实验四　线性代数实验

通过本实验,熟悉矩阵的各种运算、线性方程组的求解与初等变换、线性无关向量的正交规范化、确定线性相关向量组的一个极大线性无关向量组、矩阵的分解、矩阵特征值与特征向量的求解、Jordan 标准形及相似矩阵、二次型与标准型.

第一节　矩阵、向量及其运算

一、矩阵的建立与修改

在 MATLAB 中,最简单也是最直观的建立矩阵的方式,莫过于在 MATLAB 工作窗口中直接输入矩阵,但在大多数情况下,这并非最佳方法.我们可以利用不同方法建立矩阵,利用各种函数修改已存在的矩阵,以最快捷的方式建立和生成需要的矩阵.

1.由文件生成和保存矩阵

用文件方式来创建和保存的矩阵一般有三种:M 文件、文本文件和 MAT 文件.M 文件是由 MATLAB 软件来创建的,因为它是标准的 ASCⅡ文件,故亦可由其它文字编辑软件输入,但一般还是用 MATLAB 软件的“新建”要好些.如可建立如下 M 文件,文件名为 myfile.m ,内容为

```
A=[3  4  -1  1  -9  10;6  5  0  7  4  -6 ;1  -4  7  -16  -18;2
   -4  5  -6  12  -8;-3  6  -7  8  -1  1;8  -4  9  1  3  0]
```

再用写字板或记事本输入一文本文件,文件名为 txtfile.txt,内容如下

```
 1   2   4   6  -3   2
 7   9  16  -5   8  -7
 8  11  20   1   5   5
10  15  28  13  -1   9
12  19  36  25  -7  23
 2   4   6  -3   0   5
```

在 MATLAB 中执行如下命令

```
clear                  % 清除当前工作空间中的变量
```

```
myfile                %执行M文件
A =
      3     4    -1     1    -9    10
      6     5     0     7     4   -16
      1    -4     7    -1     6    -8
      2    -4     5    -6    12    -8
     -3     6    -7     8    -1     1
      8    -4     9     1     3     0
who                   %查看工作空间中的变量
Your variables are:
A
load txtfile.txt          %装入 txtfile.txt 文件
who
Your variables are:
A         txtfile
save matfile            %保存工作空间变量到 matfile.mat 文件中
clear
who
load matfile            %重新将 matfile.mat 中的变量装入工作空间
who
Your variables are:
A         txtfile
Txtfile                %显示变量 txtfile 的内容
txtfile =
     1    2    4     6    -3     2
     7    9   16    -5     8    -7
     8   11   20     1     5     5
    10   15   28    13    -1     9
    12   19   36    25    -7    23
     2    4    6    -3     0     5
```

由上述可知，数据既可以用M文件输入，亦可用文本文件输入，但用文本文件输入时所有数据均被放在以文件名作为变量名的变量中，用命令“LOAD 文件名”来装入工作空间.MAT 文件是在 MATLAB 中用“SAVE”命令创建的，目的是保存当前工作空间的变量以备后用，它也是用“LOAD”来调入使用的，与文本文件不同的是，MAT 文件是以二进制形式存贮的，且可存贮多个变量，亦可选择

某些变量进行存贮.

2. 由函数生成矩阵

(1) 生成单位阵函数

eye(n)	生成 n 阶单位阵
eye(m, n)或 eye([m, n])	生成 $m \times n$ 阶单位阵
eye(size($\boldsymbol{A}$))	生成与矩阵 $\boldsymbol{A}$ 大小相同的单位阵

但此函数不能生成高阶单位阵,否则会出错.

