2.1 特殊矩阵

- 通用性的特殊矩阵
- 用于专门学科的特殊矩阵

1．通用的特殊矩阵

- zeros函数：产生全0矩阵，即零矩阵。
- ones函数：产生全1矩阵，即幺矩阵。
- eye函数：产生对角线为1的矩阵。当矩阵是方阵时，得到一个单位矩阵。
- rand函数：产生（0，1）区间均匀分布的随机矩阵。
- randn函数：产生均值为0，方差为1的标准正态分布随机矩阵。

zeros函数的调用格式：

- zeros(m)：产生m×m零矩阵。
- zeros(m,n)：产生m×n零矩阵。
- zeros(size(A))：产生与矩阵A同样大小的零矩阵。

```
>> A=zeros(2,3)
A =
     0     0     0
     0     0     0
>> zeros(size(reshape(A,3,2)))
ans =
     0     0
     0     0
     0     0
```

例1　首先产生5阶两位随机整数矩阵A，再产生均值为0.6、方差为0.1的5阶正态分布随机矩阵B，最后验证(A+B)I=IA+BI（I为单位矩阵）。

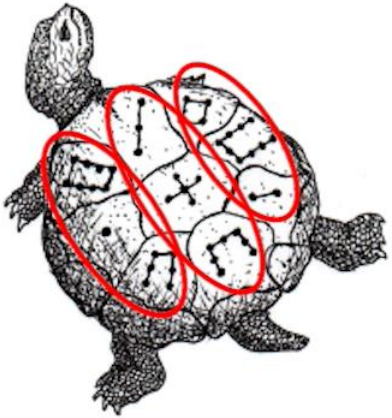
- rand函数：产生(0，1)开区间均匀分布的随机数*x*。
- fix(a+(b-a+1)*x)：产生[a，b]区间上均匀分布的随机整数。
- randn函数：产生均值为0、方差为1的标准正态分布随机数*x*。
- $\mu+\sigma x$：得到均值为μ、方差为σ^2的随机数。

```
>> A=fix(10+(99-10+1)*rand(5));
>> B=0.6+sqrt(0.1)*randn(5);
>> C=eye(5);
>> (A+B)*C==C*A+B*C
ans =
     1     1     1     1     1
     1     1     1     1     1
     1     1     1     1     1
     1     1     1     1     1
     1     1     1     1     1
```

2．用于专门学科的特殊矩阵

(1)魔方矩阵

—Magic Square



```
>> M=magic(3)
M =
     8     1     6
     3     5     7
     4     9     2
```

- n阶魔方阵由1, 2, 3, …, n^2共n^2个整数组成，且每行、每列以及主、副对角线上各n个元素之和都相等。
- n阶魔方阵每行每列元素的和为$(1+2+3+\cdots+ n^2)/n=(n+n^3)/2$
- n>2时有很多不同的n阶魔方阵，MATLAB函数magic(n)产生一个特定的魔方阵。

例2　产生8阶魔方阵，求其每行每列元素的和。

```
>> M=magic(8);
>> sum(M(1,:))
ans =
   260
>> sum(M(:,1))
ans =
   260
```

（2）范德蒙矩阵

对于向量$v=[v_1, v_2, \cdots, v_n]$， 范得蒙矩阵的一般形式为：

$$V=\begin{bmatrix} v_1^{n-1} & \cdots & v_1^2 & v_1^1 & v_1^0 \\ v_2^{n-1} & \cdots & v_2^2 & v_2^1 & v_2^0 \\ v_3^{n-1} & \cdots & v_3^2 & v_3^1 & v_3^0 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ v_n^{n-1} & \cdots & v_n^2 & v_n^1 & v_n^0 \end{bmatrix}$$

范德蒙（Vandermonde）矩阵是法国数学家范德蒙提出的一种特殊矩阵。范得蒙矩阵的最后一列全为1，即向量v各元素的零次方，倒数第二列为指定的向量v，即向量v各元素的一次方， 其他各列是其后列与倒数第二列的点乘积。

在MATLAB中，函数vander(V)生成以向量V为基础的范得蒙矩阵。

```
>> A=vander(1:5)
A =
     1     1     1     1     1
    16     8     4     2     1
    81    27     9     3     1
   256    64    16     4     1
   625   125    25     5     1
```

范德蒙矩阵常用在各种通信系统的纠错编码中，例如，常用的Reed-Solomon编码即以范德蒙矩阵为基础。

（3）希尔伯特矩阵

n阶希尔伯特（Hilbert）矩阵的一般形式为：

$$H=\begin{bmatrix} 1 & 1/2 & \cdots & 1/n \\ 1/2 & 1/3 & \cdots & 1/(n+1) \\ \vdots & \vdots & \ddots & \vdots \\ 1/n & 1/(n+1) & \cdots & 1/(2n-1) \end{bmatrix}$$

希尔伯特矩阵的元素为H(i,j)=1/(i+j-1)。

在MATLAB中，生成n阶希尔伯特矩阵的函数是hilb(n)。

```
>> format rat
>> H=hilb(4)
H =
       1          1/2        1/3        1/4
       1/2        1/3        1/4        1/5
       1/3        1/4        1/5        1/6
       1/4        1/5        1/6        1/7
```

