Metody numeryczne projekt nr 2

Kacper Wnek

11.01.2023

Contents

1 Wyznaczanie Rozkładu Crouta macierzy

1.1 Opis matematyczny

Dekompozycja Crouta jest metoda rozkładania macierzy na iloczyn dwóch macierzy dolnotrójkatnej L i górnotrójkatnej U.

$$A = L \cdot U$$

Polega ona na założeniu, że na głównej przekatnej macierzy górnej U wystepuja jedynki. Pozostałe elementy macierzy L i U wyznacza sie dla kolejnych wierszy od pierwszego do ostatniego wg zależności:

$$i, j \in 1, \ldots, n$$

$$u_{ii} = 1$$

$$l_{ij} = a_{ij} - \sum_{k=1}^{i} l_{ik} \cdot u_{kj}$$

$$u_{ij} = \frac{1}{l_{ii}}(a_{ij} - \sum_{k=i+1}^{n} l_{ik} \cdot u_{kj})$$

1.2 Opis programu

Program 'crout' jest funkcja służaca do dekompozycji macierzy kwadratowej zgodnie z metoda Crouta opisana w 1.1. Progam przyjmuje jeden argument:

- A- macierz kwadratowa do rozłożenia.

oraz zwraca:

- L - Macierz dolnotrójkatna,
- U - Macierz górnotrójkatna

```
function [L, U] = crout(A)
%Funkcja dokonuje dekompozycji kwadratowej macierzy A na macierz
%dolnotrójkątną i górnotrójkątną przy czym górnotrójkątna macierz ma na
%głównej diagonali jedynki
if size(A,1)~=size(A,2)
    error('A must be a square matrix')
end
[n, n] = size(A);
L=zeros(n,n);
U=eye(n);

        for i = 1:n
            L(i, 1) = A(i, 1);
        end
        for j = 2:n
            U(1, j) = A(1, j) / L(1, 1);
        end
        for i = 2:n
            for j = 2:i
                L(i, j) = A(i, j) - L(i, 1:j - 1) * U(1:j - 1, j);
            end

            for j = i + 1:n
                U(i, j) = (A(i, j) - L(i, 1:i - 1) * U(1:i - 1, j)) / L(i, i);
            end
        end
    end
```

Funkcja crout

1.3 Wykorzystanie rozkładu do rozwiazania równań macierzowych AX=B oraz XA=B

1.3.1 Równania w postaci Ax=B

Nastepujace równanie przekształcamy koszystajac z rozkładu:

$$A \cdot X = B$$

$$L \cdot U \cdot X = B$$

Nastepnie dokonujemy podstawienia:

$$U \cdot X = Y$$

Korzystajac z podstawienia rozwiazujemy równanie:

$$L \cdot Y = B$$

a nastepnie po obliczeniu Y wracamy do podstawienia i rozwiazujemy równanie:

$$U \cdot X = Y$$

W celu rozwiazania takiego równania wykorzystamy funkcje pomocnicze $solvecroutLYB$, $solvecroutUXY$ oraz funkcje $solvecroutLUX$ Wyżej wymienione funkcje działaja w nastepujacy sposób:

- $solvecroutLYB$

  Funkcja służy do rozwiazania równania $L \cdot Y = B$ gdzie L jest macierza dolnotrójkatna o wymiarach $n \times n$ a Y i B sa macierzami o wymiarach $n \times m$. Funkcja przyjmuje dane wejściowe:

  - $A := L$
  - $B := B$

  oraz zwraca:

  - Y

```
function [Y] = solve_crout_LYB(A,B)
%Funkcja służy do rozwiązania równania L*Y=B
% gdzie L jest macierzą dolnotrójkątną o wymiarach nxn
% a Y i B są macierzami o wymiarach nxm
% zwraca wartość Y
% dane wejściowe A->L B->B
if size(A,1)~=size(B,1)
    error('number of rows of A and B matrix must be equal')
end
[n,m]=size(B);
Y=zeros(n,m);

for i=1:n
     j=1:i-1;
     y=A(i,j)*Y(j,:);
     Y(i,:)=(B(i,:)-y)/A(i,i);

end
end
```

  Funkcja solve crout LYB

- $solvecroutUXY$

  Funkcja służy do rozwiazania równania $U \cdot X = Y$ gdzie U jest macierza górnotrójkatna o wymiarach $n \times n$ a X i Y sa macierzami o wymiarach $n \times m$. Funkcja przyjmuje dane wejściowe:

  - $A := U$
  - $B := Y$

  oraz zwraca:

  - X

```
function [X] = solve_crout_UXY(A,B)
%Funkcja służy do rozwiązania równania U*X=Y
% gdzie U jest macierzą górnotrójkątną o wymiarach nxn
% a X i Y są macierzami o wymiarach nxm
% zwraca wartość X
% dane wejściowe A->U B ->Y
if size(A,1)~=size(B,1)
    error('number of rows of A and B matrix must be equal')
end
[n,m]=size(B);
X=zeros(n,m);

for i=n:-1:1
     j=n:-1:i+1;
     x=A(i,j)*X(j,:);
     X(i,:)=(B(i,:)-x)/A(i,i);

end
end
```

Funkcja solve crout UXY

- *solvecroutLUX*

  Funkcja wykorzystuje funkcje *solvecroutUXY*, *solvecroutLYB* oraz *crout* do rozwiazania układu $A \cdot X = B$ poprzez dekompozycje metoda Crouta macierzy A

```
function [X] = solve_crout_LUX(A,B)
%Funkcja wykorzystuje funkcje solve_crout_UXY, solve_crout_LYB oraz crout
% do rozwiązania układu A*X=B poprzez dekompozycje metodą crouta macierzy A

% wykonujemy dekompozycje crouta macierzy A
[L,U] = crout(A);

% rozwiązujemy układ równań LY = B
Y = solve_crout_LYB(L, B);

% rozwiązujemy układ równań UX = Y
X = solve_crout_UXY(U, Y);

end
```

