Metody numeryczne
Wykład 4

Dr inż. Michał Łanczont
Instytut Elektrotechniki i Elektrotechnologii
E419, tel. 4293, m.lanczont@pollub.pl, http://m.lanczont.pollub.pl

Zakres wykładu

Metody iteracyjne rozwiązywania układów równań liniowych:

1) Richardsona
2) Jacobiego
3) Gaussa-Seidela
4) Nadrelaksacji
5) Czebyszewa
6) Gradientowa

Linsolve

- W środowisku obliczeniowym Scilab dostępna jest funkcja rozwiązująca dowolne układy równań liniowych zapisanych w postaci równania macierzowego **Ax+b=0**
- Składnia polecenia:

```
x=linsolve(A,b);
```

- Gdzie: parametrami wejściowymi jest macierz **A** i wektor **b**. Macierz i wektor muszą mieć zgodną liczbę kolumn i wierszy. Macierz **A** jest macierzą kwadratową.
- Przy powszechnie stosowanym zapisie układu równań liniowych **Ax=b** funkcję wywołuje się:

```
x=linsolve(A,-b);
```

- W efekcie działania otrzymujemy wektor wartości stanowiących rozwiązanie układu równań.

Oszacowanie błędu

Za pomocą funkcji Scilab'a wyznaczającego normę można w prosty sposób oszacować błąd obliczeń sprawdzając rozwiązanie równania:

d=norm(A*x-b);

```
1  clear;
2  clc;
3  A1=[1,1/2,-1/2,1;1/4,0.8,-1/2,1/4;1/2,-1/2,0.8,3/4;1/5,3/4,-1/2,1];
4  b1=[1;2;-2;-1];
5  A2=[6,-2,2,4;12,-8,6,10;3,-13,9,3;-6,4,1,-18];
6  b2=[12;34;27;-38];
7  A3=[2,0.2,1,0.5;2,6,1,-1;0.5,-1,3,-0.2;3,-2,1,8];
8  b3=[2;6;-3;-1];
9  x1=linsolve(A1,-b1);
10 x2=linsolve(A2,-b2);
11 x3=linsolve(A3,-b3);
12 p1=norm(A1*x1-b1);
13 p2=norm(A2*x2-b2);
14 p3=norm(A3*x3-b3);
15 disp(p1,'Błąd obliczeń układu nr 1');
16 disp(p2,'Błąd obliczeń układu nr 2');
17 disp(p3,'Błąd obliczeń układu nr 3');
```

```
Błąd obliczeń układu nr 1

  1.018D-15

Błąd obliczeń układu nr 2

  1.137D-14

Błąd obliczeń układu nr 3

  3.353D-15
```

Metody iteracyjne

- W praktyce obliczeniowej pojawiają się jednak dosyć często układy równań liniowych, których wymiar (macierzy **A**) jest rzędu 10^3 lub nawet większy.
- Prawie zawsze macierze wielkich układów liniowych nie są macierzami pełnymi, ale rzadkimi, tzn. mają niewiele elementów niezerowych.
- Jednym ze sposobów rozwiązywania wielkich układów równań jest stosowanie metod iteracyjnych.

Metody iteracyjne

- Metody iteracyjne zwracają przybliżony wynik rozwiązania po określonej liczbie kroków, lub po osiągnięciu żądanej dokładności
- W metodach iteracyjnych dochodzi się wyznaczając kolejne przybliżenia rozwiązania.
- Wymagane jest więc przyjęcie wartości początkowych obliczeń,

Metody iteracyjne

- Określenie ε jako warunku zbieżności uzyskanego rozwiązania (określana jest graniczna różnica pomiędzy kolejnymi przybliżeniami rozwiązania)

- Jeżeli oczekiwana dokładność ε nie jest zbyt duża można bardzo szybko dojść do rozwiązania.

Metody iteracyjne

Znalezienie rozwiązania układu równań liniowych opisanych równaniem macierzowym:

$$Ax=b$$

sprowadza się do przekształcenia powyższego równania do postaci:

$$x^*=Bx+c$$

gdzie sposób wyznaczenia macierzy **B** i wektora **c** jest uzależniony od zastosowanej metody.

Warunkiem wystarczającym zbieżności metody iteracyjnej jest aby dowolna norma macierzy **B** była mniejsza od jedności.

Ogólna metoda iteracyjna

- Dla ustalonej macierzy **Q** równanie **Ax=b** można zapisać w postaci iteracyjnej:

$$x^{(k)} = (I - Q^{-1}A)\,x^{(k-1)} + Q^{-1}b, \text{ dla } k \geq 1$$

- Wektor początkowy $x^{(0)}$ może być dowolny, ale wskazane jest ustawienie wartości przewidywanych
- Metoda iteracyjna jest zbieżna, jeżeli ciąg $\{x^{(k)}\}$ jest zbieżny do **x** dla dowolnego wektora startowego $x^{(0)}$

Ogólna metoda iteracyjna

Macierz Q powinna spełniać więc następujące warunki:

1. Obliczanie przybliżeń $x^{(k)}$ powinno być matematycznie łatwe
2. Ciąg $\{x^{(k)}\}$ powinien być szybko zbieżny

Warunek będzie spełniony w sytuacji gdy macierz Q^{-1} jest dobrym przybliżeniem macierzy A^{-1}

Aby istniało rozwiązanie macierze A i Q muszą być nieosobliwe.