希尔伯特矩阵是著名的病态矩阵，即任何一个元素发生较小的变动，整个矩阵的值和逆矩阵都会发生很大变化。病态程度和矩阵的阶数相关，随着阶数的增加病态越严重。

(4) 伴随矩阵

设多项式p(x)为$a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0$，则多项式的伴随矩阵是：

$$A = \begin{bmatrix} -\frac{a_{n-1}}{a_n} & -\frac{a_{n-2}}{a_n} & -\frac{a_{n-3}}{a_n} & \cdots & -\frac{a_1}{a_n} & -\frac{a_0}{a_n} \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix}$$

p(x)称为A的特征多项式，方程p(x)=0的根称为A的特征值。

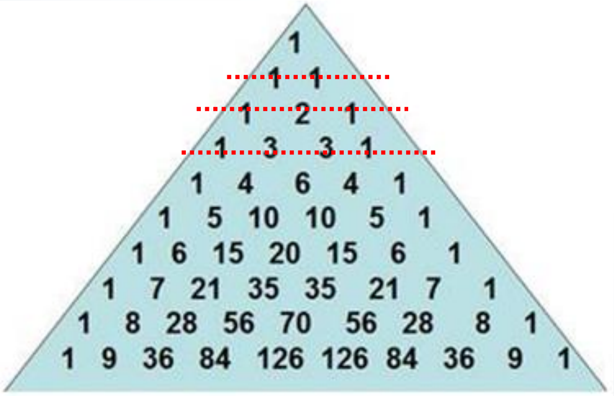
MATLAB生成伴随矩阵的函数是compan(p)，其中p是一个多项式的系数向量，高次幂系数排在前，低次幂排在后。例如，生成多项式x^3-2x^2-5x+6的伴随矩阵。

```
>> p=[1,-2,-5,6];
>> A=compan(p)
A =
     2     5    -6
     1     0     0
     0     1     0
```

可以求出伴随矩阵的特征值，该特征值等于多项式方程的根。

(5) 帕斯卡矩阵

- 根据二项式定理，(x+y)^{n}展开后的系数随着n的增大组成一个三角形表，这个三角形称为杨辉三角形。
- 把二项式系数依次填写在矩阵的左侧对角线上，然后提取左侧的n行n列元素即为n阶帕斯卡（Pascal）矩阵。



$$\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\
1 & 3 & 6 & 10 & 15 & 21 & 28 & 36 \\
1 & 4 & 10 & 20 & 35 & 56 & 84 & 120 \\
1 & 5 & 15 & 35 & 70 & 126 & 210 & 330 \\
1 & 6 & 21 & 56 & 126 & 252 & 462 & 792 \\
1 & 7 & 28 & 84 & 210 & 462 & 924 & 1716 \\
1 & 8 & 36 & 120 & 330 & 792 & 1716 & 3432
\end{bmatrix}$$

- 帕斯卡矩阵的第一行元素和第一列元素都为1，其余位置的元素是该元素的左边元素与上面元素相加，即P(i,j)=P(i,j-1)+P(i-1,j)，且P(i,1)=1,P(1,j)=1。
- 函数pascal(n)生成一个n阶帕斯卡矩阵。

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 3 & 6 & 10 & 15 & 21 & 28 & 36 \\ 1 & 4 & 10 & 20 & 35 & 56 & 84 & 120 \\ 1 & 5 & 15 & 35 & 70 & 126 & 210 & 330 \\ 1 & 6 & 21 & 56 & 126 & 252 & 462 & 792 \\ 1 & 7 & 28 & 84 & 210 & 462 & 924 & 1716 \\ 1 & 8 & 36 & 120 & 330 & 792 & 1716 & 3432 \end{bmatrix}$$

例3　生成5阶帕斯卡矩阵，验证它的逆矩阵的所有元素也为整数。

```
>> format rat
>> P=pascal(5)
P =
       1            1            1            1            1
       1            2            3            4            5
       1            3            6           10           15
       1            4           10           20           35
       1            5           15           35           70
>> inv(P)
ans =
       5          -10           10           -5            1
     -10           30          -35           19           -4
      10          -35           46          -27            6
      -5           19          -27           17           -4
       1           -4            6           -4            1
```

2.2 矩阵变换

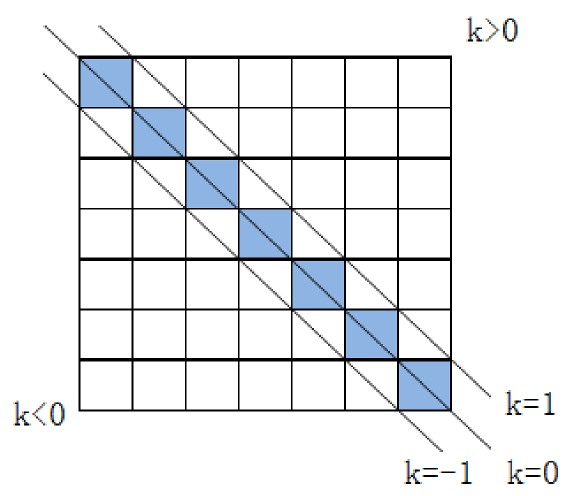
- 对角阵
- 三角阵
- 矩阵的转置
- 矩阵的旋转
- 矩阵的翻转
- 矩阵求逆

