Fachbereich Mathematik
Prof. Dr. P. Spellucci
Wintersemester 2007/2008

TECHNISCHE
UNIVERSITÄT
DARMSTADT

17.12.2007

Numerik des Matrizeneigenwertproblems
Übung 9

Präsenzübung

Ü 25 In der Vorlesung wurde gezeigt, daß sich der Winkel φ einer Jacobirotation zur Annullierung des Elementes $a_{p,q},\ q > p$ einer reell symmetrischen Matrix aus

$$\cot(2\varphi) \ = \ \frac{a_{q,q} - a_{p,p}}{2a_{p,q}}$$

berechnen lässt. Zeigen Sie, daß man

$$c \ = \ \cos(\varphi) \quad \text{und} \quad s \ = \ \sin(\varphi)$$

ohne Berechnung von φ selbst berechnen kann, wobei $\varphi \ \leq \ \pi/4$ gewählt werden kann.
Hinweis: Bestimmen Sie zunächst $t \ = \ \tan(\varphi)$, dann daraus c und schliesslich s.

Ü 26 Zeigen Sie: Wird eine Jacobirotation auf eine symmetrische Matrix angewendet wie in Ü25, dann kann die Änderung der Diagonalelemente berechnet werden aus

$$a_{p,p} \ = \ a_{p,p} - ta_{p,q} \quad \text{und} \quad a_{q,q} \ = \ a_{q,q} + ta_{p,q}$$

mit t in der gleichen Bedeutung wie dort.

Ü 27 In der Vorlesung wurde gezeigt, daß sich die Frobeniusnorm der Matrix unter der Transformation mit einer Jacobirotation nicht ändert, während die Quadratsumme der Ausserdiagonalelemente um $2a_{p,q}^2$ abnimmt. Beweisen Sie die Konvergenz des Verfahrens unter der Festlegung

$$|a_{p,q}| \ \stackrel{!}{=} \ \max_{i,j}\{|a_{i,j}| \ : \ i \ \neq \ j\}$$

Hinweis: Ist

$$2\omega(A) \ = \ \sum_{i=1}^{n} \sum_{\substack{j \ = \ 1 \\ j \ \neq \ i}}^{n} |a_{i,j}|^2$$

dann soll gelten

$$\omega(A^{(k+1)}) \ \leq \ c\omega(A^{(k)})$$

mit einer Konstanten $0 \ < \ c \ < \ 1$.

Hausübung

H 25 Zeigen Sie: das Jacobi-Eigenwertverfahren konvergiert auch in der `treshold`-Variante:

```
while omega(A) > 0
  tresh=sqrt(omega)/(4*n);
  for i = 1:n
    for j=i+1:n
       if (abs(a(i,j))) > tresh
          rotate(i,j);
       end
    end
  end
end
```

H 26 Es sei eine Jacobirotation zur Annullierung von $a_{p,q}$ anzuwenden. Die Zeilen p und q der Matrix werden dann multipliziert mit der Matrix

$$\begin{pmatrix} c & -s \\ s & c \end{pmatrix}$$

und in einem zweiten Schritt die Spalten mit der Transponierten dieser Matrix. Zeigen Sie: statt die Rotation durch

$$\begin{aligned} a_{p,\cdot} &= ca_{p,\cdot} - sa_{q,\cdot} \\ a_{q,\cdot} &= sa_{p,\cdot} + ca_{q,\cdot} \end{aligned}$$

durchzuführen kann man auch rechnen

$$\begin{aligned} a_{p,\cdot} &= a_{p,\cdot} - s(a_{q,\cdot} + \tau a_{p,\cdot}) \\ a_{q,\cdot} &= a_{q,\cdot} + s(a_{p,\cdot} - \tau a_{q,\cdot}) \end{aligned}$$

mit

$$\tau = \tan(\varphi/2) = \frac{s}{1+c} .$$

H 27 Programmieren Sie das Jacobi-Verfahren unter Berücksichtigung der rechentechnischen Modifikationen aus diesem Übungsblatt in der `treshold`–Variante, wobei Sie in jedem "sweep" die Summe der Quadrate der Ausserdiagonalelemente neu berechnen und `tresh` wie in der Hausübung 25 setzen. Erproben Sie das Verfahren an geeigneten Matrizen, z.B. denjenigen, die Sie schon für die Methoden `rqi` und `RR` benutzt haben.

