第四章 数值计算

数值计算在现代科学研究和工程技术领域中应用最为广泛。MATLAB正是凭借其卓越的数值计算功能而称雄一方。现在MATLAB拥有的数值计算功能已经非常强大。本章重点讲述在MATLAB工作环境下解决具体的数值计算问题，包含高等数学、线性代数、数据分析中的很多重要内容：矩阵的计算和分解、多项式运算、函数的零极点、数值微积分等。

本章主要内容包括：

- 矩阵的多种运算方法MATLAB实现
- 计算矩阵的秩、特征值及其对应的特征向量
- 利用矩阵操作求解线性方程组
- 利用多项式实现数据点的插值和拟合
- 利用ode()指令求初值问题的常微分方程数值解

4.1 矩阵的计算

4.1.1 矩阵的结构变换

1.矩阵转置的MATLAB运算符为单引号“’ ”。

☞ A'

☞求A的转置，其中A可以是行向量、列向量和矩阵。

【例4-1】矩阵转置的MATLAB实例。

```
>> clear,x1=[1;3;5];%3×1的列向量
>> y1=x1'    %转置结果为1×3的行向量
y1 =
   1    3    5
>> x2=[1 2 3; 4 5 6];    %2×3的矩阵
>> y2=x2'     % 转置结果为3×2的矩阵
y2 =
   1    4
   2    5
   3    6
```

4.1.1 矩阵的结构变换

```
>> x3=[1-2*i 3+4*i; 1-4*i 2+6*i;3-i -5+4*i]; % 3×2的复数矩阵
>> y3=x3'                %转置结果为2×3的共轭矩阵
y3 =
  1.0000 + 2.0000i   1.0000 + 4.0000i   3.0000 + 1.0000i
  3.0000 - 4.0000i   2.0000 - 6.0000i  -5.0000 - 4.0000i
>> x3=[1-2*i 3+4*i; 1-4*i 2+6*i;3-i -5+4*i];%3×2的复数矩阵
>> y3=x3.‘          % 的点转置结果为2×3的非共轭矩阵
y3 =
  1.0000 - 2.0000i   1.0000 - 4.0000i   3.0000 - 1.0000i
  3.0000 + 4.0000i   2.0000 + 6.0000i  -5.0000 + 4.0000i
```

4.1.1 矩阵的结构变换

- 对于复数矩阵，MATLAB完成的是复数矩阵的共轭转置。复数矩阵的非共轭转置的运算符为“.‘” 。
- 注意：矩阵转置的操作符必须在英文状态下输入。

2.对称变换

指令flipud()和fliplr()完成矩阵的对称变换，调用格式如下：

☞B = flipud(A)

☞上下方向翻转的矩阵。如果是列向量，返回相反顺序的向量；如果是行向量，返回原向量。

☞B = fliplr(A)

☞水平方向翻转的矩阵。如果是行向量，返回相反顺序的向量；如果是列向量，返回原向量。

另外，MATLAB还提供了一个以指定维翻转的函数 flipudim(). 格式为：

☞B=flipdim(A,dim)

☞返回以指定的维翻转矩阵。dim =1，以行方向翻转；dim =2，以列方向翻转。可见flipdim(A,1)操作效果等同于flipud(A), flipdim(A,2) 操作效果等同于fliplr(A)。

【例4-2】矩阵的结构变换实例。

```
>> clear,B=[1 2 3 4;5 6 7 8;9 10 11 12;13 14 15 16]
B =
     1     2     3     4
     5     6     7     8
     9    10    11    12
    13    14    15    16
>> flipud(B)    %将矩阵B上下方向翻转矩阵
ans =
    13    14    15    16
     9    10    11    12
     5     6     7     8
     1     2     3     4
>> fliplr(B)       %将矩阵B水平方向翻转
ans =
     4     3     2     1
     8     7     6     5
    12    11    10     9
    16    15    14    13
```

3.旋转

指令rot90()可以完成矩阵逆时针旋转90度。

☞rot90(A,k)