Funkcja solve crout LUX

1.3.2 Równania w postaci xA=B

Do rozwiazania równania w postaci $X \cdot A = B$ możemy skorzystać ze wzorów omówionych w poprzednim podpunkcie. Wystarczy dokonać matematycznych przekształceń.

$$X \cdot A = B$$

$$(X \cdot A)^T = B^T$$

$$A^T \cdot X^T = B^T$$

Można zauważyć, że po wykonanych przekształceniach możemy skorzystać z funkcji *solvecroutLUX* dla transponowanych macierzy i otrzymamy wtedy transponowana macierz X. W celu jasności kodu na dalszym etapie zdefiniujmy do tego funkcje *solvecroutXLU*

```
function [X] = solve_crout_XLU(A,B)
% Funkcja rozwiązująca równania xA=B poprzez funkcje solve_crout_LUX()
X=solve_crout_LUX(A',B')';
end
```

Funkcja solve crout XLU

1.3.3 Checkresult i Displayexample

Funkcje *Checkresult* oraz *Displayexample* beda służyły do wypisania rozwiazań równań $A \cdot X = B$, $X \cdot A = B$ oraz do przedstawienia w postaci tabeli nastepujacych własności:

- wskaźnik uwarunkowania macierzy $cond(A)$: $cond(A) = \|A^{-1}\| \cdot \|A\|$,
- błąd rozkładu $e_{dec} = \dfrac{\|A - BC\|}{\|A\|}$,
- błąd względny e_{rel}: jeśli z jest dokładnym rozwiązaniem układu $Ax = b$, a x obliczonym numerycznie przybliżeniem (otrzymanym naszym algorytmem), to

$$e_{rel} = \frac{\|x - z\|}{\|z\|},$$

- współczynnik stabilności wsp_{stab}: jeśli z jest dokładnym rozwiązaniem układu $Ax = b$, a x obliczonym numerycznie przybliżeniem (otrzymanym naszym algorytmem), to

$$wsp_{stab} = \frac{\|x - z\|}{\|z\|\, cond(A)},$$

- współczynnik poprawności wsp_{popr}: jeśli x jest obliczonym numerycznie przybliżeniem (otrzymanym naszym algorytmem), to

$$wsp_{popr} = \frac{\|b - Ax\|}{\|A\| \cdot \|x\|}.$$

Funkcje napisane sa w nastepujacy sposób:

- *Checkresult*

```
function [] = check_result(A,B)
%funkcja sprawdzająca poprawność przykładów dla Ax=B
%korzystająca z funkcji solve_crout_LUX, crout, oraz funkcji wbudowanych
% zwraca tabele z informacjami o wskazniku uwarunkowania, bledzie rozkladu,
% bledzie wzglednym, wspolczynniku stablinosci, wspolczynniku poprawnosci
% oraz czasy rozwiązywania równań Ax=B i xA=B (dla odpowiednio
% zmodyfikowanych jednak podobnych do siebie macierzy)
%przyjmuje macierz kwadratowa A o wymiarze n i macierz B o wymiarze nxm

X = solve_crout_LUX(A, B);
z = inv(A)*B;
[L,U]=crout(A);
wskaznik_uwar=cond(A);
blad_rozkladu= norm(A-L*U)/norm(A);
blad_wzgledny= norm(z-X)/norm(z);
wsp_stabilnosci= norm(X-z)/norm(z)*cond(A);
wsp_poprawnosci= norm(B-A*X)/norm(A)*norm(X);
g= @() solve_crout_LUX(A,B);
B=B';
f= @() solve_crout_XLU(A,B);
time_Ax=timeit(g);
time_xA=timeit(f);

tabela=table(wskaznik_uwar, blad_rozkladu, blad_wzgledny, ...
    wsp_stabilnosci, ...
    wsp_poprawnosci,time_Ax, time_xA, 'VariableNames', ...
    ["Wskaźnik uwarunkowania", "Błąd rozkładu", "Błąd względny", ...
    "Współczynnik stabilności", "Współczynnik poprawności", ...
    "czas rozwiązania Ax", "czas rozwiązania xA"])
end
```

- *Displayexample*

```
function [] = Display_example(A,B)
%Funkcja wykorzystująca funkcje solve_crout_LUX, wbudowaną funkcje inv
% oraz funkcje chceck result do zobrazowania wyników tych funkcji
X=solve_crout_LUX(A,B)
Y=inv(A)*B
B=B';
X_2=solve_crout_XLU(A,B)
Y_2=B*inv(A)
B=B';
check_result(A,B)



end
```

1.3.4 Przykłady

Działanie funkcji omówionych w poprzednim podpunkcie zobrazujemy na podstawie poniższych zróżnicowanych przykładów tj. gdy macierz A jest dużych rozmiarów, gdy macierz B jest macierza jednostkowa, gdy macierz B nie jest jednostkowa, jednak B=XA=AX, oraz dla ksiażkowych przypadków. Wyniki zwracane przez wywołanie funkcji *Displayexample* to

- X rozwiazanie równania $A \cdot X = B$ za pomoca metody $solvecroutLUX$
- Y rozwiazanie równania $A \cdot X = B$ za pomoca wbudowanej metody inv
- X_2 rozwiazanie równania $X \cdot A = B$ za pomoca metody $solvecroutXLU$
- Y_2 rozwiazanie równania $X \cdot A = B$ za pomoca wbudowanej metody inv

