Warszawa, 23.11.2018r

Politechnika Warszawska

Wydział Elektroniki i Technik Informacyjnych

Metody numeryczne

Zadanie 1: Rozwiązywanie liniowych równań algebraicznych

Wykonawca zadania:

Joanna Kiesiak #22

Prowadzący projekt:

dr inż. Andrzej Miękina

Spis treści

1.Wstęp

1.1 Środowisko programu

Program został wykonany w programie MATLAB R2017b. MATLAB to pakiet programowy, którego zadaniem jest wykonywanie złożonych obliczeń matematycznych i wizualizacja wyników. Podstawowym elementem konstrukcyjnym w tym programie jest macierz, a tablica fundamentalnym typem danych.

1.2 Rozwiązywanie układów równań linowych

Rozwiązywanie układów równań liniowych jest jednym z najczęściej występujących w praktyce inżynierskiej zagadnień numerycznych. Do układów równań linowych prowadzą między innymi następujące zagadnienia:

- analiza stanów ustalonych w liniowych obwodach elektrycznych,
- interpolacja danych pomiarowych,
- analiza pól elektromagnetycznych metodą rozwiązania równań różniczkowych cząstkowych,
- aproksymacja charakterystyk statycznych czujników pomiarowych.

2.Zadanie 1

2.1 Treść zadania

1. Opracować procedurę generacji macierzy $\boldsymbol{A}_{N,x}$ *zdefiniowanej wzorem*

$$\boldsymbol{A}_{N,x} = \begin{bmatrix} x^2 & \frac{2x}{3} & \frac{2x}{3} & \frac{2x}{3} & \cdots & \frac{2x}{3} & \frac{2x}{3} \\ \frac{2x}{3} & \frac{8}{9} & \frac{8}{9} & \frac{8}{9} & \cdots & \frac{8}{9} & \frac{8}{9} \\ \frac{2x}{3} & \frac{8}{9} & \frac{12}{9} & \frac{12}{9} & \cdots & \frac{12}{9} & \frac{12}{9} \\ \frac{2x}{3} & \frac{8}{9} & \frac{12}{9} & \frac{16}{9} & \cdots & \frac{16}{9} & \frac{16}{9} \\ \frac{2x}{3} & \frac{8}{9} & \frac{12}{9} & \frac{16}{9} & \cdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \cdots & \frac{(N-1)\times 4}{9} & \frac{(N-1)\times 4}{9} \\ \frac{2x}{3} & \frac{8}{9} & \frac{12}{9} & \frac{16}{9} & \cdots & \frac{(N-1)\times 4}{9} & \frac{N\times 4}{9} \end{bmatrix}$$

gdzie N = 3,10, 20 i x = $\sqrt{\alpha^2 + \frac{1}{2}}$ -1 *, wyznaczyć najmniejszą wartość dodatnią* α_N *dla której* *det (*$\boldsymbol{A}_{N,x}$ *) = 0.*

2.2 Opis zastosowanych algorytmów numerycznych

Za pomocą słowa kluczowego function zaprojektowano funkcję **MacierzA** generującą macierz kwadratową A dla parametrów wejściowych N i alfa. Przy czym alfa jest parametrem funkcji występującej w macierzy, a N wielkością macierzy.

Za pomocą dwóch zagnieżdżonych pętli for, którą stosuje się do wykonywania wyrażeń określoną liczbę razy, określono kolejne kolumny i wiersze macierzy. Obie pętle wykonują się od N do 1 z domyślnym licznikiem -1. Sposób generowania macierzy jest ‘od końca’, ze względu na wydajność algorytmu. Następnie za pomocą wyrażeń if oraz elseif podzielono obliczenia na 2 przypadki i dla każdego z nich wyznaczono współczynniki macierzy A.

Do wyznaczenia współczynników x występujących w macierzy A używam równania zaimplementowanego w linijce cztery (4), który jest zapisany przy pomocy zmiennej z.

2.3 Przykład działania programu

A=MacierzA(1,3)

A =

0.0505	0.1498	0.1498
0.1498	0.8889	0.8889
0.1498	0.8889	1.3333

A=MacierzA(3,5)

A =

4.3356	1.3881	1.3881	1.3881	1.3881
1.3881	0.8889	0.8889	0.8889	0.8889
1.3881	0.8889	1.3333	1.3333	1.3333
1.3881	0.8889	1.3333	1.7778	1.7778
1.3881	0.8889	1.3333	1.7778	2.2222

A = MacierzA(1,10)

A =

0.0505	0.1498	0.1498	0.1498	0.1498	0.1498	0.1498	0.1498	0.1498	0.1498
0.1498	0.8889	0.8889	0.8889	0.8889	0.8889	0.8889	0.8889	0.8889	0.8889
0.1498	0.8889	1.3333	1.3333	1.3333	1.3333	1.3333	1.3333	1.3333	1.3333
0.1498	0.8889	1.3333	1.7778	1.7778	1.7778	1.7778	1.7778	1.7778	1.7778
0.1498	0.8889	1.3333	1.7778	2.2222	2.2222	2.2222	2.2222	2.2222	2.2222
0.1498	0.8889	1.3333	1.7778	2.2222	2.6667	2.6667	2.6667	2.6667	2.6667

0.1498	0.8889	1.3333	1.7778	2.2222	2.6667	3.1111	3.1111	3.1111	3.1111
0.1498	0.8889	1.3333	1.7778	2.2222	2.6667	3.1111	3.5556	3.5556	3.5556
0.1498	0.8889	1.3333	1.7778	2.2222	2.6667	3.1111	3.5556	4.0000	4.0000
0.1498	0.8889	1.3333	1.7778	2.2222	2.6667	3.1111	3.5556	4.0000	4.4444

Macierze generowane są w poprawny sposób.

3. Zadanie 2

3.1 Treść zadania

2. Dla $N = 3,10, 20$ i $x = \sqrt{\alpha^2 + \frac{1}{2}} - 1$ wyznaczyć najmniejszą wartość dodatnią α_n dla której $det(A_{N,x}) = 0$. Sporządzić wykresy zależności wyznacznika macierzy oraz wskaźnika uwarunkowania każdej z macierzy $A_{N,x}$ od α dla $\alpha \in [\alpha_n - 0.01, \alpha_n + 0.01]$.