(2) 生成全 1 矩阵函数

ones(n)	生成 $n \times n$ 阶全 1 矩阵
ones(m, n)或 ones([m, n])	生成 $m \times n$ 阶全 1 矩阵
ones(n_1, n_2, n_3, $\cdots$)	生成 $n_1 \times n_2 \times n_3 \times \cdots$阶全 1 矩阵
ones(size($\boldsymbol{A}$))	生成与矩阵 $\boldsymbol{A}$ 大小相同的全 1 矩阵

(3) 生成全 0 阵函数

zeros(n)	生成 $n \times n$ 阶全 0 矩阵
zeros(m, n)或 zeros([m, n])	生成 $m \times n$ 阶全 0 矩阵
zeros (n_1, n_2, n_3, $\cdots$)	生成 $n_1 \times n_2 \times n_3 \times \cdots$阶全 0 矩阵
zeros(size($\boldsymbol{A}$))	生成与矩阵 $\boldsymbol{A}$ 大小相同的全 0 矩阵

(4) 生成随机阵函数

rand(n)	生成 $n \times n$ 阶随机阵
rand (n_1, n_2, n_3, $\cdots$)	生成 $n_1 \times n_2 \times n_3 \times \cdots$阶随机阵
rand (size($\boldsymbol{A}$))	生成与矩阵 $\boldsymbol{A}$ 大小相同的随机阵

函数 rand( )生成的是[0,1]区间上服从均匀分布的随机数,若要生成服从 N(0,1)正态分布的随机阵,可用函数 randn( )来生成.

(5) 生成魔方阵

魔方阵是指各行、各列及两对角线元素之和均相等的矩阵,用命令 magic(n)生成 $n \times n$ 阶魔方阵.

(6) 生成对角阵

用函数 diag(v, k)生成第 k 个对角线由向量$\boldsymbol{v}$ 组成的对角阵,k 可是正数、零或负数,当 k 为 0 时指主对角线,k 为负时$\boldsymbol{v}$ 在主对角线下,否则在主对角线上.

3. 符号矩阵的创建

符号矩阵用 sym 函数创建.如

```
exam = sym ('[ 1 , x/a , sin(x) ; y/x , 1 + 1/y , tan(x/y) ; 1 = 0 , 3 + 3 , 4 * r ]')
exam = [1          x/0        sin(x)   ]
       [y/x        1 + 1/y    tan(x/y)  ]
       [1 = 0      3 + 3      4 * r     ]
```

也可用类似创建普通数值矩阵的方法创建符号矩阵.如

```
syms x y z a b c
f = a * x^2 + b * x + c;
g = x * y * z;
h = (f + g) * b/a;
e1 = sym('a * x^2 + b * x + c = 0');
e2 = sym('x * y * z = 0');
e3 = sym('h = 0');
M = [1    2   3   x
     f    g   h   y
     e1   e2  e3  z]
```

结果显示为

```
M = [ 1,                    2,                               3,    x ]
  [ a*x^2+b*x+c, x*y*z,  (a*x^2+b*x+c+ x*y*z)*b/a,                 y]
  [ a*x^2+b*x+c=0,  x*y*z=0,              h=0,                     z ]
```

还可用 sym( ) 函数将数值矩阵转换为符号阵.值得注意的是,不管原来的数值矩阵是以分数形式给出,还是以小数形式给出,转换成符号阵后都将以最接近原数的精确有理数形式给出.

最后再介绍一种用矩阵元素通式来创建符号矩阵的方法,这对一些大型有规律的矩阵输入很有帮助.方法是先创建一个函数文件 symmat.m,其内容为

```
function M = symmat(row, col, f)
% SYMMAT 命令是利用通式来创建符号矩阵,其格式为
% symmat(row, col, f)
% 参数 row, col 分别为待创建符号矩阵的行数和列数
% 参数 f 为矩阵元素的通式
for R = 1:row
   for C = 1:col
      c = sym(C);
      r = sym(R);
      M(R, C) = subs(sym(f));
   end
end
```

