Rozwiązania zadań numerycznych z zestawu z dnia 20.11.2013r.

Autor: Mariusz Adamczyk

Komentarz: w zadaniach wykorzystywano funkcję eig() zwracająca wartości własne oraz wektory własne macierzy., tzn. zamiast implementować rozwiązywania układów równań oraz szukać wartości własnych „na piechotę” tam gdzie w treści nie zaznaczono, aby dokonać własnej implementacji wykorzystano funkcję eig().
A' - oznacza transpozycje macierzy A

Zad: 8N

```
%zad 8N zestaw z 20.11.2013r.
%autor: Mariusz Adamczyk,
%ostatnia modyfikacja: 7.12.2013r.
%-------------------------------------------------
clear all
clc

A = [ 19/12  13/12  5/6  5/6   13/12  -17/12;
      13/12  13/12  5/6  5/6  -11/12   13/12;
       5/6    5/6   5/6  -1/6   5/6     5/6;
       5/6    5/6  -1/6   5/6   5/6     5/6;
     13/12  -11/12  5/6   5/6   13/12  13/12;
     -17/12 13/12   5/6   5/6   13/12  19/12];

Y1 = [1 1 1 1 1 1]'; %dowolny o normie 1
zk = Y1';

%iter = 80;
precyzja = 0.0001;       %unormowana różnica medzy 2 kolejnymi
przyblizeniami
Yk = Y1;
licznik = 1;
%1sza ww
%for i = 1 : iter
while 1
   zkPop = zk;
   zk = A*Yk;
   Yk = zk/norm(zk);
   licznik=licznik+1;
   if rem(licznik,2) ~= 0;       %2 if dla spelniniea prezyzji, bo '
zmienne
        zk = zk';
   end
   if abs(norm(zkPop - zk')) < precyzja
       break;
   end
end
warW1 = norm(zk);
WekW1 = Y1;
WekW2 = Y1;

[V,D] = eig(A);      %liczy w. własne i odpowiadające im wektory
for i = 1 : length(A) %wybór wektora własnego odpowiadającego znalezionej
przez mnie w. własnej
```

```
    if sum(D(:,i)) > (warW1 - 0.01)
        WekW1 = V(:,i);
        break;
    end
end


licznik = 1;
%2ga ww
%for i = 1 : iter
while 1
   zkPop = zk;
   zk = A*Yk;
   zk = zk - WekW1*(WekW1'*zk);
   Yk = zk/norm(zk);
   licznik=licznik+1;
   if rem(licznik,2) == 0;
        zk = zk';
   end
   if abs(norm(zkPop - zk')) < precyzja
       break;
   end
end

warW2 = norm(zk);
for i = 1 : length(A) %wybór wektora własnego odpowiadającego znalezionej
przez mnie w. własnej
    if sum(D(:,i)) > (warW2 - 0.01)
        WekW2 = V(:,i);
        break;
    end
end
%koniec zad8N Mariusz Adamczyk
%------------------------------------------------------
```

Wejście:

A =

1.5833	1.0833	0.8333	0.8333	1.0833	-1.4167
1.0833	1.0833	0.8333	0.8333	-0.9167	1.0833
0.8333	0.8333	0.8333	-0.1667	0.8333	0.8333
0.8333	0.8333	-0.1667	0.8333	0.8333	0.8333
1.0833	-0.9167	0.8333	0.8333	1.0833	1.0833

-1.4167   1.0833   0.8333   0.8333   1.0833   1.5833

Obliczenia są wykonywane tak długo, aż zmiana między kolejnymi wektorami jest mniejsza od żądanej precyzji.

Wyniki:

warW1 = 4.0000

warW2 = 3.0000

Powyższe wartości własne znaleziono implementując metodę potęgową. Odpowiadające im wektory własne znaleziono wykorzystując funkcję eig() – nie implementowano rozwiązywania układu równań typu AX = 0:

