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

TECHNISCHE UNIVERSITÄT DARMSTADT

19.11.2007

Numerik des Matrizeneigenwertproblems
Übung 5

Präsenzübung

Ü 13 Sei L eine untere $n \times n$ Dreiecksmatrix der Form

$$L \ = \ I + \sum_{j=1}^{n-1} \alpha_j e_{k(j)} e_j^T$$

mit

$$| \ \alpha_j \ | \ \leq \ 1 \text{ und } k(j) \ > \ j, \ k(1) \ \leq \ \dots \ k(n-1) \ .$$

Zeigen Sie

$$| \ (L^{-1})_{i,j} \ | \ \leq \ 1 \quad \forall \ i, j$$

Hinweis: Betrachten Sie das Produkt von Zeile r von L^{-1} mit den Spalten $1, \dots, r-1$ von L.

Ü 14 Es sei H eine nichtzerfallende obere Hessenbergmatrix und

$$H \ = \ QR$$

eine durch Anwendung von Givensreflektoren von links auf H erzeugte `QR`-Zerlegung, d.h. Q ist unitär und R eine obere Dreiecksmatrix. Zeigen Sie

1. Es gilt
$$|r_{i,i}| \ \geq \ |h_{i+1,i}| \ > \ 0 \ , \ i = 1, \dots, n-1 \ .$$
2. Q ist eine obere Hessenbergmatrix.
3. RQ ist eine obere Hessenbergmatrix und ähnlich zu H.

Hinweis: Beachten Sie, in welcher Reihenfolge Q aus den einzelnen Givensreflektoren zusammenmultipliziert wird.

Ü 15 Bestimmen Sie die vollständige Struktur des Eigensystems einer Rang–1–Matrix

$$A \ = \ uv^H \ \neq \ O \ .$$

Wie verhält sich eine Vektorfolge

$$A^k x_0$$

mit beliebigem x_0?
Hinweis: entwickeln Sie x_0 nach v und dem Orthogonalraum zu v.

Hausübung

H 13 Zeigen Sie: Ist R eine obere Dreiecksmatrix mit Diagonale $(1,\ldots,1)$ und $|r_{i,j}| \;\le\; 1$ für $i<j$, dann gilt für die Elemente der Inversen

$$|(R^{-1})_{i,j}| \;\le\; 2^{j-i-1}$$

Hinweis: verifizieren Sie zunächst die Block-Inversionsformel

$$\begin{pmatrix} R_{11} & R_{12} \\ O & R_{22} \end{pmatrix}^{-1} = \begin{pmatrix} R_{11}^{-1} & -R_{11}^{-1}R_{12}R_{22}^{-1} \\ O & R_{22}^{-1} \end{pmatrix}$$

Dabei sind die R_{ii} quadratische Matrizen. Wenden Sie diese nun rekursiv auf R an, beginnend "rechts unten" unter Hinzufügung je einer Zeile.

H 14 Gegeben sei

$$J = \begin{pmatrix} \lambda & 1 & & 0 \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ 0 & & & \lambda \end{pmatrix} \qquad (\text{" Jordan–Kästchen "}) \text{ mit } \;\lambda \;\ne\; 0$$

Zeigen Sie, daß die Vektorfolge

$$\begin{aligned} \tilde{x}_{k+1} &= Jx_k \\ x_{k+1} &= \frac{\tilde{x}_{k+1}}{|\tilde{\xi}_{1,k+1}|} \end{aligned}$$

mit $\tilde{x}_{k+1} = (\tilde{\xi}_{1,k+1}, ..., \tilde{\xi}_{n,k+1})^T$ (direkte Iteration mit Normierung der ersten Komponente) und $x_0 = (1,...,1)^T$ als Startvektor gegen einen (welchen ?) Eigenvektor x von J konvergiert und schätzen Sie die Fehlerreduktion

