

# MATLAB 实验指导书

## 一、基础知识

### 1.1 常见数学函数

函数名	数学计算功能	函数名	数学计算功能
abs (x)	实数的绝对值或复数的幅值	floor (x)	对 x 朝 $-\infty$ 方向取整
acos (x)	反余弦 $\arcsin x$	gcd (m, n)	求正整数 m 和 n 的最大公约数
acosh (x)	反双曲余弦 $\operatorname{arccosh} x$	imag (x)	求复数 x 的虚部
angle (x)	在四象限内求复数 x 的相角	lcm (m, n)	求正整数 m 和 n 的最小公倍数
asin (x)	反正弦 $\arcsin x$	log (x)	自然对数 (以 $e$ 为底数)
asinh (x)	反双曲正弦 $\operatorname{arsinh} x$	log10 (x)	常用对数 (以 10 为底数)
atan (x)	反正切 $\arctan x$	real (x)	求复数 x 的实部
atan2 (x,y)	在四象限内求反正切	rem (m, n)	求正整数 m 和 n 的 m/n 之余数
atanh (x)	反双曲正切 $\operatorname{arctanh} x$	round (x)	对 x 四舍五入到最接近的整数
ceil (x)	对 x 朝 $+\infty$ 方向取整	sign (x)	符号函数: 求出 x 的符号
conj (x)	求复数 x 的共轭复数	sin (x)	正弦 $\sin x$
cos (x)	余弦 $\cos x$	sinh (x)	反双曲正弦 $\sinh x$
cosh (x)	双曲余弦 $\cosh x$	sqrt (x)	求实数 x 的平方根: $\sqrt{x}$
exp (x)	指数函数 $e^x$	tan (x)	正切 $\tan x$
fix (x)	对 x 朝原点方向取整	tanh (x)	双曲正切 $\tanh x$

如: 输入  $x=[-4.85 \quad -2.3 \quad -0.2 \quad 1.3 \quad 4.56 \quad 6.75]$ , 则:

```

ceil(x)=  -4    -2     0     2     5     7
fix(x)=  -4    -2     0     1     4     6
floor(x)= -5    -3    -1     1     4     6
round(x)= -5    -2     0     1     5     7

```

### 1.2 系统的在线帮助

#### 1 help 命令:

1. 当不知系统有何帮助内容时, 可直接输入 **help** 以寻求帮助:

```
>> help (回车)
```

2. 当想了解某一主题的内容时, 如输入:

```
>> help syntax (了解 Matlab 的语法规则)
```

3. 当想了解某一具体的函数或命令的帮助信息时, 如输入:

```
>> help sqrt (了解函数 sqrt 的相关信息)
```

#### 2 lookfor 命令

现需要完成某一具体操作, 不知有何命令或函数可以完成, 如输入:

>> lookfor line （查找与直线、线性问题有关的函数）

## 1.3 常量与变量

系统的变量命名规则：变量名区分字母大小写；变量名必须以字母打头，其后可以是任意字母，数字，或下划线的组合。此外，系统内部预先定义了几个有特殊意义和用途的变量，见下表：

特殊的变量、常量	取 值
ans	用于结果的缺省变量名
pi	圆周率 $\pi$ 的近似值 (3.1416)
eps	数学中无穷小 (epsilon) 的近似值 (2.2204e - 016)
inf	无穷大，如 $1/0 = \text{inf}$ (infinity)
NaN	非数，如 $0/0 = \text{NaN}$ (Not a Number)， $\text{inf} / \text{inf} = \text{NaN}$
i, j	虚数单位： $i = j = \sqrt{-1}$

### 1 数值型向量（矩阵）的输入

1. 任何矩阵（向量），可以直接按行方式输入每个元素：同一行中的元素用逗号（,）或者用空格符来分隔；行与行之间用分号（;）分隔。所有元素处于一方括号（[]）内；

例 1：

```
>> Time = [11 12 1 2 3 4 5 6 7 8 9 10]
```

```
>> X_Data = [2.32 3.43; 4.37 5.98]
```

2. 系统中提供了多个命令用于输入特殊的矩阵：

函数	功 能	函数	功 能
compan	伴随阵	toeplitz	Toeplitz 矩阵
diag	对角阵	vander	Vandermonde 矩阵
hadamard	Hadamard 矩阵	zeros	元素全为 0 的矩阵
hankel	Hankel 矩阵	ones	元素全为 1 的矩阵
invhilb	Hilbert 矩阵的逆阵	rand	元素服从均匀分布的随机矩阵
kron	Kronercker 张量积	randn	元素服从正态分布的随机矩阵
magic	魔方矩阵	eye	对角线上元素为 1 的矩阵
pascal	Pascal 矩阵	meshgrid	由两个向量生成的矩阵

上面函数的具体用法，可以用帮助命令 help 得到。如：meshgrid(x,y)

输入 x=[1 2 3 4]; y=[1 0 5]; [X,Y]=meshgrid(x, y), 则

X =

1	2	3	4
1	2	3	4

Y =

1	1	1	1
0	0	0	0

1      2      3      4                      5      5      5      5

目的是将原始数据  $x, y$  转化为矩阵数据  $X, Y$ 。

## 2 符号向量（矩阵）的输入

1. 用函数 `sym` 定义符号矩阵：

函数 `sym` 实际是在定义一个符号表达式，这时的符号矩阵中的元素可以是任何的符号或者是表达式，而且长度没有限制。只需将方括号置于单引号中。

例 2：

```
>> sym_matrix = sym ('[a b c; Jack Help_Me NO_WAY]')
sym_matrix =
      [ a,      b,      c]
 [Jack, Help_Me, NO_WAY]
```

2. 用函数 `syms` 定义符号矩阵

先定义矩阵中的每一个元素为一个符号变量，而后像普通矩阵一样输入符号矩阵。

例 3：

```
>> syms a b c ;
>> M1 = sym ('Classical');
>> M2 = sym ('Jazz');
>> M3 = sym ('Blues');
>> A = [a b c; M1, M2, M3; sym ([2 3 5])]
A =
      [ a,      b,      c]
 [Classical, Jazz, Blues]
      [ 2,      3,      5]
```

