Projekt nr 2

Krzysztof Kosz

January 11, 2023

Contents

1 Treść zadania

Rozwiązywanie układu równań liniowych $Ax = b$, gdzie $A(n \text{ x } n)$ jest macierzą postaci

$$A = \begin{pmatrix} C & S \\ -S & C \end{pmatrix},$$

gdzie $C, S(p \text{ x } p)$ i $n = 2p$. Zakładamy, że $C = diag(c_1, \ldots, c_p)$ i $detC \neq 0$ oraz $S = diag(s_1, \ldots, s_p$, gdzie $c_i^2 + s_i^2 = 1$ dla $i = 1, \ldots, p$.

Zaimplementować klasyczną metodę Jacobiego dla $Ax = b$.

2 Opis metody

Do rozwiązywania układu równań skorzystam z metody Jacobiego.

2.1 Metoda Jacobiego

Aby metoda miała sens musimy założyć, że elementy na głównej przekątnej macierzy A są niezerowe. Wiemy, z założeń, że w przypadku naszej macierzy jest to spełnione.

Zapiszmy układ równań $Ax = b$ w nastepującej postaci:

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2$$

$$\cdots$$

$$a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n$$

Wyznaczmy z pierwszego równania x_1, z drugiego x_2 itd.:

$$x_1 = (b_1 - \sum_{j=2}^{n} a_{1j}x_j)/a_{11}$$

$$x_2 = (b_2 - \sum_{j=1, j\neq 2}^{n} a_{2j}x_j)/a_{22}$$

$$\cdots$$

$$x_n = (b_n - \sum_{j=1}^{n-1} a_{nj}x_j)/a_{nn}]$$

Otrzymane wzory są podstawą iteracji. Zaczynając z danego przybliżenia początkowego $x^{(0)} = (x_1^{(0)}, \ldots, x_n^{(0)})^T \in \mathbb{R}$ obliczamy kolejne przybliżenia $x^{(k+1)}(k = 0, 1, \ldots)$ według wzorów:

$$x_1^{(k+1)} = (b_1 - \sum_{j=2}^{n} a_{1j}x_j^{(k)})/a_{11}$$

$$x_2^{(k+1)} = (b_2 - \sum_{j=1, j\neq 2}^{n} a_{2j}x_j^{(k)})/a_{22}$$

$$\cdots$$

$$x_n^{(k+1)} = (b_n - \sum_{j=1}^{n-1} a_{nj}x_j^{(k)})/a_{nn}$$

Zapiszemy te wzory w postaci macierzowej:

$$x^{(k+1)} = B_J x^{(k)} + c_J,$$

gdzie

$$x^{(k)} = \begin{pmatrix} x_1^{(k)} \\ \vdots \\ x_n^{(k)} \end{pmatrix},$$

$$B_J = \begin{pmatrix} 0 & -\frac{a_{12}}{a_{11}} & \cdots & -\frac{a_{1n}}{a_{11}} \\ -\frac{a_{21}}{a_{22}} & 0 & \cdots & -\frac{a_{2n}}{a_{22}} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{a_{n1}}{a_{nn}} & -\frac{a_{n2}}{a_{nn}} & \cdots & 0 \end{pmatrix},$$

$$c_J = \begin{pmatrix} \frac{b_1}{a_{11}} \\ \vdots \\ \frac{b_n}{a_{nn}} \end{pmatrix}.$$

Macierz B_J jest macierzą iteracji w metodzie Jacobiego.

2.2 Sprawdzanie zbieżności metody Jacobiego

Warunkiem koniecznym i dostatecznym zbieżności jest $\rho(B_J) < 1$, gdzie $\rho(B) = \max_{\lambda \in \sigma(B)} |\lambda|$.

Warunkiem zakończenia naszych obliczeń będzie natomiast $||x^{(k+1)} - x^{(k)}|| < d$ lub ilość wykonanych iteracji.

3 Opis programu

W moim programie stosuję 2 funkcje. Jedną do wyznaczania rozwiązania ukłądu równań, a drugą do tworzenia macierzy, dla których metoda Jacobiego jest zbieżna.

3.1 Funkcja Jacobi

Jako argumenty tej funkcji podajemy macierz A oraz wektor b, dla których obliczamy $Ax = b$ (argumenty bez których funkcja nie zadziała) oraz argumenty, które mają przyjętą wartość domyślną w razie ich nie podania tj. d czyli wartość warunku końcowego zakończenia obliczeń (domyślnie 10^{-40}), x czyli wartość przybliżenia początkowego (domyślnie wektor zer) oraz it czyli maksymalna ilość iteracji, po której kończymy obliczenia (domyślnie 1000).