$$\frac{\|x_{k+1}-x\|_\infty}{\|x_k - x\|_\infty}$$

größenordnungsmäßig.
<u>Hinweis</u>: Zeigen Sie zunächst, daß

$$\begin{aligned} J^k &= \sum_{\nu=0}^{\min\{n-1,k\}} \lambda_1^{k-\nu}\binom{k}{\nu}T_\nu \qquad \text{mit} \qquad T_\nu = \left(\tau_{ij}^{(\nu)}\right), \\ \tau_{ij}^{(\nu)} &= \begin{cases} 1\,, & j=i+\nu \\ 0 & \text{sonst} \end{cases} \end{aligned}$$

H 15 Schreiben Sie ein MATLAB–Unterprogramm

```
function [lambda,V]=tridiewev(a,b) % [lambda,V,resnorm]=tridiewev(a,b)
% berechnet alle Eigenwerte lambda(i) der reellen symmetrischen
Dreibandmatrix
% T mit Diagonale a(1:n) und Nebendiagonale b(1:n-1)
% nach der Bisektionsmethode auf Maschinengenauigkeit und die
% zugehörigen Eigenvektoren durch Loesung der homogenen Gleichung
% mit Normierung auf Länge 1.
% Das System werde geloest als
% P(T-lambda(i)*I) =L*R
% R*V(:,i)= R(k,k)*(0,...,0,1)' and V(k,i)=1 if R(k,k)=0
% where R(k,k) is the first under the smallest pivots, seen from the
% bottom
```

und erproben Sie es an den mit der MATLAB-Routine `tridigen` erzeugten Matrizen. Die Routine finden Sie bei den Lösungen.

```
function [a,b]=tridigen(exam)
% [a,b]=tridigen(exam) erzeugt verschiedene symmetrische
% dreibandmatrizen. hier sind vorgesehen die faelle exam='exam1'
% bis 'exam4' gemaess UE15 WS2007/2008.
%
switch lower(exam)
    case {'exam1'}
        n=21;
        b=ones(n-1,1);
        for i=1:n
          a(i)=11-i;
        end
        %eigenwerte symmetrisch um null, nahe bei +/- i, i=1,10, und 0
    case {'exam2'}
        n=21;
        a=2*ones(n,1);
        b=-ones(n-1,1);
        % eigenwerte 2(1-cos(k*pi/22))  k=1,..,21
    case {'exam3'}
        n=21;
        b=ones(n,1);
        for i=1:n
            a(i)=abs(11-i);
        end
        % extrem dicht liegende eigenwertpaare aehnlich zu exam1 absolut
    case {'exam4'}
        n=20;
```

```
        a=zeros(n+1,1);
        for i=1:n
            b(i)=sqrt(i*(n-i+1));
        end
        % eigenwerte +(-)20, +(-)18,..,,+(-)2,0
    otherwise
        error('diese eingabe ist nicht definiert');
end
```

Berechnen Sie auch die Testgrösse `norm(V'*V-I)`. Entspricht das Resultat den theoretischen Vorhersagen?

Numerik des Matrizeneigenwertproblems
Übung 5, Lösungsvorschlag

Präsenzübung

Ü 13 Sei L eine untere $n \times n$ Dreiecksmatrix der Form

$$L \;=\; I + \sum_{j=1}^{n-1} \alpha_j e_{k(j)} e_j^T$$

mit

$$\mid \alpha_j \mid \;\leq\; 1 \text{ und } k(j) \;>\; j,\; k(1) \;\leq\; \ldots\; k(n-1)\;.$$

Zeigen Sie

$$\mid (L^{-1})_{i,j} \mid \;\leq\; 1 \quad \forall\, i,j$$

Hinweis: Betrachten Sie das Produkt von Zeile r von L^{-1} mit den Spalten $1,\ldots,r-1$ von L.

