WEEIA, POLITECHNIKA ŁÓDZKA

Metody numeryczne w inżynierii

Ćwiczenie Nr.2

Algebra Liniowa

5/16/2017

Prowadzący: dr. inż. Przemysław Mosiołek

Dawid Kędzierski 202182

Mayyar Karkout 204641

1. Eliminacja Gaussa bez wyboru elementu głównego.

```
function[L U] = gauss1(A);
n = size(A, 1);
L = eye(n);
U = zeros(n);
for i = 1:n
        for j = (i+1):n
        k = A(j, i) / A(i, i);
        L(j, i) = k;
        A(j, :) = A(j, :) - k.* A(i, :);
        end
end
U = A;
```

2. Rozwiązanie metodą Gaussa z częściowym wyborem el.głównego

```
function [C] = gauss2(A, B)
  n = size(A, 1);
  C = [A B];
  for i = 1:n
         [x y] = max(C(i:end, i));
           y = y + (i - 1);
           z = C(i, :);
          C(i, :) = C(y, :);
          C(y, :) = z;
                  for j = (i+1):n;
                     x = C(j, i) / C(i, i);
                    C(j, :) = C(j, :) - x .* C(i, :);
                 end
  end
                   A = C(:, 1:n);
  B = C(:, end);
           C = zeros(n, 1);
          for i = n:-1:1
          C(i) = B(i) ./ A(i, i);
          B(1:i) = B(1:i) - C(i) .* A(1:i, i);
  end
```

3. Czas operacji: A.A, A.b, A/b, det(A), inv(A), trace(A), eig(A) zależnie od rozmiaru macierzy:

```
function[ L U] = gaussczas(A);
n= size(A, 1);
U = zeros(n);
L = eye(n);
t1=tic;
for k=1:n;
   for l = (k+1 ):n;
      m=A(l, k) / A( k, k );
      L(l, k) = m;
      A(l, :) = A(l, : ) - m .*A(k, :);
   end
end
inv(A)
det(A)
trace(A)
eig(A)
A*A
t=toc(t1)
disp(t)
end
```

macierz 2x2: 0.0019531s
macierz 4x4: 0.0010819s
macierz 8x8: 0.0015650s
macierz 16x16: 0.0063651s
macierz 32x32: 0.64109s
macierz 64x64: 0.734124s.

- Jak można było się spodziewać – łączny czas operacji był największy dla największej macierzy.

- Najbardziej wymagającą operacją okazała się operacja eig(A), gdyż czas potrzebny do jej wykonania wraz ze wzrostem rozmiaru macierzy – jest kilkukrotnie razy większy niż dla pozostałych operacji. Zauważyliśmy również, że operacja trace(A), tj. jej czas jest praktycznie stały niezależnie od rozmiaru macierzy – z bardzo małymi różnicami.

Czas wykonywania się poszczególnych operacji (w sekundach):

	A*A	A*x	Det(A)	Inv(A)	A/b	Eig(A)
2x2	0.00000989914	0.0000103307	0.000049901	0.0000801492	0.000025301	0.0000300384
4x4	0.0000100517	0.0000100112	0.000170369	0.000250449	0.0000300407	0.0000400901
8x8	0.0000100398	0.0000100303	0.000327859	0.000641129	0.0000196791	0.0000804281
16x16	0.0000102806	0.0000098395	0.00052639	0.00101296	0.0000293899	0.000172238
32x32	0.000025301	0.0000103092	0.00125445	0.00285999	0.0000497913	0.000656731
64x64	0.000100179	0.0000106883	0.00281141	0.00682036	0.00012506	0.00351142

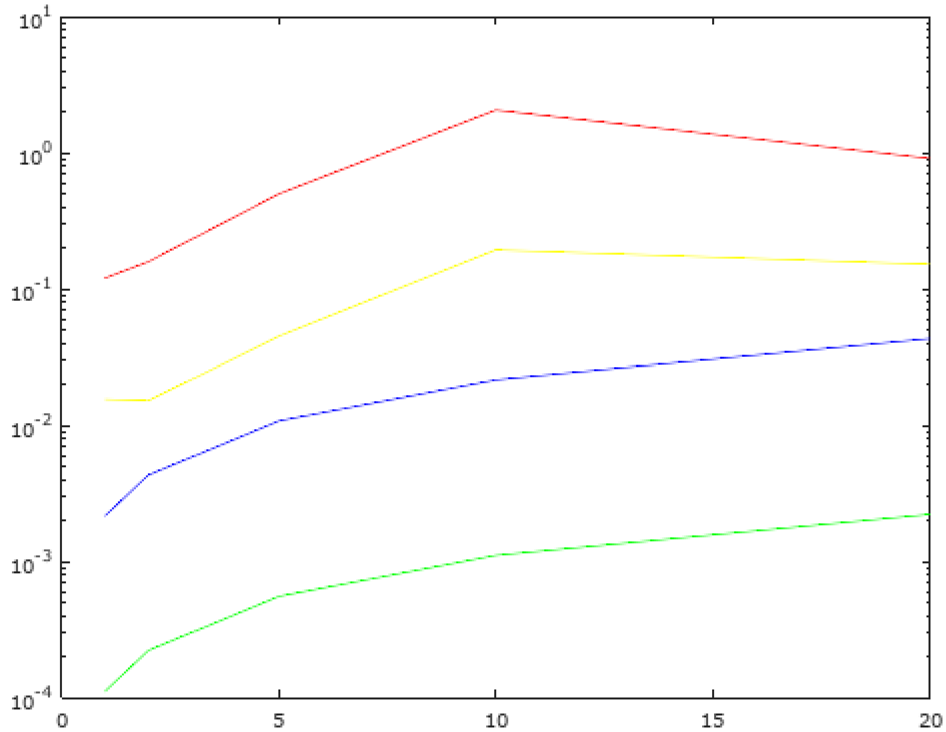
Wykres wielkości macierzy do czasu potrzebnego do wykonania obliczeń z nią związanych