Naturalnie w przypadku równań $A \cdot X = B$ oraz $X \cdot A = B$ rozmiary macierzy sie różnia, wiec macierze nie sa sobie równe, ale sa one odpowiednio transponowane tak, żeby można było porównać rozwiazania 2 równań dla odpowiadajacych sobie macierzy.

```
%Przykład pierwszy
%AX=B
%XA=B
A = gallery("gcdmat",10);
B=randn(10,4);
Display_example(A,B);


%Przykład drugi B=I
%AX=B
%XA=B
fprintf("B jednostkowa")
A = gallery("gcdmat",10);
B=eye(10);
Display_example(A,B);


%Przykład trzeci B=I
%AX=B
%XA=B
fprintf("B jednostkowa")
A = [1,7,3;4,0,6;7,1,9];
B=eye(3);
Display_example(A,B);


%Przykład czwarty
%AX=B
%XA=B

A = [1,7,3;4,0,6;7,1,9];
B=[0,2;1,4;7,15];
Display_example(A,B);



%Przykład piąty B=AX=XA
%AX=B
%XA=B
fprintf("B=AX=XA")
A=[2,0;1,-1];
B=[6,0;2,0];
Display_example(A,B);


%Przykład szósty
%AX=B
%XA=B
A = gallery("gcdmat",5);
B=randn(5,4);
Display_example(A,B);

```

PRZYKŁAD I

X =

-1.6070	4.4113	-1.1432	11.9151
1.2647	0.2942	-0.5870	-7.8315
-0.2749	-0.7778	1.4167	-4.5975
-0.5538	-0.8687	0.7989	-0.0166
-0.5298	-0.8813	-0.0860	-1.6424
0.4945	-0.1859	-0.6290	3.0843
0.3229	-0.3773	0.1843	-0.3347
-0.0665	-0.2067	-0.0351	0.2572
0.0135	0.1492	-0.3792	0.0540
0.2466	0.2851	0.2593	1.1453

Y =

-1.6070	4.4113	-1.1432	11.9151
1.2647	0.2942	-0.5870	-7.8315
-0.2749	-0.7778	1.4167	-4.5975
-0.5538	-0.8687	0.7989	-0.0166
-0.5298	-0.8813	-0.0860	-1.6424
0.4945	-0.1859	-0.6290	3.0843
0.3229	-0.3773	0.1843	-0.3347
-0.0665	-0.2067	-0.0351	0.2572
0.0135	0.1492	-0.3792	0.0540
0.2466	0.2851	0.2593	1.1453

X_2 =

-1.6070	1.2647	-0.2749	-0.5538	-0.5298	0.4945	0.3229	-0.0665	0.0135	0.2466
4.4113	0.2942	-0.7778	-0.8687	-0.8813	-0.1859	-0.3773	-0.2067	0.1492	0.2851
-1.1432	-0.5870	1.4167	0.7989	-0.0860	-0.6290	0.1843	-0.0351	-0.3792	0.2593
11.9151	-7.8315	-4.5975	-0.0166	-1.6424	3.0843	-0.3347	0.2572	0.0540	1.1453

Y_2 =

-1.6070	1.2647	-0.2749	-0.5538	-0.5298	0.4945	0.3229	-0.0665	0.0135	0.2466
4.4113	0.2942	-0.7778	-0.8687	-0.8813	-0.1859	-0.3773	-0.2067	0.1492	0.2851
-1.1432	-0.5870	1.4167	0.7989	-0.0860	-0.6290	0.1843	-0.0351	-0.3792	0.2593
11.9151	-7.8315	-4.5975	-0.0166	-1.6424	3.0843	-0.3347	0.2572	0.0540	1.1453

TABELA

Wskaźnik uwarunkowania	Błąd rozkładu	Błąd względny	Współczynnik stabilności	Współczynnik poprawności	czas rozwiązania Ax	czas rozwiązania xA
109.47	0	1.8828e-16	2.0611e-14	3.3737e-16	0.00020005	0.00018747

PRZYKŁAD II

B jednostkowa
X =

3.6667	-1.7500	-1.0000	0	-0.5000	0.5000	-0.1667	0	0	0.2500
-1.7500	2.2500	0.5000	-0.5000	0.2500	-0.5000	0	0	0	-0.2500
-1.0000	0.5000	1.1667	0	0	-0.5000	0	0	-0.1667	0
0	-0.5000	0	0.7500	0	0	0	-0.2500	0	0
-0.5000	0.2500	0	0	0.5000	0	0	0	0	-0.2500
0.5000	-0.5000	-0.5000	0	0	0.5000	0	0	0	0
-0.1667	0	0	0	0	0	0.1667	0	0	0
0	0	0	-0.2500	0	0	0	0.2500	0	0
0	0	-0.1667	0	0	0	0	0	0.1667	0
0.2500	-0.2500	0	0	-0.2500	0	0	0	0	0.2500

Y =

3.6667	-1.7500	-1.0000	0	-0.5000	0.5000	-0.1667	0	0	0.2500
-1.7500	2.2500	0.5000	-0.5000	0.2500	-0.5000	0	0	0	-0.2500
-1.0000	0.5000	1.1667	0	0	-0.5000	0	0	-0.1667	0
0	-0.5000	0	0.7500	0	0	0	-0.2500	0	0
-0.5000	0.2500	0	0	0.5000	0	0	0	0	-0.2500
0.5000	-0.5000	-0.5000	0	0	0.5000	0	0	0	0
-0.1667	0	0	0	0	0	0.1667	0	0	0
0	0	0	-0.2500	0	0	0	0.2500	0	0
0	0	-0.1667	0	0	0	0	0	0.1667	0
0.2500	-0.2500	0	0	-0.2500	0	0	0	0	0.2500