Numerik des Matrizeneigenwertproblems
Übung 9, Lösungsvorschlag

Präsenzübung

Ü 25 In der Vorlesung wurde gezeigt, daß sich der Winkel φ einer Jacobirotation zur Annullierung des Elementes $a_{p,q},\ q > p$ einer reell symmetrischen Matrix aus

$$\cot(2\varphi) \;=\; \frac{a_{q,q} - a_{p,p}}{2a_{p,q}}$$

berechnen lässt. Zeigen Sie, daß man

$$c \;=\; \cos(\varphi) \quad \text{und} \quad s \;=\; \sin(\varphi)$$

ohne Berechnung von φ selbst berechnen kann, wobei $\varphi \;\leq\; \pi/4$ gewählt werden kann. Hinweis: Bestimmen Sie zunächst $t \;=\; \tan(\varphi)$, dann daraus c und schliesslich s.

Es gilt

$$\cot(2\varphi) \;=\; \frac{\cos(2\varphi)}{\sin(2\varphi)} \;=\; \frac{\cos^2(\varphi) - \sin^2(\varphi)}{2\sin(\varphi)\cos(\varphi)}$$

und

$$\begin{aligned} 2\sin(\varphi)\cos(\varphi) \;&=\; 2\tan(\varphi)\cos^2(\varphi) \\ &=\; \frac{2\tan(\varphi)}{\frac{1}{\cos^2(\varphi)}} \\ &=\; \frac{2\tan(\varphi)}{\frac{\cos^2(\varphi)+\sin^2(\varphi)}{\cos^2(\varphi}} \\ &=\; \frac{2\tan(\varphi)}{1+\tan^2(\varphi)} \end{aligned}$$

Somit

$$\cot(2\varphi) \;=\; \frac{1+\tan^2(\varphi)}{2\tan(\varphi)}\frac{1-\tan^2(\varphi)}{1+\tan(\varphi)} \;=\; \frac{1-t^2}{2t}$$

Wir erhalten also mit

$$\zeta \;=\; \frac{a_{q,q} - a_{p,p}}{2a_{p,q}}$$

die quadratische Geichung in t

$$1 - t^2 - 2t\zeta \;=\; 0$$

mit den beiden Lösungen

$$t \;=\; \zeta \pm \sqrt{1+\zeta^2}$$

Da der Winkel $\varphi \;\leq\; \pi/4$ *erfüllen soll, muss*

$$|t| \;\leq\; 1$$

gelten und deshalb muss die betragkleinere der beiden Lösungen gewählt werden. Um diese aber numerisch stabil, d.h. ohne Stellenverlust, berechnen zu können, muss man die betragsgrössere Nullstelle berechnen und dann den Satz von Vieta benutzen:

$$t = \frac{sign(\zeta)}{|\zeta| + \sqrt{1+\zeta^2}}$$

Aus obiger Darstellung folgt

$$c^2 = \cos^2(\varphi) = \frac{1}{1+\tan^2(\varphi)}$$

also wegen $|\varphi| \le \pi/4$

$$c = \frac{1}{\sqrt{1+t^2}}$$

und schliesslich

$$s = tc\ .$$

Ü 26 Zeigen Sie: Wird eine Jacobirotation auf eine symmetrische Matrix angewendet wie in Ü25, dann kann die Änderung der Diagonalelemente berechnet werden aus

$$a_{p,p} = a_{p,p} - ta_{p,q} \quad \text{und} \quad a_{q,q} = a_{q,q} + ta_{p,q}$$

mit t in der gleichen Bedeutung wie dort.

Wir zeigen dies am 2×2 *Fall:*

$$\begin{pmatrix} c & -s \\ s & c \end{pmatrix} \begin{pmatrix} \alpha & \gamma \\ \gamma & \beta \end{pmatrix} \begin{pmatrix} c & s \\ -s & c \end{pmatrix} = \begin{pmatrix} \alpha c^2 - 2\gamma cs + \beta s^2 & (c^2-s^2)\gamma + (\alpha-\beta)sc \\ (c^2-s^2)\gamma + (\alpha-\beta)sc & \alpha s^2 + 2\gamma cs + \beta c^2 \end{pmatrix}$$

c und s sind so gewählt, daß das Ausserdiagonalelement null wird:

$$\frac{1-t^2}{t} = \frac{\beta-\alpha}{2\gamma}$$

Wir erhalten damit

$$\begin{aligned} \alpha c^2 - 2\gamma cs + \beta s^2 &= \alpha + (\beta-\alpha)s^2 - 2\gamma cs \\ &= \alpha + 2\gamma(\frac{1-t^2}{2t}s^2 - cs) \\ &= \alpha + 2\gamma cs(\frac{1-t^2}{2} - 1) \\ &= \alpha - \gamma(t^2+1)cs \\ &= \alpha - \gamma t \end{aligned}$$

und

$$\begin{aligned} \alpha s^2 + 2\gamma sc + \beta c^2 &= \beta + (\alpha-\beta)s^2 + 2\gamma sc \\ &= \beta + \gamma t \end{aligned}$$