Metoda iteracyjna

1.

$$\begin{bmatrix} 6 & -2 & 2 & 4 \\ 12 & -8 & 6 & 10 \\ 3 & -13 & 9 & 3 \\ -6 & 4 & 1 & -18 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 12 \\ 34 \\ 27 \\ -38 \end{bmatrix}$$

$$\begin{cases} 6x_1 - 2x_2 + 2x_3 + 4x_4 = 12 \\ 12x_1 - 8x_2 + 6x_3 + 10x_4 = 34 \\ 3x_1 - 13x_2 + 9x_3 + 3x_4 = 27 \\ -6x_1 + 4x_2 + x_3 - 18x_4 = -34 \end{cases} \rightarrow \begin{cases} x_1 = \dfrac{12 + 2x_2 - 2x_3 - 4x_4}{6} \\ x_2 = \dfrac{34 - 12x_1 - 6x_3 - 10x_4}{-8} \\ x_3 = \dfrac{27 - 3x_1 + 13x_2 - 3x_4}{9} \\ x_4 = \dfrac{-34 + 6x_1 - 4x_2 - x_3}{-18} \end{cases}$$

$$x = Bx + c, gdzie: \quad B = \begin{bmatrix} 0 & \frac{1}{3} & -\frac{1}{3} & -\frac{2}{3} \\ \frac{3}{2} & 0 & \frac{3}{4} & \frac{5}{4} \\ -\frac{1}{3} & \frac{13}{9} & 0 & -\frac{1}{3} \\ -\frac{1}{3} & \frac{2}{9} & -\frac{1}{18} & 0 \end{bmatrix}, c = \begin{bmatrix} 2 \\ -\frac{17}{4} \\ 3 \\ \frac{17}{9} \end{bmatrix} \rightarrow \begin{array}{l} \|B\|_1 = 2.25 \\ \|B\|_2 = 2.27 \\ \|B\|_\infty = 3.5 \end{array}$$

Dla tak wyznaczonej macierzy B, rozwiązanie iteracyjne może nie być zbieżne.

Metody skończone

$$\begin{bmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 0 & -3 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 12 \\ 10 \\ -9 \\ -3 \end{bmatrix}$$

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} x_1 = \dfrac{12 - (4 \cdot 1 + 2 \cdot -2 - 2 \cdot -3)}{6} = 1 \\ x_2 = \dfrac{10 - (2 \cdot 1 + 2 \cdot -2)}{-4} = -3 \\ x_3 = \dfrac{-9 - (-5 \cdot 1)}{2} = -2 \\ x_4 = \dfrac{-3}{-3} = 1 \end{bmatrix} = \begin{bmatrix} 1 \\ -3 \\ -2 \\ 1 \end{bmatrix}$$

Metoda iteracyjna

2.

$$\begin{bmatrix} 1 & 0.5 & -0.5 & 1 \\ 0.25 & 0.8 & -0.5 & 0.25 \\ 0.5 & -0.5 & 0.8 & 0.75 \\ 0.2 & 0.75 & -0.5 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ -2 \\ -1 \end{bmatrix}$$

$$\begin{cases} x_1 + 0.5x_2 - 0.5x_3 + x_4 = 1 \\ 0.25x_1 + 0.8x_2 - 0.5x_3 + 0.25x_4 = 2 \\ 0.5x_1 - 0.5x_2 + 0.8x_3 - 0.75x_4 = -2 \\ 0.2x_1 + 0.75x_2 - 0.5x_3 + x_4 = -1 \end{cases} \rightarrow \begin{cases} x_1 = 1 - (0.5x_2 - 0.5x_3 + x_4) \\ x_2 = \dfrac{2 - (0.25x_1 - 0.5x_3 + 0.25x_4)}{0.8} \\ x_3 = \dfrac{-2 - (0.5x_1 - 0.5x_2 - 0.75x_4)}{0.8} \\ x_4 = -1 - (0.2x_1 + 0.75x_2 - 0.5x_3) \end{cases}$$

$$x = Bx + c, gdzie\ B = \begin{bmatrix} 0 & -0.5 & 0.5 & -1 \\ -0.3125 & 0 & 0.625 & -0.3125 \\ -0.625 & 0.625 & 0 & 0.9375 \\ -0.2 & -0.75 & 0.5 & 0 \end{bmatrix} i\ c = \begin{bmatrix} 1 \\ 1.25 \\ -2.5 \\ -1 \end{bmatrix}, \begin{matrix} \|B\|_1 = 2.25 \\ \|B\|_2 = 1.78 \\ \|B\|_\infty = 2.19 \end{matrix}$$

Dla tak wyznaczonej macierzy B, rozwiązanie iteracyjne może nie być zbieżne.