☞矩阵A逆时针翻转k×90度。

【例4-3】矩阵旋转变换实例。

```
>> clear,x=[1 2;3 4;5 6];
>> A1=rot90(x)%逆时针旋转90度
A1 =
    2     4     6
    1     3     5
>> A2=rot90(x,2)%逆时针旋转2×90度
A2 =
    6     5
    4     3
    2     1
>> A3=rot90(x,3)%逆时针旋转3×90度
A3 =
    5     3     1
    6     4     2
```

4.提取三角阵

MATLAB中提取上三角矩阵的函数为triu()，提取下三角矩阵的函数为tril()，格式如下：

☞triu(X,K)

☞K=0时，提取X的主对角线及以上的元素。K>0时，提取矩阵X主对角线上方的第K条对角线以上的元素。K<0时，提取矩阵X的主对角线下方第-K条主对角线以上的元素。K缺省时默认为0。

☞tril (X,K)

☞K=0时，提取X的主对角线及以下的元素。其中K>0时，提取矩阵X主对角线上方的第K条对角线以下的元素。K<0时，提取矩阵X的主对角线下方第-K条主对角线以下的元素。K缺省时默认为0。

【例4-6】提取三角阵指令实例。

```
>> clear,x=magic(3);%创建3阶魔方矩阵
>> A=triu(x) %提取主对角线上三角矩阵
A =
    8     1     6
    0     5     7
    0     0     2
>> B=tril(x)% 提取主对角线下三角矩阵
B =
    8     0     0
    3     5     0
    4     9     2
>> B1=tril(x,1)% 提取主对角线上方第一条对角线下三角矩阵
B1 =
    8     1     0
    3     5     7
    4     9     2
```

4.1.2 矩阵分析

矩阵分析是线性代数中关于矩阵运算的重要环节。矩阵分析包括矩阵的秩、矩阵对应的行列式和逆矩阵等运算，MATLAB提供了相应的指令。用户在计算中只需要正确调用这些指令，就可以快速地得到结果。

1秩

矩阵中线性无关的行数和列数成为矩阵的秩。在MATLAB中求秩函数为rank（ ）。rank的格式为：

☞rank(A)

☞计算矩阵A的秩。

【例4-7】 求下列矩阵的秩

（1）　　（2）

（1）>> clear,a=[1 4 2;6 7 2;5 -4 -8];

>> R1=rank(a)

R1 =

3

（2）>> b=[5 4 -2 ;4 5 2;-2 2 8] ;

>> R2=rank(b)

R2 =

2

1秩

MATLAB还提供了一个**化简函数rref()**，其格式如下：

☞rref(A)

☞将矩阵A化为行阶梯阵。

根据MATLAB的输出结果，用户可以得到矩阵的秩，即行阶梯阵中非零行的行数

【例4-8】（续例4-7）行阶梯阵化简函数rref（ ）应用实例。

```
>> rref(a)   %将矩阵a化简为行阶梯阵
ans =
     1     0     0
     0     1     0
     0     0     1
>> rref(b)      %将矩阵b化简为行阶梯阵
ans =
     1     0    -2
     0     1     2
     0     0     0
```

由化简结果可知，矩阵a的秩为3，矩阵b的秩为2.。

2．行列式和逆矩阵

把方阵A看做行列式|A|，对其按照行列式的规则求值，称该值为行列式的值。MATLAB的实现指令为det()，其格式为：

☞det(A)

☞计算方阵A对应的行列式的值

行列式的值为0时，相应的矩阵成为奇异矩阵，否则称为非奇异矩阵（或满秩矩阵）。

对于非奇异矩阵A，有与其同型的非奇异矩阵B，使得A×B=B×A=I，其中I为单位矩阵，则A和B互为逆矩阵。求方阵A逆矩阵的MATLAB函数为inv()，格式为：

☞B=inv(A)