X_2 =

3.6667	-1.7500	-1.0000	0	-0.5000	0.5000	-0.1667	0	0	0.2500
-1.7500	2.2500	0.5000	-0.5000	0.2500	-0.5000	0	0	0	-0.2500
-1.0000	0.5000	1.1667	0	0	-0.5000	0	0	-0.1667	0
0	-0.5000	0	0.7500	0	0	0	-0.2500	0	0
-0.5000	0.2500	0	0	0.5000	0	0	0	0	-0.2500
0.5000	-0.5000	-0.5000	0	0	0.5000	0	0	0	0
-0.1667	0	0	0	0	0	0.1667	0	0	0
0	0	0	-0.2500	0	0	0	0.2500	0	0
0	0	-0.1667	0	0	0	0	0	0.1667	0
0.2500	-0.2500	0	0	-0.2500	0	0	0	0	0.2500

Y_2 =

3.6667	-1.7500	-1.0000	0	-0.5000	0.5000	-0.1667	0	0	0.2500
-1.7500	2.2500	0.5000	-0.5000	0.2500	-0.5000	0	0	0	-0.2500
-1.0000	0.5000	1.1667	0	0	-0.5000	0	0	-0.1667	0
0	-0.5000	0	0.7500	0	0	0	-0.2500	0	0
-0.5000	0.2500	0	0	0.5000	0	0	0	0	-0.2500
0.5000	-0.5000	-0.5000	0	0	0.5000	0	0	0	0
-0.1667	0	0	0	0	0	0.1667	0	0	0
0	0	0	-0.2500	0	0	0	0.2500	0	0
0	0	-0.1667	0	0	0	0	0	0.1667	0
0.2500	-0.2500	0	0	-0.2500	0	0	0	0	0.2500

TABELA

Wskaźnik uwarunkowania	Błąd rozkładu	Błąd względny	Współczynnik stabilności	Współczynnik poprawności	czas rozwiązania Ax	czas rozwiązania xA
109.47	0	2.0599e-17	2.255e-15	2.1224e-16	0.00019844	0.00019747

PRZYKŁAD III

```
B jednostkowa
X =

   -0.1250   -1.2500    0.8750
    0.1250   -0.2500    0.1250
    0.0833    1.0000   -0.5833


Y =

   -0.1250   -1.2500    0.8750
    0.1250   -0.2500    0.1250
    0.0833    1.0000   -0.5833


X_2 =

   -0.1250   -1.2500    0.8750
    0.1250   -0.2500    0.1250
    0.0833    1.0000   -0.5833


Y_2 =

   -0.1250   -1.2500    0.8750
    0.1250   -0.2500    0.1250
    0.0833    1.0000   -0.5833
```

TABELA

Wskaźnik uwarunkowania	Błąd rozkładu	Błąd względny	Współczynnik stabilności	Współczynnik poprawności	czas rozwiązania Ax	czas rozwiązania xA
27.275	0	2.0918e-16	5.7053e-15	1.261e-16	2.2651e-05	2.5236e-05

PRZYKŁAD IV

```
X =

    4.8750    7.8750
    0.6250    1.1250
   -3.0833   -4.5833


Y =

    4.8750    7.8750
    0.6250    1.1250
   -3.0833   -4.5833


X_2 =

    0.7083    6.7500   -3.9583
    1.5000   11.5000   -6.5000


Y_2 =

    0.7083    6.7500   -3.9583
    1.5000   11.5000   -6.5000
```

TABELA

Wskaźnik uwarunkowania	Błąd rozkładu	Błąd względny	Współczynnik stabilności	Współczynnik poprawności	czas rozwiązania Ax	czas rozwiązania xA
27.275	0	1.1295e-16	3.0808e-15	8.2244e-15	2.1557e-05	2.2546e-05

PRZYKŁAD V

```
B=AX=XA
X =

     3     0
     1     0


Y =

     3     0
     1     0


X_2 =

     4    -2
     0     0


Y_2 =

     4    -2
     0     0
```

TABELA

Wskaźnik uwarunkowania	Błąd rozkładu	Błąd względny	Współczynnik stabilności	Współczynnik poprawności	czas rozwiązania Ax	czas rozwiązania xA
2.618	0	0	0	0	1.0409e-05	1.1444e-05

PRZYKŁAD VI

```
X =

   -2.8847   -0.3609    2.3104   -3.1117
    1.8985   -0.6232   -1.8445    1.1175
    0.4999    0.8253   -0.4358   -0.0472
   -0.2234    0.0318    0.6459    0.2194
    0.2010   -0.3321   -0.0193    0.5518


Y =

   -2.8847   -0.3609    2.3104   -3.1117
    1.8985   -0.6232   -1.8445    1.1175
    0.4999    0.8253   -0.4358   -0.0472
   -0.2234    0.0318    0.6459    0.2194
    0.2010   -0.3321   -0.0193    0.5518


X_2 =

   -2.8847    1.8985    0.4999   -0.2234    0.2010
   -0.3609   -0.6232    0.8253    0.0318   -0.3321
    2.3104   -1.8445   -0.4358    0.6459   -0.0193
   -3.1117    1.1175   -0.0472    0.2194    0.5518


Y_2 =

   -2.8847    1.8985    0.4999   -0.2234    0.2010
   -0.3609   -0.6232    0.8253    0.0318   -0.3321
    2.3104   -1.8445   -0.4358    0.6459   -0.0193
   -3.1117    1.1175   -0.0472    0.2194    0.5518
```

TABELA

Wskaźnik uwarunkowania	Błąd rozkładu	Błąd względny	Współczynnik stabilności	Współczynnik poprawności	czas rozwiązania Ax	czas rozwiązania xA
26.46	0	3.9493e-17	1.045e-15	5.7334e-16	5.2284e-05	5.6031e-05

2 Porównanie równań AX=B oraz XA=B

2.1 Porównanie czasu rozwiazywania równań AX=B oraz XA=B

Do tego porównania skorzystamy z funkcji *timecomparing*, która przyjmuje dwie macierze A i B oraz zwraca czas działania funkcji *solvecroutLUX* oraz *solvecroutXLU*