Metody iteracyjne

3.

$$\begin{bmatrix} 2 & 0.2 & 1 & 0.5 \\ 2 & 6 & 1 & -1 \\ 0.5 & -1 & 3 & -0.2 \\ 3 & -2 & 1 & 8 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 2 \\ 6 \\ -3 \\ -1 \end{bmatrix}$$

$$\begin{cases} 2x_1 + 0.2x_2 + x_3 + 0.5x_4 = 2 \\ 2x_1 + 6x_2 + x_3 - x_4 = 6 \\ 0.5x_1 - x_2 + 3x_3 - 0.2x_4 = -3 \\ 3x_1 - 2x_2 + x_3 + 8x_4 = -1 \end{cases} \rightarrow \begin{cases} x_1 = \dfrac{2 - 0.2x_2 - x_3 - 0.5x_4}{2} \\ x_2 = \dfrac{6 - 2x_1 - x_3 + x_4}{6} \\ x_3 = \dfrac{-3 - 0.5x_1 + x_2 + 0.2x_4}{3} \\ x_4 = \dfrac{-1 - 3x_1 + 2x_2 - x_3}{8} \end{cases}$$

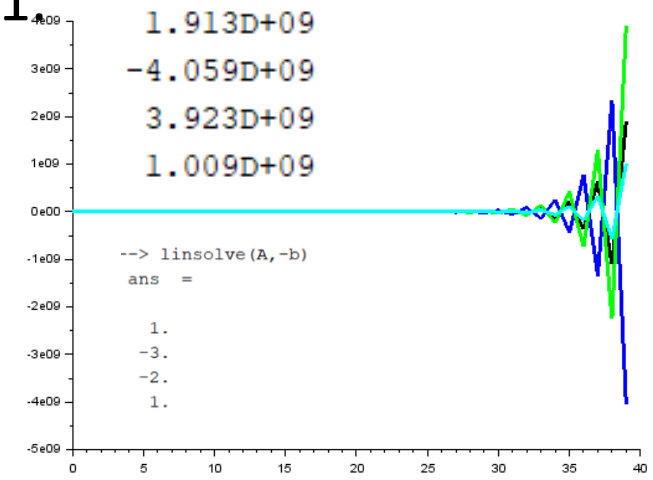
$$x = Bx + c, gdzie: \quad B = \begin{bmatrix} 0 & -0.1 & -0.5 & -0.25 \\ -\frac{1}{3} & 0 & -\frac{1}{6} & \frac{1}{6} \\ -\frac{1}{6} & \frac{1}{3} & 0 & \frac{1}{15} \\ -\frac{3}{8} & 0.25 & -0.125 & 0 \end{bmatrix}, c = \begin{bmatrix} 1 \\ 1 \\ -1 \\ -\frac{1}{8} \end{bmatrix} \rightarrow \begin{array}{l} \|B\|_1 = 0.875 \\ \|B\|_2 = 0.658 \\ \|B\|_\infty = 0.85 \end{array}$$

Dla tak wyznaczonej macierzy B, rozwiązanie iteracyjne będzie zbieżne.

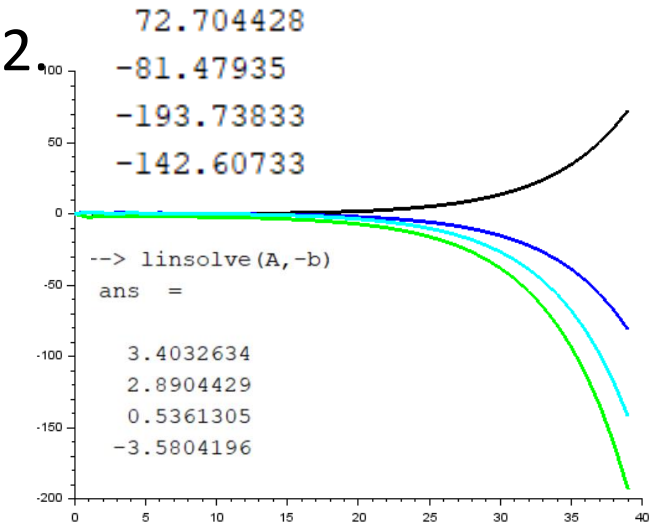
Metody iteracyjne


```
13 for k=2:M
14     for i=1:n
15         for j=1:n
16             x(i,k)=x(i,k)+B(i,j)*x(j,k-1)
17         end
18         x(i,k)=x(i,k)+c(i);
19     end
20 end
```

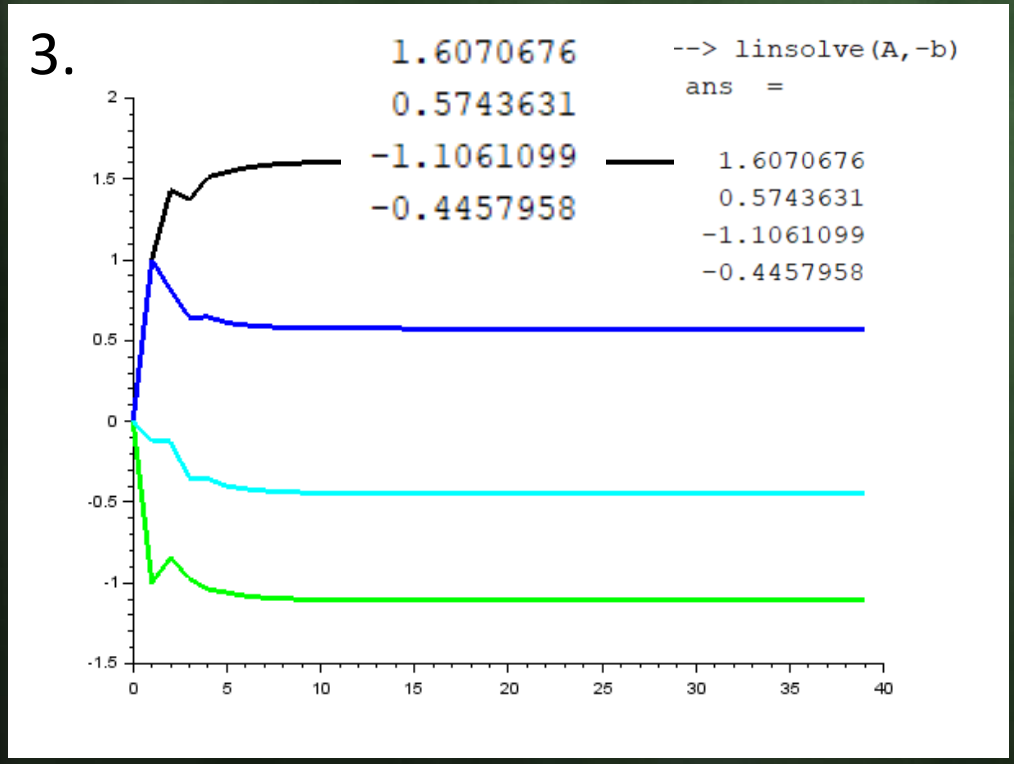
1.