## 1.4 数组（矩阵）的点运算

运算符：+（加）、-（减）、./（右除）、.\（左除）、.^（乘方），

例 4：

```
>> g = [1 2 3 4]; h = [4 3 2 1];
>> s1 = g + h, s2 = g.*h, s3 = g.^h, s4 = g.^2, s5 = 2.^h
```

## 1.5 矩阵的运算

运算符：+（加）、-（减）、\*（乘）、/（右除）、\（左除）、^（乘方）、'（转置）等；

常用函数：`det`（行列式）、`inv`（逆矩阵）、`rank`（秩）、`eig`（特征值、特征向量）、`rref`（化矩阵为行最简形）

例 5:

```
>> A=[2 0 -1;1 3 2]; B=[1 7 -1;4 2 3;2 0 1];
>> M = A*B           % 矩阵 A 与 B 按矩阵运算相乘
>> det_B = det(B)     % 矩阵 A 的行列式
>> rank_A = rank (A)  % 矩阵 A 的秩
>> inv_B = inv (B)    % 矩阵 B 的逆矩阵
>> [V,D] = eig(B)     % 矩阵 B 的特征值矩阵 V 与特征向量构成的矩阵 D
>> X = A/B             % A/B = A*B-1, 即XB=A, 求X
>> Y = B\A            % B\A = B-1*A, 即BY=A, 求Y
```

### 上机练习 (一):

1. 练习数据和符号的输入方式, 将前面的命令在命令窗口中执行通过;
2. 输入  $A=[7 \ 1 \ 5; 2 \ 5 \ 6; 3 \ 1 \ 5]$ ,  $B=[1 \ 1 \ 1; 2 \ 2 \ 2; 3 \ 3 \ 3]$ , 在命令窗口中执行下列表达式, 掌握其含义:  
 $A(2, 3)$        $A(:,2)$        $A(3,:)$        $A(:,1:2:3)$        $A(:,3).*B(:,2)$   
 $A(:,3).*B(2,:)$        $A*B$        $A.*B$        $A^{\wedge}2$        $A.^2$        $B/A$        $B./A$
3. 输入  $C=1:2:20$ , 则  $C(i)$  表示什么? 其中  $i=1,2,3,\dots,10$ ;
4. 查找已创建变量的信息, 删除无用的变量;
5. 欲通过系统做一平面图, 请查找相关的命令与函数, 获取函数的帮助信息。

## 二、编程

### 2.1 无条件循环

当需要无条件重复执行某些命令时, 可以使用 for 循环:

```
for 循环变量 t=表达式 1: 达式 2: 表达式 3
    语句体
end
```

**说明:** 表达式 1 为循环初值, 表达式 2 为步长, 表达式 3 为循环终值; 当表达式 2 省略时则默认步长为 1; for 语句允许嵌套。

例 6:

生成  $3 \times 4$  阶的 Hiltber 矩阵。

```
for i=1:3
```

```
    for j=1:4
```

```
        H (i, j) =1/ (i+j-1);
```

```
    end
```

```
end
```

如: 矩阵输入程序

```
m=input('矩阵行数: m=');
```

```
n= input('矩阵列数: n=');
```

```
for i=1:m
```

```
    for j=1:n
```

```
        disp(['输入第',num2str(i),'行, 第', num2str(j),'列元素'])
```

```
        A(i, j) = input('  ')
```

```
    end    end
```

## 2.2 条件循环

### 1) if-else-then 语句

if-else-then 语句的常使用三种形式为：

(1) if 逻辑表达式 语句体 end	(3) if 逻辑表达式 1 语句体 1 elseif 逻辑表达式 2 语句体 2 elseif 逻辑表达式 3 ... else 语句体 n end
(2) if 逻辑表达式 1 语句体 1 else 语句体 2 end	

### 2) while 循环语句

while 循环的一般使用形式为：

```
while 表达式
    语句体
end
```

例 7：

用二分法计算多项式方程  $x^3 - 2x - 5 = 0$  在  $[0, 3]$  内的一个根。

解：

```
a = 0; fa = -inf;
b = 3; fb = inf;
while b-a > eps*b
    x = (a+b) / 2;
    fx = x^3-2*x-5;
    if sign(fx)== sign(fa)
        a = x; fa = fx;
    else
        b = x; fb = fx;
    end
end
x
```

运行结果为：x = 2.0945515148154233

## 2.3 分支结构

若需要对不同的情形执行不同的操作，可用 switch 分支语句：

switch 表达式（标量或字符串）

```

case 值 1
    语句体 1
case 值 2
    语句体 2
...
otherwise
    语句体 n
end

```

说明：当表达式不是“case”所列值时，执行 otherwise 语句体。

## 2.4 建立 M 文件

将多个可执行的系统命令，用文本编辑器编辑后并存放在后缀为 .m 的文件中，若在 MATLAB 命令窗口中输入该 m-文件的文件名（不跟后缀.m!），即可依次执行该文件中的多个命令。这个后缀为.m 的文件，也称为 Matlab 的脚本文件(Script File)。

**注意：文件存放路径必须在 Matlab 能搜索的范围内。**

## 2.5 建立函数文件

对于一些特殊用户函数，系统提供了一个用于创建用户函数的命令 function，以备用户随时调用。

1. 格式：

```

function [输出变量列表]=fun_name(输入变量列表)
    用户自定义的函数体

```

2. 函数文件名为：fun\_name，注意：保存时文件名与函数名最好相同；

3. 存储路径：最好在系统的搜索路径上。

4. 调用方法：输出参量=fun\_name (输入变量)

例 8：

计算  $s = n!$ ，在文本编辑器中输入：

```

function s=pp(n);
    s=1;
    for i=1:n
        s=s*i;
    end
    s;

```

在 MATLAB 命令窗口中输入：s=pp(5)

结果为 s = 120

**上机练习（二）：**

1. 编写程序，计算  $1+3+5+7+\cdots+(2n+1)$  的值（用 input 语句输入 n 值）。

2. 编写分段函数  $f(x) = \begin{cases} x & 0 \leq x < 1 \\ 2-x & 1 \leq x \leq 2 \\ 0 & \text{其它} \end{cases}$  的函数文件，存放于文件 ff.m 中，计算出