3.2 Opis zastosowanych algorytmów numerycznych

Utworzono funkcję **detMacierzA**(macierz, rozmiarN), której na wejściu podajemy macierz i wielkość wymiarów macierzy, dla których mają powstać wykresy. Funkcja zwraca wartość alfa, dla której wyznacznik jest równy zero. W tej funkcji zostało przyjęte że wartość alfy być mniejsza niż 10^{-14}, to jest to poprawna wartość α_n.

$$0 = \sqrt{\alpha^2 + \frac{1}{2}} - 1$$

$$\alpha = \frac{\sqrt{2}}{2} \approx 0.7071$$

Utworzono funkcję **wykresZal**(N, a), która pobiera rozmiar macierzy i macierz wejściową. Następnie przy pomocy polecenia linspace zostaje utworzona zmienna OSx, która ma 1000 punktów z przedziału $\alpha \in [\alpha_n - 0.01, \alpha_n + 0.01]$. Następnie za pomocą wbudowanej funkcji det i cond w pętli for wyliczono wartość wyznacznika macierzy i wskaźnika uwarunkowania dla każdego podanego alfa z OSx.

Aby uzyskać wykresy wskaźnika uwarunkowania i wyznacznika macierzy, należy zadeklarować zmienne, które zarezerwują wystarczającą ilość pamięci. Za pomocą pętli for zostaje wygenerowana MacierzA w zależności od zmiennej N – czyli rozmiaru macierzy. Dla alfy dla której wyznacznik jest równy zero. W pętli for zostaje użyta funkcja wykresZal, która wylicza wartość wyznacznika macierzy i wskaźnika uwarunkowania od każdego punktu i zostaje to zapisane w zmiennej Osy i OSyy. W kolejnym kroku algorytm rysuje wykresy w zależności wartości wyznacznika lub wskaźnika od parametru alfa.

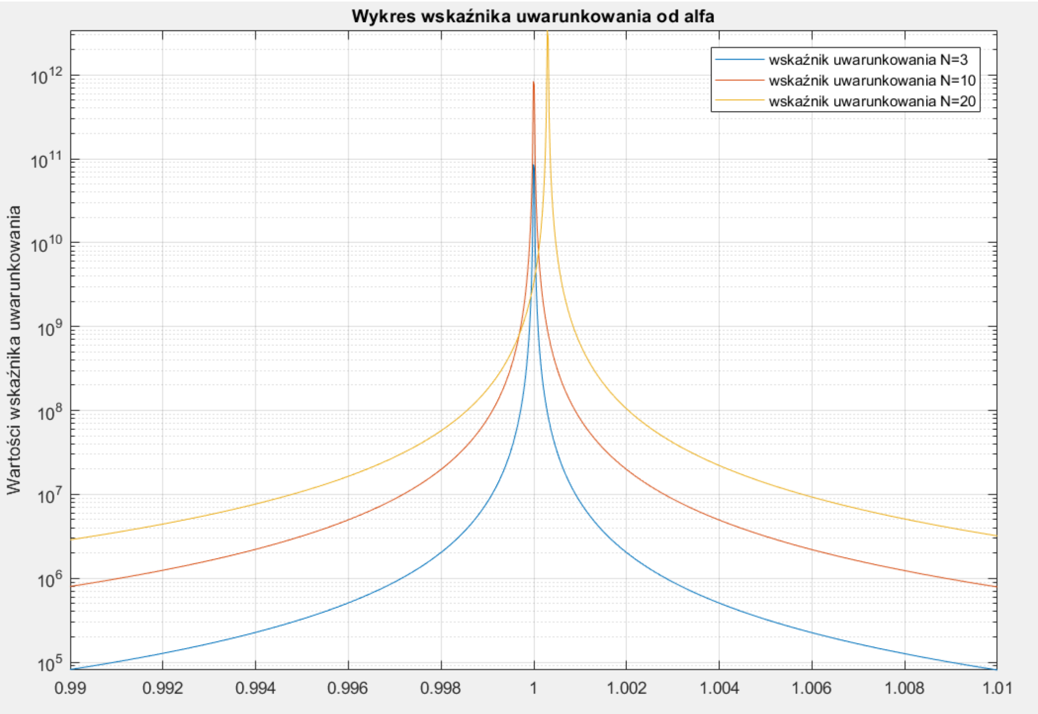
Wygenerowano nowe okno wykresu poleceniem figure i za pomocą funkcji semilogy narysowany został wykres. Używając poleceń xlabel i ylabel nazwano osie wykresu, komenda grid on stworzyła siatkę lini pomocniczych na wykresie. Analogicznie postępowano dla wykresu wskaźnika uwarunkowania macierzy.

alfa = detMacierzA(A, 5)

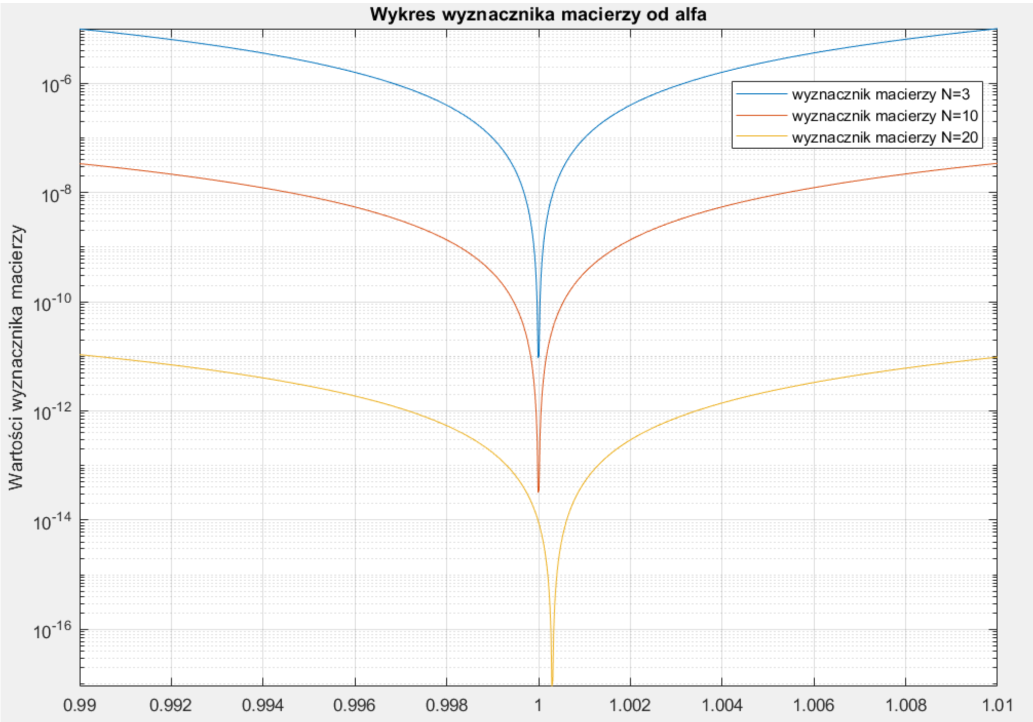
alfa =

    0.7071

3.3 Przykład działania programu



rys.1 Wykres wskaźnika uwarunkowania macierzy MacierzA(alfa,3), MacierzA(alfa,10), MacierzA(alfa,20) w funkcji α



rys.2 Wykres wyznacznika macierzy MacierzA(alfa, 3), MacierzA(alfa, 10) i MacierzA(alfa, 20) w funkcji α.