Nach Vorgabe ist

$$L_{j,j} \;=\; 1 \text{ und } L_{k(j),j} \;=\; \alpha_j$$

während alle anderen Elemente von L null sind. Also wird nach Hinweis

$$(L^{-1})_{r,j} L_{j,j} + (L^{-1}_{r,k(j)} L_{k(j),j} \;=\; 0$$

also

$$\mid (L^{-1})_{r,j} \mid \;\leq\; |\alpha_j| \mid (L^{-1}_{r,k(j)} \mid$$

Diese Rekursion kann man fortsetzen, bis $k(k(..k(j)) \;\geq\; r$ wird und erhält dann entweder ein Produkt von $\alpha's$ oder null als Resultat dieser Kette. Mit der Annahme für die α_j folgt die Behauptung.

Ü 14 Es sei H eine nichtzerfallende obere Hessenbergmatrix und

$$H \;=\; QR$$

eine durch Anwendung von Givensreflektoren von links auf H erzeugte `QR`-Zerlegung, d.h. Q ist unitär und R eine obere Dreiecksmatrix. Zeigen Sie

1. Es gilt
$$|r_{i,i}| \;\geq\; |h_{i+1,i}| \;>\; 0\;,\; i = 1,\ldots,n-1\;.$$
2. Q ist eine obere Hessenbergmatrix.
3. RQ ist eine obere Hessenbergmatrix und ähnlich zu H.

Hinweis: Beachten Sie, in welcher Reihenfolge Q aus den einzelnen Givensreflektoren zusammenmultipliziert wird.

1. *Es werden $n-1$ Givensreflektoren Ω_i von links angewendet, die dazu dienen, das Element $(i+1, i)$ von H in null zu überführen. Diese Reflektoren tangieren nur die Zeilen i und $i+1$ und ändern daher die darunter liegenden Zeilen und damit die Nebendiagonalelemente von H in den Zeilen $i+2$ folgende nicht. Weil jeder Givensreflektor unitär ist, gilt somit die erste Behauptung.*
2. *Nach der Überlegung unter 1. ist*

$$Q = \Omega_1 \cdots \Omega_{n-1}$$

Um die Struktur dieses Produktes zu ermitteln, beginnen wir mit Ω_1 und multiplizieren nun dies von rechts mit Ω_2. Dies betrifft also Spalte 2 und 3 von Ω_1. Weil diese Spalten unterhalb des Elementes 3 null sind, wird nur ein Element ungleich null in der Position (3,2) eingeführt, die Elemente oberhalb der Diagonale interessieren uns nicht, sie werden natürlich mit jeder weiteren Reflektion aufgefüllt. Der Vorgang stezt sich nun induktiv fort mit Spalte 3 und 4 usw., so daß offensichtlich Q eine obere Hessenbergmatrix wird.

3. *Nun haben wir das Produkt einer oberen Dreiecksmatrix (das ist eine (0,n-1)–Bandmatrix) mit einer oberen Hessenbergmatrix von rechts, (das ist eine (1,n-1)–Bandmatrix) und nach den Regeln für das Rechnen mit Bandmatrizen wird das Produkt eine $(0+1, \min\{n-1, 2n-2\})$–Bandmatrix, und dies ist eine obere Hessenbergmatrix. Ausserdem ist*

$$RQ \;=\; Q^T HQ$$

also unitär ähnlich zu H.

Ü 15 Bestimmen Sie die vollständige Struktur des Eigensystems einer Rang–1–Matrix

$$A \;=\; uv^H \;\neq\; O\;.$$

Wie verhält sich eine Vektorfolge

$$A^k x_0$$

mit beliebigem x_0?
Hinweis: entwickeln Sie x_0 nach v und dem Orthogonalraum zu v.