☞求非奇异矩阵A的逆矩阵B

【例4-9】（续例4-7）矩阵对应行列式的值、逆矩阵和广义逆矩阵运算实例。

```
>> det(a)
ans =
   66
>> det(b)  %计算矩阵对应行列式的值
ans =
    0
>> inv(a)   %矩阵a为非奇异矩阵，存在逆矩阵
ans =
  -0.7273    0.3636   -0.0909
   0.8788   -0.2727    0.1515
  -0.8939    0.3636   -0.2576
>> pinv(b)  %矩阵b为奇异矩阵，存在伪逆矩阵
ans =
   0.0617    0.0494   -0.0247
   0.0494    0.0617    0.0247
  -0.0247    0.0247    0.0988
```

【说明】

对于奇异矩阵，调用MATLAB指令inv()也可以获得计算结果.此时MATLAB会给出警告：Warning: Matrix is close to singular or badly scaled.用户不要将输出矩阵认为是该奇异矩阵的逆矩阵，必须通过逆矩阵定义加以验证。

【例4-10】逆矩阵的应用：

设 A= $\begin{pmatrix}1 & 2 & 3\\ 2 & 3 & 4\\ 3 & 4 & 3\end{pmatrix}$,B = $\begin{pmatrix}2 & 5\\ 4 & 3\end{pmatrix}$,C= $\begin{pmatrix}1 & 3\\ 4 & 0\\ 2 & 1\end{pmatrix}$ ，

求矩阵X，使满足: $AXB = C$

【例4-10】逆矩阵的应用

由线性代数可知：$X = A^{-1}CB^{-1}$，或者 $X = A\backslash C/B$。本题给出两种计算程序，并比较计算结果和耗时。

编写文件名为exm4_10的脚本文件：

```
clear,clc,
A=[1 2 3;2 3 4;3 4 3];B=[2,5;4 3];C=[1 3;4 0;2 1];
%显示利用逆矩阵求解时的耗时
tic %开启计时器
X1=inv(A)*C*inv(B)
toc %结束计时，并输出时间
%显示利用矩阵除法求解时的耗时
tic
X2=A\C/B
toc
```

【例4-10】逆矩阵的应用

在指令窗中执行exm4_11：

```
X1 =
   -4.7500    4.2500
    4.3571   -3.9286
   -1.1071    1.1786
Elapsed time is 0.000201 seconds.
X2 =
   -4.7500    4.2500
    4.3571   -3.9286
   -1.1071    1.1786
Elapsed time is 0.000137 seconds.
```

【说明】

利用逆矩阵求解方程组时需要两步计算，即求逆矩阵和矩阵乘法，因此较左除法求解方程组速度要慢。

- 程序在不同计算机上执行时耗时不唯一，但时长顺序不会改变。

如果矩阵A不是方阵或是奇异矩阵时，该矩阵存在与A的转置矩阵同型的矩阵B，使A×B×A=A,B×A×B=B成立，则B称为A的逆，即伪逆矩阵。求伪逆矩阵的MATLAB函数为pinv(),其格式为：

☞B=pinv(A)

☞求矩阵A的伪逆矩阵B

【例4-11】求方程组的 $\begin{cases} 2x_1 + x_2 + 7x_3 + x_4 = 3 \\ 8x_1 - 3x_2 + 2x_3 + 6x_4 = 5 \end{cases}$ 最小范数解。

分析：该方程组为欠定方程组，系数矩阵不是方阵，因此只能求其伪逆矩阵。

编写文件名为exm4_11的脚本文件：

```
clear,clc
A=[2 1 7 1;8 -3 2 6];b=[3 5]';
X=pinv(A)*b
```

在指令窗中执行exm4_11：

```
X =
    0.3426
   -0.0691
    0.3063
    0.2400
```

3.矩阵的迹和范数

矩阵的迹：矩阵的对角线元素之和，即矩阵的特征值之和。**MATLAB中求迹函数是trace()**，格式如下：

☞trace(A)

☞求矩阵A的迹。

向量种度量形式。使用范数可以测量两个向量或矩阵之间的距离。范数有多种定义形式。**MATLAB计算向量或矩阵的范数指令为norm()**，格式如下：

☞norm (A,k)

☞计算向量或矩阵A的k—范数:k可以取1、2、Inf、'fro',分别为计算矩阵的1-范数、2-范数、无穷大-范数、Frobenius-范数,k缺省时为计算2-范数。