Figure 1: Implementacja metody Jacobiego cz.I

```
function [wynik] = Jacobi(A, b, d, x, it)
%Jacobi - funkcja przyjmuje 5 argumentów i w wyniku zwraca rozwiązanie
%układu równań (za pomocą metody Jacobiego) w postaci wektora
%   Argumenty:
%   A - macierz A współczynników układu równań
%   b - wektor b współczynników układu równań
%   d - wielkość warunku końcowego (różnica norm wektora wynikowego k+1-ej i
%   k-tej iteracji musi być mniejsza od d)
%   x - wektor przybliżenia początkowego x
%   it - maksymalna ilość iteracji
n = size(A);
n = n(1);
B = zeros(n);
c = zeros(n,1);
%Sprawdzanie ilości argumentów wejściowych
switch nargin
    case 2
        x = zeros(n,1);
        d = 10^(-40);
        it = 1000;
    case 3
        x = zeros(n,1);
        it = 1000;
    case 4
        it = 1000;
end
%Wyznaczanie wartości wektora c i macierzy B służących do wyznaczania
%wektorów x w kolejnych iteracjach
```

Figure 2: Implementacja metody Jacobiego cz.II

```
for i = 1:n
    c(i) = b(i)/A(i,i);
    for j = 1:n
        if i~=j
            B(i,j) = -(A(i,j)/A(i,i));
        end
    end
end
y=eig(B);
prom=max(abs(y));
if prom >=1
    disp('Metoda Jacobiego nie jest zbieżna dla danej macierzy')
   return
end
lastx = [1:n]';
i=1;
solvex=linsolve(A,b);
%Wyznaczanie kolejnych waryości wektora x
while abs(norm(lastx-x))>d && i<it
    lastx = x;
    x = B*x + c;
    iteracja(i)=i;
    norma(i)=sum(abs(x-solvex));
    i=i+1;
end
%Wykres obrazujący zależność błędu względnego w zależności od numeru
%iteracji
plot(iteracja,norma)
xlabel("Numer iteracji");
ylabel("Błąd bezwzględny aktualnej wartości wyniku");
title("Zależność błędu bezwględnego wyniku w zależności od iteracji");
wynik=x;
end
```

W ciele naszej funkcji najpierw sprawdzamy czy zostały podane wszystkie argumenty. Jeśli nie to przyjmujemy wartości domyślne argumentów.

Następnie wyznaczamy macierz iteracji Jacobiego oraz macierz c_J.

Kolejnym krokiem jest sprawdzenie warunku koniecznego i dostatecznego zbieżności metody Jacobiego. W przypadku gdy promień spektralny macierzy iteracji jest większy bądź równy jeden od razu kończymy działanie funkcji i wypisujemy informację o braku zbieżności.

W końcu wyznaczamy kolejne wartości wektora wynikowego oraz sprawdzając warunek zakończenia obliczeń przy każdej iteracji. Dodatkowo tworzymy wektor, który potem posłuży nam do stworzenia wykresu.

Na sam koniec tworzymy wykres, który pomoże nam zwizualizować zbieżność metody Jacobiego.

3.2 Funkcja testMaker

Jako argument tej funkcji przyjmujemy n czyli ilość kolumn macierzy C oraz S.

Figure 3: Tworzenie macierzy, dla których metoda Jacobiego jest zbieżna

```
function [matrix] = testMaker(n)
%testMaker - tworzy macierz, dla której metoda Jacobiego będzie zbieżna
%   Argumenty które przyjmuje, to n czyli połowa ilości kolumn/wierszy
%   macierzy
vectc=(-pi/6)+(pi/3).*rand(1,n);
C=diag(cos(vectc));
S=diag(sin(vectc));
matrix=[C S; -S C];
end
```

Jako iż $c_i^2+s_i^2=1$ dla $i=1,\dots,p$ uznałem, że najłatwiej będzie tworzyć testy za pomocą wektorów sinusów i cosinusów dla tych samych wartości.

Metodą prób i błędów zaobserwowałem zależność, która wskazywała iż metoda Jacobiego jest zbieżna dla macierzy typu podanego w zadaniu, gdy $c_i > s_i$, dlatego tak też tworzę testy.

