第五章　矩阵的特征值与特征向量的计算

5.2　幂法及其 MATLAB 程序

幂法的 MATLAB 程序

用幂法计算矩阵 A 的主特征值和对应的特征向量的 MATLAB 主程序

```
function [k,lambda,Vk,Wc]=mifa(A,V0,jd,max1)
lambda=0;k=1;Wc =1; ,jd=jd*0.1;state=1; V=V0;
while((k<=max1)&(state==1))
Vk=A*V; [m j]=max(abs(Vk)); mk=m;
tzw=abs(lambda-mk); Vk=(1/mk)*Vk;
Txw=norm(V-Vk); Wc=max(Txw,tzw); V=Vk;lambda=mk;state=0;
  if(Wc>jd)
      state=1;
  end
      k=k+1;Wc=Wc;
end
if(Wc<=jd)
    disp('请注意：迭代次数k,主特征值的近似值lambda,主特征向量的近似向量Vk,相
    邻两次迭代的误差Wc如下：')
else
   disp('请注意：迭代次数k已经到达最大迭代次数max1,主特征值的迭代值lambda,主特
   征向量的迭代向量Vk,相邻两次迭代的误差Wc如下：')
end
 Vk=V;k=k-1;Wc;
```

例　用幂法计算以下矩阵的主特征值和对应的特征向量的近似向量，精度 $\varepsilon=10^{-5}$. 并把〔1〕和〔2〕输出的结果与例 5.1.1 中的结果进行比拟.

(1) $\boldsymbol{A}=\begin{pmatrix}1 & -1\\ 2 & 4\end{pmatrix}$；　(2) $\boldsymbol{B}=\begin{pmatrix}1 & 2 & 3\\ 2 & 1 & 3\\ 3 & 3 & 6\end{pmatrix}$；(3) $\boldsymbol{C}=\begin{pmatrix}1 & 2 & 2\\ 1 & -1 & 1\\ 4 & -12 & 1\end{pmatrix}$；〔4〕 $\boldsymbol{D}=\begin{pmatrix}-4 & 14 & 0\\ -5 & 13 & 0\\ -1 & 0 & 2\end{pmatrix}$.

解　〔1〕输入 MATLAB 程序

```
>>A=[1 -1;2 4]; V0=[1,1]';
 [k,lambda,Vk,Wc]=mifa(A,V0,0.00001,100),
[V,D] = eig (A), Dzd=max(diag(D)), wuD= abs(Dzd- lambda),
wuV=V(:,2)./Vk,
```

运行后屏幕显示结果

```
请注意：迭代次数 k,主特征值的近似值 lambda,主特征向量的近似向量 Vk,相邻两次迭代的误差 Wc 如下：
k =         lambda =                 Wc =

Vk =                    V =                          wuV =


Dzd =      wuD =
    3       1.738368038406435e-006
```

由输出结果可看出，迭代 33 次，相邻两次迭代的误差 $W_c\approx$8.69 19e-007，矩阵 $\boldsymbol{A}$ 的主特征值的近似值 $lambda\approx$3.000 00 和对应的特征向量的近似向量 $\boldsymbol{V}_k\approx$〔-0.500 00，1.000 00〕$^{\mathrm{T}}$, $lambda$ 与例中 $\boldsymbol{A}$ 的最大特征值 $\lambda_2=3$ 近似相等，绝对误差约为 1.738 37e-006，$\boldsymbol{V}_k$ 与特征向量 $\boldsymbol{X}_2^{\mathrm{T}}=k_2(\frac{1}{2},-1)^{\mathrm{T}}\quad(k_2\neq 0)$ 的第 1 个分量的绝对误差约等于 0，第 2 个分量的绝对值相同.由 $\boldsymbol{wuV}$ 可以看出，λ_2 的特征向量 $\boldsymbol{V}(:,2)$

与 $\boldsymbol{V}_k$ 的对应分量的比值近似相等. 因此，用程序 mifa.m 计算的结果到达预先给定的精度 $\varepsilon=10^{-5}$.

(2) 输入 MATLAB 程序

```
>>B=[1 2 3;2 1 3;3 3 6]; V0=[1,1,1]';
[k,lambda,Vk,Wc]=mifa(B,V0,0.00001,100), [V,D] = eig (B),
Dzd=max(diag(D)), wuD= abs(Dzd- lambda), wuV=V(:,3)./Vk,
```

运行后屏幕显示结果

```
请注意：迭代次数 k,主特征值的近似值 lambda,主特征向量的近似向量 Vk,相邻两次迭代的误差 Wc 如下:
k =        lambda =     Wc =     Dzd =     wuD =
  3               9       0        9         0
Vk =                    wuV =



V =
```

(3) 输入 MATLAB 程序

```
>> C=[1 2 2;1 -1 1;4 -12 1];V0=[1,1,1]';
[k,lambda,Vk,Wc]=mifa(C,V0,0.00001,100), [V,D] = eig (C),
Dzd=max(diag(D)), wuD=abs(Dzd-lambda),
Vzd=V(:,1),wuV=V(:,1)./Vk,
```