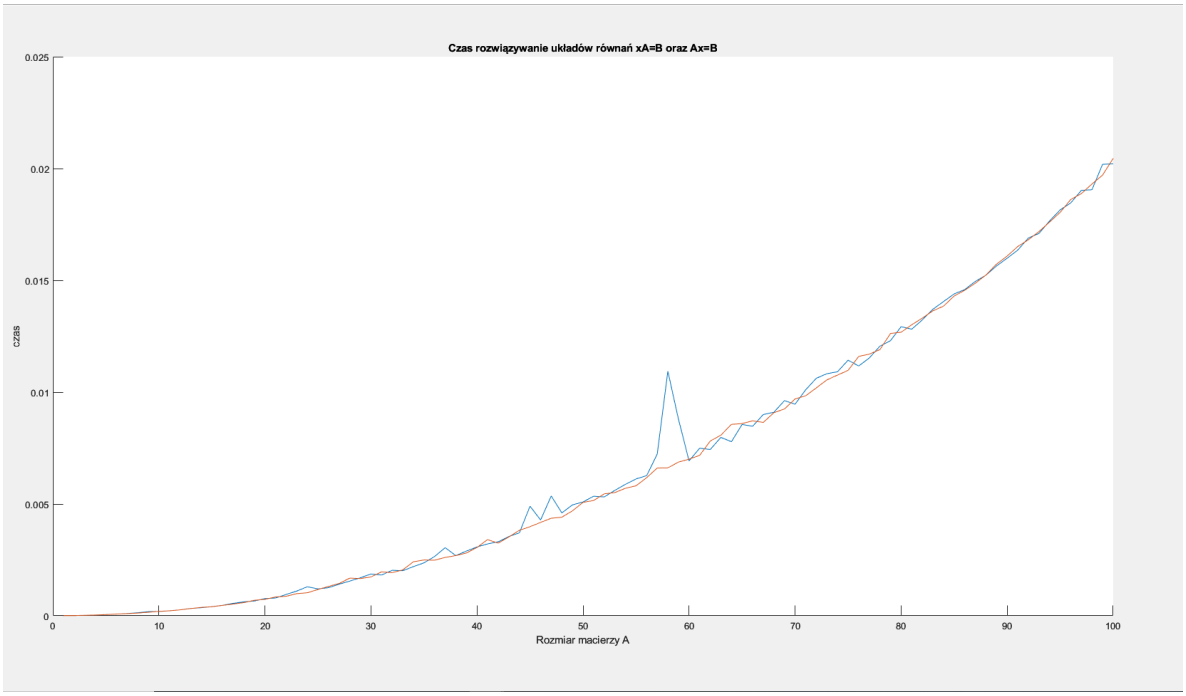
```
function [ time_Ax, time_xA] = time_comparing(A,B)
g= @() solve_crout_LUX(A,B);
B=B';
f= @() solve_crout_XLU(A,B);
time_Ax=timeit(g);
time_xA=timeit(f);

end
```

PORÓWNANIE CZASU W ZALEŻNOŚCI OD ROZMIARU MACIERZY A

Z wykorzystaniem nastepujacego kodu porównamy czas rozwiazania obu równań dla rosnacego rozmiaru macierzy A:

```
%poniższy kod rysuje czasa rozwiązywania równań Ax=B i xA=B dla
%rozmiarów macierzy A [1,100]
y=repelem(0,100);
x=1:100;
for i=x
    A=gallery("gcdmat",i);
    B=randn(i,2);
    y(i)=time_comparing(A,B);
end
hold on
plot(x,y)
title("Czas rozwiązywanie układów równań xA=B oraz Ax=B")
xlabel("Rozmiar macierzy A")
ylabel("czas")
hold off
```

Otrzymamamy wykres:



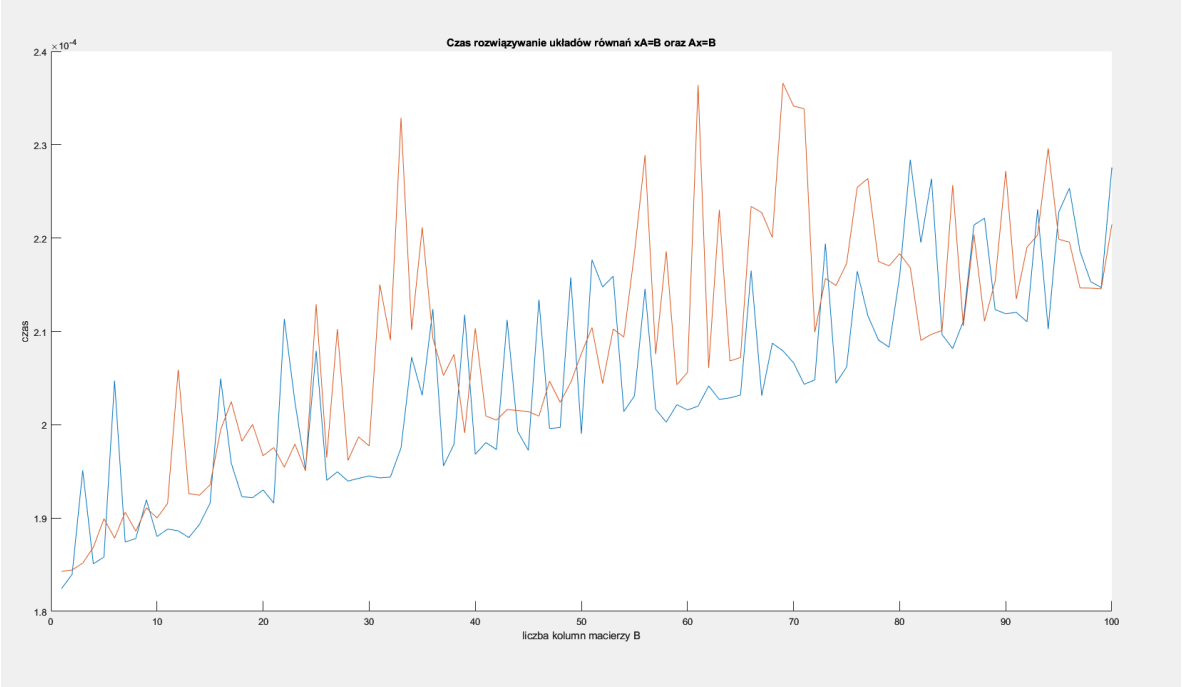
Gdzie niebieski wykres przedstawia czas dla $A \cdot X = B$, a czerwony dla $X \cdot A = B$.
Możemy zauważyć, że czas rozwiazywania jest zbliżony, jednak dla równań $X \cdot A = B$ jest on bardziej stabilny; W przypadku wykresu $A \cdot X = B$ możemy zauważyć, że dla niektórych rozmiarów macierzy czas rozwiazania sie znaczaco (oczywiście nie dla ludzkiego oka) wydłuża.