4 Przykłady

Pokażę kilka przykładów dla macierzy o różnych wymiarach, dla których metoda Jacobiego jest zbieżna oraz takie, dla których nie jest zbieżna. Zaprezentuję również wykresy zbieżności do faktycznej wartości rozwiązania.

Uznałem, że takie wykresy będą ciekawsze niż prezentowanie błędu w zależności od wielkości warunku kończącego obliczenia, ponieważ jak się można spodziewać im większy warunek kończący obliczenia tym większy błąd obliczeń.

W poniższych przykładach An oraz bn to macierze i wektory z n-tego przykładu, prom oznacza promień spektralny macierzy iteracji Jacobiego, wynikn to rozwiązanie n-tego przykładu, bladWzgledny to błędy względne komórek wektora rozwiązania względem wektora rozwiązania wbudowaną funkcją w matlabie linsolve().

Od pewnego momentu ograniczyłem się tylko do wypisywania promienia spektralnego oraz błędu względnego w formie średniego błędu bezwzględnego komórek wektora rozwiązań.

Natomiast w wykresach stosuję określenie błąd bezwzględny. Jest to suma błędów bezwględnych aktualnego wektora rozwiązania względem wektora rozwiązania wbudowaną w matlabie funkcją linsolve().

4.1 Przykład nr 1

Figure 4: Przyklad nr 1

```
A1 =

    0.1670         0         0         0         0   -0.9859         0         0         0         0
         0    0.8028         0         0         0         0   -0.5962         0         0         0
         0         0   -0.8738         0         0         0         0    0.4863         0         0
         0         0         0   -0.5105         0         0         0         0   -0.8599         0
         0         0         0         0   -0.3074         0         0         0         0   -0.9516
    0.9859         0         0         0         0    0.1670         0         0         0         0
         0    0.5962         0         0         0         0    0.8028         0         0         0
         0         0   -0.4863         0         0         0         0   -0.8738         0         0
         0         0         0    0.8599         0         0         0         0   -0.5105         0
         0         0         0         0    0.9516         0         0         0         0   -0.3074


b1 =

   91.7194
   28.5839
   75.7200
   75.3729
   38.0446
   56.7822
    7.5854
    5.3950
   53.0798
   77.9167
```

Figure 5: Przykład nr 1

```
prom =

    5.9023

Metoda Jacobiego nie jest zbieżna dla danej macierzy
```

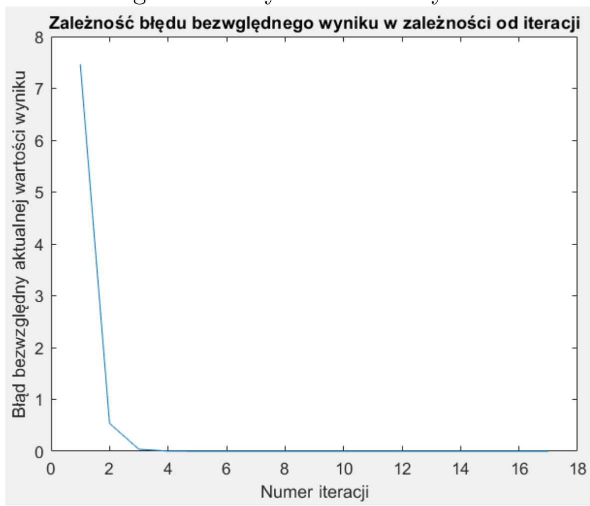
Jak widzimy promień spektralny jest większy od 1, więc funkcja wypisuje wiadomość iż Metoda jacobiego nie jest zbieżna dla tej macierzy.

4.2 Przykład nr 2

Figure 6: Przyklad nr 2

```
A2 =

    0.9974   -0.0718
    0.0718    0.9974


b2 =

   91.0648
   18.1847


prom =

    0.0719


wynik2 =

   92.1349
   11.6029


bladWzgledny =

     0
     0
```

Figure 7: Przykład nr 2 - wykres



Widzimy, że przy tak małej macierzy błąd względny jest na tyle mały, że w matlabie jest reprezentowany jako wartość 0 na obu miejscach wektora wynikowego.