2.



3.



Metoda Richardsona

W metodzie Richardsona za macierz **Q** przyjmuje się macierz jednostkową

Równanie iteracyjne przyjmuje postać:

$$x^{(k)} = (I - A)x^{(k-1)} + b$$

$$x^{(k)} = x^{(k-1)} + r^{(k-1)}$$

gdzie $r^{(k-1)} = b - Ax^{(k-1)}$

Warunkiem zbieżności rozwiązania jest spełnienie nierówności $||I-A||<1$

Algorytm metody Richardsona

Dane wejściowe $x^{(0)}$, A, b, n – liczba niewiadomych, M – liczba iteracji

$$for\ k = 1\ to\ M$$

$$for\ i = 1\ to\ n$$

$$r_i = b_i - \sum_{j=1}^{n} a_{ij} x_{i-1}$$

$$for\ i = 1\ to\ n$$

$$x_i = x_{i-1} + r_i$$

Przykład 1

11:25

Metodą Richardsona rozwiązać układ równań liniowych.

```
clear;
clc;
n=4
A=[1,1/2,-1/2,1;1/4,0.8,-1/2,1/4;1/2,-1/2,0.8,3/4;1/5,3/4,-1/2,1];
b=[1;2;-2;-1];
I=eye(n,n);
r=[0;0;0;0];
M=50;
t=[0:M-1];
x=zeros(n,M);
for k=1:M-1
    for i=1:n
        c=0;
        for j=1:n
            c=c+A(i,j)*x(j,k);
        end
        r(i)=b(i)-c;
    end;
    for i=1:n
        x(i,k+1)=x(i,k)+r(i);
    end
end
plot2d(t,[x(1,:)',x(2,:)',x(3,:)',x(4,:)']);
```

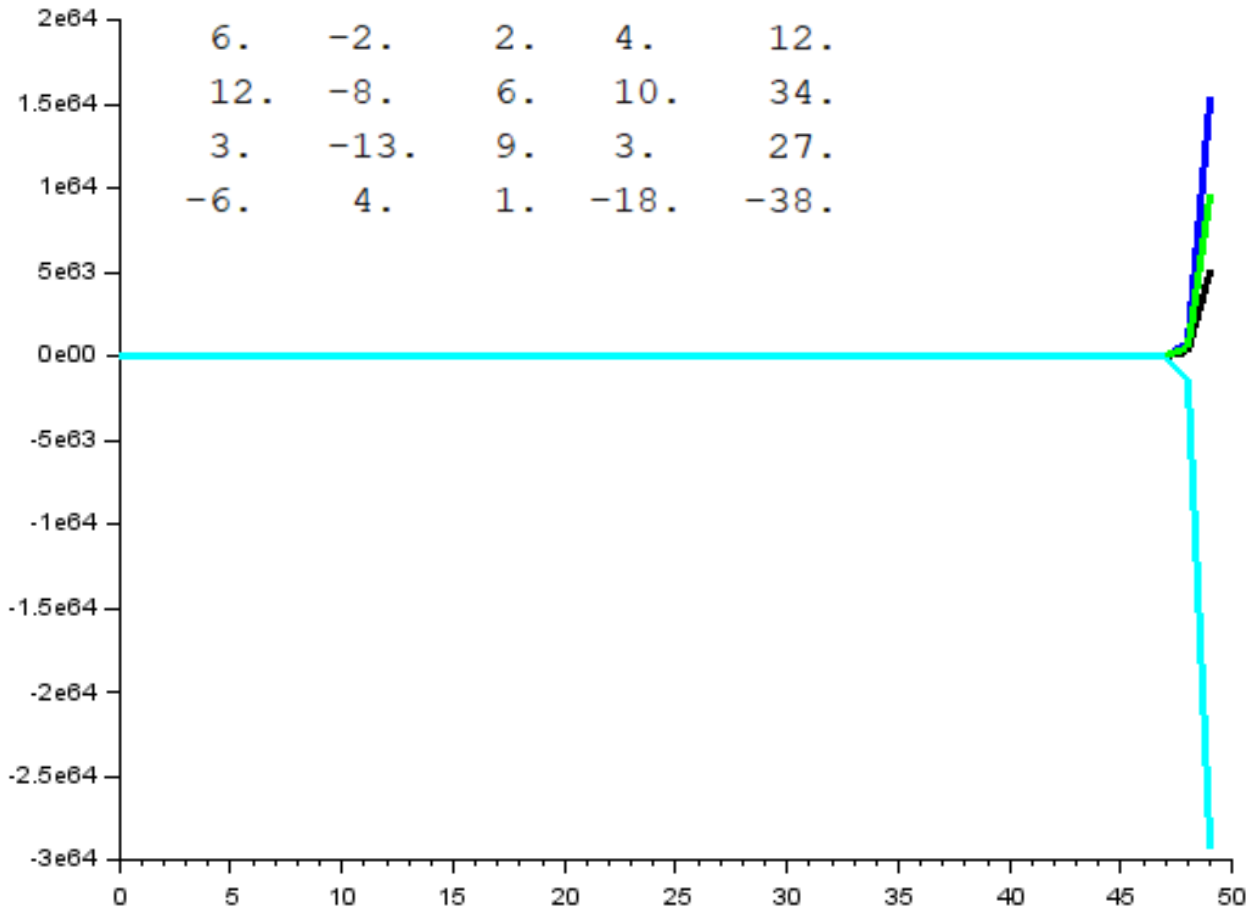
```
for k=1:M-1
    for i=1:n
        c=0;
        for j=1:n
            c=c+A(i,j)*x(j,k);
        end
        r(i)=b(i)-c;
    end;
    for i=1:n
        x(i,k+1)=x(i,k)+r(i);
    end
end
```

```
for i=1:M
    x=(I-A)*xm+b;
    xm=x;
end
```