运行后屏幕显示

```
请注意：迭代次数 k 已经到达最大迭代次数 max1,主特征值的迭代值 lambda,主特征向量的迭代向量 Vk,
相邻两次迭代的误差 Wc 如下:
k =     lambda =           Wc =
 100      0.09090909090910   2.37758124193119
Dzd =                     wuD =
  1.00000000000001          0.90909090909091
Vk=                 Vzd =                wuV =
0.99999999999993      0.90453403373329       0.90453403373335
```

由输出结果可见，迭代次数 k 已经到达最大迭代次数 $max1=100$，并且 $lambda$ 的相邻两次迭代的误差 $Wc\approx 2.377\,58>2$,由 $\boldsymbol{wuV}$ 可以看出，$lambda$ 的特征向量 $\boldsymbol{V}_k$ 与真值 Dzd 的特征向量 $\boldsymbol{Vzd}$ 对应分量的比值相差较大，所以迭代序列发散. 实际上，实数矩阵 $\boldsymbol{C}$ 的特征值的近似值为 $\lambda_1=1.000\,000\,000\,000\,01, \lambda_2=-\mathrm{i}, \lambda_3=\mathrm{i}$，并且对应的特征向量的近似向量分别为 $\boldsymbol{X}_1^{\mathrm{T}}=k_1$ (0.90453403373329,,) $^{\mathrm{T}}$,

$\boldsymbol{X}_2^{\mathrm{T}}=k_2$ (-0.72547625011001,-0.07254762501100i,

0.58038100008801-) $^{\mathrm{T}}$,

$\boldsymbol{X}_3^{\mathrm{T}}=k_3$ (-0.72547625011001, ,

) $^{\mathrm{T}}$ $(k_1\neq 0, k_2\neq 0$, $k_3\neq 0$ 是常数).

(4) 输入 MATLAB 程序

```
>> D=[-4 14 0;-5 13 0;-1 0 2]; V0=[1,1,1]';
[k,lambda,Vk,Wc]=mifa(D,V0,0.00001,100), [V,Dt] =eig (D),
Dtzd=max(diag(Dt)), wuDt=abs(Dtzd-lambda),
 Vzd=V(:,2),wuV=V(:,2)./Vk,
```

运行后屏幕显示结果

```
请注意：迭代次数 k,主特征值的近似值 lambda,主特征向量的近似向量 Vk,相邻两次迭代的误差 Wc 如下:
k =    lambda =              Wc =
  19     6.00000653949528    6.539523793591684e-006
Dtzd =                      wuDt =
  6.00000000000000         6.539495284840768e-006
Vk =                Vzd =                  wuV =
   0.79740048053564   0.79740048053564     0.79740048053564
```

5.3 反幂法和位移反幂法及其 MATLAB 程序

原点位移反幂法的 MATLAB 程序

〔一〕 原点位移反幂法的 MATLAB 主程序 1

用原点位移反幂法计算矩阵 A 的特征值和对应的特征向量的 MATLAB 主程序 1

```
function [k,lambdan,Vk,Wc]=ydwyfmf(A,V0,jlamb,jd,max1)
[n,n]=size(A); A1=A-jlamb*eye(n); jd= jd*0.1;RA1=det(A1);
if RA1==0
disp('请注意：因为A-aE的n阶行列式hl等于零，所以A-aE不能进行LU分解.')
return
end
lambda=0;
if RA1~=0
       for p=1:n
          h(p)=det(A1(1:p, 1:p));
        end
          hl=h(1:n);
        for i=1:n
          if h(1,i)==0
             disp('请注意: 因为A-aE的r阶主子式等于零,所以A-aE不能进行LU分解.')
            return
          end
        end
       if h(1,i)~=0
           disp('请注意：因为A-aE的各阶主子式都不等于零，所以A-aE能进行LU分解.')
             k=1;Wc =1;state=1; Vk=V0;
          while((k<=max1)&(state==1))
               [L U]=lu(A1); Yk=L\Vk;Vk=U\Yk; [m j]=max(abs(Vk));
               mk=m;Vk1=Vk/mk; Yk1=L\Vk1;Vk1=U\Yk1;
               [m j]=max(abs(Vk1));
               mk1=m;Vk2=(1/mk1)*Vk1;tzw1=abs((mk-mk1)/mk1);
               tzw2=abs(mk1-mk);Txw1=norm(Vk)-norm(Vk1);
               Txw2=(norm(Vk)-norm(Vk1))/norm(Vk1);
               Txw=min(Txw1,Txw2); tzw=min(tzw1,tzw2); Vk=Vk2;
               mk=mk1; Wc=max(Txw,tzw); Vk=Vk2;mk=mk1;state=0;
              if(Wc>jd)
                 state=1;
              end
                 k=k+1;%Vk=Vk2,mk=mk1,
          end
          if(Wc<=jd)
             disp('A-aE的秩R(A-aE)和各阶顺序主子式值hl、迭代次数k,按模最小特征值的近似值lambda,特征向量的近似向量Vk,相邻两次迭代的误差Wc如下：')
          else
             disp('A-aE的秩R(A-aE)和各阶顺序主子式值hl、迭代次数k已经到达最大迭代次数max1,按模最小特征值的迭代值lambda,特征向量的迭代向量Vk,相邻两次迭代的误差Wc如下：')
      end
        hl,RA1
    end
end
[V,D]=eig(A,'nobalance'),Vk;k=k-1;Wc;lambdan=jlamb+1/mk1;
```

例 用原点位移反幂法的迭代公式（5.28），根据给定的以下矩阵的特征值λ_n的初始值$\tilde{\lambda}_n$，计算与λ_n对应的特征向量$\boldsymbol{X}_n$的近似向量,精确到0.000 1.