1．对角阵

- 对角阵：只有对角线上有非零元素的矩阵。
- 数量矩阵：对角线上的元素相等的对角矩阵。
- 单位矩阵：对角线上的元素都为1的对角矩阵。

(1) 提取矩阵的对角线元素

- diag(A)：提取矩阵A主对角线元素，产生一个列向量。
- diag(A,k)：提取矩阵A第k条对角线的元素，产生一个列向量。



矩阵的对角线：与主对角线平行，往上为第1条、第2条、一直到第n条对角线，往下为第-1条、-2条、一直到-n条对角线。主对角线为第0条对角线。

(2) 构造对角阵

- diag(V)：以向量 V为主对角线元素，产生对角矩阵。
- diag(V,k)：以向量 V为第k条对角线元素，产生对角矩阵。

例1　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=[7,0,1,0,5;3,5,7,4,1;4,0,3,0,2;1,1,9,2,3;1,8,5,2,9]
A =
     7     0     1     0     5
     3     5     7     4     1
     4     0     3     0     2
     1     1     9     2     3
     1     8     5     2     9
>> D=diag(1:5);
>> D*A
ans =
     7     0     1     0     5
     6    10    14     8     2
    12     0     9     0     6
     4     4    36     8    12
     5    40    25    10    45
```

用一个对角阵左乘一个矩阵时，相当于用对角阵对角线的第1个元素乘以该矩阵的第一行，用对角阵对角线的第2个元素乘以该矩阵的第二行，…，依此类推。

要将A的各列元素分别乘以对角阵的对角线元素，如何实现？

```
>> A=[7,0,1,0,5;3,5,7,4,1;4,0,3,0,2;1,1,9,2,3;1,8,5,2,9]
A =
     7     0     1     0     5
     3     5     7     4     1
     4     0     3     0     2
     1     1     9     2     3
     1     8     5     2     9
>> D=diag(1:5);
>> A*D
ans =
     7     0     3     0    25
     3    10    21    16     5
     4     0     9     0    10
     1     2    27     8    15
     1    16    15     8    45
```

要将A的各列元素分别乘以对角阵的对角线元素，可以用一个对角阵右乘矩阵A。

2．三角阵

- 上三角阵：矩阵的对角线以下的元素全为零的矩阵。
- 下三角阵：对角线以上的元素全为零的矩阵。

（1）上三角矩阵

- triu(A)：提取矩阵A的主对角线及以上的元素。
- triu(A,k)：提取矩阵A的第k条对角线及以上的元素。

```
>> triu(ones(4),-1)
ans =
     1     1     1     1
     1     1     1     1
     0     1     1     1
     0     0     1     1
```

（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，其用法与triu函数完全相同。

3. 矩阵的转置

- 转置运算符是小数点后面接单引号（.'）。
- 共轭转置，其运算符是单引号（'），它在转置的基础上还要取每个数的复共轭。

```
>> A=[1,3;3+4i,1-2i]
A =
    1.0000 + 0.0000i   3.0000 + 0.0000i
    3.0000 + 4.0000i   1.0000 - 2.0000i
>> A.'
ans =
    1.0000 + 0.0000i   3.0000 + 4.0000i
    3.0000 + 0.0000i   1.0000 - 2.0000i
>> A'
ans =
    1.0000 + 0.0000i   3.0000 - 4.0000i
    3.0000 + 0.0000i   1.0000 + 2.0000i
```

- 矩阵的转置：把源矩阵的第一行变成目标矩阵的第一列，第二行变成第二列，…，依此类推。
- 如果矩阵的元素是实数，那么转置和共轭转置的结果是一样的。

4．矩阵的旋转

rot90(A,k)：将矩阵A逆时针方向旋转90°的k倍，当k为1时可省略。

```
>> A=[1,3,2;-3,2,1;4,1,2]
A =
       1         3         2
      -3         2         1
       4         1         2
>> rot90(A)
ans =
       2         1         2
       3         2         1
       1        -3         4
>> rot90(A,2)
ans =
       2         1         4
       1         2        -3
       2         3         1
```

5．矩阵的翻转

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，…，依此类推。

- fliplr(A)：对矩阵A实施左右翻转
- flipud(A)：对矩阵A实施上下翻转。

例2　验证魔方阵的主对角线、副对角线元素之和相等。

```
>> A=magic(5);
>> D1=diag(A);
>> sum(D1)
ans =
    65
>> B=flipud(A);
>> D2=diag(B);
>> sum(D2)
ans =
    65
```

- 对矩阵A实施上下翻转得到矩阵B，这样A的副对角线就移到了B的主对角线
- 5阶魔方阵的主对角线、副对角线元素之和相等，都为65。

6．矩阵的求逆

- 对于一个方阵A，如果存在一个与其同阶的方阵B，使得AB=BA=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。
- inv(A)：求方阵A的逆矩阵。

例3 用求逆矩阵的方法解线性方程组。

$$\begin{cases} x + 2y + 3z = 5 \\ x + 4y + 9z = -2 \\ x + 8y + 27z = 6 \end{cases}$$

在线性方程组Ax=b两边各左乘A^{-1}，得$x=A^{-1}b$。