Przykład 1

1.

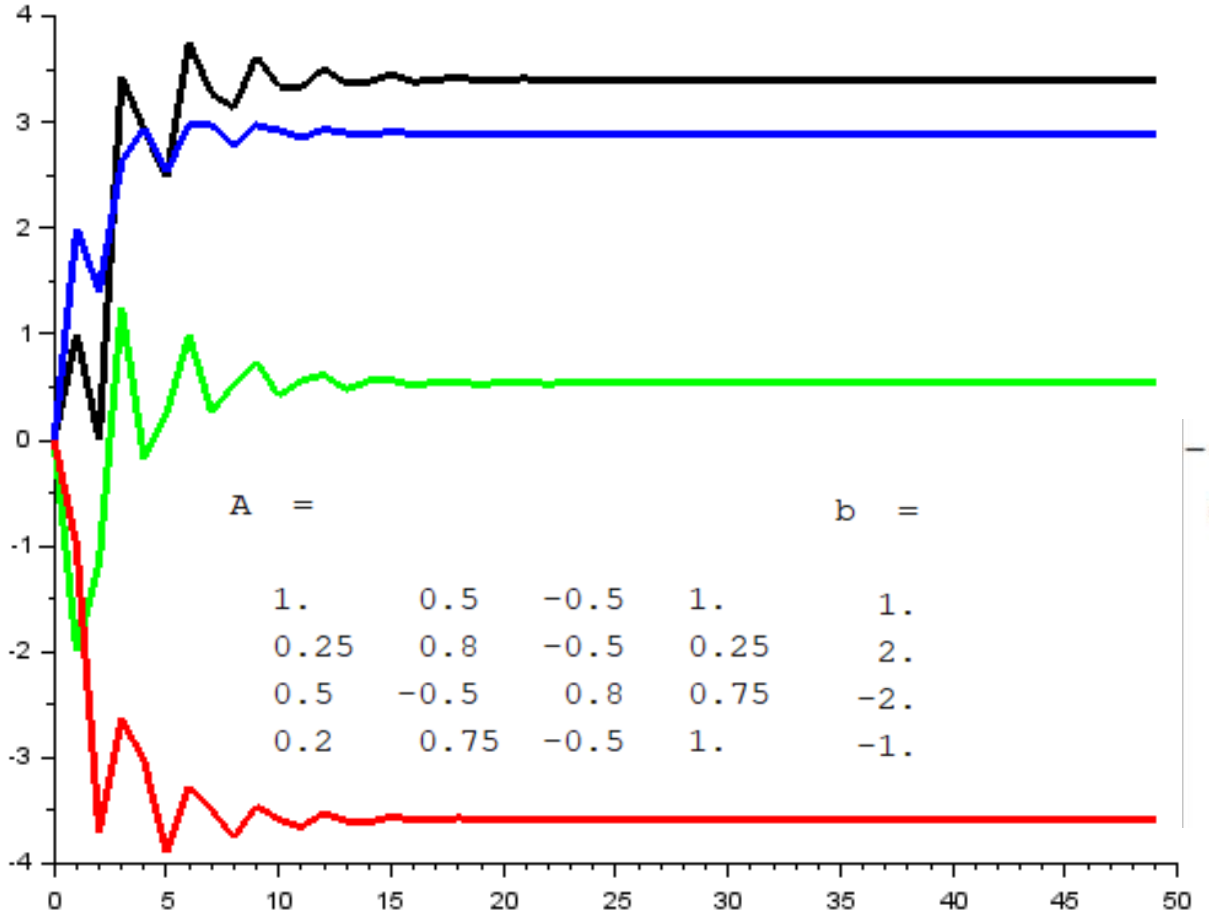
```
A  =                              b  =

    6.    -2.    2.    4.      12.
   12.    -8.    6.   10.      34.
    3.   -13.    9.    3.      27.
   -6.     4.    1.  -18.     -38.
```




```
--> x(:,40)
 ans  =

   4.103D+50
   1.241D+51
   7.758D+50
  -2.371D+51


--> linsolve(A,-b)
 ans  =

   1.
  -3.
  -2.
   1.
```

Przykład 1

2.