然后由通式 a,b,c 建立 $\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}$ 三个符号矩阵

```
syms x  y  c  r
a = sin((c + (r - 1) * 3));
```

```
b = exp(r + (c - 1) * 4);
c = (c + (r - 1) * 3) * x + (r + (c - 1) * 4) * y;
A = symmat(3, 3, a)
A =
[  sin(1),  sin(2),  sin(3)]
[  sin(4),  sin(5),  sin(6)]
[  sin(7),  sin(8),  sin(9)]
B = symmat(4, 3, b)
B =
[  exp(1),  exp(5),  exp(9)]
[  exp(2),  exp(6),  exp(10)]
[  exp(3),  exp(7),  exp(11)]
[  exp(4),  exp(8),  exp(12)]
C = symmat(5, 5, c)
C =
[         x + y, 2 * x + 5 * y, 3 * x + 9 * y, 4 * x + 13 * y, 5 * x + 17 * y]
[  4 * x + 2 * y, 5 * x + 6 * y, 6 * x + 10 * y, 7 * x + 14 * y, 8 * x + 18 * y]
[  7 * x + 3 * y, 8 * x + 7 * y, 9 * x + 11 * y, 10 * x + 15 * y, 11 * x + 19 * y]
[ 10 * x + 4 * y, 11 * x + 8 * y, 12 * x + 12 * y, 13 * x + 16 * y, 14 * x + 20 * y]
[ 13 * x + 5 * y, 14 * x + 9 * y, 15 * x + 13 * y, 16 * x + 17 * y, 17 * x + 21 * y]
```

4. 矩阵的修改

(1) 矩阵的标识　矩阵的标识在对矩阵的单个或多个元素进行引用、赋值和修改中起着非常重要的作用.熟练掌握各种标识方式可方便灵活地对矩阵进行修改和引用.矩阵的标识主要有元素标识方式、向量标识方式、0 - 1 标识方式和与 find() 函数连用的矩阵标识方式.① 用 $\boldsymbol{A}(i\ ,\ j)$可标识矩阵 $\boldsymbol{A}$ 第 i 行 j 列的元素;② $A(\boldsymbol{vr},\boldsymbol{vc})$可标识矩阵 $\boldsymbol{A}$ 部分行与部分列的元素,其中 $\boldsymbol{vr}$ 和 $\boldsymbol{vc}$ 分别为含有矩阵 $\boldsymbol{A}$ 的行号和列号的单调向量,也可用“:”表示全部;③0 - 1 向量标识方式可用 $\boldsymbol{A}(\boldsymbol{v}_1,\boldsymbol{v}_2)$, $\boldsymbol{A}(:\ ,\boldsymbol{v}_2)$, $\boldsymbol{A}(\boldsymbol{v}_1,:)$标识,其中 $\boldsymbol{v}_1$ 与 $\boldsymbol{v}_2$ 是由 0 或 1 组成的长度分别为 A 的行维和列维的向量,其元素如果是 1,则表示取此相应位置上的行或列,是 0 则不取.必须注意, $\boldsymbol{v}_1$ 与 $\boldsymbol{v}_2$ 中的元素不能取除 0 和 1 以外的值;④用 $A(\text{find}(\boldsymbol{B}))$ 标识时,矩阵 $\boldsymbol{B}$ 与 $\boldsymbol{A}$ 必须是大小相同的矩阵.若 $\boldsymbol{B}$ 的元素非 0,则取当前位置 $\boldsymbol{A}$ 的元素,否则不取.

(2) 矩阵的修改　矩阵的修改包括对矩阵进行扩充、矩阵元素的部分删除、矩阵元素的抽取以及矩阵结构变化等操作.这些操作在矩阵的构造和生成中具

有非常重要的作用.灵活地运用这些操作可以非常方便地生成所需矩阵,特别是对元素较多的矩阵,可以避免冗长的矩阵输入,既节约时间,又不易出错.矩阵的扩充可用“[ ]”将小矩阵扩充成大矩阵,如可用命令

```
D=[A
   B  C]
```

构造矩阵 $\boldsymbol{D}$,其中 $\boldsymbol{B}$ 与 $\boldsymbol{C}$ 必须有相同的行数,且 $\boldsymbol{B}$ 与 $\boldsymbol{C}$ 列数之和与 $\boldsymbol{A}$ 的列数相同.也可用直接赋值的办法对超出矩阵维数的元素赋值来扩充矩阵,如果还有某些元素未赋值,则用0补齐,如 $\boldsymbol{A}$ 为 2×4 矩阵,用命令

```
A(1:3,1:3)=eye(3)
```