Ü 27 In der Vorlesung wurde gezeigt, daß sich die Frobeniusnorm der Matrix unter der Transformation mit einer Jacobirotation nicht ändert, während die Quadratsumme der Ausserdiagonalelemente um $2a_{p,q}^2$ abnimmt. Beweisen Sie die Konvergenz des Verfahrens unter der Festlegung

$$|a_{p,q}| \stackrel{!}{=} \max_{i,j}\{|a_{i,j}| \ : \ i \neq j\}$$

Hinweis: Ist

$$2\omega(A) \ = \ \sum_{i=1}^{n} \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{i,j}|^2$$

dann soll gelten

$$\omega(A^{(k+1)}) \ \leq \ c\omega(A^{(k)})$$

mit einer Konstanten $0 \ < \ c \ < \ 1$.

Nach Definition von p und q ist

$$\tfrac{n^2-n}{2}|a_{p,q}^{(k)}|^2 \ \geq \ \omega(A^{(k)})$$

also

$$\omega(A^{(k+1)}) \ \leq \ \omega(A^{(k)})(1 - \frac{2}{n^2-n})$$

d.h. $\omega(A^{(k)})$ nimmt ab wie eine geometrische Folge und konvergiert daher gegen null. Somit ist jeder Häufungswert des Produktes der Rotationsmatrizen eine Eigenvektormatrix und jeder Häufungswert der entstehenden Diagonalmatrix enthält die Eigenwerte von A.

Hausübung

H 25 Zeigen Sie: das Jacobi-Eigenwertverfahren konvergiert auch in der `treshold`-Variante:

```
while omega(A) > 0
  tresh=sqrt(omega)/(4*n);
  for i = 1:n
    for j=i+1:n
       if (abs(a(i,j))) > tresh
          rotate(i,j);
       end
    end
  end
end
```

Weil

$$\omega(A) \;\leq\; \tfrac{n(n-1)}{2} \max\{|a_{i,j}|^2 \;:\; j > i\}$$

wird nach der Definition von `tresh` *zumindest ein Element, das größer oder gleich* `tresh` *ist, pro sweep annulliert. Ausserdem ist nach wie vor die Folge* $\omega(A^{(k)})$ *streng monoton fallend. Schlimmstenfalls reduziert sich somit die Konvergenzrate auf*

$$(1 - \frac{1}{4n^2})^{2/(n(n-1))} \;<\; 1$$

die Nullfolgeneigenschaft bleibt aber erhalten. Durch diese Vorgehensweise entfällt die aufwendige Buchhaltung und Neubestimmung des betragsmaximalen Ausserdiagonalelementes.

H 26 Es sei eine Jacobirotation zur Annullierung von $a_{p,q}$ anzuwenden. Die Zeilen p und q der Matrix werden dann multipliziert mit der Matrix

$$\begin{pmatrix} c & -s \\ s & c \end{pmatrix}$$

und in einem zweiten Schritt die Spalten mit der Transponierten dieser Matrix. Zeigen Sie: statt die Rotation durch

$$\begin{aligned} a_{p,.} &= ca_{p,.} - sa_{q,.} \\ a_{q,.} &= sa_{p,.} + ca_{q,.} \end{aligned}$$

durchzuführen kann man auch rechnen

$$\begin{aligned} a_{p,.} &= a_{p,.} - s(a_{q,.} + \tau a_{p,.}) \\ a_{q,.} &= a_{q,.} + s(a_{p,.} - \tau a_{q,.}) \end{aligned}$$

mit

$$\tau \;=\; \tan(\varphi/2) \;=\; \frac{s}{1+c} \;.$$

Diese Behauptung ist jedenfalls richtig, wenn

$$1 - s\tau \;=\; c\;.$$

Nun gilt wegen der Seztung von τ

$$1 - \frac{s^2}{1+c} \;=\; \frac{1+c-s^2}{1+c} \;=\; \frac{c^2+c}{1+c} \;=\; c\;,$$

womit schon alles gezeigt ist.