$f(-3)$ ， $f(\sqrt{2})$ ， $f(\infty)$  的值。

## 三、矩阵及其运算

### 3.1 矩阵的创建

#### 1. 加、减运算

运算符：“+”和“-”分别为加、减运算符。

运算规则：对应元素相加、减，即按线性代数中矩阵的“+”、“-”运算进行。

例 3-1 在 Matlab 编辑器中建立 m 文件：LX0701.m

```
A=[1, 1, 1; 1, 2, 3; 1, 3, 6]
```

```
B=[8, 1, 6; 3, 5, 7; 4, 9, 2]
```

```
A+B=A+B
```

```
A-B=A-B
```

在 Matlab 命令窗口建入 LX0701，则

结果显示：A+B=

```
9    2    7
```

```
4    7   10
```

```
5   12    8
```

A-B=

```
-7    0   -5
```

```
-2   -3   -4
```

```
-3   -6    4
```

#### 2. 乘法

运算符：\*

运算规则：按线性代数中矩阵乘法运算进行，即放在前面的矩阵的各行元素，分别与放在后面的矩阵的各列元素对应相乘并相加。

(1) 两个矩阵相乘

例 3-2 在 Matlab 编辑器中建立 M 文件：LX0702.m

```
X= [2  3  4  5
```

```
1  2  2  1];
```



```
Y=[0 1 1
    1 1 0
    0 0 1
    1 0 0];
```

```
Z=X*Y
```

存盘

在命令行中建入 LX0702，回车后显示：

```
Z=
    8    5    6
    3    3    3
```

(2) 矩阵的数乘：数乘矩阵

上例中：a=2\*X

则显示：a =

```
    4    6    8   10
    2    4    4    2
```

(3) 向量的点乘（内积）：维数相同的两个向量的点乘。

**命令：dot** 向量点乘函数

例：X=[-1 0 2];

```
Y=[-2 -1 1];
```

```
Z=dot(X,Y)
```

则显示：Z =

```
4
```

还可用另一种算法：

```
sum(X.*Y)
```

```
ans=
```

```
4
```

(4) 向量叉乘

在数学上，两向量的叉乘是一个过两相交向量的交点且垂直于两向量所在平面的向量。

在 Matlab 中，用函数 cross 实现。

**命令 cross** 向量叉乘函数

例 3-3 计算垂直于向量(1, 2, 3)和(4, 5, 6)的向量。

在 Matlab 编辑器中建立 M 文件：LX0703.m

```
a=[1 2 3];
```

```
b=[4 5 6];
```

```
c=cross(a,b)
```

结果显示：

```
c=
   -3    6   -3
```

可得垂直于向量(1, 2, 3)和(4, 5, 6)的向量为±(-3, 6, -3)

(5) 混合积

混合积由以上两函数实现:

例 3-4 计算向量  $a=(1, 2, 3)$ 、 $b=(4, 5, 6)$ 和  $c=(-3, 6, -3)$  的混合积  $a \cdot (b \times c)$

在 Matlab 编辑器中建立 M 文件: LX0704.m

```
a=[1 2 3]; b=[4 5 6]; c=[-3 6 -3];
```

```
x=dot(a, cross(b, c))
```

结果显示:  $x =$

54

**注意:** 先叉乘后点乘, 顺序不可颠倒。

### 3.矩阵的除法

Matlab 提供了两种除法运算: 左除 ( $\backslash$ ) 和右除 ( $/$ )。一般情况下,  $x=a\backslash b$  是方程  $a*x=b$  的解, 而  $x=b/a$  是方程  $x*a=b$  的解

例:  $a=[1 \ 2 \ 3; 4 \ 2 \ 6; 7 \ 4 \ 9]$

```
b=[4; 1; 2];
```

```
x=a\b
```

则显示:  $x=$

-1.5000

2.0000

0.5000

如果  $a$  为非奇异矩阵, 则  $a\backslash b$  和  $b/a$  可通过  $a$  的逆矩阵与  $b$  阵得到:

$$a\backslash b = \text{inv}(a)*b$$
$$b/a = b*\text{inv}(a)$$

### 4.矩阵乘方

运算符:  $\wedge$

运算规则:

(1) 当  $A$  为方阵,  $p$  为大于 0 的整数时,  $A^p$  表示  $A$  的  $p$  次方, 即  $A$  自乘  $p$  次;  $p$  为小于 0 的整数时,  $A^p$  表示  $A^{-1}$  的  $p$  次方。

(2) 当  $A$  为方阵,  $p$  为非整数时, 则  $A^p = V \begin{bmatrix} d_{11}^p & & \\ & \ddots & \\ & & d_{nn}^p \end{bmatrix} V^{-1}$  其中  $V$  为  $A$  的特征向

量,  $\begin{bmatrix} d_{11} & & \\ & \ddots & \\ & & d_{nn} \end{bmatrix}$  为特征值矩阵

### 5.矩阵的转置

运算符:  $'$

运算规则: 与线性代数中矩阵的转置相同。

## 6. 矩阵的逆矩阵

例 3-5 求  $A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 2 & 1 \\ 3 & 4 & 3 \end{pmatrix}$  的逆矩阵

方法一：在 Matlab 编辑器中建立 M 文件：LX07051.m

```
A=[1 2 3; 2 2 1; 3 4 3];
```

```
inv(A)或 A^(-1)
```

则结果显示为

```
ans =  
    1.0000    3.0000   -2.0000  
   -1.5000   -3.0000    2.5000  
    1.0000    1.0000   -1.0000
```

方法二：由增广矩阵  $B = \begin{pmatrix} 1 & 2 & 3 & 1 & 0 & 0 \\ 2 & 2 & 1 & 0 & 1 & 0 \\ 3 & 4 & 3 & 0 & 0 & 1 \end{pmatrix}$  进行初等行变换

在 Matlab 编辑器中建立 M 文件：LX07052.m

```
B=[1, 2, 3, 1, 0, 0; 2, 2, 1, 0, 1, 0; 3, 4, 3, 0, 0, 1];
```

```
C=rref(B) %化行最简形
```

```
X=C(:, 4:6)
```