可将矩阵 $\boldsymbol{A}$ 变成 3×4 矩阵,且前三列为单位阵(不管矩阵原来前3列是什么),第4列的前两行不变,第3行元素用0补齐.用 $\boldsymbol{A}(:,j)=[\,]$ 或 $A(i,:)=[\,]$ 可删除矩阵的第 j 列或第 i 行(减少维数).亦可用类似的办法删除矩阵相邻的若干行与列,但不能用这种办法删除矩阵的某一个元素,否则会出错.修改矩阵元素可直接对矩阵下标操作,亦可用 $\boldsymbol{A}(:)$ 将矩阵 $\boldsymbol{A}$ 变成一列,元素个数不变.用 diag($\boldsymbol{A}$,k) 将矩阵 $\boldsymbol{A}$ 的第 k 个对角线的元素抽取组成列向量,若 k 省略等价于 $k=0$,表明是主对角线元素;用 tril($\boldsymbol{A}$,k)与 tril($\boldsymbol{A}$) 提取矩阵的下三角部分,用 triu($\boldsymbol{A}$,k)与 triu($\boldsymbol{A}$) 提取矩阵的上三角部分,未提取部分用0补齐;用 reshape($\boldsymbol{A}$,n_1,n_2,…)将矩阵 $\boldsymbol{A}$ 变成 $n_1\times n_2\times n_3\times\cdots$ 维矩阵;用 rot90($\boldsymbol{A}$)将 $\boldsymbol{A}$ 逆时针旋转 90°;用 rot90($\boldsymbol{A}$,k)将 $\boldsymbol{A}$ 逆时针旋转 $k\times90°$,其中 k 为整数;用 fliplr($\boldsymbol{A}$)将矩阵 $\boldsymbol{A}$ 左右翻转;flipud($\boldsymbol{A}$) 将矩阵 $\boldsymbol{A}$ 上下翻转.

二、矩阵与向量的基本运算

1. 矩阵的运算

(1) 矩阵的四则运算 仍用第一目矩阵输入时提及的两个 6×6 矩阵 $\boldsymbol{A}$ 和 txtfile 为例,作矩阵四则运算.矩阵的加减乘运算在线性代数中是有明确定义的.数乘矩阵和矩阵的转置也是最常见的基本运算.矩阵的除法在线性代数中是没有定义的,但在 MATLAB 中为了便于计算,专门定义了矩阵的除法运算,并且还有左除与右除之分,这些在第一章中均已述及,这里不再赘述.如:

```
A+txtfile              % 矩阵加法
ans =    4   6   3    7  -12   12
        13  14  16    2   12  -23
         9   7  27    0   11   -3
        12  11  33    7   11    1
         9  25  29   33   -8   24
        10   0  15   -2    3    5
```

```
A - txtfile                    % 矩阵减法
ans =     2     2    - 5    - 5   - 6     8
        - 1   - 4   - 16     12   - 4   - 9
        - 7  - 15   - 13    - 2     1  - 13
        - 8  - 19   - 23   - 19    13  - 17
       - 15  - 13   - 43   - 17     6  - 22
          6   - 8      3      4     3   - 5
A * txtfile                    % 矩阵乘法
ans =  - 55  - 85  - 180  - 245    80  - 175
        127   174    348    250  - 13     52
         75   110    220    194  - 41    154
         82   129    260    283  - 91    239
         53    76    138     21    21   - 29
         98   151    284    165  - 33    167
3 * A + 7 * txtfile                   % 数乘矩阵
ans =  16   26   25    45  - 48    44
       67   78  112  - 14    68  - 97
       59   65  161     4    53    11
       76   93  211    73    29    39
       75  151  231   199  - 52   164
       38   16   69  - 18     9    35
A'                             % 矩阵转置
ans =     3     6     1     2   - 3     8
          4     5   - 4   - 4     6   - 4
        - 1     0     7     5   - 7     9
          1     7   - 1   - 6     8     1
        - 9     4     6    12   - 1     3
         10  - 16   - 8   - 8     1     0
txtfile/A                          % 矩阵右除
ans =     0.9337  - 0.1687   1.5921  - 0.2383  0.7938    0.0596
          5.9273  - 0.0162   5.5138    3.0371  1.8754  - 2.0808
          7.2382  - 0.7096   6.9955    3.2673  3.3674  - 1.6106
          9.1056  - 1.0471  10.1798    2.7908  4.9550  - 1.4914
         11.3503  - 1.9092  13.2537    2.7827  7.2407  - 0.9615
          2.8057  - 0.3272   2.3154    1.3410  0.9586  - 0.8219
```

```
A\txtfile                  %矩阵左除
ans =   -2.1783  -3.1414  -6.0051  -3.2105   0.1122  -2.9871
         5.0944   7.5348  14.2952   7.7724  -0.7515   5.4291
         3.6894   5.4000  10.1410   4.5299   0.0113   4.1883
         0.2344   0.4858   0.8027   0.2483  -0.2744   1.4962
         2.1216   3.3950   6.3834   4.2520  -1.2436   3.8074
         0.9706   1.6754   3.1624   2.7091  -1.1237   2.6204
```