4.3 Przykład nr 3

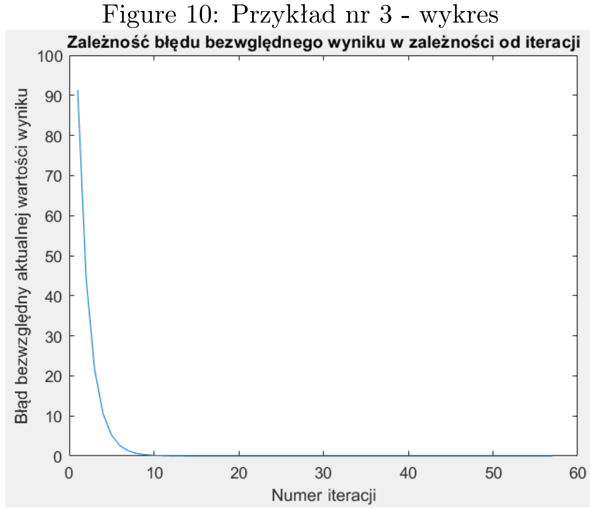
Figure 8: Przyklad nr 3

```
A3 =

    0.8908         0   -0.4543         0
         0    0.9124         0    0.4093
    0.4543         0    0.8908         0
         0   -0.4093         0    0.9124


b3 =

   94.4787
   49.0864
   48.9253
   33.7719


prom =

    0.5100


wynik3 =

  106.3930
   30.9617
    0.6618
   50.9057
```

Figure 9: Przyklad nr 3

```
bladWzgledny =

   1.0e-14 *

         0
         0
    0.4865
         0
```

Figure 10: Przykład nr 3 - wykres



W tym przykładzie mamy macierz $2x2$. Widzimy, że pojawia nam się błąd względny, jednak jest on rzędu 10^{-15}, więc jest niewielki.

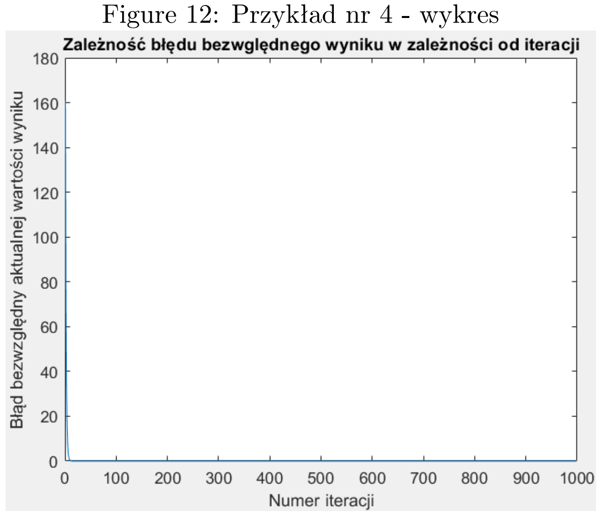
4.4 Przykład nr 4

Od tego przykładu wypisywałem tylko promień spektralny macierzy oraz błąd względny w postaci średniego błędu względnego.

Figure 11: Przyklad nr 4

```
prom =

    0.5195


bladWzgledny =

   3.3128e-17
```

Figure 12: Przykład nr 4 - wykres



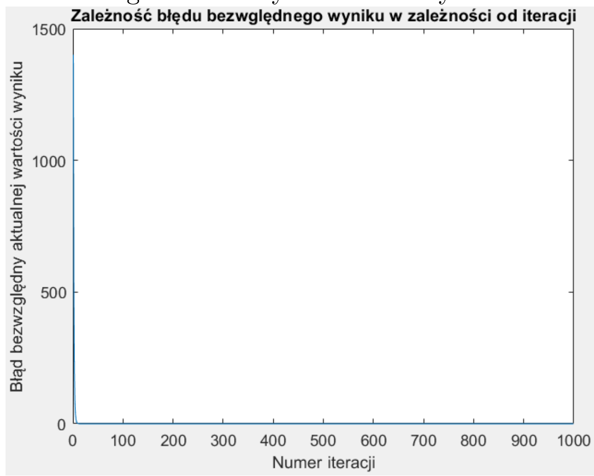
W tym przypadku mamy macierz rozmiaru $10x10$. Błąd względny jest rzędu 10^{-17}.