Es sei $\mathcal{W}$ Orthogonalraum zu v. Dann ist offenbar

$$Aw \;=\; 0 \text{ falls } w \;\in\; \mathcal{W}\;.$$

D.h. $\mathcal{W}$ ist invarianter Unterraum von A zum $n-1$–fachen Eigenwert null. Ferner

$$v^h u \;\neq\; 0 \;\Rightarrow\; Au \;=\; (uv^H)u \;=\; (v^H u)u$$

also ist u Eigenvektor zum Eigenwert $(v^H u)$ und A ist diagonalisierbar. Ist jedoch $v^H u = 0$, dann ist $u \in \mathcal{W}$, also auch wieder ein Eigenvektor zu null, aber A nicht diagonalähnlich. v ist dann ein Linkseigenvektor zum Eigenwert null. Schreiben wir

$$x_0 = \alpha u + Wa \ , \quad \text{mit } W \text{ als Basismatrix zu } \mathcal{W} \ ,$$

dann wird offenbar

$$A^k x_0 = (v^H u)^k \alpha u$$

d.h. alle diese Vektoren sind Vielfache eines Eigenvektors u von A.

Hausübung

H 13 Zeigen Sie: Ist R eine obere Dreiecksmatrix mit Diagonale $(1,\dots,1)$ und $|r_{i,j}| \;\le\; 1$ für $i<j$, dann gilt für die Elemente der Inversen

$$|(R^{-1})_{i,j}| \;\le\; 2^{j-i-1}$$

Hinweis: verifizieren Sie zunächst die Block-Inversionsformel

$$\begin{pmatrix} R_{11} & R_{12} \\ O & R_{22} \end{pmatrix}^{-1} = \begin{pmatrix} R_{11}^{-1} & -R_{11}^{-1}R_{12}R_{22}^{-1} \\ O & R_{22}^{-1} \end{pmatrix}$$

Dabei sind die R_{ii} quadratische Matrizen. Wenden Sie diese nun rekursiv auf R an, beginnend "rechts unten" unter Hinzufügung je einer Zeile.

Die Blockinversionsformel ist offensichtlich richtig, da die Multiplikation der beiden Matrizen die Einheitsmatrix ergibt. Die Behauptung ist auch offensichtlich richtig für den Fall $n=2$ mit Hilfe der Block-Inversionsformel:

$$\begin{pmatrix} 1 & r_{12} \\ 0 & 1 \end{pmatrix}^{-1} = \begin{pmatrix} 1 & -r_{12} \\ 0 & 1 \end{pmatrix}$$

Sei die Behauptung nun erfüllt für eine Matrix der Dimension $n-1$. Wir wählen R_{22} als solch eine Matrix und wenden wieder die Blockinversionsformel für eine solche Matrix der Dimension n an, d.h. $R_{11} \;=\; 1$ und $R_{12} \;=\; (r_{1,2},\dots,r_{1,n})$. Das Element $(1,1)$ der Inversen ist wieder 1 und für die Elemente 2 bis n der ersten Zeile erhalten wir den Zeilenvektor

$$-R_{12}(R_{22})^{-1}$$

und nach der Dreieckungleichung und der Induktionsannahme für die erste Zeile, also $i=1$, für $j \;\ge\; 3$

$$\begin{aligned} |(-R_{12}(R_{22})^{-1})_j| &\le 1+\sum_{k=1}^{j-2} 2^{j-k-2} \\ &= 1+1+\dots 2^{j-3} \\ &= 1+\frac{2^{j-2}-1}{2-1} \;=\; 2j-2 \;=\; 2^{j-1-1} \end{aligned}$$

während sich die Behauptung für $j=2$ wieder aus der Induktionsannahme ergibt. Man beachte die Indexverschiebung um 1, wenn man die Untermatrix R_{22} in R einbettet. Bei der speziellen Matrix mit $R_{i,i} \;=\; 1$ und $R_{i,j}=-1$ für $j>i$ werden die Schranken gerade angenommen, z.B. mit $n=10$

```
inv(R)



    1     1     2     4     8    16    32    64   128   256
    0     1     1     2     4     8    16    32    64   128
    0     0     1     1     2     4     8    16    32    64
    0     0     0     1     1     2     4     8    16    32
    0     0     0     0     1     1     2     4     8    16
    0     0     0     0     0     1     1     2     4     8
    0     0     0     0     0     0     1     1     2     4
    0     0     0     0     0     0     0     1     1     2
    0     0     0     0     0     0     0     0     1     1
    0     0     0     0     0     0     0     0     0     1
```