PORÓWNANIE CZASU W ZALEŻNOŚCI OD LICZBY KOLUMN|WIERSZY MACIERZY B

Analogicznie za pomoca poniższego kodu zbadamy różnice czasu gdy modyfikowana jest odpowiednio liczba kolumn wierszy macierzy B.

```
166  %poniższy kod rysuje roznice czasu rozwiązywania równań Ax=B i xA=B dla
167  %liczby kolumn\wierszy macierzy B[1,100]
168  y=repelem(0,100);
169  x=1:100;
170  for i=x
171      A=gallery("gcdmat",10);
172      B=randn(size(A,1),i);
173      y(i)= time_comparing(A,B);
174  end
175  hold on
176  plot(x,y)
177  title("Czas rozwiązywanie układów równań xA=B oraz Ax=B")
178  xlabel("liczba kolumn macierzy B")
179  ylabel("czas")
180  hold off
```

Otrzymamy wykres:



Gdzie ponownie niebieski wykres przedstawia czas dla $A \cdot X = B$ a czerwony dla $X \cdot A = B$.
Na wykresie wyraźnie widać, że w przypadku jednakowych rozmiarów n macierzy A oraz macierzy B o odpowienio rozmiarach $n \times m$ i $m \times n$ czas rozwiazywania równania (ponownie raczej niewidoczny dla ludzkiego oka) dla równania $X \cdot A = B$ jest zazwyczaj wiekszy chociaż nie zachodzi tak zawsze.

2.2 Zależność błedu wzglednego dla równań AX=B oraz XA=B

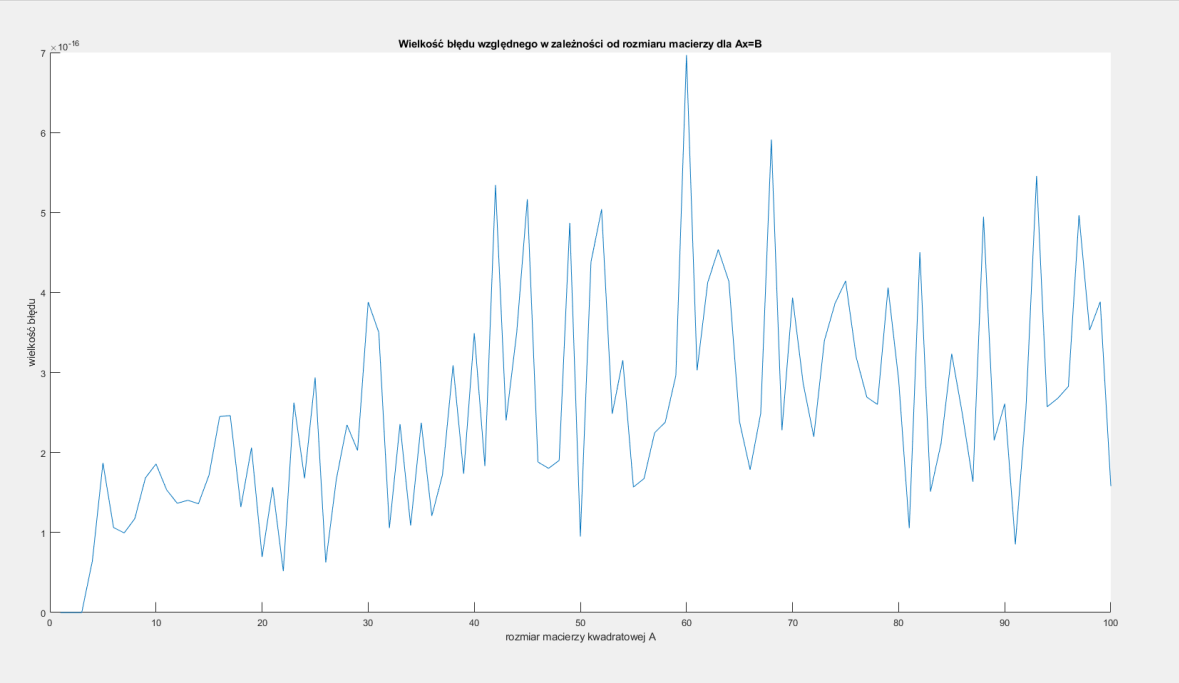
2.2.1 Rozmiar macierzy A

Poniżej znajduje sie kod, za pomoca którego można zobaczyć jak zmienia sie błąd wzgledny w zależności od rozmiaru macierzy A dla równania $A \cdot X = B$