在 Matlab 命令窗口建入 LX07052，则显示结果如下：

```
C =  
    1.0000         0         0    1.0000    3.0000   -2.0000  
         0    1.0000         0   -1.5000   -3.0000    2.5000  
         0         0    1.0000    1.0000    1.0000   -1.0000
```

```
X =  
    1.0000    3.0000   -2.0000  
   -1.5000   -3.0000    2.5000  
    1.0000    1.0000   -1.0000
```

这就是 A 的逆矩阵。

## 7. 方阵的行列式

**命令：det** 计算行列式的值

例 3-6 计算上例中 A 的行列式的值

在 Matlab 编辑器中建立 M 文件：LX0706.m

```
A=[1 2 3; 2 2 1; 3 4 3];
```

```
D=det(A)
```

则结果显示为

```
D =  
    2
```

## 3.2 符号矩阵的运算

### 1. 符号矩阵的四则运算

Matlab 5.x 抛弃了在 4.2 版中为符号矩阵设计的复杂函数形式,把符号矩阵的四则运算简化为与数值矩阵完全相同的运算方式,其运算符为:加(+),减(-)、乘(×)、除(/、\ )等或:符号矩阵的和(symadd),差(symsub),乘(symmul)。

例 3-7  $A = \text{sym}('1/x, 1/(x+1); 1/(x+2), 1/(x+3)')$ ;

$B = \text{sym}('x, 1; x+2, 0')$ ;

$C=B-A$

$D=a\b b$

则显示:

$C=$

$$\begin{array}{cc} x-1/x & 1-1/(x+1) \\ x+2-1/(x+2) & -1/(x+3) \end{array}$$

$D=$

$$\begin{array}{cc} -6*x-2*x^3-7*x^2 & 1/2*x^3+x+3/2*x^2 \\ 6+2*x^3+10*x^2+14*x & -2*x^2-3/2*x-1/2*x^3 \end{array}$$

### 2. 其他基本运算

符号矩阵的其他一些基本运算包括转置(')、行列式(det)、逆(inv)、秩(rank)、幂(^)和指数(exp 和 expm)等都与数值矩阵相同

### 3. 符号矩阵的简化

符号工具箱中提供了符号矩阵因式分解、展开、合并、简化及通分等符号操作函数。

#### (1) 因式分解

**命令: factor** 符号表达式因式分解函数

格式: factor(s)

**说明:** s 为符号矩阵或符号表达式。常用于多项式的因式分解

例 3-8 将  $x^9-1$  分解因式

在 Matlab 命令窗口建入

`syms x`

`factor(x^9-1)`

则显示: ans =

$$(x-1)*(x^2+x+1)*(x^6+x^3+1)$$

例 3-9 问入取何值时, 齐次方程组

$$\begin{cases} (1-\lambda)x_1 - 2x_2 + 4x_3 = 0 \\ 2x_1 + (3-\lambda)x_2 + x_3 = 0 \\ x_1 + x_2 + (1-\lambda)x_3 = 0 \end{cases}$$

有非 0 解

解：在 Matlab 编辑器中建立 M 文件：LX0709.m

```
syms k
A=[1-k -2 4;2 3-k 1;1 1 1-k];
D=det(A)
factor(D)
```

其结果显示如下：

```
D =
-6*k+5*k^2-k^3
ans =
-k*(k-2)*(-3+k)
```

从而得到：当  $k=0$ 、 $k=2$  或  $k=3$  时，原方程组有非 0 解。

(2) 符号矩阵的展开

**命令** **expand** 符号表达式展开函数

格式：expand(s)

**说明：**s 为符号矩阵或表达式。常用在多项式的因式分解中，也常用于三角函数，指数函数和对数函数的展开中

例 3-10 将  $(x+1)^3$ 、 $\sin(x+y)$  展开

在 Matlab 编辑器中建立 M 文件：LX0710.m

```
syms x y
p=expand((x+1)^3)
q=expand(sin(x+y))
则结果显示为
p =
x^3+3*x^2+3*x+1
q =
sin(x)*cos(y)+cos(x)*sin(y)
```

(3) 同类式合并

**命令：**Collect 合并系数函数

格式：Collect(s,v) 将 s 中的变量 v 的同幂项系数合并。

Collect(s) s — 矩阵或表达式，此命令对由命令 findsym 函数返回的默认变量进行同类项合并。

(4) 符号简化

**命令：**simple 或 simplify 寻找符号矩阵或符号表达式的最简型

格式：Simple(s) s — 矩阵或表达式

**说明：**Simple(s) 将表达式 s 的长度化到最短。若还想让表达式更加精美，可使用函数 Pretty。

格式：Pretty(s) 使表达式 s 更加精美

例 3-11 计算行列式

$$\begin{vmatrix} 1 & 1 & 1 & 1 \\ a & b & c & d \\ a^2 & b^2 & c^2 & d^2 \\ a^4 & b^4 & c^4 & d^4 \end{vmatrix}$$

的值。

在 Matlab 编辑器中建立 M 文件: LX0711.m

```
syms a b c d
```

```
A=[1 1 1 1;a b c d;a^2 b^2 c^2 d^2;a^4 b^4 c^4 d^4];
```

```
d1=det(A)
```

```
d2=simple(d1) %化简表达式 d1
```

```
pretty(d2) %让表达式 d2 符合人们的书写习惯
```

则显示结果如下:

```
d1 =
```

```
b*c^2*d^4-b*d^2*c^4-b^2*c*d^4+b^2*d*c^4+b^4*c*d^2-b^4*d*c^2-a*c^2*d^4+a*d^2*c^4+a*b^2*d^4-a*b^2*c^4-a*b^4*d^2+a*b^4*c^2+a^2*c*d^4-a^2*d*c^4-a^2*b*d^4+a^2*b*c^4+a^2*b^4*d-a^2*b^4*c-a^4*c*d^2+a^4*d*c^2+a^4*b*d^2-a^4*b*c^2-a^4*b^2*d+a^4*b^2*c
```

```
d2 =
```

```
(-d+c)*(b-d)*(b-c)*(-d+a)*(a-c)*(a-b)*(a+c+d+b)
```

```
(-d+c)(b-d)(b-c)(-d+a)(a-c)(a-b)(a+c+d+b)
```