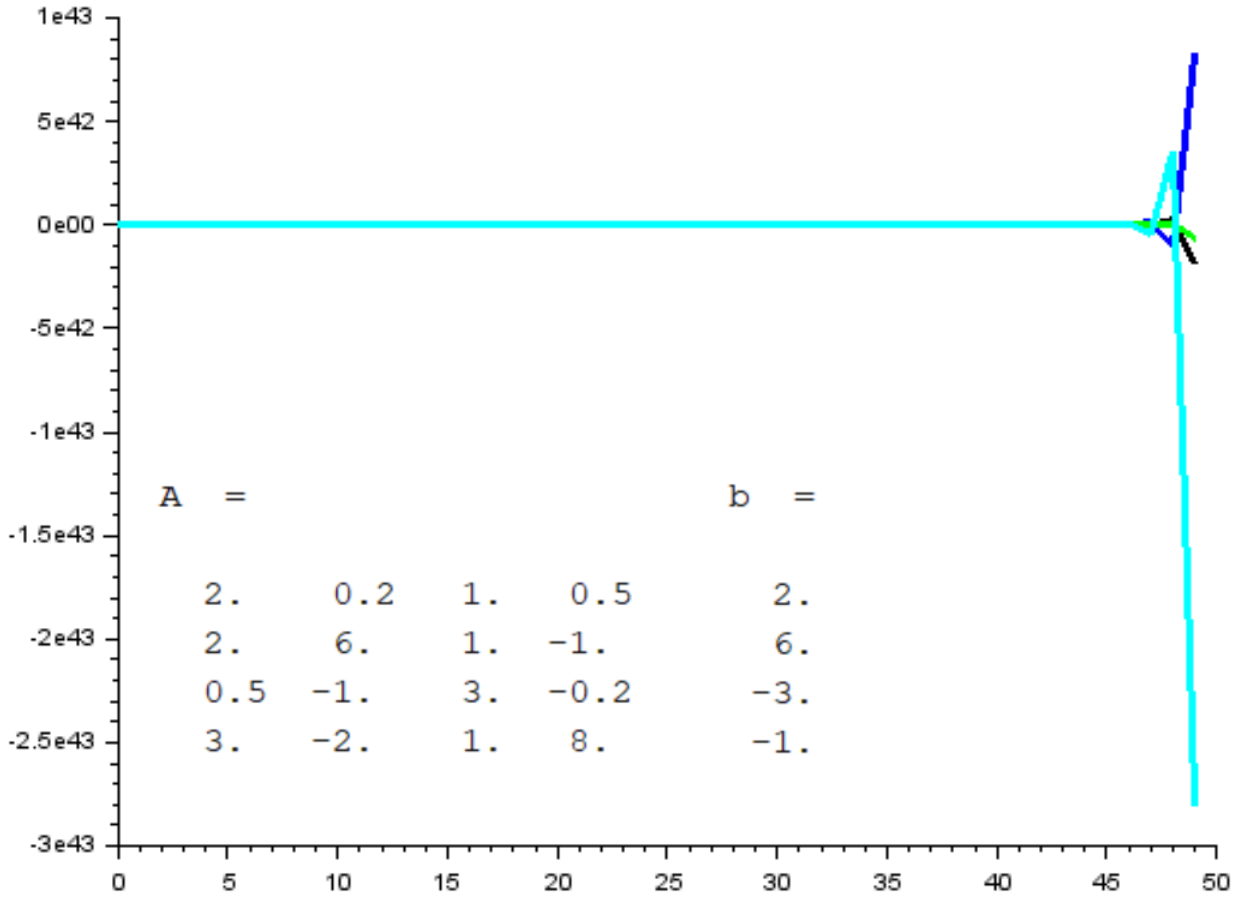



```
 3.4032637
 2.8904423
 0.5361324
-3.5804201
```

```
--> linsolve(A,-b)
 ans  =

   3.4032634
   2.8904429
   0.5361305
  -3.5804196
```

Przykład 1

3.




```
--> x(:,40)
 ans  =

  -2.219D+33
   9.746D+33
  -7.415D+32
  -3.269D+34
```

```
--> linsolve(A,-b)
 ans  =

   1.6070676
   0.5743631
  -1.1061099
  -0.4457958
```

Metoda Jacobiego

W pierwszym kroku zastępuje się macierz A układem równoważnym w postaci L + D + U

Gdzie L jest macierzą trójkątną dolną, D – macierzą diagonalną, a U macierzą trójkątną górną.

Układ równań w postaci Ax=b można więc zapisać jako:

$$(L+D+U)x=b$$

Metoda Jacobiego

Układ równań $(L+D+U)x=b$ można przekształcić do postaci:

$$Dx^{*}=-(L+U)x+b$$

Jeżeli macierz D jest nieosobliwa możemy układ równań zapisać jako:

$$x^{*}=-D^{-1}(L+U)x+D^{-1}b \rightarrow Bx+c$$

Istotne jest więc takie przekształcenie macierzy A (działaniami elementarnymi) aby na przekątnej nie było wyrazów 0.