```
>> A=[1,2,3;1,4,9;1,8,27];
>> b=[5;-2;6];
>> x=inv(A)*b
x =
   23.0000
  -14.5000
    3.6667
>> x=A\b
x =
   23.0000
  -14.5000
    3.6667
```

2.3 矩阵求值

- 矩阵的行列式值
- 矩阵的秩
- 矩阵的迹
- 矩阵的范数
- 矩阵的条件数

1. 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，这个值就称为所对应的行列式的值。
- det(A)：求方阵A所对应的行列式的值。

例1　验证$\det(A^{-1})=1/\det(A)$。

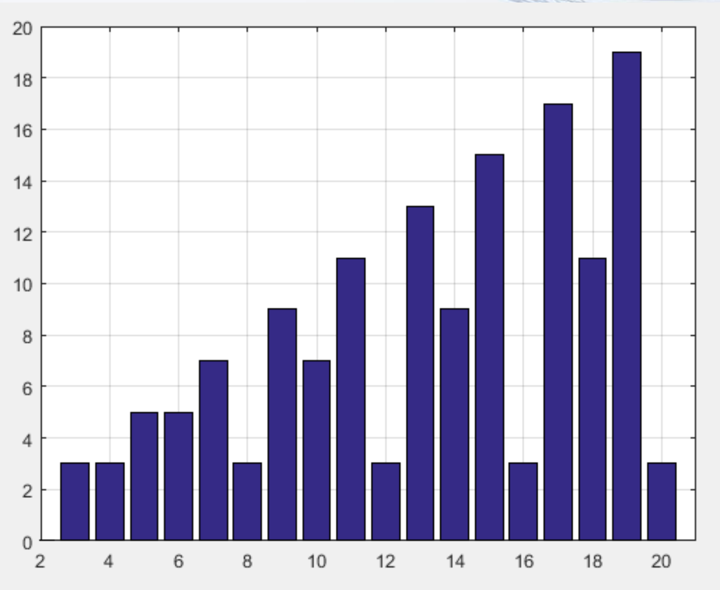
```
>> format rat
>> A=[1,3,2;-3,2,1;4,1,2]
A =
       1              3              2
      -3              2              1
       4              1              2
>> det(inv(A))
ans =
       1/11
>> 1/det(A)
ans =
       1/11
```

2. 矩阵的秩

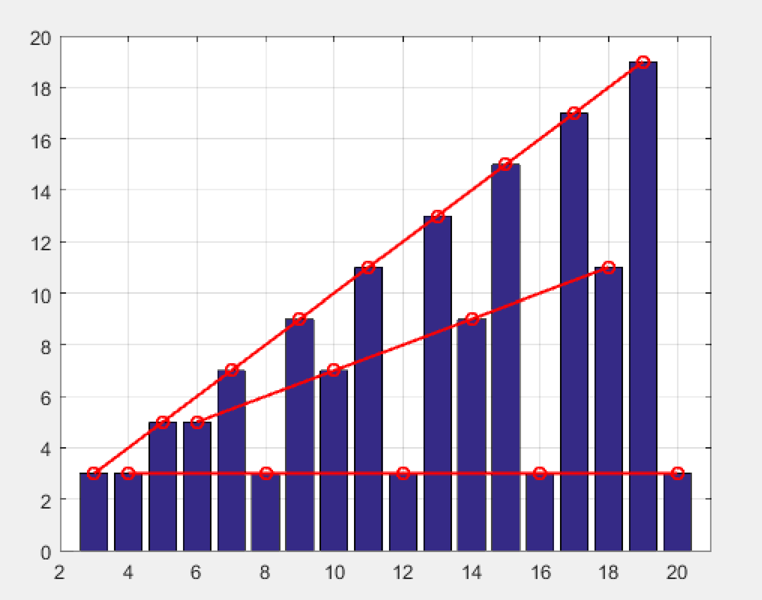
- 矩阵线性无关的行数或列数称为矩阵的秩。
- rank(A)：求矩阵A的秩。

例2　求3~20阶魔方阵的秩。

```
for n=3:20
    r(n)=rank(magic(n));
end
bar(r)
grid on
axis([2,21,0,20])
[3:20;r(3:20)]
```



n=3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
r=3	3	5	5	7	3	9	7	11	3	13	9	15	3	17	11	19	3



- 奇数阶魔方阵秩为n，即奇数阶魔方阵是满秩矩阵。
- 一重偶数阶魔方阵秩为n/2+2（n是2的倍数，但非4的倍数）。
- 双重偶数阶魔方阵秩均为3 （阶数是4的倍数） 。

3．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。
- trace(A)：求矩阵A的迹。