（1）$\begin{pmatrix} 1 & -1 & 0 \\ -2 & 4 & -2 \\ 0 & -1 & 2 \end{pmatrix}$, $\tilde{\lambda}_2 = 0.2$；（2）$\begin{pmatrix} 1 & -1 \\ 2 & 4 \end{pmatrix}$，$\tilde{\lambda}_2 = 2.001$；（3）$\begin{pmatrix} -11 & 2 & 15 \\ 2 & 58 & 3 \\ 15 & 3 & -3 \end{pmatrix}$，$\tilde{\lambda}_3 = 8.26$.

解 （1）输入 MATLAB 程序

```
>> A=[1 -1 0;-2 4 -2;0 -1 2];V0=[1,1,1]';
[k,lambda,Vk,Wc]=ydwyfmf(A,V0,0.2,0.0001,10000)
```

运行后屏幕显示结果

```
请注意：因为A-aE的各阶主子式都不等于零，所以A-aE能进行LU分解.
A-aE的秩R(A-aE)和各阶顺序主子式值hl、迭代次数k,按模最小特征值的近似值lambda,特征向量的近似向量Vk,相邻两次迭代的误差Wc如下：
k =    lambda =     Wc =            hl =
  3      0.2384     1.0213e-007    0.8000    1.0400    0.2720
Vk =         V =                          D =
 1.0000     -0.2424  -1.0000   -0.5707      5.1249        0         0
 0.7616     1.0000   -0.7616   0.3633          0    0.2384        0
 0.4323     -0.3200  -0.4323   1.0000           0        0    1.6367
```

（2）输入 MATLAB 程序

```
>> A=[1 -1;2 4];V0=[20,1]';
 [k,lambda,Vk,Wc]=ydwyfmf(A,V0,2.001,0.0001,100)
```

运行后屏幕显示结果

```
请注意：因为A-aE的各阶主子式都不等于零，所以A-aE能进行LU分解.
A-aE的秩R(A-aE)和各阶顺序主子式值hl、迭代次数k,按模最小特征值的近似值lambda,特征向量的近似向量Vk,相邻两次迭代的误差Wc如下：
k =      lambda =      Wc =              hl =
  2          2.0020      5.1528e-007      -1.0010   -0.0010
Vk =               V =                        D =
   1.0000            -1.0000    0.5000           2     0
  -1.0000             1.0000   -1.0000            0     3
```

（3）输入 MATLAB 程序

```
>> A=[-11 2 15;2 58 3;15 3 -3];V0=[1,1,-1]';
 [k,lambdan,Vk,Wc]=ydwyfmf(A,V0,8.26, 0.0001,100)
```

运行后屏幕显示结果

```
请注意：因为A-aE的各阶主子式都不等于零，所以A-aE能进行LU分解.
A-aE的秩R(A-aE)和各阶顺序主子式值hl、迭代次数k,按模最小特征值的近似值lambda,特征向量的近似向量Vk,相邻两次迭代的误差Wc如下：
k =      lambdan=       Wc =           hl =
  2        8.2640        6.9304e-008    -19.2600 -961.9924   -6.1256
Vk =        V =                               D =
   -0.7692     0.7928   0.6081  0.0416    -22.5249         0         0
    0.0912     0.0030  -0.0721  0.9974          0    8.2640         0
   -1.0000    -0.6095  0.7906  0.0590          0         0   58.2609
```

例 用原点位移反幂法的迭代公式（5.28），计算$\boldsymbol{A} = \begin{pmatrix} 0 & 11 & -5 \\ -2 & 17 & -7 \\ -4 & 26 & -10 \end{pmatrix}$的分别对应于特征值$\lambda_1 \approx \tilde{\lambda}_1 = 1.001$，$\lambda_2 \approx \tilde{\lambda}_2 = 2.001$，$\lambda_3 \approx \tilde{\lambda}_3 = 4.001$的特征向量$\boldsymbol{X}_1$，$\boldsymbol{X}_2$，$\boldsymbol{X}_3$的近似向量，相邻迭代误差为0.001. 将计算结果与精确特征向量比拟.

解 （1）计算特征值$\lambda_1 \approx \tilde{\lambda}_1 = 1.001$ 对应的特征向量$\boldsymbol{X}_1$的近似向量. 输入 MATLAB 程序

```
>> A=[0 11 -5;-2 17 -7;-4 26 -10];V0=[1,1,1]';
[k,lambda,Vk,Wc]= ydwyfmf(A,V0,1.001, 0.001,100),
[V,D]=eig(A);
Dzd=min(diag(D)), wuD= abs(Dzd- lambda),
```

```
VD=V(:,1),wuV=V(:,1)./Vk,
```

运行后屏幕显示结果

请注意：因为 A-aE 的各阶主子式都不等于零，所以 A-aE 能进行 LU 分解.