Metoda Jacobiego

Równanie macierzowe można zapisać w postaci iteracyjnej:

$$Dx^{(i+1)} = -(L+U)x^{(i)} + b$$

otrzymujemy:

$$a_{kk}x_k^{(i+1)} = -\left(\sum_{j=1}^{k-1} a_{kj}x_j^{(i)} + \sum_{j=k+1}^{n} a_{kj}x_j^{(i)}\right) + b_k, dla\ k = 1,2,\dots,n$$

w efekcie końcowy dochodzimy do:

$$x_k^{(i+1)} = \frac{-\sum_{(j=1, j\neq k)}^{n} a_{kj}x_j^{(i)} + b_k}{a_{kk}}, dla\ k = 1,2,\dots,n;\ a_{kk} \neq 0$$

Algorytm metody Jacobiego

Dane wejściowe $x^{(0)}$, A, b, n – liczba niewiadomych, M – liczba iteracji

$$for\ k = 1\ to\ M$$
$$for\ i = 1\ to\ n$$
$$for\ j = 1\ to\ n$$
$$if\ j = i\ then\ continue$$
$$z_i = z_i + \frac{a_{ij}x_j}{a_{ii}}$$
$$x_i = b_i - z_i$$

$$z_i = \sum_{j=1(j\neq i)}^{n} \frac{a_{ij}x_j}{a_{ii}}$$

Przykład 2

Metodą Jacobiego rozwiązać układ równań liniowych.


```
clear;
clc;
M=100;
n=4;
x=zeros(n,M);
t=[0:M-1];
A=[1,1/2,-1/2,1;1/4,0.8,-1/2,1/4;1/
b=[1;2;-2;-1];
for k=2:M
    for i=1:n
        z=0;
        for j=1:n
            if j==i then
                continue;
            end
            z=z+(A(i,j)*x(j,k-1));
        end
        x(i,k)=(b(i)-z)/A(i,i);
    end
end
plot2d(t,[x(1,:)',x(2,:)',x(3,:)',x(4,:)'])
```

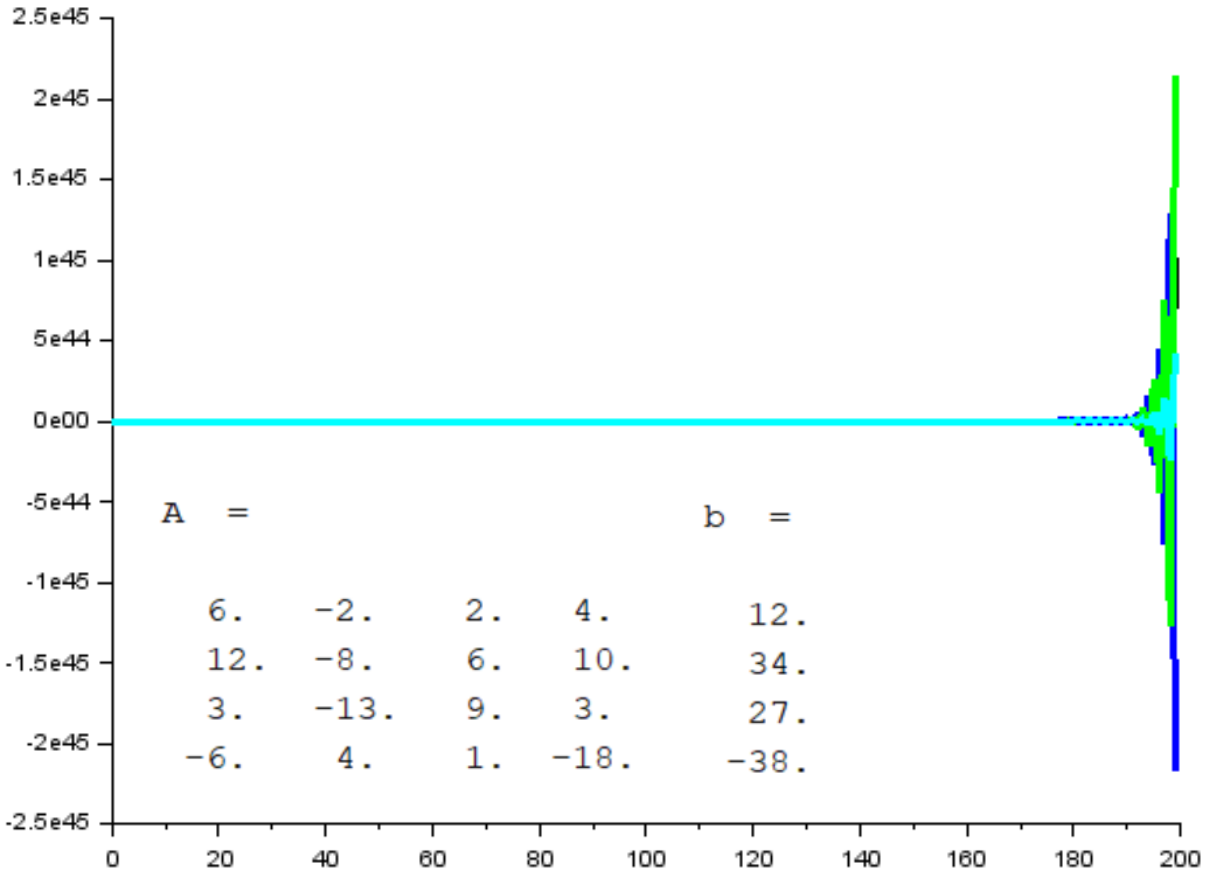
```
for k=2:M
    for i=1:n
        z=0;
        for j=1:n
            if j==i then
                continue;
            end
            z=z+(A(i,j)*x(j,k-1));
        end
        x(i,k)=(b(i)-z)/A(i,i);
    end
end
```