3.4 Wnioski

Na wykresach obserwujemy, że otrzymane ekstrema znajdują się w wartości odpowiadającej parametrowi α, dla którego wyznacznik macierzy jest równy 0.

4.1 Treść zadania

3. Opracować procedurę wyznaczania macierzy $A_{N,x}^{-1}$ - odwrotnej do $A_{N,x}$ wg schematu przedstawionego na slajdzie 2-16 przy zastosowaniu obu rozkładów: LU i LL ^{T}. Sprawdzić jej działanie dla kilku niewielkich dodatnio określonych macierzy.

4.2 Opis zastosowanych algorytmów numerycznych

Funkcja **wyznaczOdwrotnaLLt i wyznaczOdwrotnaLU** zwraca macierz Y dla podanej na wejście dodatnio określonej macierzy kwadratowej A. Macierz Y jest macierzą odwrotną do macierzy wejściowej i jest oznaczana jest jako A^{-1}. Mnożenie macierzy odwrotnych prowadzi do równania:

$$A * A^{-1} = A^{-1} * A = I$$

gdzie macierz I to macierz jednostkowa. Macierz jednostkowa ma ogólną postać:

$$I = \begin{bmatrix} 1 & 0 & & \dots & 0 \\ 0 & 1 & & \dots & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & i_{n,n} & = 1 \end{bmatrix}$$

Aby uzyskać macierz odwrotną korzystamy z dwóch metod: rozkładu Cholesky'ego-Banachiewicza oraz rozkładu wg metody Dolitttle'a.

Rozkład Cholesky'ego-Banachiewicza to iloczyn macierzy trójkątnej dolnej oraz macierzy trójkątnej dolnej transponowanej. Ogólny wzór to:

$$A=L*L^T$$

i ma postać:

$$L = \begin{bmatrix} l_{1,1} & 0 & \dots & 0 \\ l_{2,1} & l_{2,2} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ l_{n,1} & l_{n,2} & \cdots & l_{n,n} \end{bmatrix}$$

Mnożenie kolejnych wartości macierzy L i L^T oraz porównywanie z odpowiednimi elementami macierzy A prowadzi do algorytmu, w postaci:

$$l_{n,n} = \sqrt{a_{n,n} - \sum_{i=1}^{n-1} l_{n,i}^2} \; dla\; n = 1, \dots, N$$

$$l_{v,n} = \frac{a_{v,n} - \sum_{i=1}^{n-1} l_{v,i} \cdot l_{n,i}}{l_{n,n}} \; dla\; n = 1, \dots, N, v = n+1, \dots, N$$

W celu wyznaczenia dolnej macierzy trójkątnej L skorzystano z polecenia L = chol(A, 'lower').

Metoda Dolittle'a zwraca macierz L i U dla podanej na wejście dodatnio określonej macierzy kwadratowej A. Macierz L jest dolną macierzą trójkątną, która ma wartość 1.0 na przekątnej, natomiast macierz U jest górną macierzą trójkątną i mają ogólną postać:

$$L = \begin{bmatrix} 1 & 0 & \dots & 0 \\ l_{2,1} & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ l_{n,1} & l_{n,2} & \cdots & l_{n,n} = 1 \end{bmatrix}$$

$$U = \begin{bmatrix} u_{1,1} & u_{1,2} & \cdots & u_{1,n} \\ 0 & u_{2,2} & \cdots & u_{2,n} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & u_{n,n} \end{bmatrix}$$

Mnożenie kolejnych wartości macierzy L i U oraz porównywanie z odpowiednimi elementami macierzy A prowadzi do algorytmu, w postaci:

$$u_{i,j} = a_{i,j} - \sum_{k=1}^{i-1} l_{i,k} u_{k,j} \ dla\ i = 1, \dots, N$$

$$l_{i,j} = \frac{1}{u_{i,i}} \left(a_{i,j} - \sum_{k=1}^{i-1} l_{j,k} \cdot u_{k,i} \right) dla\ j = 1, \dots, N\ ;\ k = n+1, \dots, N$$

W celu wyznaczenia macierzy L i U skorzystano z polecenia [L, U, P] = lu(A), gdzie P oznacza macierz permutacji.

W celu obliczenia macierzy odwrotnej A^{-1} wykonujemy metodę rozwiązywania układu równań z udostępnionych Materiałów Pomocniczych z slajdu 2-19. Algorytm sprowadza się do wzoru:

$$L*L^T*Y=I$$

$$L*U*Y=I$$

Gdzie I to macierz jednostkowa. W algorytmie oznaczamy U*Y = X i L^T*Y = X.

Przy pomocy polecenia size pobrano wymiar macierzy wejściowej A i zapisano go do zmiennej N. Następne tworzymy macierze X i Y, które mają wymiary N i są wypełnione zerami. Następnie przy pomocy trzech zagnieżdżonych pętli for zostaje wyznaczona macierz X z równania i prowadzi do poniższego algorytmu:

$$L*X = E$$

$$x_{j,i} = \frac{1}{l_{j,j}} \left(e_{j,i} - \sum_{k=1}^{j-1} x_{k,i} \cdot l_{j,k} \right) dla\ j = i = 1, \dots, N\ ;\ k = 1, \dots, j-1$$

W analogiczny sposób, również przy pomocy trzech zagnieżdżonych pętli for wyznaczamy macierz Y z równania i określa to ogólny wzór:

$$U*Y = X$$

$$y_{j,i} = \frac{1}{u_{j,j}}\left(x_{j,i} - \sum_{v=N}^{j+1} y_{v,i} \cdot u_{j,v} \right) dla\ j = i = 1, \dots, N\ ;\ v = N, \dots, j+1$$

Macierz wynikowa to macierz Y, czyli macierz odwrotna do macierzy wejściowej.

W analogiczny sposób postępujemy w przypadku L^T *Y = X.