【例4-12】计算矩阵 $\begin{pmatrix} 8 & 6 & -1 & 1 \\ 4 & 3 & -1 & 2 \\ 5 & 3 & -3 & 4 \\ 3 & 4 & -7 & -2 \end{pmatrix}$ 的迹和范数。

```
>> clear,A=[8 6 -1 1;4 3 -1 2;5 3 -3 4;3 4 -7 -2];
>> trace(A) %计算矩阵的迹
ans =
    6
>> norm(A,1) %计算矩阵的1-范数
ans =
   20
>> norm(A) %计算矩阵的2-范数
ans =
  14.6918
```

【例4-12】计算矩阵 $\begin{pmatrix} 8 & 6 & -1 & 1 \\ 4 & 3 & -1 & 2 \\ 5 & 3 & -3 & 4 \\ 3 & 4 & -7 & -2 \end{pmatrix}$ 的迹和范数。

```
>> norm(A,inf) %计算矩阵的无穷大-范数
ans =
   16
>> norm(A,'fro') %计算矩阵的Frobenius-范数
ans =
  16.4012
>> norm(A(:,2),1) %计算向量的1-范数
ans =
   16
```

4.条件数

矩阵的条件数是解线性方程组是判断系数矩阵的变化对解的影响程度的一个参数。矩阵的条件数的定义依赖于范数。矩阵的条件数接近1时表示方程为良性的，否则为病态。

MATLAB计算矩阵条件数的指令为cond(),格式如下：

☞cond(A,p)

☞计算矩阵A的p—范数:p可以取1、2、Inf、'fro',分别为计算矩阵的1-范数条件数、2-范数条件数、无穷大-范数条件数、Frobenius-范数条件数,p缺省时为计算2-范数条件数。

【例4-13】（续例4-12）计算矩阵A的条件数。

```
>> cond(A,1) %计算矩阵的1-范数条件数
ans =
 108.1720
>> cond(A) %计算矩阵的2-范数条件数
ans =
  51.7487
>> cond(A,inf) %计算矩阵的无穷大-范数条件数
ans =
  67.0968
>> cond(A,'fro') %计算矩阵的Frobenius-范数条件数
ans =
  58.0285
```

4.1.3矩阵的特征值分析

设n阶方阵A，如果数 λ 和n维非零列向量 x，使关系式 $AX=\lambda X$ 成立，则数 λ 称为方阵A的特征值，非零向量 X 称为矩阵A的对应于特征值 λ 的特征向量，MATLAB计算矩阵特征值和特征向量的函数为eig()，格式为：

- [V,D]=eig(A)
- 计算矩阵A的特征值D和对应的特征向量V，使得AV=VD。如果函数只有一个输出宗量，则只给出特征值。

【例4-14】计算3阶魔方矩阵的特征值和特征向量。

```
>>clear, A=magic(3); %创建3阶魔方矩阵
>> d=eig(A)
d =
   15.0000
    4.8990
   -4.8990
>> [V,D]=eig(A)
V =
   -0.5774   -0.8131   -0.3416
   -0.5774    0.4714   -0.4714
   -0.5774    0.3416    0.8131
D =
   15.0000         0         0
         0    4.8990         0
         0         0   -4.8990
```