H 14 Gegeben sei

$$J = \begin{pmatrix} \lambda & 1 & & 0 \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ 0 & & & \lambda \end{pmatrix} \qquad (\text{“ Jordan–Kästchen ”}) \text{ mit } \lambda \neq 0$$

Zeigen Sie, daß die Vektorfolge

$$\begin{aligned} \tilde{x}_{k+1} &= J x_k \\ x_{k+1} &= \frac{\tilde{x}_{k+1}}{|\tilde{\xi}_{1,k+1}|} \end{aligned}$$

mit $\tilde{x}_{k+1} = (\tilde{\xi}_{1,k+1}, ..., \tilde{\xi}_{n,k+1})^T$ (direkte Iteration mit Normierung der ersten Komponente) und $x_0 = (1, ..., 1)^T$ als Startvektor gegen einen (welchen ?) Eigenvektor x von J konvergiert und schätzen Sie die Fehlerreduktion

$$\frac{\|x_{k+1} - x\|_\infty}{\|x_k - x\|_\infty}$$

größenordnungsmäßig.
<u>Hinweis</u>: Zeigen Sie zunächst, daß

$$\begin{aligned} J^k &= \sum_{\nu=0}^{\min\{n-1,k\}} \lambda_1^{k-\nu} \binom{k}{\nu} T_\nu \qquad \text{mit} \qquad T_\nu = \left(\tau_{ij}^{(\nu)}\right), \\ \tau_{ij}^{(\nu)} &= \begin{cases} 1\,, & j = i + \nu \\ 0 & \text{sonst} \end{cases} \end{aligned}$$

Die Zwischenbehauptung lautet in Komponentenschreibweise

$$(J^k)_{i,j} \;=\; \lambda^{k-(j-i)}\binom{k}{j-i} \quad \text{für } j \;\geq\; i$$

d.h.u.a. die Elemente auf jeder Nebendiagonale sind gleich. Daß J^k eine obere Dreiecksmatrix ist, ist klar und daß die Hauptdiagonalelemente λ^k lauten ebenfalls, wie man an der Formel für die Matrixmultiplikation ablesen kann. Wir betrachten nun ein allgemeines Element oberhalb der Diagonalen: Wieder nach der Formel für die Matrixmultiplikation "Zeile i mal Spalte j" folgt für $n \;\geq\; j \;>\; i$

$$\begin{aligned}
(J^{k+1})_{i,j} &= (JJ^k)_{i,j} \\
&= \lambda(J^k)_{i,j} + (J^k)_{i+1,j} \\
&= \lambda\lambda^{k-(j-i)}\binom{k}{j-i} + \lambda^{k-(j-i-1)}\binom{k}{j-i-1} \\
&= \lambda^{k+1-(j-i)}(\binom{k}{j-i} + \binom{k}{j-i-1}) \\
&= \lambda^{k+1-(j-i)}\binom{k+1}{j-i}
\end{aligned}$$

wie behauptet.
J hat nur einen einzigen Eigenvektor, nämlich ein Vielfaches des ersten Koordinateneinheitsvektors, mit der angegebenen Normierung also $x = e_1$. Wegen der Grenzwertbetrachtung können wir $k \;\geq\; n$ annehmen, sodaß J^k nun eine vollbesetzte obere Dreiecksmatrix ist. Da es irrelevant ist, ob wir die Normierung während der Rekursion immer von neuem oder nur abschliessend vornehmen, haben wir

