多种方法定义，其定义不同，范数值也就不同。

1. 向量的3种常用范数及其计算函数

在MATLAB中，求向量范数的函数为：

(1) `norm(V)`或`norm(V,2)`：计算向量 V 的2—范数。

(2) `norm(V,1)`：计算向量 V 的1—范数。

(3) `norm(V,inf)`：计算向量 V 的 ∞ —范数。

2. 矩阵的范数及其计算函数

MATLAB提供了求3种矩阵范数的函数，其函数调用格式与求向量的范数的函数完全相同。

2.4.7 矩阵的条件数

在MATLAB中，计算矩阵A的3种条件数的函数是：

(1) `cond(A,1)` 计算A的1—范数下的条件数。

(2) `cond(A)`或`cond(A,2)` 计算A的2—范数数下的条件数。

(3) `cond(A,inf)` 计算A的 ∞ —范数下的条件数。

2.4.8 矩阵的特征值与特征向量

在MATLAB中，计算矩阵A的特征值和特征向量的函数是`eig(A)`，常用的调用格式有3种：

(1) `E=eig(A)`：求矩阵A的全部特征值，构成向量E。

(2) `[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。

(3) $[V,D]=\text{eig}(A, 'nobalance')$: 与第2种格式类似，但第2种格式中先对A作相似变换后求矩阵A的特征值和特征向量，而格式3直接求矩阵A的特征值和特征向量。

例2-12 用求特征值的方法解方程。

$$3x^5-7x^4+5x^2+2x-18=0$$

`p=[3,-7,0,5,2,-18];`

`A=compan(p);`

`%A的伴随矩阵`

`x1=eig(A)`

`%求A的特征值`

`x2=roots(p)`

`%直接求多项式p的零`

`点`

2.5 矩阵的超越函数

1. 矩阵平方根sqrtm

`sqrtm(A)`计算矩阵A的平方根。

2. 矩阵对数logm

`logm(A)`计算矩阵A的自然对数。此函数输入参数的条件与输出结果间的关系和函数`sqrtm(A)`完全一样

3. 矩阵指数`expm`、`expm1`、`expm2`、`expm3`
`expm(A)`、`expm1(A)`、`expm2(A)`、`expm3(A)`的功能都求矩阵指数 e^A 。

4. 普通矩阵函数`funm`

`funm(A,'fun')`用来计算直接作用于矩阵 A 的由‘`fun`’指定的超越函数值。当`fun`取`sqrt`时，`funm(A,'sqrt')`可以计算矩阵 A 的平方根，与`sqrtm(A)`的计算结果一样。

2.6 字符串

在MATLAB中，字符串是用单撇号括起来的字符序列。

MATLAB将字符串当作一个行向量，每个元素对应一个字符，其标识方法和数值向量相同。也可以建立多行字符串矩阵。

字符串是以ASCII码形式存储的。**abs**和**double**函数都可以用来获取字符串矩阵所对应的ASCII码数值矩阵。相反，**char**函数可以把ASCII码矩阵转换为字符串矩阵。

例2-13 建立一个字符串向量，然后对该向量做如下处理：

- (1) 取第1~5个字符组成的子字符串。
- (2) 将字符串倒过来重新排列。
- (3) 将字符串中的小写字母变成相应的大写字母，其余字符不变。
- (4) 统计字符串中小写字母的个数。

命令如下：

```
ch='ABc123d4e56Fg9';  
subch=ch(1:5)          %取子字符串  
revch=ch(end:-1:1)     %将字符串倒排  
k=find(ch>='a'&ch<='z'); %找小写字母的位置  
ch(k)=ch(k)-('a'-'A'); %将小写字母变成相应  
的大写字母  
char(ch)  
length(k)              %统计小写字母的个数
```

与字符串有关的另一个重要函数是eval，其调用格式为：

eval(t)