H 27 Programmieren Sie das Jacobi-Verfahren unter Berücksichtigung der rechentechnischen Modifikationen aus diesem Übungsblatt in der `treshold`–Variante, wobei Sie in jedem “sweep“ die Summe der Quadrate der Ausserdiagonalelemente neu berechnen und `tresh` wie in der Hausübung 25 setzen. Erproben Sie das Verfahren an geeigneten Matrizen, z.B. denjenigen, die Sie schon für die Methoden `rqi` und `RR` benutzt haben.

```
function [v,d,history,historyend,numrot]=jacobi(a_in,itermax)
% [v,d,history,historyend,numrot]=jacobi(a_in,itermax)
% computes the eigenvalues d and
% eigenvectors v of the real symmetric matrix a_in,
% using rutishausers modfications of the classical
% jacobi rotation method with treshold pivoting.
% history(1:historyend) is a column vector of the length of
% total sweeps  used containing the sum of squares of
% strict upper diagonal elements of a. a is not
% touched but copied locally
% the upper triangle is used only
% itermax is the maximum number of total sweeps allowed
% numrot is the number of rotations applied in total
% check arguments
 siz=size(a_in);
 if siz(1) ~= siz(2)
   error('jacobi : matrix must be square ' );
 end
 if norm(a_in-a_in',inf) ~= 0
   error('jacobi ; matrix must be symmetric ');
 end
 if ~isreal(a_in)
  error(' jacobi : valid for real matrices only');
 end
 n=siz(1);
 v=eye(n);
 a=a_in;
 history=zeros(itermax,1);
```

```
d=diag(a);
bw=d;
zw=zeros(n,1);
iter=0;
numrot=0;
while iter < itermax
  iter=iter+1;
  history(iter)=sqrt(sum(sum(triu(a,1).^2)));
  historyend=iter;
  tresh=history(iter)/(4*n);
  if tresh ==0
    return;
  end
  for p=1:n
    for q=p+1:n
       gapq=10*abs(a(p,q));
       termp=gapq+abs(d(p));
       termq=gapq+abs(d(q));
       if iter>4 & termp==abs(d(p)) & termq==abs(d(q))
       % annihilate tiny elements
          a(p,q)=0;
       else
         if abs(a(p,q)) >= tresh
         %apply rotation
            h=d(q)-d(p);
            term=abs(h)+gapq;
            if term == abs(h)
               t=a(p,q)/h;
            else
               theta=0.5*h/a(p,q);
               t=1/(abs(theta)+sqrt(1+theta^2));
               if theta < 0
                  t=-t;
               end
            end
            c=1/sqrt(1+t^2);
            s=t*c;
            tau=s/(1+c);
            h=t*a(p,q);
            zw(p)=zw(p)-h;  %accumulate corrections to diagonal elements
            zw(q)=zw(q)+h;
            d(p)=d(p)-h;
            d(q)=d(q)+h;
```

```
            a(p,q)=0;
            %rotate, use information from the upper triangle of a only
            for j=1:p-1
               g=a(j,p);
               h=a(j,q);
               a(j,p)=g-s*(h+g*tau);
               a(j,q)=h+s*(g-h*tau);
            end
            for j=p+1:q-1
               g=a(p,j);
               h=a(j,q);
               a(p,j)=g-s*(h+g*tau);
               a(j,q)=h+s*(g-h*tau);
            end
            for j=q+1:n
               g=a(p,j);
               h=a(q,j);
               a(p,j)=g-s*(h+g*tau);
               a(q,j)=h+s*(g-h*tau);
            end
            for j=1:n
              g=v(j,p);
              h=v(j,q);
              v(j,p)=g-s*(h+g*tau);
              v(j,q)=h+s*(g-h*tau);
            end
            numrot=numrot+1;
         end
       end %if
     end % for q
   end % for p
   bw=bw+zw;
   d=bw;
   zw=zeros(n,1);
end %while


%  [v,d,history,historyend,numrot]=jacobi(a_in,itermax)