```
>> A=[1,3,2;-3,2,1;4,1,2]
A =
       1          3          2
      -3          2          1
       4          1          2
>> b = trace(A)
b =
       5
>> t = sum(diag(A))
t =
       5
```

4．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。

（1）向量的3种常用范数

- 向量1—范数：向量元素的绝对值之和。

$$\|V\|_1 = \sum_{i=1}^{n} |v_i|$$

- 向量2—范数：向量元素绝对值的平方和的平方根。

$$\|V\|_2 = \sqrt{\sum_{i=1}^{n} |v_i|^2}$$

- 向量∞—范数：所有向量元素绝对值中的最大值。

$$\|V\|_\infty = \max_{1\le i\le n}\{|v_i|\}$$

在MATLAB中，求向量范数的函数为：

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

（2）矩阵的范数

- 矩阵A的1—范数：所有矩阵列元素绝对值之和的最大值。

$$\|A\|_1 = \max_{1\le j\le n}\{\sum_{i=1}^{m}|a_{ij}|\}$$

- 矩阵A的2—范数：A’A矩阵的最大特征值的平方根。

$$\|A\|_2 = \sqrt{\lambda_1}$$

其中λ1为A’A的最大特征值。

- 矩阵A的∞—范数：所有矩阵行元素绝对值之和的最大值。

$$\|A\|_\infty = \max_{1\le i\le m}\{\sum_{j=1}^{n}|a_{ij}|\}$$

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。

```
>> x=[2 0 1;-1 1 0;-3 3 0]
x =
     2     0     1
    -1     1     0
    -3     3     0
>> n = norm(x)
n =
    4.7234
>> n = norm(x,1)
n =
     6
```

5．矩阵的条件数

- 矩阵A的条件数等于A的范数与A的逆矩阵的范数的乘积。
- 条件数越接近于1，矩阵的性能越好，反之，矩阵的性能越差。

在MATLAB中，计算矩阵A的3种条件数的函数是：

- cond(A,1)：计算A的1—范数下的条件数。
- cond(A)或cond(A,2)：计算A的2—范数数下的条件数。
- cond(A,inf)：计算A的∞—范数下的条件数。

例3 求2~10阶希尔伯特矩阵的条件数。

```
for n=2:10
    c(n)=cond(hilb(n));
end
format long
c'
```

```
   1.0e+13 *

                 0
   0.000000000001928
   0.000000000052406
   0.000000001551374
   0.000000047660725
   0.000001495105864
   0.000047536735631
   0.001525757554777
   0.049315394619572
   1.602490962516758
```

随着阶数的增加，希尔伯特矩阵的条件数不断增大，矩阵性能变差。

2.4 矩阵的特征值与特征向量

- 矩阵特征值的数学定义
- 求矩阵的特征值与特征向量
- 特征值的几何意义

1．矩阵特征值的数学定义

设A是n阶方阵，如果存在常数λ和n维非零列向量x，使得等式Ax=λx成立，则称λ为A的特征值，x是对应特征值λ的特征向量。

2．求矩阵的特征值与特征向量

在MATLAB中，计算矩阵的特征值和特征向量的函数是eig，常用的调用格式有两种：

- E=eig(A)：求矩阵A的全部特征值，构成向量E。
- [X,D]=eig(A)：求矩阵A的全部特征值，构成对角阵D，并产生矩阵X，X各列是相应的特征向量。

```
>> A=[1,1,0;1,0,5;1,10,2]
A =
     1     1     0
     1     0     5
     1    10     2
>> [X,D]=eig(A)
X =
    0.0722    0.9751    0.0886
    0.5234   -0.0750   -0.6356
    0.8490   -0.2089    0.7669
D =
    8.2493         0         0
         0    0.9231         0
         0         0   -6.1723
```

```
>> A*X(:,1)
ans =
    0.5956
    4.3174
    7.0040
>> D(1)*X(:,1)
ans =
    0.5956
    4.3174
    7.0040
```

例1　设

$$\mathbf{A}=\begin{bmatrix} \boldsymbol{R}_{3\times3} & \boldsymbol{O}_{3\times2} \\ \boldsymbol{O}_{2\times3} & \boldsymbol{S}_{2\times2} \end{bmatrix}$$

又设$\boldsymbol{\lambda}_i$为R的特征值，$\boldsymbol{\lambda}_j$为S的特征值，$\boldsymbol{x}_{\mathrm{i}}=(\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,\boldsymbol{\alpha}_3)'$是R对应于$\boldsymbol{\lambda}_i$的特征向量，$\boldsymbol{y}_{\mathrm{j}}=(\boldsymbol{\beta}_1,\boldsymbol{\beta}_2)'$是S对应于$\boldsymbol{\lambda}_j$的特征向量，试验证：

（1）$\boldsymbol{\lambda}_i$、$\boldsymbol{\lambda}_j$为A的特征值。

（2）$\mathbf{p}_{\mathrm{i}}=(\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,\boldsymbol{\alpha}_3,\mathbf{0},\mathbf{0})'$是A对应于$\boldsymbol{\lambda}_i$的特征向量，$\boldsymbol{q}_{\mathrm{j}}=(\mathbf{0},\mathbf{0},\mathbf{0},\boldsymbol{\beta}_1,\boldsymbol{\beta}_2)'$是A对应于$\boldsymbol{\lambda}_j$的特征向量。