例 3-12 设  $A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 2 & 1 \\ 3 & 4 & 3 \end{pmatrix}$ ,  $B = \begin{pmatrix} 2 & 1 \\ 5 & 3 \end{pmatrix}$ ,  $C = \begin{pmatrix} 1 & 3 \\ 2 & 0 \\ 3 & 1 \end{pmatrix}$

求矩阵 X, 使满足:  $AXB = C$

在 Matlab 编辑器中建立 M 文件: LX0712.m

```
A=[1 2 3;2 2 1;3 4 3];
```

```
B=[2,1;5 3];
```

```
C=[1 3;2 0;3 1];
```

```
X=A\C/B
```

则结果显示如下:

```
X =
```

```
-2.0000 1.0000
```

```
10.0000 -4.0000
```

```
-10.0000 4.0000
```

例 3-13 计算  $\begin{pmatrix} \cos(t) & -\sin(t) \\ \sin(t) & \cos(t) \end{pmatrix}^5$

在 Matlab 编辑器中建立 M 文件: LX0713.m

```
syms t
```

```
A=[cos(t) -sin(t); sin(t), cos(t)];
```

```
B=sympow(A, 5) %计算 A 的 5 次幂
```

```

C=simple(B)          %化简
pretty(C)
则显示结果如下
B =
[ cos(t)*(cos(t)*(cos(t)*(cos(t)^2-sin(t)^2)-2*sin(t)^2*cos(t))-sin(t)*(sin(t)*(cos(t)^2-sin(t)^2)+2*cos(t)^2*sin(t)))-sin(t)*(sin(t)*(cos(t)*(cos(t)^2-sin(t)^2)-2*sin(t)^2*cos(t))+cos(t)*(sin(t)*(cos(t)^2-sin(t)^2)+2*cos(t)^2*sin(t))),
cos(t)*(cos(t)*(-2*cos(t)^2*sin(t)-sin(t)*(cos(t)^2-sin(t)^2))-sin(t)*(cos(t)*(cos(t)^2-sin(t)^2)-2*sin(t)^2*cos(t)))-sin(t)*(sin(t)*(-2*cos(t)^2*sin(t)-sin(t)*(cos(t)^2-sin(t)^2))+cos(t)*(cos(t)*(cos(t)^2-sin(t)^2)-2*sin(t)^2*cos(t)))]
[ sin(t)*(cos(t)*(cos(t)*(cos(t)^2-sin(t)^2)-2*sin(t)^2*cos(t))-sin(t)*(sin(t)*(cos(t)^2-sin(t)^2)+2*cos(t)^2*sin(t)))+cos(t)*(sin(t)*(cos(t)*(cos(t)^2-sin(t)^2)-2*sin(t)^2*cos(t))+cos(t)*(sin(t)*(cos(t)^2-sin(t)^2)+2*cos(t)^2*sin(t))),
sin(t)*(cos(t)*(-2*cos(t)^2*sin(t)-sin(t)*(cos(t)^2-sin(t)^2))-sin(t)*(cos(t)*(cos(t)^2-sin(t)^2)-2*sin(t)^2*cos(t)))+cos(t)*(sin(t)*(-2*cos(t)^2*sin(t)-sin(t)*(cos(t)^2-sin(t)^2))+cos(t)*(cos(t)*(cos(t)^2-sin(t)^2)-2*sin(t)^2*cos(t)))]
C =
[ cos(5*t), -sin(5*t)]
[ sin(5*t), cos(5*t)]

[cos(5 t)    -sin(5 t)]
[
[ sin(5 t)    cos(5 t) ]

```

## 四、秩与线性相关性

### 4.1 矩阵和向量组的秩以及向量组的线性相关性。

矩阵 A 的秩是矩阵 A 中最高阶非零子式的阶数；向量组的秩通常由该向量组构成的矩阵来计算。

**命令：rank**

格式：rank(A) A 为矩阵式向量组构成的矩阵

例 4-1 求向量组  $\alpha_1 = (1 \ -2 \ 2 \ 3)$ ， $\alpha_2 = (-2 \ 4 \ -1 \ 3)$ ， $\alpha_3 = (-1 \ 2 \ 0 \ 3)$ ，

$\alpha_4 = (0 \ 6 \ 2 \ 3)$ ， $\alpha_5 = (2 \ -6 \ 3 \ 4)$  的秩，并判断其线性相关性。

在 Matlab 编辑器中建立 M 文件：LX0714.m

```
A=[1 -2 2 3;-2 4 -1 3;-1 2 0 3;0 6 2 3;2 -6 3 4];
```

```
B=rank(A)
```

运行后结果如下：

```
B =
```

由于秩为  $3 <$  向量个数, 因此向量组线性相关。

## 4.2 向量组的最大无关组

矩阵的初等行变换有三条:

1. 交换两行  $r_i \leftrightarrow r_j$  (第  $i$ 、第  $j$  两行交换)

2. 第  $i$  行的  $K$  倍  $kr_i$

3. 第  $i$  行的  $K$  倍加到第  $j$  行上去  $r_j + kr_i$

通过这三条变换可以将矩阵化成行最简形, 从而找出列向量组的一个最大无关组, Matlab 将矩阵化成行最简形的命令是:

**命令: rref**

格式: `rref(A)`  $A$  为矩阵

例 4-2 求向量组  $a_1=(1,-2,2,3)$ ,  $a_2=(-2,4,-1,3)$ ,  $a_3=(-1,2,0,3)$ ,  $a_4=(0,6,2,3)$ ,  $a_5=(2,-6,3,4)$  的一个最大无关组。

在 Matlab 编辑器中建立 M 文件: LX0715.m

```
a1=[1 -2 2 3]';
```

```
a2=[-2 4 -1 3]';
```

```
a3=[-1 2 0 3]';
```

```
a4=[0 6 2 3]';
```

```
a5=[2 -6 3 4]';
```

```
A=[a1 a2 a3 a4 a5]
```

```
format rat %以有理格式输出
```

```
B=rref(A) %求 A 的行最简形
```