%  computes the eigenvalues d and
%  eigenvectors v of the real symmetric matrix a_in,
%  using rutishausers modfications of the classical
%  jacobi rotation method with treshold pivoting.
%  history(1:historyend) is a column vector of the length of
```

```
%  total sweeps  used containing the sum of squares of
%  strict upper diagonal elements of a. a is not
%  touched but copied locally
%  itermax is the maximum number of total sweeps
%  check arguments

maxiter=100;
diary on
% 1. Beispiel
A=rosser;
disp('rosser');
[V,D,history,historyend,numrot]=jacobi(A,100);
disp('eigenvalues');
for i=1:length(D)
disp([num2str(i,'%4.4d'),' ',num2str(D(i),'%23.16e')]);
end
disp(['total number of rotations used :',num2str(numrot,'%10d')]);
disp('sequence  of sweeps');
disp(['step',' omega(A)  ']);
for i=1:historyend
disp([num2str(i,'%4.4d'),' ',num2str(history(i),'%12.5e')]);
end
disp(['norm(V*D*V^T-A)',' ',num2str(norm(V*diag(D)*V'-A),'% 12.5e')]);
disp('=================================================================');
disp('hankel([1;-4;5;6],[6,3,-9,1])');
A=hankel([1;-4;5;6],[6,3,-9,1]);
[V,D,history,historyend,numrot]=jacobi(A,100);
disp('eigenvalues');
for i=1:length(D)
disp([num2str(i,'%4.4d'),' ',num2str(D(i),'%23.16e')]);
end
disp(['total number of rotations used :',num2str(numrot,'%10d')]);
disp('sequence  of sweeps');
disp(['step',' omega(A)  ']);
for i=1:historyend
disp([num2str(i,'%4.4d'),' ',num2str(history(i),'%12.5e')]);
end
disp(['norm(V*D*V^T-A)',' ',num2str(norm(V*diag(D)*V'-A),'% 12.5e')]);
disp('=================================================================');
A=gallery('ris',6);
disp('ris(6)');
[V,D,history,historyend,numrot]=jacobi(A,100);
disp('eigenvalues');
```

```
for i=1:length(D)
disp([num2str(i,'%4.4d'),' ',num2str(D(i),'%23.16e')]);
end
disp(['total number of rotations used :',num2str(numrot,'%10d')]);
disp('sequence  of sweeps');
disp(['step',' omega(A)  ']);
for i=1:historyend
disp([num2str(i,'%4.4d'),' ',num2str(history(i),'%12.5e')]);
end
disp(['norm(V*D*V^T-A)',' ',num2str(norm(V*diag(D)*V'-A),'% 12.5e')]);
disp('===========================================================================');
A=pascal(10);
disp('pascal(10)');
[V,D,history,historyend,numrot]=jacobi(A,100);
disp('eigenvalues');
for i=1:length(D)
disp([num2str(i,'%4.4d'),' ',num2str(D(i),'%23.16e')]);
end
disp(['total number of rotations used :',num2str(numrot,'%10d')]);
disp('sequence  of sweeps');
disp(['step',' omega(A)  ']);
for i=1:historyend
disp([num2str(i,'%4.4d'),' ',num2str(history(i),'%12.5e')]);
end
disp(['norm(V*D*V^T-A)',' ',num2str(norm(V*diag(D)*V'-A),'% 12.5e')]);
disp('===========================================================================');
```

```
rosser
eigenvalues
??? Undefined function or variable 'd'.

Error in ==> <a href="error:/home/spellucci/matlab/jacobitest.m,20,0">jacobitest at
for i=1:length(d)

uiopen('/home/spellucci/matlab/jacobitest.m', true);
jacobitest
rosser
eigenvalues
0001
0002
0003
0004
```

```
0005
0006
0007
0008
total number of rotations used :0000000121
sequence  of sweeps
step omega(A)
0001 1.31170e+03
0002 6.56511e+02
0003 8.40796e+01
0004 7.92426e+00
0005 3.17893e-01
0006 1.45396e-01
0007 6.17436e-03
0008 8.27372e-05
0009 2.51673e-06
0010 9.21963e-09
0011 3.07194e-10
0012 1.23143e-11
0013 3.25480e-13
0014 8.58563e-15
0015 1.27647e-20
0016 2.61775e-23
0017 1.39102e-26
0018 0.00000e+00
norm(V*D*V^T-A) 6.43093e-13