4.5 Przykład nr 5

Figure 13: Przyklad nr 5

```
prom =

    0.5665


bladWzgledny =

   1.0973e-16
```

Figure 14: Przykład nr 5 - wykres



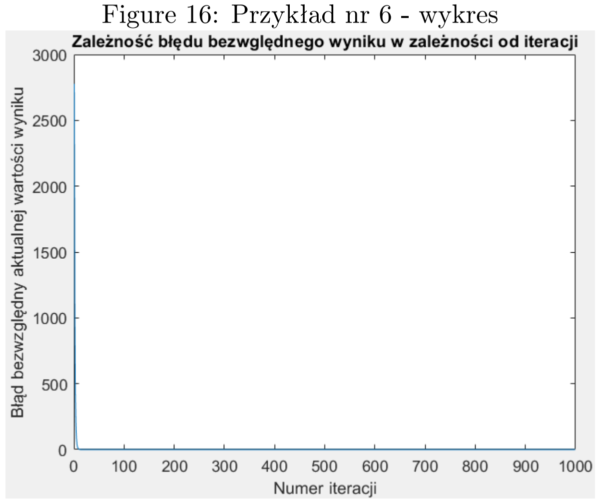
W tym przypadku mamy macierz rozmiaru $100x100$. Błąd względny jest rzędu 10^{-16}.

4.6 Przykład nr 6

Figure 15: Przyklad nr 6

```
prom =

    0.5563


bladWzgledny =

   1.0693e-16
```

Figure 16: Przykład nr 6 - wykres



W tym przypadku mamy macierz rozmiaru $200x200$. Błąd względny jest rzędu 10^{-16}.

4.7 Przykład nr 7

Figure 17: Przyklad nr 7

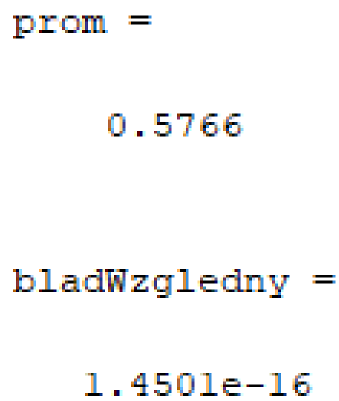
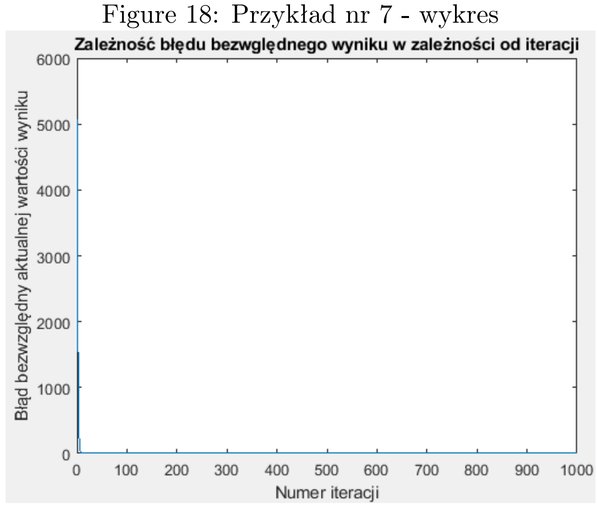


Figure 18: Przykład nr 7 - wykres



W tym przypadku mamy macierz rozmiaru $400x400$. Błąd względny jest rzędu 10^{-16}.

5 Analiza uzyskanych wyników

Jak widzimy błędy w przypadku naszych macierzy są naprawdę małe, co pokazuje że metoda Jacobiego jest naprawdę całkiem dokładna.

Jednakże przy tworzeniu testów zauważyłem, że w wielu przypadkach metoda po prostu nie jest zbieżna. Można to bez problemu również zauważyć, ponieważ promień spektralny ma być mniejszy od 1.

Podsumowując metoda Jacobiego jest zbieżna dla małej, specyficznej grupy macierzy, ale gdy już jest zbieżna jest całkiem dokładna.

6 Dodatek

Chciałem skorzystać z tego, że typ macierzy, z której korzystałem w tym zadaniu ma w wielu miejscach 0, tak więc myślałem, że może szukanie rozwiązania oddzielnie dla każdego układu równań może być szybsze:

$$c_{i,i}x_i + s_{i,n+i}x_{n+i} = b_i$$

$$-s_{n+i,i}x_i + c_{n+i,n+i}x_{n+i} = b_{n+i}$$

Dla $i = 1, \ldots, n$.

Jednak po zmierzeniu czasu dla zwykłego rozwiązania i dla rozwiązania, które rozdziela układ równań w sposób jaki zaproponowałem okazało się, że ten sposób jest trochę wolniejszy, więc porzuciłem ten pomysł.