4.3 Przykład działania programu

Przetestujmy działanie funkcji dla pewniej dodatnio określonej macierzy A w programie MATLAB

```
A = [1 2 3; 2 5 8; 3 8 14]

A =

    1   2   3

    2   5   8

    3   8   14

X =

   1.0000        0          0

  -0.3333   1.0000        0

  -0.5000  -0.5000   1.0000

Y =

   1.0000   6.0000  -4.0000

  -2.0000  -4.0000   5.0000

   1.0000   1.0000  -2.0000

>> a = P*A*Y

a =

   1.0000  -0.0000  -0.0000

   0.0000   1.0000        0

   0.0000  -0.0000   1.0000
```

Komentarz do przykładu: Macierz A i Y po zastosowaniu mnożenia generują macierz jednostkową, zatem możemy stwierdzić, że funkcja wyznaczOdwrotnaLU i wyznaczOdwrotnaLLt działa prawidłowo.

5. Zadanie 4

5.1 Treść zadania

4. Zastosować powyższą procedurę do wyznaczenia macierzy A^{-1} dla N = 3, 10, 20 oraz $x = \frac{2^k}{300}$*, gdzie k = 0, 1, ... 21.*

5.2 Opis zastosowanych algorytmów numerycznych

Macierz zostaje wyznaczona w sposób analogiczny do macierzy z zadania 1. W algorytmie została zaimplementowana funkcja **MacierzB**, która generuje macierz B z wartościami x wyznaczonymi wg wzoru z polecenia.

5.3 Przykład działania programu

```
>> B=MacierzB(2,10)

B =
```

0.0002	0.0089	0.0089	0.0089	0.0089	0.0089	0.0089	0.0089	0.0089	0.0089
0.0089	0.8889	0.8889	0.8889	0.8889	0.8889	0.8889	0.8889	0.8889	0.8889
0.0089	0.8889	1.3333	1.3333	1.3333	1.3333	1.3333	1.3333	1.3333	1.3333
0.0089	0.8889	1.3333	1.7778	1.7778	1.7778	1.7778	1.7778	1.7778	1.7778
0.0089	0.8889	1.3333	1.7778	2.2222	2.2222	2.2222	2.2222	2.2222	2.2222
0.0089	0.8889	1.3333	1.7778	2.2222	2.6667	2.6667	2.6667	2.6667	2.6667
0.0089	0.8889	1.3333	1.7778	2.2222	2.6667	3.1111	3.1111	3.1111	3.1111
0.0089	0.8889	1.3333	1.7778	2.2222	2.6667	3.1111	3.5556	3.5556	3.5556
0.0089	0.8889	1.3333	1.7778	2.2222	2.6667	3.1111	3.5556	4.0000	4.0000
0.0089	0.8889	1.3333	1.7778	2.2222	2.6667	3.1111	3.5556	4.0000	4.4444

6. Zadanie 5

6.1 Treść zadania

Dla każdego rozwiązania $\hat{A}_{N,x}^{-1}$ *wyznaczyć wskaźnik błędu średniokwadratowego:*

$$\delta_2 = || A_{N,x} * \hat{A}_{N,x}^{-1} - I_N ||_2$$

Oraz wskaźnik błędu maksymalnego:

$$\delta_\infty = || A_{N,x} * \hat{A}_{N,x}^{-1} - I_N ||_\infty$$

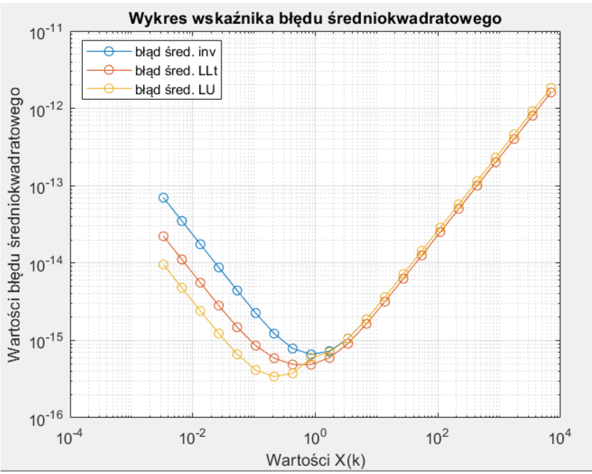
Normy macierzy należy liczyć w sposób przedstawiony na slajdzie 1-5. Porównać otrzymane wyniki dla rozkładów LU i LLt z wartościami tych wskaźników, wyznaczonymi dla macierzy odwrotnych uzyskanych za pomocą operatora wyznaczania odwrotności macierzy inv, *zaimplementowanego w MATLABie. Sporządzić odpowiednie wykresy. Porównać wyznaczone normy z wynikami uzyskanymi za pomocą funkcji norm zaimplementowanej w Matlabie.*

6.2 Opis zastosowanych algorytmów numerycznych

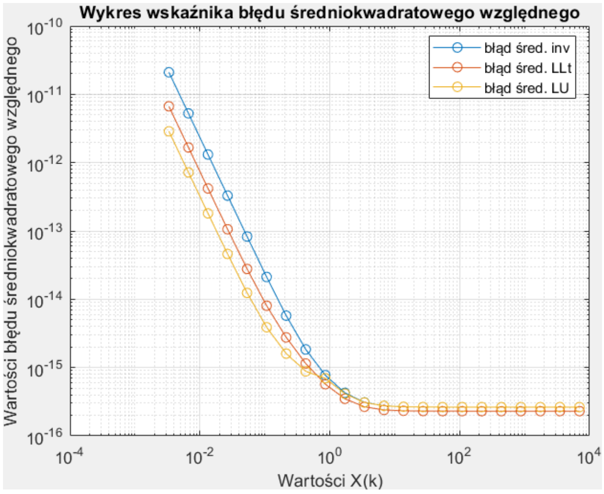
Funkcja **wskBlad** zwraca dwie wartości: błąd średniokwadratowy oraz błąd maksymalny.

Funkcja **wyznaczBledy** zwraca nam wykresy na których możemy porównać trzy metody uzyskania macierzy odwrotnej. Jako pierwszy krok zaimplementowano zmienne, do których będziemy zapisywać wyniki. Za pomocą dwóch zagnieżdżonych pętli for, generujemy macierz B, która jest rozkładana na 2 sposoby(metodę LLt i LU) oraz za pomocą funkcji (inv). Dla każdego sposobu rozłożenia macierzy wejściowej są wyliczone błędy średniokwadratowe oraz maksymalne. Błędy są wyliczone za pomocą funkcji wskBlad oraz wbudowanej funkcji w Matlabie norm. Wykresy są rysowane w funkcji loglog, została dopisana legenda oraz zaznaczone kółeczkami konkretne błędy w celu interpretacji wyników.