4.1.4 矩阵的分解

矩阵分解是线性代数中比较重要的内容，针对不同的分解方法，MATLAB提供了多个关于矩阵分解的函数，具体见表4-1所示。

表4-1 矩阵分解函数及其功能表

分解类别	调用格式	功能
LU分解	[L,U]=lu(A)	计算上三角矩阵U和交换下三角矩阵L，使LU=A
	[L,U,P]=lu(A)	计算上三角矩阵U、有单位对角线的下三角矩阵L和交换矩阵P，满足LU=PA
Cholesky因式分解	R=chol(A)	计算正定矩阵A的Cholesky因子，是一个上三角矩阵，满足R'*R=A。如果A不是一个正定矩阵，则给出一个错误信息。
	[G,p]=chol(A)	计算矩阵A的Cholesky因子G。如果A不是一个正定矩阵，则不给出错误信息，而是将p设为正整数。
qr因式分解	[Q,R]=qr(A)	计算m×n矩阵A分解得到的m×n酉矩阵Q和m×n的上三角矩阵R。Q的列形成了一个正交基。Q和R满足A=Q*R
	[Q,R,P]=qr(A)	计算矩阵Q、上三角矩阵R和交换矩阵P。Q的列形成一个正交基，R的对角线元素按大小降序排列，满足A*P=Q*R
SVD分解	s=svd(A)	计算一个包含矩阵A的奇异值的向量s
	[U,S,V]=svd(A)	计算奇异值为对角线且与A同型的矩阵S，以及m×m和n×n的酉矩阵U和V，使A=U*S*V’

【例4-15】利用LU分解法求欠定方程组的一个特解。

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

分析：方程组形式为AX=b，A=LU，即LUX=b，由矩阵除法可知X=U\(L\b)。

```
>>clear,clc,
>> A=[3 2 -1;5 3 0];b=[2;7];[L,U]=lu(A);
>> X=U\(L\b)
X =
   1.4000
        0
2.2000
```

4.1.5 线性方程组的求解

线性方程组分为齐次方程组和非齐次方程组两类。对于系数矩阵为满秩矩阵的方程组，可以将方程组表述为Ax=b,其解为

$x=A^{-1}b$.MATLAB语句为

x=inv(A)*b 或者 x=A\b.

【例4-16】解方程组

$$\begin{cases} 2x_1 + x_2 - x_3 + x_4 = 2 \\ 5x_1 + 3x_2 - x_3 + 2x_4 = 3 \\ 2x_1 + 3x_2 - 2x_3 + 4x_4 = 5 \\ 3x_1 + 3x_2 - 3x_3 + 2x_4 = 3 \end{cases}$$

【例4-16】解方程组

```
>>clear, A=[2 1 -1 1;5 3 -1 2;2 3 -2 4;3 3 -3 2];
>> det(A)   %由行列式值是否为零判断A是否存在逆矩阵
ans =
   16
>> b=[2 3 5 3]';
>> x=inv(A)*b
x =
   0.5000
  -1.0000
  -0.5000
1.5000
```

由执行结果可知，方程组的解为：x1=0.5;x2=-1;
x3=-0.5;x4=1.5.

该题也可以通过执行语句 x=A\b求得。

对于系数矩阵为非满秩矩阵，即奇异矩阵，MATLAB提供了针对齐次线性方程组和非齐次线性方程组的不同求解函数。

齐次线性方程组求解

对于系数阵A为奇异阵的齐次线性方程组，可以利用函数rref(A)将系数阵化为行阶梯阵，对照化简结果可以直接写出方程组的解。

【例4-17】解方程组

$$\begin{cases} 2x_1 + x_2 + x_3 - x_4 = 0 \\ 4x_1 + 2x_2 + 2x_3 - 2x_4 = 0 \\ x_1 + 3x_2 + 2x_3 - 4x_4 = 0 \\ x_1 + 2x_2 + 2x_3 + 2x_4 = 0 \end{cases}$$

【例4-17】解方程组

MATLAB求解过程如下：

```
>> clear, A=[2 1 1 -1;4 2 2 -2;1 3 2 -4;1 2 2 2];
>> format rat     %有理格式输出
>> x=rref(A)
x =
     1          0          0          -4/3
     0          1          0          -6
     0          0          1          23/3
     0          0          0           0
```

由执行结果可知，该方程组有无穷多解，其通解为：

$$
\begin{cases}
x_1 = \dfrac{4}{3} x_4 \\
x_2 = 6 x_4 \\
x_3 = \dfrac{23}{3} x_4
\end{cases}
$$

【例4-17】解方程组

MATLAB还提供了函数null（ ）用来求解齐次线性方程组解空间的一组基（即基础解系），格式如下：

☞X=null（A，‘r’ ）

☞求系数矩阵为A的齐次方程组一组基础解系（即一组基）。r是可选参数：当有r时，输出有理基；当无r时，输出正交规范基。

【例4-18】（续例4-17）利用函数null（ ）求解齐次线性方程组示例。

```
>> x=null(A,'r')
x =
      4/3
      6
     -23/3
      1
```