A-aE 的秩 R(A-aE)和各阶顺序主子式值 hl、迭代次数 k,按模最小特征值的近似值 lambda,特征向量的近似向量 Vk,相邻两次迭代的误差 Wc 如下：

```
hl =

k =          lambda =                    RA1 =
   5    1.00200000000000         -0.00299600100000
Vk =                    VD =                  wuV =



Wc =                          Dzd =            wuD =
1.378794763695562e-009 1.00000000000000    0.00200000000000
```

从输出的结果可见，迭代 5 次，特征向量 $\boldsymbol{X}_1$ 的近似向量 $\widetilde{\boldsymbol{X}}_1$ 的相邻两次迭代的误差 $Wc \approx 1.379$ e-009，由 wuV 可以看出，$\widetilde{\boldsymbol{X}}_1 = Vk$ 与 VD 的对应分量的比值相等.特征值 λ_1 的近似值 $lambda \approx 1.002$ 与初始值 $\widetilde{\lambda}_1 = 1.001$ 的绝对误差为 0.001，而与 λ_1 的绝对误差为 0.002，其中

$$\boldsymbol{X}_1 = (-0.50000000000000\ ,\ -0.50000000000000\ ,\ 1.00000000000000\ )^{\mathrm{T}},$$

$$\widetilde{\boldsymbol{X}}_1 = (-0.500\,000\,000\,000\,00,\ -0.500\,000\,000\,000\,00,\ 1.000\,000\,000\,000\,00)^{\mathrm{T}}.$$

（2）计算特征值 $\lambda_2 \approx \widetilde{\lambda}_2 = 2.001$ 对应特征向量 $\boldsymbol{X}_2$ 的近似向量.

输入 MATLAB 程序

```
>> A=[0 11 -5;-2 17 -7;-4 26 -10];V0=[1,1,1]';
[k,lambda,Vk,Wc]=ydwyfmf(A,V0,2.001, 0.001,100) ,
[V,D]=eig(A); WD=lambda-D(2,2),VD=V(:,2),wuV=V(:,2)./Vk,
```

运行后屏幕显示结果

请注意：因为 A-aE 的各阶主子式都不等于零，所以 A-aE 能进行 LU 分解.

A-aE 的秩 R(A-aE)和各阶顺序主子式值 hl、迭代次数 k,按模最小特征值的近似值 lambda,特征向量的近似向量 Vk,相邻两次迭代的误差 Wc 如下：

```
hl =

k =  Wc =                  lambda =          WD =
 2      2.00200000000016  0.00200000000016
  Vk =                   VD =                 wuV =

 -0.49999999999999    0.43643578047198   -0.87287156094398
  -1.00000000000000   0.87287156094397   -0.87287156094397
```

从输出的结果可见，迭代 2 次，特征向量 $\boldsymbol{X}_2$ 的近似向量 $\widetilde{\boldsymbol{X}}_2$ 的相邻两次迭代的误差 $Wc \approx$3.131e-007，$\widetilde{\boldsymbol{X}}_2$ 与 $\boldsymbol{X}_2$ 的对应分量的比值近似相等.特征值 λ_2 的近似值 *lambda*≈2.002 与初始值 $\widetilde{\lambda}_2 =$2.001 的绝对误差约为 0.001，而 *lambda* 与 λ_2 的绝对误差约为 0.002，其中

$$\widetilde{\boldsymbol{X}}_2 = (-0.249\,999\,999\,999\,99, -0.499\,999\,999\,999\,99, -1.000\,000\,000\,000\,00)^{\mathrm{T}},$$

$$\boldsymbol{X}_2 = (-0.24999999999999\ ,\ -0.50000000000000\ ,\ \ -1.00000000000000\ )^{\mathrm{T}}.$$

（3）计算特征值 $\lambda_3 \approx \widetilde{\lambda}_3 = 4.001$ 对应特征向量 $\boldsymbol{X}_3$ 的近似向量. 输入 MATLAB 程序

```
>> A=[0 11 -5;-2 17 -7;-4 26 -10];V0=[1,1,1]';
[k,lambda,Vk,Wc]=ydwyfmf(A,V0,4.001, 0.001,100)
[V,D]=eig(A); WD=lambda-max(diag(D)),VD=V(:,3),wuV=V(:,3)./Vk,
```

运行后屏幕显示结果

请注意：因为 A-aE 的各阶主子式都不等于零，所以 A-aE 能进行 LU 分解.