```
R=[-1,2,0;2,-4,1;1,1,-6];
S=[1,2;2,3];
A=[R,zeros(3,2);zeros(2,3),S];
[X1,d1]=eig(R)
[X2,d2]=eig(S)
[X3,d3]=eig(A)
```

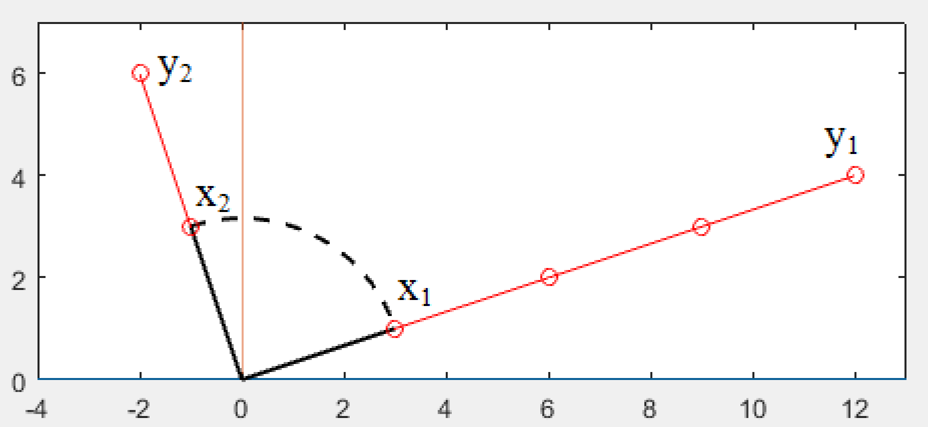
A矩阵的特征值由R矩阵的特征值和S矩阵的特征值组成，关于A矩阵每个特征值的特征向量，前三个特征向量的前三个元素是R的特征向量，后两个特征向量的后两个元素是S的特征向量，运算结果与结论相符。

```
X1 =
    0.8553    0.4517    0.1899
    0.4703   -0.8395   -0.5111
    0.2173   -0.3021    0.8383
d1 =
    0.0996         0         0
         0   -4.7165         0
         0         0   -6.3832
X2 =
   -0.8507    0.5257
    0.5257    0.8507
d2 =
   -0.2361         0
         0    4.2361
```

