Rozwiązania zadań numerycznych z zestawu z dnia 6.11.2013r.

Autor: Mariusz Adamczyk

Zad: 10N

```
%zad 10N zestaw z 06.11.2013r.
%autor: Mariusz Adamczyk,
%ostatnia modyfikacja: 10.11.2013r.
%-----------------------------------------------
clear all
clc

A = eye(7)*4;
for i = 1 : 7
    A(i+1,i) = 1;
    A(i,i+1) = 1;
end
A = A(1:7,1:7);
B = 1 : 7;

%wynik z gotowych funkcji programu dla porównania
x_zProgramu = B*inv(A);
x_zProgramu = x_zProgramu';

%wektor x ma same jedynki, należy pamiętać,...
%że te jedynki to współczynniki
%przed zmiennymi x1, x2, ...
x = ones(7);
x = x(:,1);

%moja implementacja metody Gaussa
%bez pivoting'u
% i - nr wiersza
% j - nr kolumny
for j = 1: length(A) - 1
    for i = 1 + j : length(A)
        if(A(i,j) ~=0 )
            a = A(i,j)/A(j,j);
            C = A(j, :)*a;
            A(i, :) = A(i, :) - C;
            B(i) = B(i) - B(j) * a;
        else continue
        end
    end
end

%wyliczanie wektora x
for k = length(A) : -1 : 1
    x(k) = B(k)/A(k,k);
    A(1:k-1, k) = A(1:k-1, k) * x(k);
    B(1:k-1) = B(1:k-1) - A(1:k-1, k)';
end

%wyswietlenie wynikow
x
x_zProgramu
```

```
%porównie mojej funkcji
x - x_zProgramu

%koniec zad10N Mariusz Adamczyk
%-----------------------------------------------------
```

Dane wejściowe:

A =

4	1	0	0	0	0	0
1	4	1	0	0	0	0
0	1	4	1	0	0	0
0	0	1	4	1	0	0
0	0	0	1	4	1	0
0	0	0	0	1	4	1
0	0	0	0	0	1	4

B =

1 2 3 4 5 6 7

Uzyskane wyniki:

x =

0.1668

0.3328

0.5018

0.6598

0.8590

0.9043

1.5239

Wyniki otrzymane dzięki gotowym funkcjom:
x_zProgramu =

0.1668

0.3328

0.5018

0.6598

0.8590

0.9043

1.5239

Różnice między moimi wynikami a wynikami uzyskanym dzięki gotowym funkcjom:

ans =

1.0e-015 *

0.0278

-0.0555

0.1110

-0.2220

-0.2220

0.2220

0

Wyniki uzyskane dzięki mojej implementacji nie są identyczne z wynikami uzyskanymi dzięki gotowym funkcjom. Są jednak bardzo zbliżone. Związane jest to zapewne z procedurą odwracania macierzy. Warto zwrócić uwagę, że program Matlab dokonuje zaokrągleń na bardzo dalekich miejscach po przecinku, co może prowadzić do różnic w wynikach.

Zad: 12N

```
%zad 12N zestaw z 06.11.2013r.
%autor: Mariusz Adamczyk,
%ostatnia modyfikacja: 19.11.2013r.
%-----------------------------------------------
clear all
clc

%a, A_1, u, v jak na wykładzie tzn A - niezab, A_1 - zab
A = eye(7)*4;
for i = 1 : 7
    A(i+1,i) = 1;
    A(i,i+1) = 1;
end
A = A(1:7,1:7);
A(1,1) = A(1,1) - 1;
A(7,7) = A(7,7) - 1;
%A jest teraz niezaburzone
B = 1 : 7;

u = [ 1; 0; 0; 0; 0; 0; 1];
v = u;

%A_1 to macierz zaburzona
A_1 = A + u*v';

%wynik z gotowych funkcji programu dla porównania
x_zProgramu = B*inv(A_1);
x_zProgramu = x_zProgramu';

%wektor x ma same jedynki, należy pamiętać,...
%że te jedynki to współczynniki
%przed zmiennymi x1, x2, ...
%x = ones(7);
%x = x(:,1);

%wzor Shermana-Morissona (mozna wyk. got. alg. do rozw. ukła
niezaburzonego)
%zaburzenie to tylko A(1,7), A(7,1) - jak jeszcze od A(1,1) i A(7,7) -1 to
%jest ukła niezaburzony + "narożniki" = 1
%bez pivoting'u
%algorytm  - slajdy 22-24 z Wykładu 04 2012

%trochę zamieszane transpozycje, ale to tylko po to, żeby koncowo wynik
był
%pionowym wektorem
z = B*inv(A);       %czyli odwracamy tylko niezaburzona macierz
z = z';
q = u'*inv(A);
q = q';
%wyliczanie wektora x
x = z - v'*z*q/(1+v'*q);

%wyswietlenie wynikow
x
x_zProgramu
```

```
%porównie mojej funkcji
x - x_zProgramu

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

Dane wejściowe:

A = (macierz niezaburzona)

3	1	0	0	0	0	0
1	4	1	0	0	0	0
0	1	4	1	0	0	0
0	0	1	4	1	0	0
0	0	0	1	4	1	0
0	0	0	0	1	4	1
0	0	0	0	0	1	3

u = v =

1

0

0

0

0

0

1

A_1 = (macierz zaburzona)

4	1	0	0	0	0	1
1	4	1	0	0	0	0
0	1	4	1	0	0	0
0	0	1	4	1	0	0
0	0	0	1	4	1	0
0	0	0	0	1	4	1
1	0	0	0	0	1	4

B =

1	2	3	4	5	6	7

<u>Uzyskane wyniki:</u>

x =

-0.2602

0.4472

0.4715

0.6667

0.8618

0.8862

1.5935

Wyniki otrzymane dzięki gotowym funkcjom:

x_zProgramu =

-0.2602

0.4472

0.4715

0.6667

0.8618

0.8862

1.5935

Różnice między moimi wynikami a wynikami uzyskanym dzięki gotowym funkcjom:

ans =

1.0e-015 *

-0.0555

0.0555

0

0

0.1110

0.2220

0.4441

Wyniki uzyskane dzięki mojej implementacji nie są identyczne z wynikami uzyskanymi dzięki gotowym funkcjom. Są jednak bardzo zbliżone. Związane jest to zapewne z procedurą odwracania macierzy. Warto zwrócić uwagę, że program Matlab dokonuje zaokrągleń na bardzo dalekich miejscach po przecinku, co może prowadzić do różnic w wynikach.

Zad: 13N