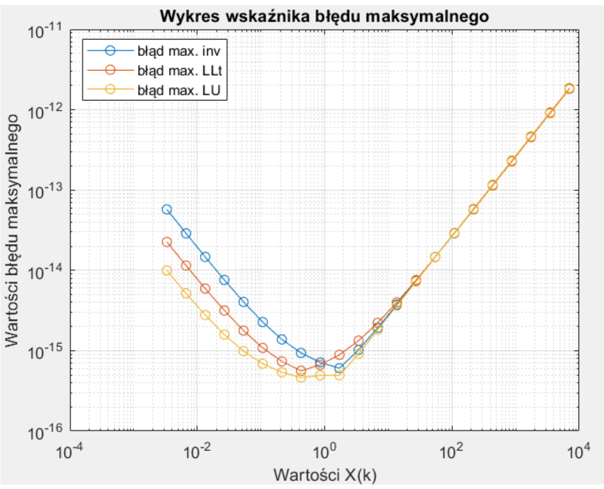
6.3 Przykład działania programu



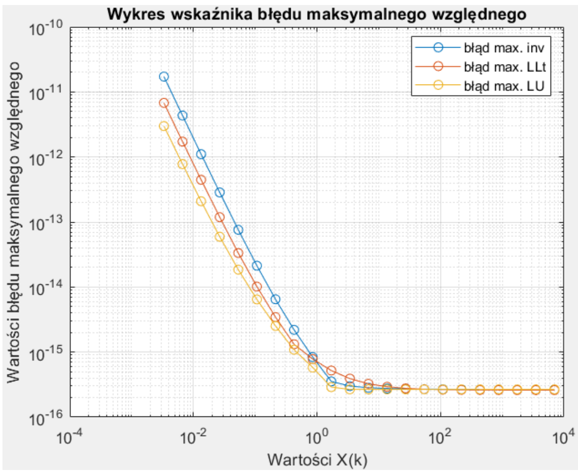
Rys.3 Wykres wskaźnika błędu średniokwadratowego dla N=3, obliczony za pomocą funkcji wskBlad



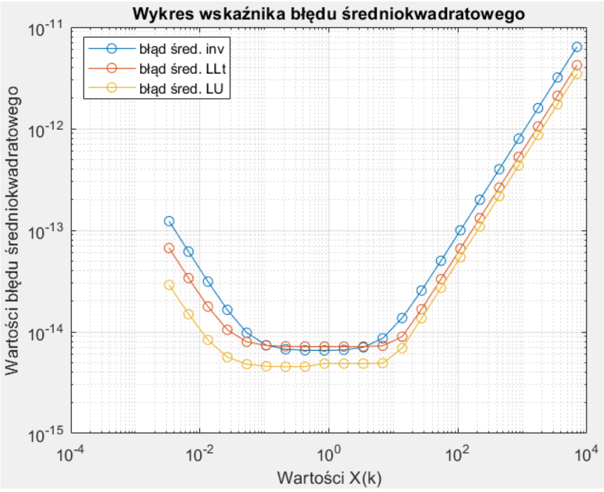
Rys.4 Wykres wskaźnika błędu średniokwadratowego względnego dla N=3



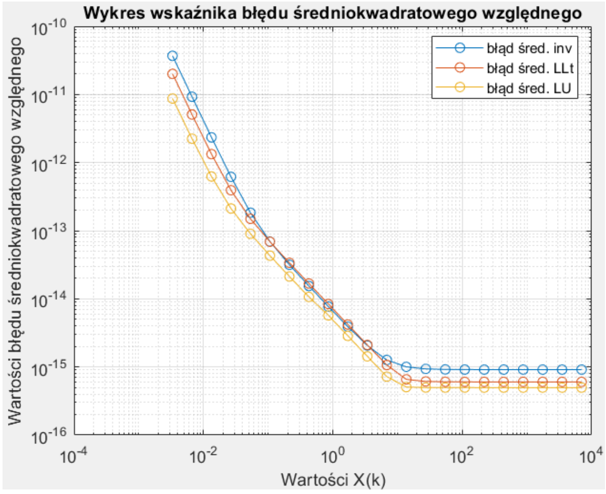
Rys.5 Wykres wskaźnika błędu maksymalnego dla N=3, obliczony za pomocą funkcji wskBlad



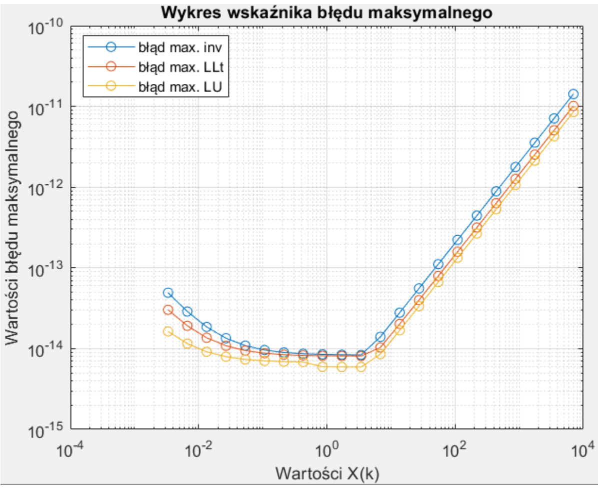
Rys.6 Wykres wskaźnika błędu maksymalnego względnego dla N=3



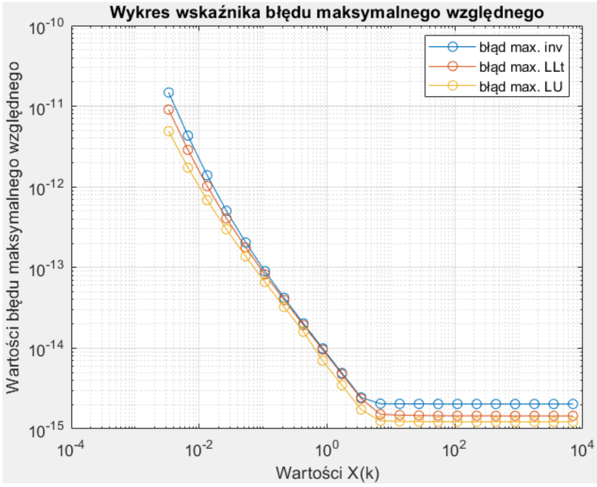
Rys.7 Wykres wskaźnika błędu średniokwadratowego dla N=10, obliczony za pomocą funkcji wskBlad