$$\frac{(J^k x_0)_i}{(J^k x_0)_1} \;=\; \frac{\sum_{j=0}^{n-i} \lambda^{k-j}\binom{k}{j}}{\sum_{j=0}^{n-1} \lambda^{k-j}\binom{k}{j}}$$

und deshalb für den Fehler $||x^k - x||_\infty$

$$\max_{i=2,\ldots,n}\left|\frac{\sum_{j=0}^{n-i} \lambda^{-j}\binom{k}{j}}{\sum_{j=0}^{n-1} \lambda^{-j}\binom{k}{j}}\right|$$

während der Fehler in der ersten Komponente null ist. Die einzelnen Summanden in den Summen mit der Struktur

$$\frac{k \cdot (k-1) \cdot \ldots \cdot (k-j+1)}{\lambda^j \cdot 2 \cdot \ldots \cdot j}$$

wachsen für grosses k sehr schnell an, d.h. der Wert der Summe wird im wesentlichen durch den letzten Summanden bestimmt, sodass der grösste Fehler schliesslich für $i = 2$ auftritt mit einer Grössenordnung von $1/k$ (wie im Skript behauptet). Die Fehlerreduktion verschlechtert sich also immer mehr und der Grenzwert liegt bei 1.

H 15 Schreiben Sie ein `MATLAB`–Unterprogramm

```
function [lambda,V]=tridiewev(a,b) % [lambda,V,resnorm]=tridiewev(a,b)
% berechnet alle Eigenwerte lambda(i) der reellen symmetrischen
Dreibandmatrix
% T mit Diagonale a(1:n) und Nebendiagonale b(1:n-1)
% nach der Bisektionsmethode auf Maschinengenauigkeit und die
% zugehörigen Eigenvektoren durch Loesung der homogenen Gleichung
% mit Normierung auf Länge 1.
% Das System werde geloest als
% P(T-lambda(i)*I) =L*R
% R*V(:,i)= R(k,k)*(0,...,0,1)' and V(k,i)=1 if R(k,k)=0
% where R(k,k) is the first under the smallest pivots, seen from the
% bottom
```

und erproben Sie es an den mit der `MATLAB`-Routine `tridigen` erzeugten Matrizen. Die Routine finden Sie bei den Lösungen.

```
function [a,b]=tridigen(exam)
% [a,b]=tridigen(exam) erzeugt verschiedene symmetrische
% dreibandmatrizen. hier sind vorgesehen die faelle exam='exam1'
% bis 'exam4' gemaess UE15 WS2007/2008.
%
switch lower(exam)
    case {'exam1'}
        n=21;
        b=ones(n-1,1);
        for i=1:n
          a(i)=11-i;
        end
        %eigenwerte symmetrisch um null, nahe bei +/- i, i=1,10, und 0
    case {'exam2'}
        n=21;
        a=2*ones(n,1);
        b=-ones(n-1,1);
        % eigenwerte 2(1-cos(k*pi/22))  k=1,..,21
    case {'exam3'}
        n=21;
        b=ones(n,1);
        for i=1:n
            a(i)=abs(11-i);
        end
        % extrem dicht liegende eigenwertpaare aehnlich zu exam1 absolut
    case {'exam4'}
```

```
        n=20;
        a=zeros(n+1,1);
        for i=1:n
            b(i)=sqrt(i*(n-i+1));
        end
        % eigenwerte +(-)20, +(-)18,..,,+(-)2,0
    otherwise
        error('diese eingabe ist nicht definiert');
end
```