```
%zad 13N zestaw z 06.11.2013r.
%autor: Mariusz Adamczyk,
%ostatnia modyfikacja: 11.11.2013r.
%------------------------------------------------
clear all
clc

%funkcja obliczająca współczynnik uwarunkowania macierzy A,
%definiowany jako cond(A) = ||A||?||A-1||, stanowiący
%współczynnik z jakim błędy wejściowe przenoszą się na wyjście
%podczas operacji macierzowej; im większa jest wartość
%współczynnika uwarunkowania macierzy tym większa jest jej
%wra?liwość na błędy zaokrągleń podczas wykonywania operacji
%arytmetycznych

A = [-116.66654   583.33346  -333.33308  100.00012  100.00012;
      583.33346  -116.66654  -333.33308  100.00012  100.00012;
     -333.33308  -333.33308   133.33383  200.00025  200.00025;
      100.00012   100.00012   200.00025  50.000025 -649.99988;
      100.00012   100.00012   200.00025 -649.99988  50.000025];

wspUwarunkowania_A = cond(A);

b1 = [-0.33388066;  1.08033290; -0.98559856; 1.31947922; -0.09473435];
b2 = [-0.33388066;  1.08033290; -0.98559855; 1.32655028; -0.10180541];
b3 = [ 0.72677951;  0.72677951; -0.27849178; 0.96592583;  0.96592583];
b4 = [ 0.73031505;  0.73031505; -0.27142071; 0.96946136;  0.96946136];



z1 = inv(A)*b1;
z2 = inv(A)*b2;
z3 = inv(A)*b3;
z4 = inv(A)*b4;

%długości wektorów
D_b1_minus_b2 = sqrt(sum((b1-b2).^2));
D_b3_minus_b4 = sqrt(sum((b3-b4).^2));

Il_1 = sqrt(sum((z1-z2).^2))/D_b1_minus_b2;
Il_2 = sqrt(sum((z3-z4).^2))/D_b3_minus_b4;

%układ źle uwarunkowany?

%koniec zad13N Mariusz Adamczyk
%-------------------------------------------------------
```

Dane wejściowe:

```
A = [-116.66654   583.33346  -333.33308  100.00012  100.00012;
      583.33346  -116.66654  -333.33308  100.00012  100.00012;
     -333.33308  -333.33308   133.33383  200.00025  200.00025;
      100.00012   100.00012   200.00025  50.000025 -649.99988;
      100.00012   100.00012   200.00025 -649.99988  50.000025];
```

```
b1 = [-0.33388066;  1.08033290; -0.98559856; 1.31947922; -0.09473435];
b2 = [-0.33388066;  1.08033290; -0.98559855; 1.32655028; -0.10180541];
b3 = [ 0.72677951;  0.72677951; -0.27849178; 0.96592583;  0.96592583];
b4 = [ 0.73031505;  0.73031505; -0.27142071; 0.96946136;  0.96946136];
```

Uzyskane wyniki:

z1 =

0,00196304996370100

-5,72551219644168e-05

-0,000259268641844557

0,000220741637443567

-0,00179956373674273

z2 =

0,00196562396121500

-5,46811244781509e-05

-0,000254120634252786

0,000233417158618465

-0,00180709124689400

z3 =

364,019900458131

364,019900458131

728,037635852779

364,018105821937

364,018105821937

z4 =

367,660091053431

367,660091053431

735,318017043367

367,658295893964

367,658295893965

Il_1 = ||z1 – z2||/||b1 – b2|| =

0.001603389293339

Il_2 = ||z3 – z4||/||b3 – b4|| =

1.029601026288040e+03

D_b1_minus_b2 = ||b1 – b2|| =

0.009999988952359

D_b3_minus_b4 = ||b3 – b4|| =

0.010000003094495

Współczynnik uwarunkowania macierzy A:

823680,896110094

Macierz jest źle uwarunkowana (ma wysoki współczynnik uwarunkowania), co za tym idzie otrzymujemy bardzo rozbieżne wyniki dla podobnych zestawów danych. Błąd szczególnie dobrze widoczny jest podczas dzielenia przez różnicę, ponieważ dla różnicy względny błąd obliczeń jest bardzo wysoki.