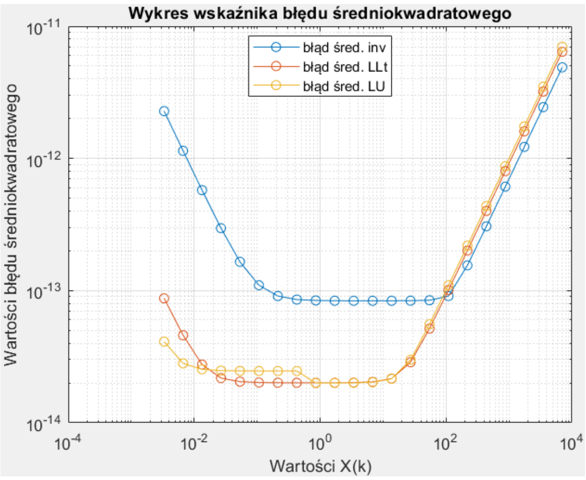
Rys.8 Wykres wskaźnika błędu średniokwadratowego względnego dla N=10



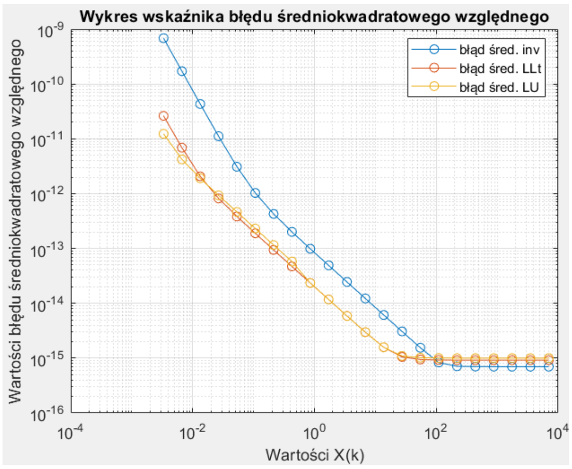
Rys.9 Wykres wskaźnika błędu maksymalnego dla N=10, obliczony za pomocą funkcji wskBlad



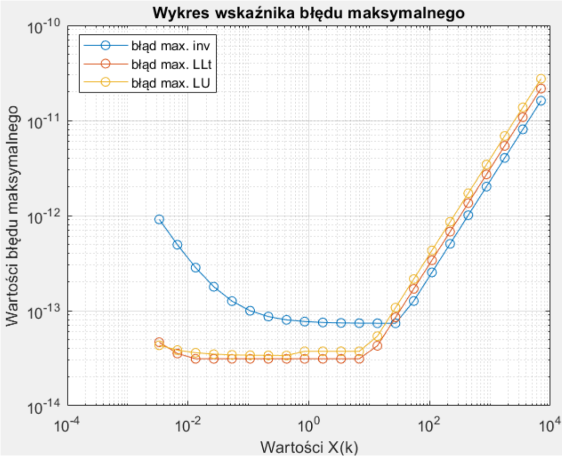
Rys.10 Wykres wskaźnika błędu maksymalnego względnego dla N=10



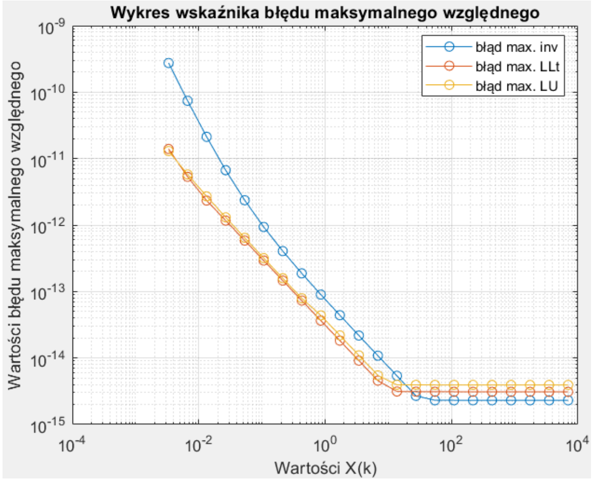
Rys.11 Wykres wskaźnika błędu średniokwadratowego dla N=20, obliczony za pomocą funkcji wskBlad



Rys.12 Wykres wskaźnika błędu średniokwadratowego względnego dla N=20



Rys.13 Wykres wskaźnika błędu maksymalnego dla N=20, obliczony za pomocą funkcji wskBlad



Rys.14 Wykres wskaźnika błędu maksymalnego względnego dla N=20

6.4 Wnioski

Wykres wskaźnika błędu średniokwadratowego oraz maksymalnego dla N=3 maleje do x(k) = 10^0, wtedy błąd osiąga wartość 10^{-15}, następnie błąd zaczyna wzrastać proporcjonalnie do wartości x. Wykres błędu średniokwadratowego i maksymalnego dla N=10 jest malejący w obszarze 10^{-3} do 10^{-2}, następnie stabilizuje się aż do wartości x(k) = 10^0, od tej wartości błąd zaczyna wzrastać proporcjonalnie do x. Dla N=20 błąd średniokwadratowy i maksymalny stabilizuje się w zakresie 10^{-2} do 10^1, a następnie proporcjonalnie wzrasta.

W celu interpretacji wyników został obliczony błąd względny dla każdego przypadku, który globalnie pokazuje jak wartości x(k) mają wpływ na dokładność metody. Dla N=3 błąd stabilizuje się gdy x(k)=10^0, co oznacza że duże liczby nie będą mieć wpływu na wzrost rozkładu macierzy, w przeciwnym wypadku - małe wartości x będą warunkować wzrost błędu. Podobną sytuację mamy dla N=10, w którym wartości stabilizują się dla x(k)=10 i przy wzroście x, błąd pozostaje taki sam. N=10 małe wartości x powodują wzrost błędu względnego. Male wartości x powodują wzrost błędu względnego dla N= 20, błąd względny zaczyna stabilizować się od x(k)=10^2.

Porównując metody rozkładu w zależności od wielkości macierzy wejściowej, dla najmniejszego N, najmniej błędów zwraca wbudowana funkcja inv, ale do x=1, ponieważ metoda przestaje mieć znaczenie - wykresy LU, LLu i inv nakładają się na siebie. Dla N=10 najlepsza metoda to rozkład przez funkcję inv, jednakże wybór innej metody nie będzie powodował że błąd radykalnie wzrośnie. W przypadku największego N metoda inv okazuje się najlepsza, błąd dla LU i LLt będzie większy o niecały jeden rząd wielkości.