Berechnen Sie auch die Testgrösse `norm(V'*V-I)`. Entspricht das Resultat den theoretischen Vorhersagen?

```
function [lambda,V]=tridiewev(a,b)
% [lambda,V]=tridiewev(a,b);
% computes all eigenvalues lambda and the corresponding
% eigenvectors
% V(:,i) of a symmetric unreduced tridiagonal matrix
% with main diagonal a and subdiagonal b
% using bisection and a rudimentary form a invers iteration
% namely solving
% P(T-lambda(i)*I) =L*R
% R*V(:,i)= R(k,k)*(0,...,0,1)' and V(k,i)=1 if R(k,k)=0
% where R(k,k) is the first under the smallest pivot, seen from the
% bottom
%
%
  n=length(a);
  lambda=zeros(n,1);
  V=zeros(n);
  b(n)=0;
  for i=1:n-1
      if b(i)==0
          error('tridiewev: subdiagonal element=0');
      end
  end
  siz=size(a);
  if siz(1)==1
      a=a';
  end
  siz=size(b);
  if siz(1)==1
      b=b';
  end
```

```
q=zeros(n,1);
A=zeros(n,4);
% gerschgorin circles
low=min([a(1)-abs(b(1)),min(a(2:n-1)-abs(b(1:n-2))-abs(b(2:n-1))), ...
        a(n)-abs(b(n-1))]);
up=max([a(1)+abs(b(1)),max(a(2:n-1)+abs(b(1:n-2))+abs(b(2:n-1))), ...
        a(n)+abs(b(n-1))]);
% iteration
for i=1:n
    %determine lambda(i)
    if i>1
        low=lambda(i-1);
    end
    lowl=low;
    upl=up;
    while abs(upl-lowl) > eps*(max(abs(upl),abs(lowl))+eps)
        mu=(lowl+upl)/2;
        count=0;
        q(1)=a(1)-mu;
        if q(1)==0
            q(1)=eps;
        end
        if q(1) <0
            count=1;
        end
        for k=2:n
            q(k)=a(k)-mu-b(k-1)^2/q(k-1);
            if k<n && q(k)==0
                q(k)=eps;
            end
            if q(k) < 0
                count=count+1;
            end
        end
        if count < i
            lowl=mu;
        else
            upl=mu;
        end
    end
    lambda(i)=mu;
    % set matrix A, banded storage scheme
    A(1,1)=0; % undefined value
```

```
A(2:n,1)=b(1:n-1);
A(1:n,2)=a-mu;
A(1:n-1,3)=b(1:n-1);
A(n,3)=0;
A(1:n,4)=zeros(n,1);
%additional  superdiagonal in
% case of row interchange
% elimination with pivots for a tridiagonal,
% written in full form
% for i=1:n-1
%  if abs(A(i+1,i)) > abs(A(i,i))
%    for k=0:2
%       term=A(i+1,i+k);
%       A(i+1,i+k)=A(i,i+k);
%       A(i,i+k)=term;
%    end
%   end
%   A(i+1,i)=A(i+1,i)/A(i,i);
%   for k=1:2
%     A(i+1,i+k)=A(i+1,i+k)-A(i+1,i)*A(i,i+k);
%   end
% end
% indextransformation A(i,j)->  A(i,2-(i-j)) , j=i-1,i,i+1,i+2
for j=1:n-1
    if abs(A(j+1,1)) > abs(A(j,2))
       for k=0:2
           term=A(j+1,k+1);
           A(j+1,k+1)=A(j,k+2);
           A(j,k+2)=term;
       end
   end
   A(j+1,1)=A(j+1,1)/A(j,2);
   for k=1:2
       A(j+1,k+1)=A(j+1,k+1)-A(j+1,1)*A(j,k+2);
   end
end
% upper triangular matrix contained in the three columns
% A(i,j),j=2,4
k=n;
pivmin=abs(A(n,2));
for j=n-1:-1:1
    if abs(A(j,2)) < pivmin
        k=j;
```

```
                pivmin=abs(A(j,2));
            end
        end
        V(k,i)=1;
        if k>1
          V(k-1,i)=-A(k-1,3)/A(k-1,2);
        end
        for j=k-2:-1:1
           V(j,i)=-(A(j,3)*V(j+1,i)+A(j,4)*V(j+2,i))/A(j,2);
        end
        V(:,i)=V(:,i)/norm(V(:,i));
    end


[a,b]=tridigen('exam1');
[lambda,V]=tridiewev(a,b);
lambda=lambda';