wekWł1 =
0,408248290463863

0,408248290463863

0,408248290463863

0,408248290463863

0,408248290463863

0,408248290463863

wekWł2 =

0,707106781186547

7,23834180561173e-17

-1,23395830893539e-16

-1,37273618701354e-16

-1,70152606688119e-16

-0,707106781186548

Zad: 9N

```
%zad 9N zestaw z 20.11.2013r.
%autor: Mariusz Adamczyk,
%ostatnia modyfikacja: 8.12.2013r.
%-----------------------------------------------
clear all
clc

A = [ 19/12  13/12  5/6  5/6   13/12  -17/12;
      13/12  13/12  5/6  5/6  -11/12   13/12;
       5/6    5/6   5/6  -1/6   5/6     5/6;
       5/6    5/6  -1/6   5/6   5/6     5/6;
     13/12  -11/12  5/6   5/6   13/12  13/12;
     -17/12 13/12   5/6   5/6   13/12  19/12];

 Awej = A;

% A = [1 2 -1;
%      1 4 5;
%      1 4 1 ];

[V,D] = eig(A);     %liczy w. własne i odpowiadające im wektory dla
sprawdzenia

%najpierw Householder do macierzy trójkątnej

for i = 1:length(Awej)-1
%for i = 1:2
   a = A(:,i);
   n = 1;
   while n < i
      a(n) = 0;
      n = n+1;
   end
   e = zeros(length(A));
   e = e(:,1);
   e(i) = 1;
   v = a + norm(a)*e;
   b = v'*v;
   c = v*v';
   H = eye(length(A)) - 2*c/b;
   A = H*A;
   if i == 1
       Q_trans =H;             %Góra w4 slajd 8
   else
       Q_trans = H*Q_trans;
   end
end
%macierz A po hausholderze
Awej;
Q = Q_trans';
R = A;

%czyli mamy ortogonale Q

%Anew = Q'*Awej*Q;  = Q'*Q*R*Q   bo T = RQ
```

```
T = R*Q;
%tylko, że t nie jest trójkątna ????

[wekW ww] = eig(T);

%koniec zad9N Mariusz Adamczyk
%-------------------------------------------------------
```

Wejście:

Awej =

1.5833	1.0833	0.8333	0.8333	1.0833	-1.4167
1.0833	1.0833	0.8333	0.8333	-0.9167	1.0833
0.8333	0.8333	0.8333	-0.1667	0.8333	0.8333
0.8333	0.8333	-0.1667	0.8333	0.8333	0.8333
1.0833	-0.9167	0.8333	0.8333	1.0833	1.0833
-1.4167	1.0833	0.8333	0.8333	1.0833	1.5833

Wyniki:

Zastosowano transformację Hausholdera oraz faktoryzacje QR, otrzymano:
Q =

-0.5512	-0.3223	-0.0561	-0.0322	-0.2519	-0.7243
-0.3772	-0.3680	-0.1354	-0.0776	0.8156	0.1811
-0.2901	-0.2831	-0.2394	0.6778	-0.3377	0.4527
-0.2901	-0.2831	0.4640	-0.5491	-0.3377	0.4527
-0.3772	0.4943	-0.6506	-0.3730	-0.1351	0.1811
0.4932	-0.5967	-0.5317	-0.3048	-0.1584	0.0000

R =

-2.8723	-0.6093	-0.8704	-0.8704	-0.6093	0.2611
-0.0000	-2.3192	-0.8493	-0.8493	-0.5945	-0.8232
-0.0000	0.0000	-1.4217	-0.7183	-1.0303	-1.4266
-0.0000	0.0000	0.0000	-1.2269	-0.5907	-0.8179
-0.0000	0.0000	0.0000	0.0000	-1.9013	0.2805
-0.0000	0.0000	0.0000	-0.0000	-0.0000	2.1729

Q jest macierzą ortogonalną:
>> Q*Q'

ans =

1.0000	0	0.0000	0.0000	-0.0000	-0.0000
0	1.0000	-0.0000	0	0.0000	0.0000
0.0000	-0.0000	1.0000	-0.0000	0.0000	0.0000
0.0000	0	-0.0000	1.0000	0.0000	-0.0000
-0.0000	0.0000	0.0000	0.0000	1.0000	-0.0000
-0.0000	0.0000	0.0000	-0.0000	-0.0000	1.0000

zatem Q’AQ = T
powinno dać macierz trój diagonalną, ponadto ponieważ A = QR zatem T= RQ

T =

2.6768	1.1858	0.3058	0.1753	0.8555	1.0717
1.1858	1.5317	0.9477	0.5434	-1.1072	-1.2965
0.3058	0.9477	1.4359	0.2499	1.0878	-1.1553

0.1753	0.5434	0.2499	1.1433	0.6236	-0.6623
0.8555	-1.1072	1.0878	0.6236	0.2123	-0.3443
1.0717	-1.2965	-1.1553	-0.6623	-0.3443	0.0000