```
X3 =
    0.8553    0.4517    0.1899         0         0
    0.4703   -0.8395   -0.5111         0         0
    0.2173   -0.3021    0.8383         0         0
         0         0         0   -0.8507   -0.5257
         0         0         0    0.5257   -0.8507
d3 =
    0.0996         0         0         0         0
         0   -4.7165         0         0         0
         0         0   -6.3832         0         0
         0         0         0   -0.2361         0
         0         0         0         0    4.2361
```

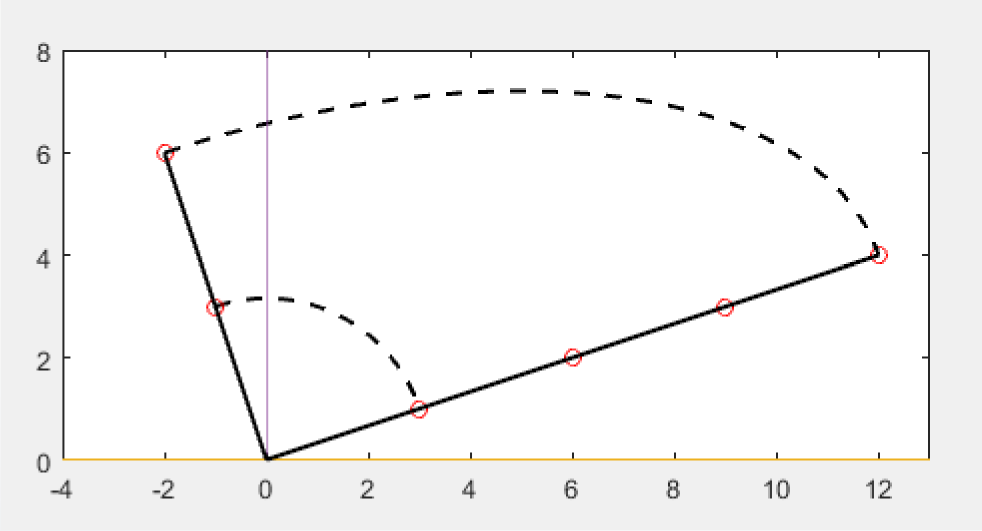
3．特征值的几何意义

设$A=\begin{bmatrix}3.8 & 0.6\\0.6 & 2.2\end{bmatrix}$，其特征向量有$x_1=\begin{bmatrix}3\\1\end{bmatrix}$，$x_2=\begin{bmatrix}-1\\3\end{bmatrix}$，对应的特征值分别为$\lambda_1=4$和$\lambda_2=2$，令$y_1=Ax_1=\lambda_1x_1$，$y_2=Ax_2=\lambda_2x_2$，我们讨论$y_1$与$x_1$，$y_2$与$x_2$之间的关系。

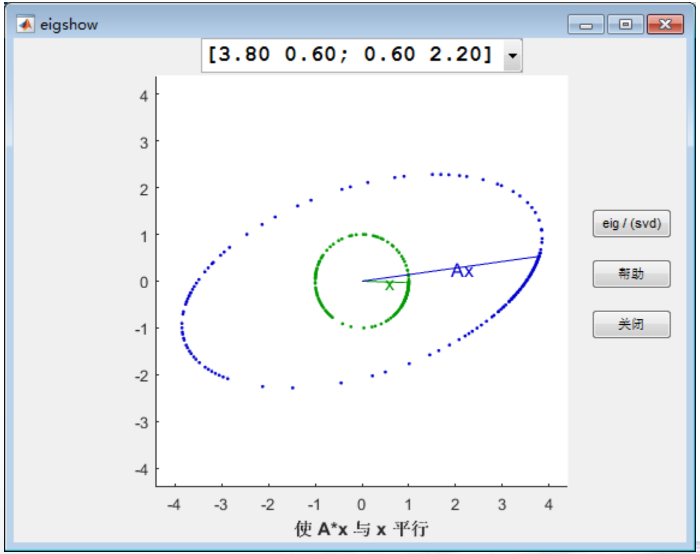


y_1、y_2是x_1、x_2经过矩阵A变换以后的结果，A相当于一个变换矩阵。把λ_1、λ_2当作伸缩因子，y_1、y_2是x_1、x_2经过λ_1、λ_2伸缩以后的结果，如图所示。黑色部分代表向量x_1和x_2，红色部分代表对x_1和x_2进行拉伸的结果。

更进一步，连续取单位向量x，让它大小保持为1，那么Ax就将四分之一圆弧进行拉伸，变成四分之一椭圆。



MATLAB提供了一个eigshow命令，可以演示向量x和Ax之间的关系。用鼠标拖动绿色的单位向量x绕原点转动，图中同步出现蓝色的Ax向量。Ax的大小在变化，方向也在变化，而且Ax的方向与x不一定相同。在变化过程中，x与Ax共线的位置称为特征方向。在特征方向上有Ax等于λx。



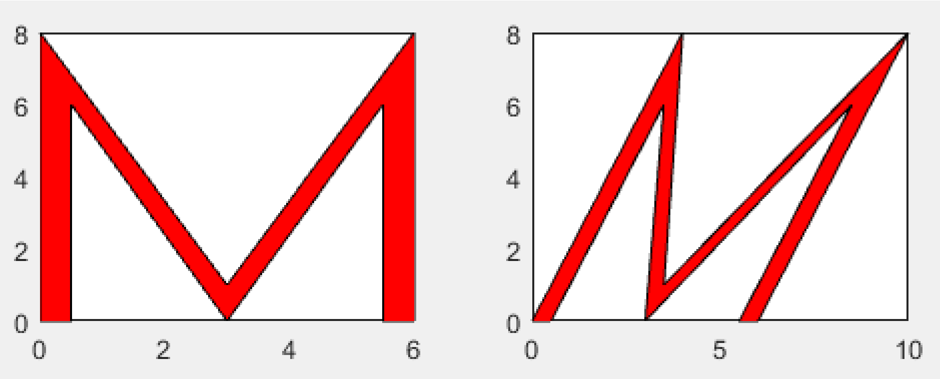
例2 已知大写字母M的各个结点坐标如表所示（第一行代表横坐标，第二行代表纵坐标）。

x	0	0.5	0.5	3	5.5	5.5	6	6	3	0
y	0	0	6	0	6	0	0	8	1	8

（1）绘制M的图形。

（2）设$A=\begin{bmatrix}1 & 0.5\\0 & 1\end{bmatrix}$，用A对M的结点坐标进行变换，并绘制变换后的图形。

```
x=[0,0.5,0.5,3,5.5,5.5,6,6,3,0;0,0,6,0,6,0,0,8,1,8];
A=[1,0.5;0,1];
y=A*x;
subplot(2,2,1);                 %选择1号子图，详见专题四
fill(x(1,:),x(2,:),'r');        %绘制M的图形，并用红色（red）填充
subplot(2,2,2);                 %选择2号子图
fill(y(1,:),y(2,:),'r');        %绘制变换后的M图形，并用红色填充
```



- 定义变换矩阵A，再利用A对x进行变换，得到y矩阵，最后分别绘制变换前后的图形，M原来是正体，变换后改为斜体。
- **启示：**在构建字库时，不必单独创建斜体字库，而只需对正体字库进行适当的线性变换即可，这样可以大大节省存储空间。

2.5 稀疏矩阵

- 矩阵的存储方式
- 稀疏存储方式的产生
- 稀疏矩阵的应用实例

1．矩阵的存储方式

- 完全存储方式：将矩阵的全部元素按列存储。
- 稀疏存储方式：只存储矩阵的非零元素的值及其位置，即行号和列号。

注意，采用稀疏存储方式时，矩阵元素的存储顺序并没有改变，也是按列的顺序进行存储。

$$A=\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 5 & 0 & 0\\ 2 & 0 & 0 & 7\end{bmatrix}$$

A矩阵的稀疏存储方式：

（1，1），1

（3，1），2

（2，2），5

（3，4），7

当矩阵的规模很大时，采用稀疏存储方式可以大大节约存储空间。

2．稀疏存储方式的产生

（1）完全存储方式与稀疏存储方式之间的转化

- A=sparse(S)：将矩阵S转化为稀疏存储方式的矩阵A。
- S=full(A)：将矩阵A转化为完全存储方式的矩阵S。