A-aE 的秩 R(A-aE)和各阶顺序主子式值 hl、迭代次数 k,按模最小特征值的近似值 lambda,特征向量的近似向量 Vk,相邻两次迭代的误差 Wc 如下：

```
hl =
```

```
k =  lambda =           Wc =              WD =

      Vk =                 VD =              wuV =
```

从输出的结果可见，迭代 2 次，特征向量 $\boldsymbol{X}_3$ 的近似向量 $\widetilde{\boldsymbol{X}}_3$ 的相邻两次迭代的误差 $Wc \approx 1.996\text{e-}007$，$\widetilde{\boldsymbol{X}}_3$ 与 $\boldsymbol{X}_3$ 的对应分量的比值近似相等.特征值 λ_3 的近似值 $lambda \approx 4.002$ 与初始值 $\widetilde{\lambda}_2 = 4.001$ 的绝对误差近似为 0.001，而 *lambda* 与 λ_3 的绝对误差约为 0.002，其中

$$\boldsymbol{X}_3 = (\,\text{-0.400 000 000 000 00，-0.600 000 000 000 00，-1.000 000 000 000 00})^{\mathrm{T}},$$

$$\widetilde{\boldsymbol{X}}_3 = (0.400\,000\,000\,000\,1, 0.600\,000\,000\,000\,01, 1.000\,000\,000\,000\,00)^{\mathrm{T}}.$$

（二）原点位移反幂法的 MATLAB 主程序 2

用原点位移反幂法计算矩阵 A 的特征值和对应的特征向量的 MATLAB 主程序 2

```
function [k,lambdan,Vk,Wc]=wfmifa1(A,V0,jlamb,jd,max1)
[n,n]=size(A); jd= jd*0.1;A1=A-jlamb*eye(n);nA1=inv(A1);
lambda1=0;k=1;Wc =1;state=1; U=V0;
while((k<=max1)&(state==1))
    Vk=A1\U; [m j]=max(abs(Vk)); mk=m; Vk=(1/mk)*Vk; Vk1=A1\Vk;
    [m1 j]=max(abs(Vk1)); mk1=m1,Vk1=(1/mk1)*Vk1;U=Vk1,
    Txw=(norm(Vk1)-norm(Vk))/norm(Vk1);
    tzw=abs((lambda1-mk1)/mk1);
    Wc=max(Txw,tzw); lambda1=mk1;state=0;
  if(Wc>jd)
    state=1;
  end
    k=k+1;
end
if(Wc<=jd)
    disp('请注意迭代次数k,特征值的近似值lambda,对应的特征向量的近似向量Vk,相
   邻两次迭代的误差Wc如下：')
else
   disp('请注意迭代次数k已经到达最大迭代次数max1, 特征值的近似值lambda,对应的
   特征向量的近似向量Vk,相邻两次迭代的误差Wc如下：')
end
[V,D] =eig(A,'nobalance'), Vk=U;k=k-1;Wc;lambdan=jlamb+1/mk;
```

例 用原点位移反幂法的迭代公式（5.27），计算例题 5.3.3，并且将这两个例题的计算结果进行比拟. 再用两种原点位移反幂法的 MATLAB 主程序，求 $\widetilde{\lambda}_1 = 0.999\,999\,999\,999\,97$ 对应的特征向量.

解 （1）计算特征值 $\lambda_1 \approx \widetilde{\lambda}_1 = 1.001$ 对应特征向量 $\boldsymbol{X}_1$ 的近似向量.

输入 MATLAB 程序

```
>> A=[0 11 -5;-2 17 -7;-4 26 -10];V0=[1,1,1]';
   [k,lambda,Vk,Wc]=wfmifa1(A,V0,1.001,0.001,100)
```

运行后屏幕显示结果

请注意迭代次数 k，特征值的近似值 lambda，对应的特征向量的近似向量 Vk，相邻两次迭代的误差 Wc 如下：

```
k =   lambda =             Wc =
 5    1.00200000000138     1.376344154436924e-006
Vk' =  -0.50000000000000  -0.50000000000000  -1.00000000000000
```

同理可得，另外与两个特征值对应的特征向量.

（2）再用两种原点位移反幂法的MATLAB主程序，求 $\widetilde{\lambda}_1 = 0.999\,999\,999\,999\,97$ 对应的特征向量.

输入MATLAB程序

```
>> A=[0 11 -5;-2 17 -7;-4 26 -10];V0=[1,1,1]';
```

```
[k,lambda,Vk,Wc]=ydwyfmf(A,V0,0.99999999999997,0.001,100)
```

运行后屏幕显示结果

请注意：因为A-aE的各阶主子式都不等于零，所以A-aE能进行LU分解.