运行后的结果为

$A =$

1	-2	-1	0	2
-2	4	2	6	-6
2	-1	0	2	3
3	3	3	3	4

$B =$

1	0	1/3	0	16/9
0	1	2/3	0	-1/9
0	0	0	1	-1/3
0	0	0	0	0

从  $B$  中可以得到: 向量  $a_1$   $a_2$   $a_4$  为其中一个最大无关组



## 五、线性方程的组的求解

我们将线性方程的求解分为两类：一类是方程组求唯一解或求特解，另一类是方程组求无穷解即通解。可以通过系数矩阵的秩来判断：

若系数矩阵的秩  $r=n$  ( $n$  为方程组中未知变量的个数)，则有唯一解

若系数矩阵的秩  $r<n$ ，则可能有无穷解。

线性方程组的无穷解 = 对应齐次方程组的通解+非齐次方程组的一个特解；其特解的求法属于解的第一类问题，通解部分属第二类问题。

### 5.1 求线性方程组的唯一解或特解（第一类问题）

这类问题的求法分为两类：一类主要用于解低阶稠密矩阵 —— 直接法；另一类是解大型稀疏矩阵 —— 迭代法。

#### 5.1.1 利用矩阵除法求线性方程组的特解（或一个解）

方程： $AX=b$

解法： $X=A\backslash b$

例 5-1 求方程组 
$$\begin{cases} 5x_1 + 6x_2 & = 1 \\ x_1 + 5x_2 + 6x_3 & = 0 \\ x_2 + 5x_3 + 6x_4 & = 0 \\ x_3 + 5x_4 + 6x_5 & = 0 \\ x_4 + 5x_5 & = 1 \end{cases} \text{ 的解}$$

解：在 Matlab 编辑器中建立 M 文件：LX0716.m

```
A=[5 6 0 0 0
    1 5 6 0 0
    0 1 5 6 0
    0 0 1 5 6
    0 0 0 1 5];
```

```
B=[1 0 0 0 1]';
```

```
R_A=rank(A) %求秩
```

```
X=A\B %求解
```

运行后结果如下

```
R_A =
```

```
5
```

```
X =
```

```
2.2662
```

```
-1.7218
```

```
1.0571
```

```
-0.5940
```

```
0.3188
```

这就是方程组的解。

例 5-2 求方程组

$$\begin{cases} x_1 + x_2 - 3x_3 - x_4 = 1 \\ 3x_1 - x_2 - 3x_3 + 4x_4 = 4 \\ x_1 + 5x_2 - 9x_3 - 8x_4 = 0 \end{cases}$$

的一个特解

解：在 Matlab 编辑器中建立 M 文件：LX0717.m

```
A=[1 1 -3 -1;3 -1 -3 4;1 5 -9 -8];
B=[1 4 0]';
X=A\B
X =
    0
    0
 -0.5333
 0.6000
```

### 5.1.2 利用矩阵的 LU、QR 和 cholesky 分解求方程组的解

1.LU 分解：

LU 分解又称 Gauss 消去分解，可把任意方阵分解为下三角矩阵的基本变换形式（行交换）和上三角矩阵的乘积。即  $A=LU$ ， $L$  为下三角阵， $U$  为上三角阵。

则： $A*X=b$  变成  $L*U*X=b$

$\therefore X=U\backslash(L\backslash b)$  这样可以大提高运算速度。

命令  $[L, U]=lu(A)$

例 5-3 求方程组

$$\begin{cases} 4x_1 + 2x_2 - x_3 = 2 \\ 3x_1 - x_2 + 2x_3 = 10 \\ 11x_1 + 3x_2 = 8 \end{cases}$$

的一个特解

解

$$A = \begin{pmatrix} 4 & 2 & -1 \\ 3 & -1 & 2 \\ 11 & 3 & 0 \end{pmatrix} \quad b = [2, 10, 8]'$$

在 Matlab 编辑器中建立 M 文件：LX0718.m

```
A=[4 2 -1;3 -1 2;11 3 0];
B=[2 10 8]';
D=det(A)
[L,U]=lu(A)
X=U\ (L\B)
```

在 Matlab 命令窗口建入 LX0718，回车后显示结果如下：

D =

```

0
L =
    0.3636   -0.5000    1.0000
    0.2727    1.0000         0
    1.0000         0         0
U =
   11.0000    3.0000         0
         0   -1.8182    2.0000
         0         0    0.0000

```

Warning: Matrix is close to singular or badly scaled.

Results may be inaccurate. RCOND = 2.018587e-017.

> In D:\Matlab\pujun\lx0720.m at line 4

```

X =
   1.0e+016 *
   -0.4053
    1.4862
    1.3511

```

**说明：**结果中的警告是由于系数行列式为零产生的。可以通过  $A \cdot X$  验证其正确性。

## 2. Cholesky 分解

若  $A$  为对称正定矩阵，则 Cholesky 分解可将矩阵  $A$  分解成上三角矩阵和其转置的乘积，即： $A = R' \cdot R$  其中  $R$  为上三角阵。

方程  $A \cdot X = b$  变成  $R' \cdot R \cdot X = b$

所以 
$$X = R \setminus (R' \setminus b)$$

**命令**  $R = \text{chol}(A)$

## 3. QR 分解

对于任何长方矩阵  $A$ ，都可以进行 QR 分解，其中  $Q$  为正交矩阵， $R$  为上三角矩阵的初等变换形式，即： $A = QR$

方程  $A \cdot X = b$  变形成  $QRX = b$

所以 
$$X = R \setminus (Q \setminus b)$$

**命令**  $[Q, R] = \text{qr}(A)$

上例中  $[Q, R] = \text{qr}(A)$   
 $X = R \setminus (Q \setminus B)$

**说明：**这三种分解，在求解大型方程组时很有用。其优点是运算速度快、可以节省磁盘空间、节省内存。

## 5.2 求线性齐次方程组的通解

在 Matlab 中，函数 `null` 用来求解零空间，即满足  $A \cdot X = 0$  的解空间，实际上是求出解

空间的一组基（基础解系）。

格式：`z = null`                      %  $z$  的列向量为方程组的正交规范基，满足  $Z' \times Z = I$

`z = null(A,'r')`            %  $z$  的列向量是方程  $AX=0$  的有理基

例 5-4 求解方程组的通解：

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

解：在 Matlab 编辑器中建立 M 文件：LX0719.m

```
A=[1 2 2 1;2 1 -2 -2;1 -1 -4 -3];
```

```
format rat            %指定有理式格式输出
```

```
B=null(A,'r')        %求解空间的有理基
```