(2) 方阵的行列式,逆与伪逆　在 MATLAB 中,方阵行列式的值是用 Gauss 消去法得到的.如

```
det(A) = 245295
det(txtfile) = 0
```

由此可以看出为何上述矩阵左除用 A\txtfile,而不用 txtfile\A,因为 txtfile 的逆不存在,或者说矩阵 txtfile 可能是奇异的或病态的.当 $\boldsymbol{A}$ 为一非奇异矩阵时,可用 inv($\boldsymbol{A}$) 求矩阵 $\boldsymbol{A}$ 的逆,求方阵的逆时也是采用 Gauss 消去法.对于奇异阵或病态阵,虽不存在逆矩阵,但可用函数 pinv($\boldsymbol{A}$) 求其伪逆,该函数返回一个与矩阵 $\boldsymbol{A}$ 具有相同大小的矩阵 $\boldsymbol{X}$,且满足 $\boldsymbol{AXA}=\boldsymbol{A}$, $\boldsymbol{XAX}=\boldsymbol{X}$,并且 $\boldsymbol{XA}$ 与 $\boldsymbol{AX}$ 均为 Hermitian 矩阵.当 $\boldsymbol{A}$ 非奇异时, inv($\boldsymbol{A}$) 与 pinv($\boldsymbol{A}$)等价.

```
inv(A)
ans =
   -0.0737   0.0604  -0.2297   0.0067  -0.0804   0.1042
    0.3142   0.0036   0.2408   0.1605   0.1259  -0.1436
    0.2099  -0.0395   0.3155   0.0364   0.0834  -0.0663
   -0.0827  -0.0123   0.0088  -0.0777   0.0779   0.0878
    0.0134  -0.0335  -0.0159   0.1129   0.1061   0.0337
    0.0377  -0.0525  -0.0110   0.0469   0.0698   0.0411
pinv(txtfile)
ans =
   -0.0233  -0.0038   0.0359  -0.0106   0.0059  -0.0398
    0.0449   0.0296  -0.0588   0.0309  -0.0126   0.0849
    0.0262   0.0274  -0.0283   0.0240  -0.0056   0.0387
   -0.0315  -0.0236   0.0491  -0.0140   0.0272  -0.1694
   -0.1599  -0.0640   0.2255  -0.0943   0.0353  -0.2439
   -0.0769  -0.0692   0.0962  -0.0577   0.0308   0.0000
```

(3) 矩阵的迹、范数、条件数与秩:用函数 trace($\boldsymbol{A}$)求矩阵 $\boldsymbol{A}$ 的迹,如 trace($\boldsymbol{A}$) =8, trace(txtfile)=41.用 norm( )求矩阵的范数; norm($\boldsymbol{A}$)或 norm($\boldsymbol{A}$,2)给出

矩阵的普范数或 2－范数；norm(**A**,1)给出矩阵列和最大值；norm(**A**,inf)给出矩阵行和最大值；norm(**A**,' fro ')给出矩阵的 F－ 范数；norm(**A**,p) 给出矩阵的 p－范数．用 cond( ) 函数求矩阵的条件数，其用法与 norm( ) 函数完全相同．用 rank()求矩阵的秩，如 rank(**A**)＝6，而 rank(txtfile)＝4，从另一方面也反映了 txtfile 奇异的原因．