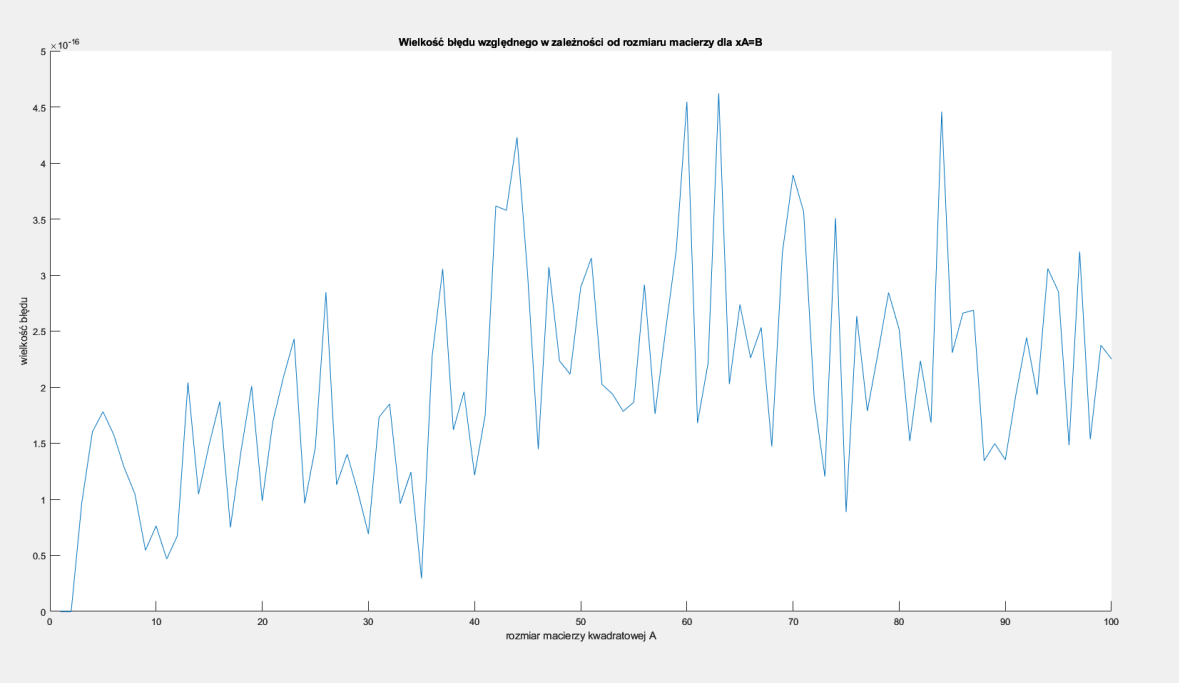
```
57  % poniższy kod pokazuje na wykresie jak zmienia się błąd względny
58  %w zaleznośći od rozmiaru macierzy A
59  % błąd wzgledny:
60  % z- dokładne rozwiązanie układu Ax=B,
61  % X obliczone numerycznie przybliżenie
62  % błąd względny ||X-z||/||z||
63
64  y=repelem(0,100);
65  x=1:100;
66  for i=x
67      A=gallery("gcdmat",i);
68      B=randn(i,1);
69      X = solve_crout_LUX(A, B);
70      z = inv(A)*B;
71      y(i)= norm(z-X)/norm(z);
72  end
73  hold on
74  plot(x,y)
75  title("Wielkość błędu względnego w zależności od rozmiaru macierzy dla xA=B")
76
77  xlabel("rozmiar macierzy kwadratowej A")
78  ylabel("wielkość błędu")
79  hold off
```

Dokonujac odpowienich drobnych zmian, tak aby wymiary macierzy sie zgadzały otrzymamy wykres dla równania $X \cdot A = B$
Wykresy wygladaja nastepujaco:

$$A \cdot X = B$$



$$X \cdot A = B$$



Z wykresu patrzac na skale y możemy wyczytać, że średnio bład wzgledny jest mniejszy dla rozwiazywania równania $X \cdot A = B$