7.Kod funkcji w języku programu MATLAB

```
clear all
close all
  %Zadanie 1
A = MacierzA(alfa,N);

% %Zadanie 2
alfa = detMacierzA(A, N);

%zadanie 3
Y1 = wyznaczOdwrotnaLLt(A);
Y = wyznaczOdwrotnaLU(A);
```

```
%zadanie 4 i 5
B = MacierzB(k,N);

%% Zadanie 1
%Wygenerowanie macierzy o zadanym rozmiarze n i zadanej
wartosci x
function A=MacierzA(alfa,N)
%A-zwracana wygenerowana zgodnie ze wzorem macierz
A=zeros(N);
x = sqrt((alfa)^2 + 0.5) - 1; %zadana w poleceniu funkcja
A(1,1) = (x)^2; %macierz min. 1x1, więc wyrzucam przed pętlę
for i=N:-1:1   %Iteracja wierszy (od końca)
   for j=N:-1:1   %Iteracja kolumn (od końca)
      if (i>=2 && j>=2)
         A(i,j) = min(i,j)*4/9;
      elseif ((i==1 && j>1) || (j==1 && i>1 ))
         A(i,j)=(2/3)*x;
      end
   end
end
end

%% Zadanie 2

%Znalezienie najmniejszej dodatniej wartosci alfa, dla ktorej
wartosc
%wyznacznika macierzy o danym rozmiarze wynosi 0
function alfa = detMacierzA(A, N)
[N, N] = size(A);
A = @(z) det(MacierzA(z, N));
for i = linspace(0, 5 , 100000)
  if (A(i) < 10^(-14))           %W praktyce wartość wyznacznika
musi dążyć do zera
     alfa = i;
     break;
  end
end
end

%blok wylicza wartości z zależności wyznacznika macierzy oraz
%wskaźnika uwarunkowania każdej z macierzy od alfa
function [OSy, OSyy] = wykresyZal(N, A)
alfan = detMacierzA(A, N);
% alfan = sqrt(2*2)/2;
OSx = linspace(alfan-0.01,alfan+0.01,1000); % x - tworzy
wektor punktów
                              %dla których rysowane są wykresy
OSy = zeros(1,1000);
OSyy = zeros(1,1000);
A = @(z) det(MacierzA(z,N));
 % pętla przypisuje wartosć wyznacznika macierzy dla każdego
x
   for i=1:1000
       OSy(i) = A(OSx(i));
   end

 A = @(z) cond(MacierzA(z,N));
% pętla przypisuje wartość wskaźnika uwarunkowania macierzy
dla każdego x
   for i=1:1000
       OSyy(i) = A(OSx(i));
   end
end

%blok rysuje wykresy z zależności wyznacznika macierzy oraz
%wskaźnika uwarunkowania każdej z macierzy od alfa
% alfa = 0;
N = [3 10 20];
z = 1000;
alfan = sqrt(2*2)/2;

OSx = linspace(alfan-0.01,alfan+0.01,1000);
OSy = zeros(3,z);
OSyy = zeros(3,z);
OSp = zeros(3,z);
OSpp = zeros(3,z);
j = 0;

for i = 1 : length(N)
   A = MacierzA(alfan,N(i));
   alfa = detMacierzA(A, N(i));
   [OSp, OSpp] = wykresyZal(N(i), A);
   j = j+1;
   OSy(j,:) = OSp;
   OSyy(j,:) = OSpp;
end

figure;
semilogy(OSx, OSy(1,:), OSx, OSy(2,:), OSx, OSy(3,:));
axis([alfan-0.01 alfan+0.01 0 inf])
xlabel('Alfa w przedziale alfa-0.01,alfa+0.01');
ylabel('Wartości wyznacznika macierzy');
title('Wykres wyznacznika macierzy od alfa');
legend('wyznacznik macierzy N=3','wyznacznik macierzy
N=10','wyznacznik macierzy N=20', 'Best', 'Best');
grid on;
hold on;

figure;
semilogy(OSx, OSyy(1,:), OSx, OSyy(2,:), OSx, OSyy(3,:));
axis([alfan-0.01 alfan+0.01 0 inf])
xlabel('Alfa w przedziale alfa-0.01,alfa+0.01');
ylabel('Wartości wskaźnika uwarunkowania');
title('Wykres wskaźnika uwarunkowania od alfa');
legend('wskaźnik uwarunkowania N=3','wskaźnik
uwarunkowania N=10','wskaźnik uwarunkowania N=20', 'Best',
'Best');
grid on;
hold on;



%% Zadanie 3
function Y1 = wyznaczOdwrotnaLLt(A)
%rozkład Cholewskiego-Banachiewicza
L = chol(A, 'lower');
Lt=L';          %transpozycja dolnej macierzy trójkątnej
[N, N] = size(L);
[N, N] = size(Lt);
Y1 = zeros(N);
E1 = eye(N);        %macierz jednostkowa
X1 = zeros(N);
% L*X = E              ;wyszukujemy X
for ii = 1:N
   for jj = 1:N
         s1 = 0;
         for kk = 1 : jj-1
            s1 = s1 + X1(kk,ii)*L(jj,kk);
         end
         X1(jj,ii) = ( E1(jj,ii) - s1 ) * ( 1 / L(jj,jj) );
   end
end
% checkEye1 = L*X1 ;
% Lt*Y=X              ;wyszukujemy Y
for i = 1 : N
   for j = N : -1 : 1
         s2 = 0;
         for v =  N: -1 : j+1
            s2 = s2 + Y1(v,i)*Lt(j,v);
         end
         Y1(j,i) = ( X1(j,i) - s2 ) * ( 1 / Lt(j,j) );
   end
end
% check2shouldbeX1 = Lt*Y1 ;
% X1 = X1 ;
% checkbeEye1 = A*Y1 ;
% matrixA1 = L*Lt;
% Eye1 = L*Lt*Y1 ;
end

function Y = wyznaczOdwrotnaLU(A)
[L1, U, P] = lu(A);        %P - macierz permutacji
```

```
[N, N] = size(L1);
[N, N] = size(U);
E = eye(N);
X = zeros(N);
Y = zeros(N);
% L*X = E            ;wyszukujemy X
for ii = 1:N
   for jj = 1:N
      s1 = 0;
         for kk = 1 : jj-1
            s1 = s1 + X(kk,ii)*L1(jj,kk);
         end
      X(jj,ii) = ( E(jj,ii) - s1 ) * ( 1 / L1(jj,jj) );
   end
end
checkShouldBeEye2 = L1*X ;
% U*Y=X              ;wyszukujemy Y
for i = 1 : N
   for j = N : -1 : 1
         s2 = 0;
         for v =  N: -1 : j+1
            s2 = s2 + Y(v,i)*U(j,v);
         end
         Y(j,i) = ( X(j,i) - s2 ) * ( 1 / U(j,j) );
   end
end
% check2shouldbeX2 = U*Y ;
% X2 = X ;
% checkbeEye2 = L1*U*Y ;
end

%% Zadanie 4 i 5
function B = MacierzB(k, N)
%B-zwracana wygenerowana zgodnie ze wzorem macierz
;analogicznie do
%MacierzA
B=zeros(N);
x = 2^k / 300; %zadana w poleceniu funkcja
B(1,1) = (x)^2; %macierz min. 1x1, więc wyrzucam przed pętlę
for i=N:-1:1   %Iteracja wierszy (od końca)
   for j=N:-1:1   %Iteracja kolumn (od końca)
      if (i>=2 && j>=2)
         B(i,j) = min(i,j)*4/9;
      elseif ((i==1 && j>1) || (j==1 && i>1 ))
         B(i,j)=(2/3)*x;
      end
   end
end
end

%Norma druga i nieskonczona
function [deltaSr, deltaMx ]= wskBlad(B)
   deltaSr = max(sqrt(eigs(B' * B)));
   deltaMx = max(sum(abs(B), 2));
end

%Wykresy wzkaźników
%inicjalizacja zmiennych
z = 1000;
N = [3 10 20];
k = 0:1:21;
K = 22;
B = zeros(z);
X = 2.^k / 300;

det2_inv = zeros(2, K);
det2_LLt = zeros(2, K);
det2_LU = zeros(2, K);
detInf_inv = zeros(2, K);
detInf_LLt = zeros(2, K);
detInf_LU = zeros(2, K);

for i = 1 : length(N)
   z=0;
   for j = 1 : length(k)
      z=z+1;
      %generacja macierzy jednostkowej i permutacji w
zależności od
      %rozmiaru N
      I = eye(N(i));
      P = zeros(N(i));
      B = MacierzB(k(j),N(i));

% dla polecenia wbudowanego inv
      B_inv = inv(B);
      BB_inv = B*B_inv - I;
      [deltaSr, deltaMx ]= wskBlad(BB_inv);
      det2_inv(1, z) = deltaSr;
      detInf_inv(1, z) = deltaMx;
      det2_inv(2, z) = norm(BB_inv, 2);
      detInf_inv(2, z) = norm(BB_inv, inf);


% dla metody LLt
      B_LLt = wyznaczOdwrotnaLLt(B);
      BB_LLt = B*B_LLt - I;
      [deltaSr, deltaMx ]= wskBlad(BB_LLt);
      det2_LLt(1, z) = deltaSr;
      detInf_LLt(1,z) = deltaMx;
      det2_LLt(2, z) = norm(BB_LLt, 2);
      detInf_LLt(2, z) = norm(BB_LLt, inf);

% dla metody LU
      [B_LU, P] = wyznaczOdwrotnaLU(B);
      BB_LU = B*B_LU*P - I;
      [deltaSr, deltaMx ]= wskBlad(BB_LU);
      det2_LU(1, z) = deltaSr;
      detInf_LU(1, z) = deltaMx;
      det2_LU(2, z) = norm(BB_LU, 2);
      detInf_LU(2,z) = norm(BB_LU, inf);
   end

   figure;
%     loglog(X, det2_inv(1,:),'-o', X, det2_LLt(1,:),'-o', X,
det2_LU(1,:),'-o');
   loglog(X, det2_inv(1,:)./X,'-o', X, det2_LLt(1,:)./X,'-o', X,
det2_LU(1,:)./X,'-o');
      %dla bledu wzglednego
   xlabel('Wartości X(k)');
   ylabel('Wartości błędu średniokwadratowego względnego');
   title('Wykres wskaźnika błędu średniokwadratowego
względnego');
   legend('błąd śred. inv','błąd śred. LLt','błąd śred.
LU','Location', 'Best');
%       legend('błąd śred. inv','błąd śred. LLt','błąd śred.
LU','Location', 'Best');
   grid on;
   hold on;

   figure;
%     loglog(X, detInf_inv(1,:),'-o', X, detInf_LLt(1,:),'-o', X,
detInf_LU(1,:),'-o');
   loglog(X, detInf_inv(1,:)./X,'-o', X, detInf_LLt(1,:)./X,'-o', X,
detInf_LU(1,:)./X,'-o');
   xlabel('Wartości X(k)');
   ylabel('Wartości błędu maksymalnego względnego');
   title('Wykres wskaźnika błędu maksymalnego względnego');
   legend('błąd max. inv','błąd max. LLt','błąd max. LU',
'Location', 'Best');
%       legend('błąd max. inv','błąd max. LLt','błąd max. LU',
'Location', 'Best');
   grid on;
   hold on;

%       figure;
%     semilogy(X, det2_inv(2,:), X, det2_LLt(2,:), X,
det2_LU(2,:));
%     xlabel('Alfa w przedziale alfa-0.01,alfa+0.01');
%     ylabel('Wartości błędu');
%     title('Wykres wskaźnika błędu średniokwadratowego');
%     legend('wyznacznik macierzy N=3','wyznacznik macierzy
N=10','wyznacznik macierzy N=20', 'Best', 'Best');
```

```
%    grid on;
%    hold on;
%
%    figure;
%    semilogy(X, detInf_inv(2,:), X, detInf_LLt(2,:), X,
detInf_LU(2,:));
%    xlabel('Alfa w przedziale alfa-0.01,alfa+0.01');
%    ylabel('Wartości błędu maksymalnego');
%    title('Wykres wskaźnika błędu maksymalnego');
%    legend('wskaźnik uwarunkowania N=3','wskaźnik
uwarunkowania N=10','wskaźnik uwarunkowania N=20', 'Best',
'Best');
%    grid on;
%    hold on;
end
```