====================================================================
hankel([1;-4;5;6],[6,3,-9,1])
eigenvalues
0001
0002
0003
0004
total number of rotations used :0000000021
sequence  of sweeps
step omega(A)
0001 1.42478e+01
0002 2.12228e+00
0003 1.07145e-02
0004 6.81612e-04
0005 8.49103e-06
0006 4.11829e-07
0007 1.47217e-08
```

```
0008 4.24770e-12
0009 2.13383e-13
0010 0.00000e+00
norm(V*D*V^T-A) 4.27307e-15
===================================================================
ris(6)
eigenvalues
0001
0002
0003
0004
0005
0006
total number of rotations used :0000000060
sequence  of sweeps
step omega(A)
0001 2.37015e+00
0002 8.72754e-01
0003 7.97298e-02
0004 5.45261e-03
0005 2.56469e-04
0006 9.72497e-06
0007 1.35305e-07
0008 4.87374e-09
0009 5.83492e-11
0010 5.68082e-13
0011 1.97498e-15
0012 3.48681e-17
0013 1.94548e-34
0014 0.00000e+00
norm(V*D*V^T-A) 9.37068e-16
===================================================================
pascal(10)
eigenvalues
0001
0002
0003
0004
0005
0006
0007
0008
0009
```

```
0010
total number of rotations used :0000000214
sequence  of sweeps
step omega(A)
0001 2.84220e+04
0002 1.28553e+03
0003 6.82890e+01
0004 5.38039e+00
0005 2.04152e+00
0006 1.98608e-01
0007 1.85450e-02
0008 1.64820e-03
0009 6.25114e-05
0010 8.55386e-07
0011 2.21476e-08
0012 6.44251e-10
0013 1.51523e-11
0014 3.72743e-13
0015 4.98154e-15
0016 1.19569e-16
0017 3.29292e-18
0018 2.88354e-21
0019 0.00000e+00
norm(V*D*V^T-A) 4.17057e-11
====================================================================
jacobitest
rosser
eigenvalues
0001 9.8048640721568783e-02
0002 1.0000000000000000e+03
0003 9.9999999999999989e+02
0004 1.0200490184299967e+03
0005 -1.0200490184299969e+03
0006 1.0200000000000001e+03
0007 -5.7125090019584102e-14
0008 1.0199019513592787e+03
total number of rotations used :121
sequence  of sweeps
step omega(A)
0001 1.31170e+03
0002 6.56511e+02
0003 8.40796e+01
0004 7.92426e+00
```

```
0005 3.17893e-01
0006 1.45396e-01
0007 6.17436e-03
0008 8.27372e-05
0009 2.51673e-06
0010 9.21963e-09
0011 3.07194e-10
0012 1.23143e-11
0013 3.25480e-13
0014 8.58563e-15
0015 1.27647e-20
0016 2.61775e-23
0017 1.39102e-26
0018 0.00000e+00
norm(V*D*V^T-A) 6.43093e-13
===================================================================
hankel([1;-4;5;6],[6,3,-9,1])
eigenvalues
0001 -1.4630747266227672e+01
0002 5.7422368626987863e+00
0003 1.2323494477365315e+01
0004 6.5650159261635670e+00
total number of rotations used :21
sequence  of sweeps
step omega(A)
0001 1.42478e+01
0002 2.12228e+00
0003 1.07145e-02
0004 6.81612e-04
0005 8.49103e-06
0006 4.11829e-07
0007 1.47217e-08
0008 4.24770e-12
0009 2.13383e-13
0010 0.00000e+00
norm(V*D*V^T-A) 4.27307e-15
===================================================================
ris(6)
eigenvalues
0001 1.5621803049948124e+00
0002 1.5707898035154457e+00
0003 7.0804607698542066e-01
0004 -1.5707962709288075e+00
```

```
0005 -1.5704741904113677e+00
0006 -1.4437572681670479e+00
total number of rotations used :60
sequence  of sweeps
step omega(A)
0001 2.37015e+00
0002 8.72754e-01
0003 7.97298e-02
0004 5.45261e-03
0005 2.56469e-04
0006 9.72497e-06
0007 1.35305e-07
0008 4.87374e-09
0009 5.83492e-11
0010 5.68082e-13
0011 1.97498e-15
0012 3.48681e-17
0013 1.94548e-34
0014 0.00000e+00
norm(V*D*V^T-A) 9.37068e-16
====================================================================
pascal(10)
eigenvalues
0001 5.2803793085887085e-01