2．向量的运算

向量实际上是单行或单列矩阵，因而，作为一种特殊矩阵，向量的运算与矩阵运算是完全相同的，这里仅就三维空间的向量运算作以下简单介绍．

(1) 两向量共线与共面的判断：先输入下述两个函数

```
function y = iscline(a, b)
 % 共线性判别
 % 判断向量 a 与 b 是否共线
 % 语法 iscline(a, b), 返回 0 不共线,返回 1 共线
n = length(a);
y = 1;
if n~ = length(b)
   y = 0;
   return;
end
if rank([reshape(a, n, 1)  reshape(b, n, 1)])~ = 1
   y = 0;
   return;
end

function y = iscface(a, b, c)
 % 共面性判别
 % 判别向量 a, b, c 是否共面
 % 语法 iscface(a, b, c) , 返回 0 不共面,返回 1 共面
n = length(a);
y = 1;
if n~ = length(b)&n~ = length(c)
   y = 0;
   return;
end
```

```
if rank([reshape(a,n,1)  reshape(b,n,1)  reshape(c,n,1)]) == 3
   y = 0;
   return;
end
```

设

$$a = [1\quad 2\quad 3], b = [2\quad 4\quad 6], c = [1\quad 5\quad 3], d = [2\quad 4\quad 7]$$

用如下方式调用该函数,结果为

```
iscline(a,b) = 1          % a与b共线;
iscface(a,b,d) = 1        % a、b与d共面;
iscface(a,c,d) = 0        % a、c与d不共面.
```

(2) 向量的长度(2-范数)　可用下述函数求向量的长度(2-范数)

```
function y = radvec(v)
y = sqrt(sum(sum(v.^2)));
```

(3) 向量的方向角　可用下述函数求向量的方向角

```
function y = dangle(v,flag)
y = zeros(3,1);
if nargin == 1
   flag = 0;
end
d = radvec(v);
if d == 0
   error('d为一零向量!');
end
y = v./d;
if flag~ = 0
   y = acos(y);
end
```

(4) 向量的点乘积　可用下述函数求向量的点乘积

```
function y = dotab(a,b)
y = 0;
for i = 1:length(a)
   y = y + a(i) * b(i);
end
```

(5) 两向量的夹角　可用下述函数求两向量的夹角

```
function y = vtov(v1,v2,flag)
```

```
if nargin == 2
   flag = 0;
end
if flag == 0
   y = dotab(v1, v2)/radvec(v1)/radvec(v2);
elseif flag == 1
   y = acos(dotab(v1, v2) /radvec(v1)/radvec(v2));
elseif flag == 2
   y = acos(dotab(v1, v2) /radvec(v1)/radvec(v2)) * 180/pi;
end
```

(6) 两点间的距离　可用下述函数求两点间的距离

```
function y = dist(v1, v2)
y = radvec(v1 - v2);
```

(7) 向量的正交投影　可用下述函数求向量的正交投影

```
function [c, d] = orthprj(a, b)
k = dotab(a, b)/(radvec(b)^2);
c = k * b;
d = a - c;
```

(8) 两向量的向量积　可用下述函数求两向量的向量积

```
function y = cross_product(a, b)
a = reshape(a, 3, 1);
b = reshape(b, 3, 1);
c = eye(3);
y(1) = -det([b, a, c(:, 1)]);
y(2) = -det([b, a, c(:, 2)]);
y(3) = -det([b, a, c(:, 3)]);
```

(9) 向量的混合积　可用下述函数求向量的混合积

```
function y = mixed-product(a, b, c)
y = dotab(cross_product(a, b), c);
```

请同学们自己练习以上 9 个有关向量运算的函数.

第二节　解线性方程组

线性方程组是贯穿于线性代数始终的一个非常重要的内容,也是实际应用中使用非常广泛的内容,因而,学会求解线性方程组是非常必要的.一般情况下,