则方程组的解为 $x=k\begin{pmatrix}\frac{4}{3}\\6\\-\frac{23}{3}\\1\end{pmatrix}$ 结果与【例4-14】一致。

【例4-19】利用函数null（ ）求解齐次线性方程组

$$\begin{cases} x_1 + 2x_2 + 2x_3 + x_4 = 0 \\ 2x_1 + x_2 - 2x_3 - 2x_4 = 0 \\ x_1 - x_2 - 4x_3 - 3x_4 = 0 \end{cases}$$

```
>> clear, A=[1 2 2 1;2 1 -2 -2;1 -1 -4 -3];
>> format  rat     %有理式格式输出
>> x=null(A,'r')  %求解空间的有理基
x =
     2          5/3
    -2         -4/3
     1          0
     0          1
```

则方程组的解为 $x = k_1 \begin{pmatrix} 2 \\ -2 \\ 1 \\ 0 \end{pmatrix} + k_2 \begin{pmatrix} \frac{5}{3} \\ -\frac{4}{3} \\ 0 \\ 1 \end{pmatrix}$。

2.非齐次线性方程组求解

非齐次线性方程组可以表述为AX=b，其中A为系数矩阵，其解存在3种可能：

（1）无解：系数矩阵A的秩n小于增广阵(A,b)的秩，则方程组无解；

（2）唯一解：系数阵A的秩n等于增广阵(A,b)的秩且等于未知变量的个数m

（3）无穷多解：系数阵A的秩n等于增广阵(A,b)的秩且小于未知变量的个数m，则方程组有无穷多解。

对于情况（3），MATLAB必须要给出该方程的一个特解和对应齐次方程的(n-m)个通解

求解通解用函数null()，特解可以利用矩阵LU分解法求得。

编写一个文件名为equs()函数文件:

```
function X=equs(A,b,n)
%EQUS  Solving the homogeneous or inhomogeneous linear equations
%  A 方程组的系数矩阵
%  b 常数项对应的列向量
%  n 方程组中变量个数
%  X 线性方程组的解
if nargin>3
   error('输入宗量太多，请重新输入')
end
B=[A,b];                          %增广矩阵
R_A=rank(A);                      %求系数矩阵的秩
R_B=rank(B);                      %求增广矩阵的秩
if R_A>n|R_B>n
   error('输入变量个数有误，请重新输入')
end
```

编写一个文件名为equs()函数文件:

```
format rat %有理格式显示
if R_A~=R_B        %是否无解
    X='Equations have no solution!';
elseif  R_A==R_B&R_A==n                  %是否是唯一解
   X=A\b;
 else
   [L,U]=lu(A);
   X=U\(L\b);                      %求特解
   C=null(A,'r')                    %求通解
end
end
```

用户利用该函数文件，可以用来求解任意类型的齐次或者非齐次线性方程组。

【例4-20】求解方程组 $\begin{cases} x_1 + x_2 - 3x_3 - x_4 = 1 \\ 3x_1 - x_2 - 3x_3 + 4x_4 = 4 \\ x_1 + 5x_2 - 9x_3 - 8x_4 = 0 \end{cases}$

在指令窗中执行：

```
>> clear,
>> A=[1 1 -3 -1;3 -1 -3 4;1 5 -9 -8];b=[1 4 0]';n=4;
>> X=equs(A,b,n)
Warning: Rank deficient,rank = 2, tol = 9.0189e-015.  In equs at 23
C =
     3/2          -3/4
     3/2           7/4
     1             0
     0             1
X =
     0
     0
    -8/15
     3/5
```

则该方程组的解为：

$$x = k_1 \begin{pmatrix} \frac{3}{2} \\ \frac{3}{2} \\ 1 \\ 0 \end{pmatrix} + k_2 \begin{pmatrix} -\frac{3}{4} \\ \frac{7}{4} \\ 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -\frac{8}{15} \\ \frac{3}{5} \end{pmatrix}$$