```
>> A=sparse(eye(5))
A =
   (1,1)        1
   (2,2)        1
   (3,3)        1
   (4,4)        1
   (5,5)        1
>> B=full(A)
B =
     1     0     0     0     0
     0     1     0     0     0
     0     0     1     0     0
     0     0     0     1     0
     0     0     0     0     1
>> whos
  Name      Size            Bytes  Class     Attributes
  A         5x5               128  double    sparse
  B         5x5               200  double
```

（2）直接建立稀疏存储矩阵

sparse函数的其他调用格式：

- sparse(m,n)：生成一个m×n的所有元素都是零的稀疏矩阵。
- sparse(u,v,S)：其中u、v、S是3个等长的向量。S是要建立的稀疏存储矩阵的非零元素，u(i)、v(i)分别是S(i)的行和列下标。

```
>> A=sparse([1,2,2],[2,1,4],[4,5,-7])
A =
    (2,1)          5
    (1,2)          4
    (2,4)         -7
>> B=full(A)
B =
      0     4     0     0
      5     0     0    -7
```

使用spconvert函数直接建立稀疏存储矩阵，其调用格式为：

B=spconvert(A)

其中，A为一个m×3或m×4的矩阵，其每行表示一个非零元素，m是非零元素的个数。

- A(i,1)表示第i个非零元素所在的行。
- A(i,2)表示 第i个非零元素所在的列。
- A(i,3)表示第i个非零元素值的实部。
- A(i,4)表示第i个非零元素值的虚部。

若矩阵的全部元素都是实数，则无须第4列。

```
>> A=[2,2,1;2,1,-1;2,4,3]
A =
     2     2     1
     2     1    -1
     2     4     3
>> B=spconvert(A)
B =
   (2,1)       -1
   (2,2)        1
   (2,4)        3
```

（3）带状稀疏矩阵的稀疏存储

- 稀疏矩阵有两种基本类型：无规则结构的稀疏矩阵与有规则结构的稀疏矩阵。
- 带状稀疏矩阵就是一种十分典型的具有规则结构的稀疏矩阵，它是指所有非零元素集中在对角线上的矩阵。

- [B,d]=spdiags(A)：从带状稀疏矩阵A中提取全部非零对角线元素赋给矩阵B及其这些非零对角线的位置向量d。
- A=spdiags(B,d,m,n)：产生带状稀疏矩阵的稀疏存储矩阵A，其中m、n为原带状稀疏矩阵的行数与列数，矩阵B的第i列即为原带状稀疏矩阵的第i条非零对角线，向量d为原带状稀疏矩阵所有非零对角线的位置。

```
>> A =[11,0,0,12,0,0;0,21,0,0,22,0;0,0,31,0,0,32;41,0,0,42,0,0;0,51,0,0,52,0]
A =
    11     0     0    12     0     0
     0    21     0     0    22     0
     0     0    31     0     0    32
    41     0     0    42     0     0
     0    51     0     0    52     0
>> [B,d]=spdiags(A)
B =
     0    11    12
     0    21    22
     0    31    32
    41    42     0
    51    52     0
d =
    -3
     0
     3
```

利用带状稀疏矩阵非零对角线元素组成的矩阵B，以及对角线位置组成的向量d，命令执行后产生一个稀疏存储矩阵A。

```
>> A=spdiags(B,d,5,6)
A =
   (1,1)        11
   (4,1)        41
   (2,2)        21
   (5,2)        51
   (3,3)        31
   (1,4)        12
   (4,4)        42
   (2,5)        22
   (5,5)        52
   (3,6)        32
```

总　结

用spdiags函数产生带状稀疏矩阵的稀疏存储A：

A=spdiags(B,d,m,n)

其中，m、n为原带状矩阵的行数与列数。B为r×p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。取值方法是：若非零对角线上元素个数等于r，则取全部元素；若非零对角线上元素个数小于r，则应该用零补足到r个元素。补零的原则是：若m<n（行数<列数），则d<0时（主对角线以下）在前面补0，d>0时（主对角线以上）在后面补0；当m≥n（行数≥列数），则d<0时在后面补0；d>0时在前面补0。

（4）单位矩阵的稀疏存储

speye(m,n)返回一个m×n的稀疏存储单位矩阵。

```
>> speye(3)
ans =
   (1,1)        1
   (2,2)        1
   (3,3)        1
```

3．稀疏矩阵应用举例

求下列三对角线性方程组的解。

$$\begin{bmatrix} 2 & 3 & & & \\ 1 & 4 & 1 & & \\ & 1 & 6 & 4 & \\ & & 2 & 6 & 2 \\ & & & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 2 \\ 1 \\ 5 \end{bmatrix}$$

```
>> kf1=[1;1;2;1;0];
>> k0=[2;4;6;6;1];
>> k1=[0;3;1;4;2];
>> B=[kf1,k0,k1];
>> d=[-1;0;1];
>> A=spdiags(B,d,5,5);
>> f=[0;3;2;1;5];
>> x=A\f
x =
   -0.1667
    0.1111
    2.7222
   -3.6111
    8.6111
```