4. Wpływ zaburzeń danych wejściowych na dokładność wyników.

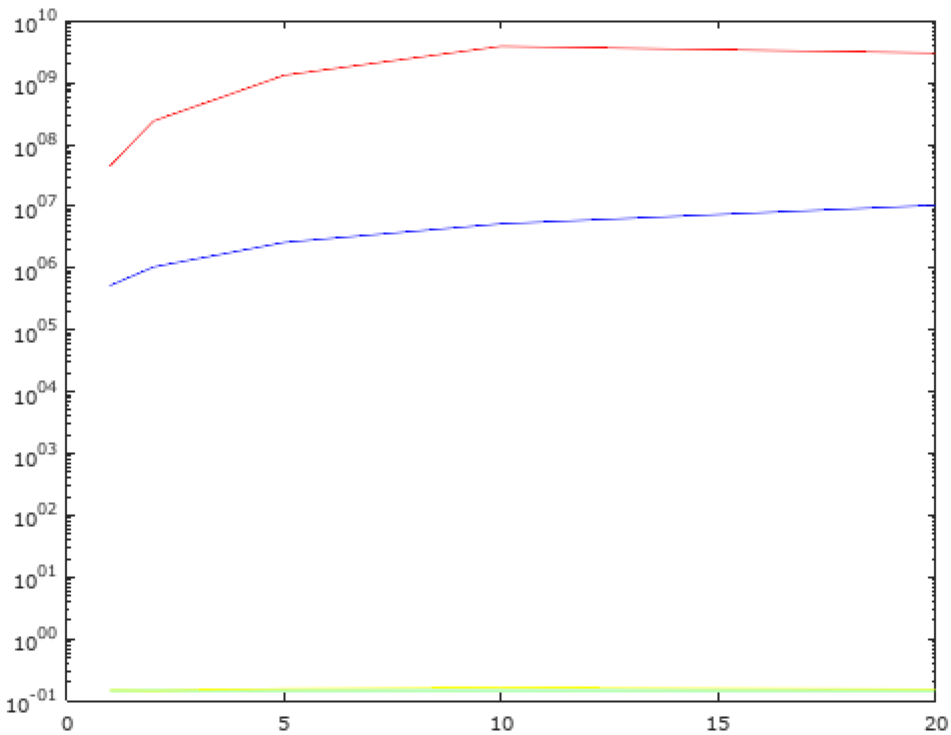
- zaburzenia, które wprowadziliśmy wynosiły: 1%, 2%, 5%, 10% oraz 20%.

tstmat.m – dobrze uwarunkowany problem
tstcnd.m - źle uwarunkowany problem

Dobrze uwarunkowany problem:



Źle uwarunkowany problem:



5. Listing m-pliku

```
function zab(n);

%[A b x] = tstmat(10);
[A b x] = tstcnd(10);
b_proc = [1 2 5 10 20];
%-------------------------
tabe = [];
for i = b_proc
  bn = b;
  bn(1) = bn(1) + bn(1) .* (i/100);
  xn = A \ bn;
  en = norm(x -xn) ./ norm(x);
  tabe = [tabe en];
end
semilogy(b_proc, tabe, 'b-');
hold on
%-------------------------
tabe = [];
for i = b_proc
  bn = b +(i./100) .* randn(10,1) .* norm(b);
  xn = A \ bn;
  en = norm(x - xn) ./ norm(x);
  tabe = [tabe en];
end
semilogy(b_proc, tabe, 'r-');
%-------------------------
tabe = [];
for i = b_proc
  An = A;
  An(1) = An(1) + An(1) .* (i/100);
  xn = An \ b;
  en = norm(x - xn) ./ norm(x);
  tabe = [tabe en];
end
semilogy(b_proc, tabe, 'g-');
%-------------------------
tabe = [];
for i = b_proc
  An = A;
  An(:,1) = An(:,1) + (i/100) .* randn(10,1) .* norm(A(:,1));
  xn = An \ b;
  en = norm(x - xn) ./ norm(x);
  tabe = [tabe en];
end
semilogy(b_proc, tabe, 'y-');
```