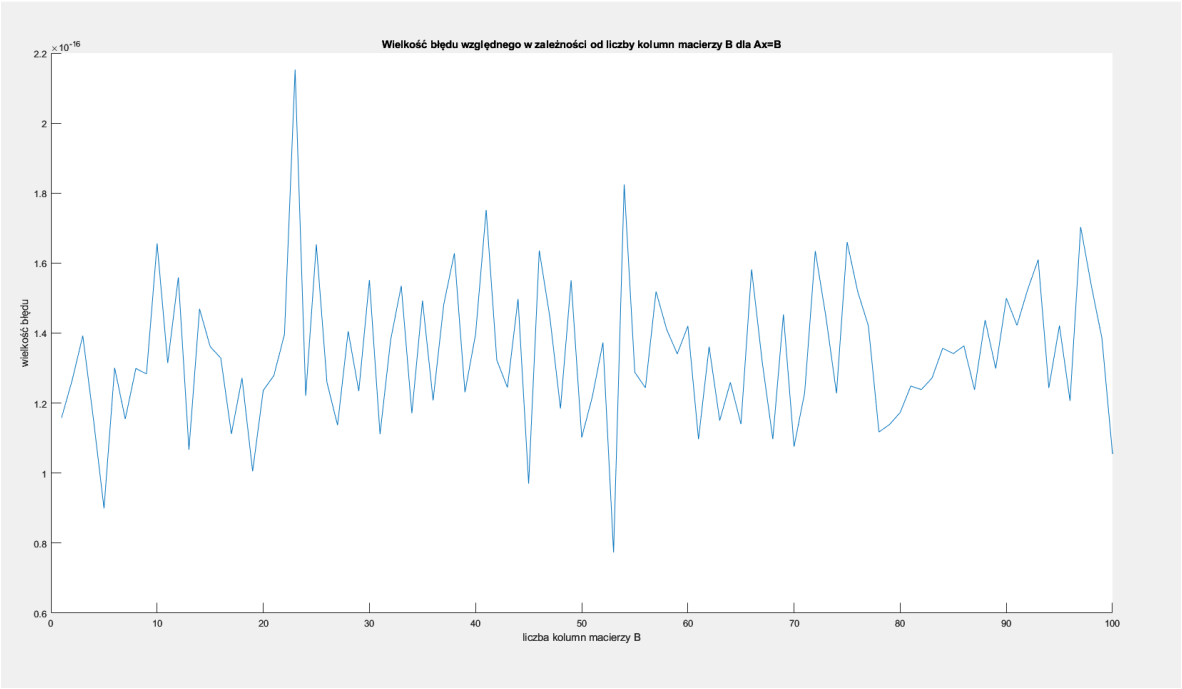
2.2.2 Liczba kolumn—wierszy macierzy B

Analogicznie porównamy błąd względny w zależności od liczby kolumn—wierszy macierzy B. Dokonamy tego za pomoca poniższego kodu:

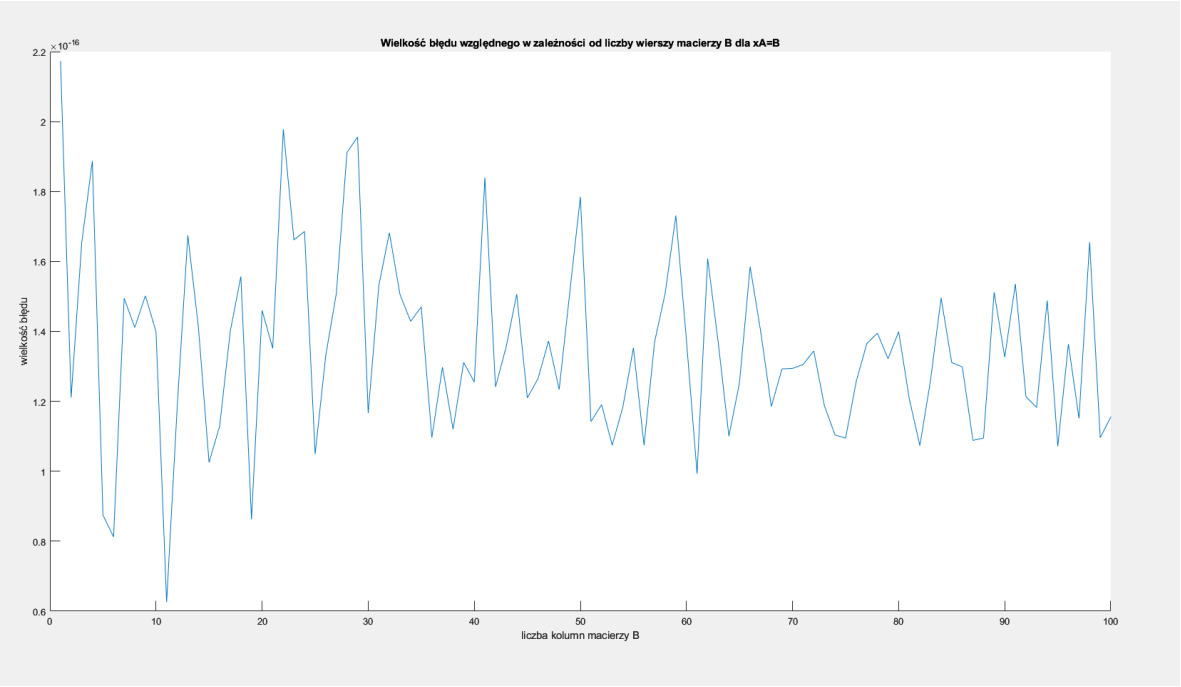
```
%poniższy kod pokazuje na wykresie jak zmienia się błąd względny
%w zaleznośći od liczby kolumn macierzy B
% błąd wzgledny:
% z- dokładne rozwiązanie układu AX=B,
% X obliczone numerycznie przybliżenie
% błąd względny ||X-z||/||z||

y=repelem(0,100);
x=1:100;
for i=x
    A=gallery("gcdmat",10);
    B=randn(size(A,1),i);
    X = solve_crout_LUX(A, B);
    z = inv(A)*B;
    y(i)= norm(z-X)/norm(z);
end
hold on
plot(x,y)
title("Wielkość błędu względnego w zależności od liczby kolumn macierzy B dla Ax=B")
xlabel("liczba kolumn macierzy B")
ylabel("wielkość błędu")
hold off
```

Dokonujac odpowienich drobnych zmian, tak aby wymiary macierzy sie zgadzały otrzymamy wykres dla równania $X \cdot A = B$
Tak powstałe wykresy wygladaja nastepujaco:

$$A \cdot X = B$$



$$X \cdot A = B$$



Wykresy zasadniczo dla tych samych argumentów sie różnia, jednak nie da sie stwierdzić, które wyniki sa bardziej satysfakcjonujace. Wyniki sa bardzo do siebie zbliżone. Raz błąd jest wiekszy dla równania $A \cdot X = B$ a raz dla $X \cdot A = B$.

2.3 Podsumowanie porównania

Cieżko jednoznacznie stwierdzić, które równania sa lepsze do rozwiazywania w przypadku $X \cdot A = B$ błąd wzgledny obliczeń z reguły mniejszy, jednak czas obliczeń był dłuższy. Różnice czasowe jak i błedu wzglednego, mimo że wystepuja to i tak nie sa wychwytywane dla ludzkiego oka. Jednak z perspektywy rosnacego czasu dla wiekszych macierzy i raczej niezależnego od rozmiaru macierzy błedu wzglednego. Do rozwiazywania dużej ilości równań lepsze sa równania w postaci $A \cdot X = B$