0002 1.0485735682587949e-02
0003 9.8964981858692333e-02
0004 1.0104584280400505e+01
0005 1.5513283764036471e-05
0006 1.8938033454779930e+00
0007 6.1420880271248111e-04
0008 9.5367652806524362e+01
0009 1.6281108241796608e+03
0010 6.4460885017017448e+04
total number of rotations used :214
sequence  of sweeps
step omega(A)
0001 2.84220e+04
0002 1.28553e+03
0003 6.82890e+01
0004 5.38039e+00
0005 2.04152e+00
0006 1.98608e-01
0007 1.85450e-02
```

```
0008 1.64820e-03
0009 6.25114e-05
0010 8.55386e-07
0011 2.21476e-08
0012 6.44251e-10
0013 1.51523e-11
0014 3.72743e-13
0015 4.98154e-15
0016 1.19569e-16
0017 3.29292e-18
0018 2.88354e-21
0019 0.00000e+00
norm(V*D*V^T-A) 4.17057e-11
=====================================================================
diary on
jacobitest
rosser
eigenvalues
0001 9.8048640721568783e-02
0002 1.0000000000000000e+03
0003 9.9999999999999989e+02
0004 1.0200490184299967e+03
0005 -1.0200490184299969e+03
0006 1.0200000000000001e+03
0007 -5.7125090019584102e-14
0008 1.0199019513592787e+03
total number of rotations used :121
sequence  of sweeps
step omega(A)
0001 1.31170e+03
0002 6.56511e+02
0003 8.40796e+01
0004 7.92426e+00
0005 3.17893e-01
0006 1.45396e-01
0007 6.17436e-03
0008 8.27372e-05
0009 2.51673e-06
0010 9.21963e-09
0011 3.07194e-10
0012 1.23143e-11
0013 3.25480e-13
0014 8.58563e-15
```

```
0015 1.27647e-20
0016 2.61775e-23
0017 1.39102e-26
0018 0.00000e+00
norm(V*D*V^T-A) 6.43093e-13
====================================================================
hankel([1;-4;5;6],[6,3,-9,1])
eigenvalues
0001 -1.4630747266227672e+01
0002 5.7422368626987863e+00
0003 1.2323494477365315e+01
0004 6.5650159261635670e+00
total number of rotations used :21
sequence  of sweeps
step omega(A)
0001 1.42478e+01
0002 2.12228e+00
0003 1.07145e-02
0004 6.81612e-04
0005 8.49103e-06
0006 4.11829e-07
0007 1.47217e-08
0008 4.24770e-12
0009 2.13383e-13
0010 0.00000e+00
norm(V*D*V^T-A) 4.27307e-15
====================================================================
ris(6)
eigenvalues
0001 1.5621803049948124e+00
0002 1.5707898035154457e+00
0003 7.0804607698542066e-01
0004 -1.5707962709288075e+00
0005 -1.5704741904113677e+00
0006 -1.4437572681670479e+00
total number of rotations used :60
sequence  of sweeps
step omega(A)
0001 2.37015e+00
0002 8.72754e-01
0003 7.97298e-02
0004 5.45261e-03
0005 2.56469e-04
```

```
0006 9.72497e-06
0007 1.35305e-07
0008 4.87374e-09
0009 5.83492e-11
0010 5.68082e-13
0011 1.97498e-15
0012 3.48681e-17
0013 1.94548e-34
0014 0.00000e+00
norm(V*D*V^T-A) 9.37068e-16
=================================================================
pascal(10)
eigenvalues
0001 5.2803793085887085e-01
0002 1.0485735682587949e-02
0003 9.8964981858692333e-02
0004 1.0104584280400505e+01
0005 1.5513283764036471e-05
0006 1.8938033454779930e+00
0007 6.1420880271248111e-04
0008 9.5367652806524362e+01
0009 1.6281108241796608e+03
0010 6.4460885017017448e+04
total number of rotations used :214
sequence  of sweeps
step omega(A)
0001 2.84220e+04
0002 1.28553e+03
0003 6.82890e+01
0004 5.38039e+00
0005 2.04152e+00
0006 1.98608e-01
0007 1.85450e-02
0008 1.64820e-03
0009 6.25114e-05
0010 8.55386e-07
0011 2.21476e-08
0012 6.44251e-10
0013 1.51523e-11
0014 3.72743e-13
0015 4.98154e-15
0016 1.19569e-16
0017 3.29292e-18
```

```
0018 2.88354e-21
0019 0.00000e+00
norm(V*D*V^T-A) 4.17057e-11
=====================================================================
quit
```