运行后显示结果如下：

```
B =  
      2      5/3  
     -2     -4/3  
      1      0  
      0      1
```

写出通解：

```
syms k1 k2
```

```
X=k1*B(:,1)+k2*B(:,2)            %写出方程组的通解
```

```
pretty(X)            %让通解表达式更加精美
```

运行后结果如下：

```
X =  
      [ 2*k1+5/3*k2]  
      [-2*k1-4/3*k2]  
      [          k1]  
      [          k2]  
      % 下面是其简化形式  
      [2k1 + 5/3k2 ]  
      [          ]  
      [-2k1 - 4/3k2]  
      [          ]  
      [      k1      ]  
      [          ]  
      [      k2      ]
```

### 5.3 求非齐次线性方程组的通解

非齐次线性方程组需要先判断方程组是否有解，若有解，再去求通解。因此，步骤为：第一步：判断  $AX=b$  是否有解，若有解则进行第二步

第二步：求  $AX=b$  的一个特解

第三步：求  $AX=0$  的通解

第四步： $AX=b$  的通解= $AX=0$  的通解+ $AX=b$  的一个特解。

例 5-5 求解方程组

$$\begin{cases} x_1 - 2x_2 + 3x_3 - x_4 = 1 \\ 3x_1 - x_2 + 5x_3 - 3x_4 = 2 \\ 2x_1 + x_2 + 2x_3 - 2x_4 = 3 \end{cases}$$

解：在 Matlab 编辑器中建立 M 文件：LX0720.m

```
A=[1 -2 3 -1;3 -1 5 -3;2 1 2 -2];
```

```
b=[1 2 3]';
```

```
B=[A b];
```

```
n=4;
```

```
R_A=rank(A)
```

```
R_B=rank(B)
```

```
format rat
```

```
if R_A==R_B&R_A==n %判断有唯一解
```

```
    X=A\b
```

```
elseif R_A==R_B&R_A<n %判断有无穷解
```

```
    X=A\b %求特解
```

```
    C=null(A,'r') %求  $AX=0$  的基础解系
```

```
else X='equition no solve' %判断无解
```

```
end
```

运行后结果显示：

```
R_A =
```

```
2
```

```
R_B =
```

```
3
```

```
X =
```

```
equition no solve
```

说明该方程组无解

例 5-6 求解方程组的通解：

$$\begin{cases} x_1 + x_2 - 3x_3 - x_4 = 1 \\ 3x_1 - x_2 - 3x_3 + 4x_4 = 4 \\ x_1 + 5x_2 - 9x_3 - 8x_4 = 0 \end{cases}$$

解法一：在 Matlab 编辑器中建立 M 文件：LX07211.m

```
A=[1 1 -3 -1;3 -1 -3 4;1 5 -9 -8];
```

```
b=[1 4 0]';
```

```
B=[A b];
```

```
n=4;
```

```
R_A=rank(A)
```

```

R_B=rank(B)
format rat
if R_A==R_B&R_A==n
    X=A\b
elseif R_A==R_B&R_A<n
    X=A\b
    C=null(A,'r')
else X='Equation has no solves'
end

```

运行后结果显示为：

R\_A =

2

R\_B =

2

Warning: Rank deficient, rank = 2 tol = 8.8373e-015.

> In D:\Matlab\pujun\lx0723.m at line 11

X =

0

0

-8/15

3/5

C =

3/2            -3/4

3/2            7/4

1              0

0              1

所以原方程组的通解为

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

解法二：在 Matlab 编辑器中建立 M 文件：LX07212.m

```
A=[1 1 -3 -1;3 -1 -3 4;1 5 -9 -8];
```

```
b=[1 4 0]';
```

```
B=[A b];
```

```
C=rref(B) %求增广矩阵的行最简形，可得最简同解方程组。
```

运行后结果显示为：

C =

1            0            -3/2            3/4            5/4

0            1            -3/2            -7/4            -1/4

0            0            0            0            0  
 对应齐次方程组的基础解系为：

$$\xi_1 = \begin{pmatrix} 3/2 \\ 3/2 \\ 1 \\ 0 \end{pmatrix}, \quad \xi_2 = \begin{pmatrix} -3/4 \\ 7/4 \\ 0 \\ 1 \end{pmatrix}$$

非齐次方程组的特解为：

$$\eta^* = \begin{pmatrix} 5/4 \\ -1/4 \\ 0 \\ 0 \end{pmatrix}$$

所以，原方程组的通解为：

$$X = k_1 \xi_1 + k_2 \xi_2 + \eta^*$$

## 六、特征值与二次型

工程技术中的一些问题，如振动问题和稳定性问题，常归结为求一个方阵的特征值和特征向量。

### 6.1 方阵的特征值与特征向量

设  $A$  为  $n$  阶方阵，如果数  $\lambda$  和  $n$  维列向量  $x$  使得关系式

$$Ax = \lambda x$$

成立，则称  $\lambda$  为方阵  $A$  的特征值，非零向量  $x$  称为  $A$  对应于特征值  $\lambda$  的特征向量。

在 Matlab 中，用如下几种调用格式来求  $A$  的特征值和特征向量。

1. **d=eig(A)**                      %  $d$  为矩阵  $A$  的特征值排成的向量

2. **[V,D]=eig(A)**                %  $D$  为  $A$  的特征值对角阵， $V$  的列向量为对应特征值的特征向量（且为单位向量）

3. **[V,D]=eig(A, 'nobalance')**    % 当  $A$  中有小到和截断误差相当的元素时，用 **nobalance** 选项，其作用是减少计算误差

例 6-1 求矩阵  $A = \begin{pmatrix} -2 & 1 & 1 \\ 0 & 2 & 0 \\ -4 & 1 & 3 \end{pmatrix}$  的特征值和特征向量