4.2 多项式

多项式运算是数学运算中最基本运算方式之一，也是线性代数分析和设计中的重要内容。MATLAB提供了多个函数可以帮助用户完成多项式的运算。

4.2.1 多项式的表达和创建

MATLAB中将n阶多项式p(x)存储在长度为n+1的行向量p中，行向量p的元素为多项式的系数，并按x的降幂排列。行向量

$p=[a_n,a_{n-1},\ldots,a_1,a_0]$代表的多项式为：

$$p(x)=a_nx^n+a_{n-1}x^{n-1}+\ldots+a_1x+a_0$$

注意：多项式中缺少某个幂次项，则在创建行向量时将该幂次项的系数写为0.

【例4-21】创建多项式 $p(x)=4x^5-3x^4+x$

```
>> p=[4,-3,0,0,1,0] %三次项、二次项、常数项系数均为0
p =
   4   -3   0   0   1   0
```

4.2.2 多项式的运算

1.多项式的加法和减法运算

多项式加减运算实质上为两个向量的加减运算。如果两个多项式向量长度相同，标准的向量加法和减法是有效的。当两个多项式阶次不同时，必须将较短的（即低阶多项式）向量前面用若干个零填补，使得两个向量长度相同，然后再进行运算。

【例4-22】将多项式 $a(x)=x^3-2x^2+6$ 与 x^2+2x+3 相加减。

```
>> a=[1 -2 0 6];
>> b=[0 1 2 3];%将低阶多项式对于行向量补零
>> c=a+b %进行加法运算
c =
    1    -1    2    9
```

【例4-22】将多项式 $a(x)=x^3-2x^2+6$ 与 $b(x)=x^2+2x+3$ 相加减。

```
>> sum=poly2str(c,'x') %标准形式输出
sum =
  x^3 - 1 x^2 + 2 x + 9
>> d=a-b %进行减法运算
d =
    1    -3    -2     3
>> dif=poly2str(d,'x') %标准形式输出
dif =
  x^3 - 3 x^2 - 2 x + 3
```

2.多项式的乘法和除法运算

两个多项式乘法运算实质是两个多项式系数的卷积运算。MATLAB中计算多项式卷积的指令为conv(),其调用格式为：

☞conv(a,b)

☞计算两个多项式的乘积，其中a,b分别为两个多项式的系数向量。

多项式除法为乘法的逆运算，MATLAB中的除法运算指令为deconv()，其调用格式为：

☞[q,r]=deconv(a,b),

☞计算两个多项式的除法，其中a,b为被除数多项式和除数多项式的系数向量，q、r分别为商多项式和余数多项式的系数向量。

【例4-23】（续例4-22）运用函数conv()和deconv()进行多项式运算实例。

```
>> a=[1 -2 0 6];
>> b=[1 2 3];
>> m=conv(a,b) %进行乘法运算
m =
    1    0    -1     0    12    18
>> pro=poly2str(m,'x') %标准形式输出
pro =
   x^5 - 1 x^3 + 12 x + 18
```

【例4-23】（续例4-22）运用函数conv()和deconv()进行多项式运算实例。

```
>> [q,r]=deconv(a,b) %进行除法运算
q =
    1    -4
r =
    0     0     5    18
>> q=poly2str(q,'x') %输出商多项式的标准形式
q =
   x - 4
>> r=poly2str(r,'x') %输出余多项式的标准形式
r =
   5 x + 18
```

3.多项式求根

MATLAB中求多项式全部根的函数为roots()，其格式为：

☞x=roots(p)

☞计算多项式的全部根，并以列向量形式赋给变量x。其中p为多项式的系数向量。

【说明】

在MATLAB中，多项式和根都表示为向量，其中多项式是行向量，根为列向量。

如果已知多项式的根x，可以用函数poly()创建多项式：

☞p=poly(x)

☞由多项式的根向量x创建多项式系数向量p，其中p的第一个元素为1，对应多项式最高阶系数为1。