A-aE的秩R(A-aE)和各阶顺序主子式值hl、迭代次数k,按模最小特征值的近似值lambda,特征向量的近似向量Vk,相邻两次迭代的误差Wc如下：

```
hl =
  -0.99999999999997   6.00000000000045   0.00000000000010
RA1 =

k =
     2
lambda =
      1.00000000000000
Vk =
   0.50000000000000
   0.50000000000000
   1.00000000000000
Wc =
```

输入MATLAB程序

```
>>  A=[0 11 -5;-2 17 -7;-4 26 -10];V0=[1,1,1]';
   [k,lambda,Vk,Wc]=wfmifa1(A,V0, 0.99999999999997,0.001,100)
```

运行后屏幕显示结果

请注意迭代次数 k,特征值的近似值 lambda,对应的特征向量的近似向量 Vk,相邻两次迭代的误差 Wc 如下：

```
k =
     3
lambda =
   1.00000000000000
Wc =
Vk =
   0.50000000000000
   0.50000000000000
   1.00000000000000
```

5.4　雅可比(Jacobi)方法及其 MATLAB 程序

雅可比方法的 MATLAB 程序

用雅可比方法计算对称矩阵 A 的特征值和对应的特征向量的 MATLAB 主程序

```
function [k,Bk,V,D,Wc]=jacobite(A,jd,max1)
[n,n]=size(A);Vk=eye(n);Bk=A;state=1;k=0;P0=eye(n);
Aij=abs(Bk-diag(diag(Bk)));[m1 i]=max(Aij);
[m2 j]=max(m1);i=i(j);
while ((k<=max1)&(state==1))
   k=k+1,aij=abs(Bk-diag(diag(Bk)));[m1 i]=max(abs(aij));
   [m2 j]=max(m1);i=i(j),j,Aij=(Bk-diag(diag(Bk)));
   mk=m2*sign(Aij(i,j)),
   Wc=m2,Dk=diag(diag(Bk));Pk=P0;
   c=(Bk(j,j)-Bk(i,i))/(2*Bk(i,j)),
   t=sign(c)/(abs(c)+sqrt(1+c^2)),
   pii=1/( sqrt(1+t^2)), pij=t/( sqrt(1+t^2)),
   Pk(i,i)=pii;Pk(i,j)=pij;
   Pk(j,j)=pii; Pk(j,i)=-pij;
   Pk,B1=Pk'*Bk;B2=B1*Pk; Vk=Vk*Pk,Bk=B2,
   if(Wc>jd)
     state=1;
     else
          return
   end
   Pk;Vk;Bk=B2;Wc;
end
if(k>max1)
   disp('请注意迭代次数k已经到达最大迭代次数max1,迭代次数k,对称矩阵Bk,以特征向量为列向量的矩阵V,特征值为对角元的对角矩阵D如下：')
   else
   disp('请注意迭代次数k,对称矩阵Bk,以特征向量为列向量的矩阵V,
```

```
    特征值为对角元的对角矩阵D如下：')
end
Wc;k=k; V=Vk;Bk=B2;D=diag(diag(Bk));[V1,D1] =eig(A,'nobalance')
```

例　用雅可比方法的 MATLAB 程序计算矩阵 $A=\begin{pmatrix}12 & -56 & 3 & -1\\ -56 & 7 & 2 & 0\\ 3 & 2 & 5 & 1\\ -1 & 0 & 1 & 12\end{pmatrix}$ 的特征值 λ_i 和对应的特征向量 X_i 〔$i=1,2,3,4$〕.

解　〔1〕保存名为jacobite.m为M文件；

〔2〕输入MATLAB程序

```
>> A=[12 -56 3 -1;-56 7 2 0;3 2 5 1;-1 0 1 12];
[k,B,V,D,Wc]=jacobite(A,0.001,100)
```

〔3〕运行后屏幕显示如下：

```
k =        i =        j =        mk =        Wc =
 1          2          1          -56          56
c =                          t =
 -0.04464285714286             -0.95635313919972
pii =                        pij =
  0.72270271801843
Pk =


          0              0    1.00000000000000              0
          0              0                   0    1.00000000000000
Vk =


          0              0    1.00000000000000              0
          0              0                   0    1.00000000000000
Bk =
 65.55577579518456              0    0.78579012788509   -0.72270271801843


k =    i =     j =     mk =                        Wc =
 2      3       2
c =                          t =
 -7.32558932518824           -0.06793885568129
pii =                        pij =
  0.99770011455446           -0.06778260409592
Pk =
  1.00000000000000              0                   0                   0
          0    0.99770011455446    0.06778260409592                   0
          0   -0.06778260409592    0.99770011455446                   0
          0              0                   0    1.00000000000000
Vk =
  0.72270271801843    0.68956942653035    0.04684855775127                   0

          0   -0.06778260409592    0.99770011455446                   0
          0              0                   0    1.00000000000000
Bk =
 65.55577579518456   -0.05326290114092    0.78398290060672   -0.72270271801843
 -0.05326290114093 -46.79484464383285                   0   -0.75735203062627
  0.78398290060672    0.00000000000000    5.23906884864829    0.95085155680318
 -0.72270271801843   -0.75735203062627    0.95085155680318   12.00000000000000
k =    i =     j =        mk =                     Wc =
 3      4       3     0.95085155680318        0.95085155680318
c = t =

pii =                        pij =
  0.99061693994324
Pk =
  1.00000000000000              0                   0                   0
          0    1.00000000000000                   0                   0
```

Vk =
0.72270271801843 0.68956942653035 0.04640897492032 0.00640268773403

Bk =
65.55577579518456 -0.05326290114092 0.87539690801061 -0.60877636330628

k = i = j = mk = Wc =
4 1 3 0.87539690801061 0.87539690801061
c = t =
-34.52598931799430
pii = pij =
0.99989519853186
Pk =