其中t为字符串。它的作用是把字符串的内容作为对应的MATLAB语句来执行。

2.7 结构数据和单元数据

2.7.1 结构数据

1. 结构矩阵的建立与引用

结构矩阵的元素可以是不同的数据类型，它可将一组具有不同属性的数据纳入到一个统一的变量名下进行管理。建立一个结构矩阵可采用给结构成员赋值的办法。具体格式为：

结构矩阵名.成员名=表达式

其中表达式应理解为矩阵表达式。

2. 结构成员的修改

可以根据需要增加或删除结构的成员。例如要给结构矩阵a增加一个成员x4，可给a中任意一个元素增加成员x4：

`a(1).x4='410075';`

但其他成员均为空矩阵，可以使用赋值语句给它赋确定的值。

要删除结构的成员，则可以使用rmfield函数来完成。例如，删除成员x4：

`a=rmfield(a,'x4');`

3. 关于结构的函数

除了一般的结构数据的操作外，MATLAB还提供了部分函数来进行结构矩阵的操作。

2.7.2 单元数据

1. 单元矩阵的建立与引用

建立单元矩阵和一般矩阵相似，只是矩阵元素用大括号括起来。

可以用带有大括号下标的形式引用单元矩阵元素。例如**`b{3,3}`**。单元矩阵的元素可以是结构或单元数据。

可以使用**`celldisp`**函数来显示整个单元矩阵，如**`celldisp(b)`**。还可以删除单元矩阵中的某个元素。

2. 关于单元的函数

MATLAB还提供了部分函数用于单元的操作。

2.8 稀疏矩阵

2.8.1 矩阵存储方式

MATLAB的矩阵有两种存储方式：完全存储方式和稀疏存储方式。

1. 完全存储方式

完全存储方式是将矩阵的全部元素按列存储。以前讲到的矩阵的存储方式都是按这个方式存储的，此存储方式对稀疏矩阵也适用。

2. 稀疏存储方式

稀疏存储方式仅存储矩阵所有的非零元素的值及其位置，即行号和列号。在MATLAB中，稀疏存储方式也是按列存储的。

注意，在讲稀疏矩阵时，有两个不同的概念，一是指矩阵的0元素较多，该矩阵是一个具有稀疏特征的矩阵，二是指采用稀疏方式存储的矩阵。

2.8.2 稀疏存储方式的产生

1. 将完全存储方式转化为稀疏存储方式

函数`A=sparse(S)`将矩阵`S`转化为稀疏存储方式的矩阵`A`。当矩阵`S`是稀疏存储方式时，则函数调用相当于`A=S`。

`sparse`函数还有其他一些调用格式：

`sparse(m,n)`：生成一个 $m \times n$ 的所有元素都是0的稀疏矩阵。

`sparse(u,v,S)`--： `u,v,S`是3个等长的向量。`S`是要建立的稀疏矩阵的非0元素，`u(i)`、`v(i)`分别是`S(i)`的行和列下标，该函数建立一个`max(u)`行、`max(v)`列并以`S`为稀疏元素的稀疏矩阵。

此外，还有一些和稀疏矩阵操作有关的函数。例如

`[u,v,S]=find(A)`：返回矩阵`A`中非0元素的下标和元素。这里产生的`u,v,S`可作为`sparse(u,v,S)`的参数。

`full(A)`：返回和稀疏存储矩阵`A`对应的完全存储方式矩阵。

2. 产生稀疏存储矩阵

只把要建立的稀疏矩阵的非0元素及其所在行和列的位置表示出来后由MATLAB自己产生其稀疏存储，这需要使
用spconvert函数。调用格式为：

B=spconvert(A)

其中A为一个 $m \times 3$ 或 $m \times 4$ 的矩阵，其每行表示一个非0元素，m是非0元素的个数，A每个元素的意义是：

(i,1) 第i个非0元素所在的行。

(i,2) 第i个非0元素所在的列。

(i,3) 第i个非0元素值的实部。

(i,4) 第i个非0元素值的虚部，若矩阵的全部元素都是实数，则无须第四列。

该函数将A所描述的一个稀疏矩阵转化为一个稀疏存储矩阵。

例2-15 根据表示稀疏矩阵的矩阵A，产生一个稀疏存储方式矩阵B。

命令如下：

```
A=[2,2,1;3,1,-1;4,3,3;5,3,8;6,6,12];  
B=spconvert(A)
```

3. 带状稀疏存储矩阵

用spdiags函数产生带状稀疏矩阵的稀疏存储，调用格式是：

$A = \text{spdiags}(B, d, m, n)$

其中，参数 m, n 为原带状矩阵的行数与列数。 B 为 $r \times p$ 阶矩阵，这里 $r = \min(m, n)$ ， p 为原带状矩阵所有非零对角线的条数，矩阵 B 的第 i 列即为原带状矩阵的第 i 条非零对角线。

4. 单位矩阵的稀疏存储

单位矩阵只有对角线元素为1，其他元素都为0，是一种具有稀疏特征的矩阵。函数 `eye` 产生一个完全存储方式的单位矩阵。

MATLAB还有一个产生稀疏存储方式的单位矩阵的函数，这就是 `speye`。函数 `speye(m,n)` 返回一个 $m \times n$ 的稀疏存储单位矩阵。

2.8.3 稀疏矩阵应用举例

稀疏存储矩阵只是矩阵的存储方式不同，它的运算规则与普通矩阵是一样的。所以，在运算过程中，稀疏存储矩阵可以直接参与运算。当参与运算的对象不全是稀疏存储矩阵时，所得结果一般是完全存储形式。