解：在 Matlab 编辑器中建立 M 文件：LX0722.m

**A=[-2 1 1;0 2 0;-4 1 3];**

**[V,D]=eig(A)**

运行后结果显示：

**V =**

**-0.7071    -0.2425    0.3015**

$$\begin{array}{ccc} 0 & 0 & 0.9045 \\ -0.7071 & -0.9701 & 0.3015 \end{array}$$

D =

$$\begin{array}{ccc} -1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{array}$$

即：特征值-1 对应特征向量 $(-0.7071 \ 0 \ -0.7071)^T$

特征值 2 对应特征向量 $(-0.2425 \ 0 \ -0.9701)^T$ 和 $(-0.3015 \ 0.9045 \ -0.3015)^T$

例 6-2 求矩阵  $A = \begin{pmatrix} -1 & 1 & 0 \\ -4 & 3 & 0 \\ 1 & 0 & 2 \end{pmatrix}$  的特征值和特征向量

解：在 Matlab 编辑器中建立 M 文件：LX0723.m

A=[-1 1 0;-4 3 0;1 0 2];

[V,D]=eig(A)

运行后结果显示为

V =

$$\begin{array}{ccc} 0 & 0.4082 & -0.4082 \\ 0 & 0.8165 & -0.8165 \\ 1.0000 & -0.4082 & 0.4082 \end{array}$$

D =

$$\begin{array}{ccc} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array}$$

**说明：**当特征值为 1（二重根）时，对应特征向量都是  $k(0.4082 \ 0.8165 \ -0.4082)^T$  为任意常数。

## 6.2 正交矩阵及二次型

A 为 n 阶方阵，且满足： $A'A=E$ （即  $A'=A^{-1}$ ），则 A 为正交矩阵

A 为正交矩阵  $\Leftrightarrow$  A 的各列(行)向量的长度为 1，而且 A 的各列(行)向量两两正交。

### 6.2.1 向量的长度（范数）

**命令 norm**

格式 norm(X) % 求 X 的范数

### 6.2.2 求矩阵的正交矩阵

**命令：orth**

格式：orth(A) %将矩阵 A 正交规范化



例如：求  $A = \begin{pmatrix} 4 & 0 & 0 \\ 0 & 3 & 1 \\ 0 & 1 & 3 \end{pmatrix}$  的正交矩阵

在 Matlab 命令窗口键入

```
A=[4 0 0;0 3 1;0 1 3];
```

```
P=orth(A)
```

```
Q=P'*P
```

则显示结果为

P =

```
1.0000    0    0
    0    0.7071 -0.7071
    0    0.7071  0.7071
```

Q =

```
1.0000    0    0
    0    1.0000    0
    0    0    1.0000
```

### 6.2.3 矩阵的 schur 分解

格式 [U,T]=schur(A) %U 为正交矩阵，使得  $A = U T U'$  和  $U' U = E$

T = schur(A) %生成 schur 矩阵 T。当 A 为实对称阵时，T 为特征值对角阵

例 6-3 设  $A = \begin{pmatrix} 4 & 0 & 0 \\ 0 & 3 & 1 \\ 0 & 1 & 3 \end{pmatrix}$ ，求一个正交矩阵 P，使  $P^{-1}AP = \Lambda$  为对角阵

解法 1：在 Matlab 编辑器中建立 M 文件：LX07241.m

```
A=[4 0 0;0 3 1;0 1 3];
```

```
[V,D]=eig(A)
```

运行后结果如下：

V =

```
1.0000    0    0
    0    0.7071  0.7071
    0    0.7071 -0.7071
```

D =

```
4    0    0
0    4    0
0    0    2
```

这里，V 就是所求的正交矩阵 P，D 就是对角矩阵  $\Lambda$

解法 2：在 Matlab 编辑器中建立 M 文件：LX07242.m

```
A=[4 0 0;0 3 1;0 1 3];
```

```
[U,T]=schur(A)
```

运行后结果显示如下:

U =  
1.0000 0 0  
0 0.7071 0.7071  
0 0.7071 -0.7071

T =  
4 0 0  
0 4 0  
0 0 2

这里, U 就是所求的正交矩阵 P, T 就是对角矩阵  $\Lambda$

**说明:** 对于实对称矩阵, 用 eig 和 schur 分解效果一样。

例 6-4 求一个正交变换  $X=PY$ , 把二次型

$$f = 2x_1x_2 + 2x_1x_3 - 2x_1x_4 - 2x_2x_3 + 2x_2x_4 + 2x_3x_4$$

化成标准形。

解: 先写出二次型的实对称矩阵

$$A = \begin{pmatrix} 0 & 1 & 1 & -1 \\ 1 & 0 & -1 & 1 \\ 1 & -1 & 0 & 1 \\ -1 & 1 & 1 & 0 \end{pmatrix}$$

在 Matlab 编辑器中建立 M 文件: LX0725.m

A=[0 1 1 -1;1 0 -1 1;1 -1 0 1;-1 1 1 0];

[P,D]=schur(A)

syms y1 y2 y3 y4

y=[y1;y2;y3;y4];

X=vpa(P,2)\*y %vpa 表示可变精度计算, 这里取 2 位精度

f=[y1 y2 y3 y4]\*D\*y

运行后结果显示如下:

P =  
780/989 780/3691 1/2 -390/1351  
780/3691 780/989 -1/2 390/1351  
780/1351 -780/1351 -1/2 390/1351  
0 0 1/2 1170/1351

D =  
1 0 0 0  
0 1 0 0  
0 0 -3 0  
0 0 0 1

X =  
[.79\*y1+.21\*y2+.50\*y3-.29\*y4]

$$\begin{aligned}
 & \begin{bmatrix} .21*y_1+.79*y_2-.50*y_3+.29*y_4 \\ .56*y_1-.56*y_2-.50*y_3+.29*y_4 \\ .50*y_3+.85*y_4 \end{bmatrix} \\
 f = & \\
 & y_1^2+y_2^2-3*y_3^2+y_4^2
 \end{aligned}$$

$$\text{即} \quad f = y_1^2 + y_2^2 - 3y_3^2 + y_4^2$$