0 1.00000000000000 0 0

0 0 0 1.00000000000000
Vk =
0.72329885394465 0.68956942653035 0.03594133368062 0.00640268773403

Bk =

k = i = j = mk = Wc =
5 4 2 -0.75024575103880 0.75024575103880
c = t =

pii = pij =
0.99991898429114
Pk =
1.00000000000000 0 0 0

0 0 1.00000000000000 0

Vk =
0.72329885394465 0.68959505973603 0.03594133368062 -0.00237529014628

Bk =
65.56845049923633 -0.05950288535679 -0.00000000000000 -0.60800441437674

k = i = j = mk = Wc =
6 4 1 -0.60800441437674 0.60800441437674
c = t =
-43.93694931878409
pii = pij =
0.99993527149402
Pk =

0 1.00000000000000 0 0
0 0 1.00000000000000 0

Vk =
0.72327906130899 0.68959505973603 0.03594133368062 0.00585436528595

Bk =

65.57536865930122 -0.05949903382392 -0.00008516835377 -0.00000000000000

k = i = j = mk = Wc =
7 3 2

c = t =
-2.486337309269764e+002 -0.00201098208240

pii = pij =
0.99999797798167 -0.00201097801616

Pk =

1.00000000000000	0	0	0
0	0.99999797798167	0.00201097801616	0
0	-0.00201097801616	0.99999797798167	0
0	0	0	1.00000000000000

……………………………………………………………………………

请注意迭代次数 k,对称矩阵 Bk,以特征向量为列向量的矩阵 V,特征值为对角元的对角矩阵 D 如下：

V1 =
0.68990429476497 -0.03732423222484 0.00588594854431 -0.72291377173450

D1 =

-46.80463661419736	0	0	0
0	5.09541442877727	0	0
0	0	0	65.57540016115307

k =
10

B =

65.57540016045945	0.00000000000175	-0.00020481967566	0.00000014862836
0.00000000000175	-46.80463661419739	0.00000062739984	0.00000000000000
-0.00020481967566	0.00000062739984	5.09541442947090	-0.00000000000737

V =
0.72291389811507 0.68990429521617 0.03732177568689 0.00588595055487

D =

65.57540016045945	0	0	0
0	-46.80463661419739	0	0
0	0	5.09541442947090	0
0			

Wc =
6.920584967017158e-004

5.5 豪斯霍尔德(Householder)方法及其 MATLAB 程序

豪斯霍尔德方法及其 MATLAB 程序

求初等反射矩阵 P，使得 PX 的第一个分量以外的其余的分量都为零的 MATLAB 主程序

```
function [xigema,rou,miou,P,PX]=Householder(X)
n=size(X);nX=norm(X,2);
xigema=nX*sign(X(1));
rou=xigema*(xigema+X(1));
miou=[xigema,zeros(1,n-1)]'+X,
E=eye(n,n); C=2*miou*(miou)';
P=E-C/(norm(miou,2)^2); PX=P*X;
```

例 设向量 $X=(2,2,1)^{T}$，确定一个初等反射矩阵 P，使得 PX 的后两个分量为零.

解 输入 MATLAB 程序

```
>> X=[2 2 1]'; [xigema,rou,miou,P,PX]=Householder(X)
```

运行后屏幕显示结果

P = PX =

-0.6667	-0.6667	-0.3333	-3.0000
-0.6667	0.7333	-0.1333	0.0000
-0.3333	-0.1333	0.9333	0.0000

矩阵约化为上豪斯霍尔德矩阵及其 MATLAB 程序

用豪斯霍尔德变换将 n 阶矩阵 A 规约成上豪斯霍尔德矩阵的 MATLAB 主程序

```
function [k,Sk,uk,ck,Pk,Uk,Ak]=Householder1(A)
n=size(A); Ak=A;
for k=1:n-2
k,Sk=norm(Ak(k+1:n,k))*sign(Ak(k+1,k)),
uk= Ak(k+1:n,k)+ Sk*eye(n-k,1),
ck=(norm(uk,2)^2)/2,
Pk= eye(n-k,n-k)-uk*uk'/ck,
Uk=[eye(k,k),zeros(k,n-k);zeros(n-k, k),Pk],
A1=Uk*Ak;Ak=A1,
end
```

例 用初等反射矩阵正交相似约化实矩阵 A 为上豪斯霍尔德矩阵.其中

$$A=\begin{pmatrix} 12 & -52 & 34 & -12 & 17 & -51\\ -56 & 7 & 2 & 0 & 32 & -17\\ 3 & 2 & 5 & 1 & 72 & -63\\ -1 & 0 & 1 & 12 & 21 & -94\\ -32 & -78 & -10.2 & 98 & -72 & 11\\ 31 & -41 & -78 & 37 & -19 & 34 \end{pmatrix}.$$

解 输入 MATLAB 程序

```
>> A=[12 -52 34 -12 17 -51;-56 7 2 0 32 -17;3 2 5 1 72 -63;-1 0 1 12
21  -94;-32  -78  -10.2  98  -72  11;31  -41  -78  37  -19  34];
[k,Sk,uk,ck,Pk,Uk,Ak]=Householder1(A)
```