lambda

lambda =

  Columns 1 through 4

 -10.74619418290336  -9.21067864733305  -8.03894111930644  -7.00395200266536

  Columns 5 through 8

  -6.00022568018517  -5.00000815867295  -4.00000020507044  -3.00000000380813

  Columns 9 through 12

  -2.00000000005449  -1.00000000000062   0.00000000000000   1.00000000000062

  Columns 13 through 16

   2.00000000005449   3.00000000380813   4.00000020507044   5.00000815867295

  Columns 17 through 20

   6.00022568018517   7.00395200266536   8.03894111930644   9.21067864733305

  Column 21
```

```
  10.74619418290336

norm(V'*V-eye(21))

ans =

     1.386166867156835e-15

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


[a,b]=tridigen('exam2');
[lambda,V]=tridiewev(a,b);
lambda=lambda';
lambda

lambda =

  Columns 1 through 4

   0.02035711623813   0.08101405277101   0.18073600929096   0.31749293433764

  Columns 5 through 8

   0.48850085129148   0.69027853210943   0.91871836508880   1.16916997399623

  Columns 9 through 12

   1.43653488631714   1.71537032345343   2.00000000000000   2.28462967654657

  Columns 13 through 16

   2.56346511368286   2.83083002600377   3.08128163491119   3.30972146789057

  Columns 17 through 20

   3.51149914870852   3.68250706566236   3.81926399070904   3.91898594722899

  Column 21

   3.97964288376187

norm(V'*V-eye(21))
```

```
ans =

     4.808286486673760e-15

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++


[a,b]=tridigen('exam3');
[lambda,V]=tridiewev(a,b);
lambda=lambda';
lambda

lambda =

  Columns 1 through 4

  -1.12544152211998   0.25380581709668   0.94753436752929   1.78932135269508

  Columns 5 through 8

   2.13020921936251   2.96105888418573   3.04309929257882   3.99604820138362

  Columns 9 through 12

   4.00435402344086   4.99978247774290   5.00024442500191   6.00021752225710

  Columns 13 through 16

   6.00023403158417   7.00395179861638   7.00395220952868   8.03894111581427

  Columns 17 through 20

   8.03894112282902   9.21067864730492   9.21067864736133  10.74619418290332

  Column 21

  10.74619418290339

norm(V'*V-eye(21))

ans =
```

```
   0.00703590714035

++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

[a,b]=tridigen('exam4');
[lambda,V]=tridiewev(a,b);
lambda=lambda';
lambda

lambda =

  Columns 1 through 4

 -20.00000000000000 -18.00000000000000 -16.00000000000000 -14.00000000000000

  Columns 5 through 8

 -12.00000000000000 -10.00000000000000  -8.00000000000000  -6.00000000000000

  Columns 9 through 12

  -4.00000000000000  -2.00000000000000   0.00000000000000   2.00000000000000

  Columns 13 through 16

   4.00000000000000   6.00000000000000   8.00000000000000  10.00000000000000

  Columns 17 through 20

  12.00000000000000  14.00000000000000  16.00000000000000  18.00000000000000

  Column 21

  20.00000000000000

norm(V'*V-eye(21))

ans =

     2.329262734588907e-15
```

Das Resulat entspricht genau den erwarteten Effekten: Im Fall der extrem dicht liegenden Eigenwerte erhält man nur sehr ungenaue Eigenvektoren, in allen anderen Fällen ist das

Eigenvektorsystem bis auf Maschinengeanuigkeit unitär. Die Eigenwerte werde nstest auf Maschinengenauigkeit bestimmt.