Nie otrzymano tak jak zakładano macierzy trójkątnej, jednak wykorzystując funkcję eig() znaleziono wartości i wektory własne macierzy T:

wartości własne (na przekątnej poniższej macierzy)=

-2.0000	0	0	0	0	0
0	-1.0000	0	0	0	0
0	0	4.0000	0	0	0
0	0	0	3.0000	0	0
0	0	0	0	2.0000	0
0	0	0	0	0	1.0000

Wektory własne (w kolejnych kolumnach wektory odpowiadające kolejnym wartościom własnym)=

-0.3482	0.1005	-0.5685	-0.7385	-0.0000	0.0000
0.5226	0.0981	-0.5548	0.1940	-0.6098	-0.0000
-0.0991	-0.5262	-0.4692	0.3363	0.3643	0.4974
-0.0568	-0.3017	-0.2690	0.1928	0.2089	-0.8675
0.5454	0.4679	-0.1654	-0.0661	0.6722	0.0000
0.5432	-0.6273	0.2218	-0.5121	-0.0000	-0.0000

Niestety nie otrzymano macierzy T jako trój diagonalnej, jednakże wartości własne T zgadzają się z wartościami własnymi macierzy wejściowej. Wektory własne nie zgadzają się z wektorami własnymi macierzy wejściowej.
Rozkład QR zaimplementowano poprawnie, sprawdzono, że Awej = Q*R

Awej - Q*R=

1.0e-015 *

0	-0.2220	0	0	-0.2220	-0.6661
0	-0.4441	-0.1110	-0.1110	0	-0.2220
0.1110	0	-0.2220	-0.2776	-0.2220	0.9992
0.1110	0	0.0555	0.1110	0.3331	0.6661
-0.2220	-0.3331	0.3331	0.4441	0.2220	0.6661
0	0.6661	0.3331	0.5551	0.8882	0

Zad: 10N

```
%zad 10N zestaw z 20.11.2013r.
%autor: Mariusz Adamczyk,
%ostatnia modyfikacja: 8.12.2013r.
%-----------------------------------------------
clear all
clc

H = [0 1 0 -i;
     1 0 -i 0;
     0 i 0 1;
     i 0 1 0];

%H = A + iB
A = H;
A(1,4) = 0;
A(2,3) = 0;
A(3,2) = 0;
A(4,1) = 0;

B = zeros(length(H));
B(1,4) = -1;
B(2,3) = -1;
B(3,2) = 1;
B(4,1) = 1;

H_ = [A -B;
      B A];

[V D] = eig(H_);
```

```
%usnięcie "klonów"

V_H = V(3:6,1:2:end);
D_H = D(1:2:end,1:2:end);
%koniec zad10N Mariusz Adamczyk
%-----------------------------------------------------
```

Wejście:
H =

0 + 0i	1+ 0i	0 + 0i	-0 - 1i
1 + 0i	0 + 0i	-0 - 1i	0 + 0i
0 + 0i	0 + 1i	0 + 0i	1 + 0i
0 + 1i	0 + 0i	1+ 0i	0 + 0i

Wyniki:

Odrzucono zdublowane (ponieważ implementowana metoda wymaga dwukrotnego zwiększania rozmiaru macierzy) wartości własne i wektory własne, otrzymano:

wartości własne (na przekątnej):

-2.0000	0	0	0
0	0	0	0
0	0	0.0000	0
0	0	0	2.0000

Wektory własne:

0.5000	0 .0000	0.7071	0.5000
-0.5000	-0.7071	0.0000	0.5000
-0.5000	0.0000	0.7071	-0.5000
0.5000	0.7071	-0.0000	-0.5000

Zad: 12N

```
%zad 12N zestaw z 20.11.2013r.
%autor: Mariusz Adamczyk,
%ostatnia modyfikacja: 8.12.2013r.
%-----------------------------------------------
clear all
clc

A = [2 -1 0 0 1;
     1 2 1 0 0;
     0 1 1 1 0;
     0 0 1 2 -1;
     1 0 0 -1 2];

 lam = 0.38197;

 B = lam*eye(length(A));
C = A - B;
%rozwiazac Cx = 0     x-wekWł

[U S V] = svd(C);
x = V(:,end);

%koniec zad12N Mariusz Adamczyk
%-------------------------------------------------------
```

Wejście:

A =

2	-1	0	0	1
1	2	1	0	0
0	1	1	1	0
0	0	1	2	-1
1	0	0	-1	2

lam =

0.3820

Wyniki:

Do rozwiązania równania A-lam*I = 0 wykorzystano gotowe funkcje.

Przybliżony wektor własny dla lam = 0.3820:

-0.3477

-0.3040

0.8174

-0.3165

0.1358