运行后屏幕显示结果

```
k =       Sk =             ck =
  1          -71.6310         9.1423e+003
uk =              Pk =
 -127.6310       -0.7818    0.0419   -0.0140   -0.4467    0.4328
    3.0000        0.0419    0.9990    0.0003    0.0105   -0.0102
   -1.0000       -0.0140    0.0003    0.9999   -0.0035    0.0034
  -32.0000       -0.4467    0.0105   -0.0035    0.8880    0.1085
   31.0000        0.4328   -0.0102    0.0034    0.1085    0.8949
Uk =
    1.0000         0         0         0         0         0
         0   -0.7818    0.0419   -0.0140   -0.4467    0.4328
         0    0.0419    0.9990    0.0003    0.0105   -0.0102
         0   -0.0140    0.0003    0.9999   -0.0035    0.0034
         0   -0.4467    0.0105   -0.0035    0.8880    0.1085
         0    0.4328   -0.0102    0.0034    0.1085    0.8949
Ak =
   12.0000  -52.0000   34.0000  -12.0000   17.0000  -51.0000
   71.6310   11.7128  -30.5678  -27.8930    1.6473   21.7643
    0.0000    1.8892    5.7655    1.6556   72.7134  -63.9112
   -0.0000    0.0369    0.7448   11.7815   20.7622  -93.6963
   -0.0000  -76.8184  -18.3655   91.0066  -79.6101   20.7191
    0.0000  -42.1447  -70.0897   43.7749  -11.6277   24.5846
k =      Sk =              ck =
   2      87.6402             7.8464e+003
uk =                Pk =
   89.5295              -0.0216   -0.0004    0.8765    0.4809
    0.0369              -0.0004    1.0000    0.0004    0.0002
  -76.8184               0.8765    0.0004    0.2479   -0.4126
  -42.1447               0.4809    0.0002   -0.4126    0.7736
```

```
Uk =
     1.0000         0         0         0         0         0
          0    1.0000         0         0         0         0
          0         0   -0.0216   -0.0004    0.8765    0.4809
          0         0   -0.0004    1.0000    0.0004    0.0002
          0         0    0.8765    0.0004    0.2479   -0.4126
          0         0    0.4809    0.0002   -0.4126    0.7736
Ak =
   12.0000  -52.0000   34.0000  -12.0000   17.0000  -51.0000
   71.6310   11.7128  -30.5678  -27.8930    1.6473   21.7643
   -0.0000  -87.6402  -49.9272  100.7790  -76.9476   31.4002
   -0.0000   -0.0000    0.7219   11.8223   20.7005  -93.6570
   -0.0000    0.0000   29.4202    5.9564   48.8026  -61.0603
    0.0000    0.0000  -43.8731   -2.8860   58.8230  -20.2818
………………………………………………………………………………
k =      Sk =            ck =
  4       -12.2088          195.0398
uk =                   Pk =
   -15.9753                -0.3085    0.9512
    11.6133                 0.9512    0.3085
Uk =
     1.0000         0         0         0         0         0
          0    1.0000         0         0         0         0
          0         0    1.0000         0         0         0
          0         0         0    1.0000         0         0
          0         0         0         0   -0.3085    0.9512
          0         0         0         0    0.9512    0.3085
Ak =
   12.0000  -52.0000   34.0000  -12.0000   17.0000  -51.0000
   71.6310   11.7128  -30.5678  -27.8930    1.6473   21.7643
   -0.0000  -87.6402  -49.9272  100.7790  -76.9476   31.4002
    0.0000   -0.0000  -52.8292   -5.8754   21.3902   18.4403
    0.0000    0.0000    0.0000   12.2088   40.2435 -106.8134
    0.0000    0.0000   -0.0000    0.0000   64.7555  -34.0909
```

实对称矩阵的三对角化及其 MATLAB 程序

将 n 阶实对称矩阵 A 规约成三对角形式的 MATLAB 主程序

```
function T=house(A)
[n,n]=size(A);
for k=1:n-2
  s=norm(A(k+1:n,k),2);
  if (A(k+1,k)<0)
    s=-s;
  end
  r=sqrt(2*s*(A(k+1,k)+s));
  U(1:k)=zeros(1,k);
  U(k+1)=(A(k+1,k)+s)/r;
  U(k+2:n)=A(k+2:n,k)'/r;
  V(1:k)=zeros(1,k);
  V(k+1:n)=A(k+1:n,k+1:n)*U(k+1:n)';
  C=U(k+1:n)*V(k+1:n)';
  P(1:k)=zeros(1,k);
  P(k+1:n)=V(k+1:n)-C*U(k+1:n);
  A(k+2:n,k)=zeros(n-k-1,1);
  A(k,k+2:n)=zeros(1,n-k-1);
  A(k+1,k)=-s; A(k,k+1)=-s;
  A(k+1:n,k+1:n)=A(k+1:n,k+1:n)-2*U(k+1:n)'*P(k+1:n)-2*P(k+1:n)'*U(k
+1:n);
end
```