8. Bibliografia

1. Morawski Roman, „Metody numeryczne (MNUB) Materiały do wykładu prowadzonego w semestrze zimowym 2018/2019"
2. Morawski Roman, „Zadania z rozwiązaniami do przedmiotów WNUM MNUB"
3. Pratap Rudra, „MATLAB 7 dla naukowców i inżynierów", Warszawa, PWN, 2013
4. „Wstęp do metod numerycznych dla studentów elektroniki i technik informacyjnych" pod red. Romana Z. Morawskiego, Warszawa, Oficyna wydawnicza Politechniki Warszawskiej, 2009
5. www.mathworks.com/help/matlab/ref/axis.html?searchHighlight=axis&s_tid=doc_srchtitle
6. www.mathworks.com/help/matlab/ref/for.html?searchHighlight=for&s_tid=doc_srchtitle
7. www.mathworks.com/help/matlab/ref/grid.html?searchHighlight=grid&s_tid=doc_srchtitle
8. www.mathworks.com/help/matlab/ref/linspace.html?searchHighlight=linspace&s_tid=doc_srchtitle
9. www.mathworks.com/help/matlab/ref/norm.html?searchHighlight=norm&s_tid=doc_srchtitle
10. www.mathworks.com/help/matlab/ref/size.html?s_tid=srchtitle