Przykład 2

```
clear;
clc;
A=[1,1/2,-1/2,1;1/4,0.8,-1/2,1/4;1/2,-1/2,0.8,3/4;1/5,3/4,-1/2,1];
b=[1;2;-2;-1];
M=100;
xm=[0;0;0;0];
L=[0,0,0,0;1/4,0,0,0;1/2,-1/2,0,0;1/5,3/4,-1/2,0];
D=diag([1,0.8,0.8,1]);
U=[0,1/2,-1/2,1;0,0,-1/2,1/4;0,0,0,3/4;0,0,0,0];
for i=1:M
    x=-D^-1*(L+U)*xm+D^-1*b;
    xm=x;
end
disp(x);
```

Przykład 2

1.

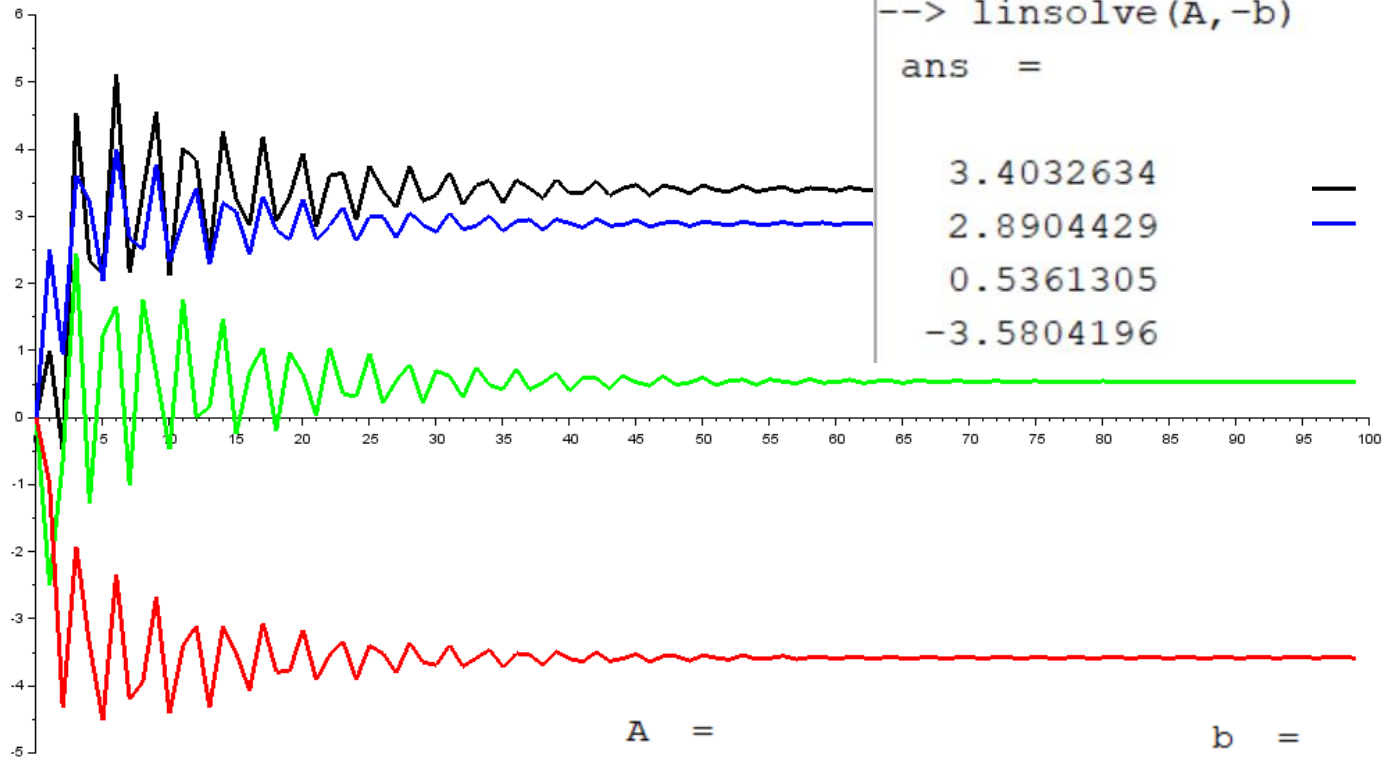


1.018D+45
-2.170D+45
2.144D+45
4.170D+44

1.
-3.
-2.
1.

Przykład 2

2.



```
   3.4048085
   2.8913311
   0.5368935
  -3.5793304

--> linsolve(A,-b)
 ans  =

   3.4032634
   2.8904429
   0.5361305
  -3.5804196
```

A =

```
   1.     0.5    -0.5    1.
   0.25   0.8    -0.5    0.25
   0.5   -0.5     0.8    0.75
   0.2    0.75   -0.5    1.
```

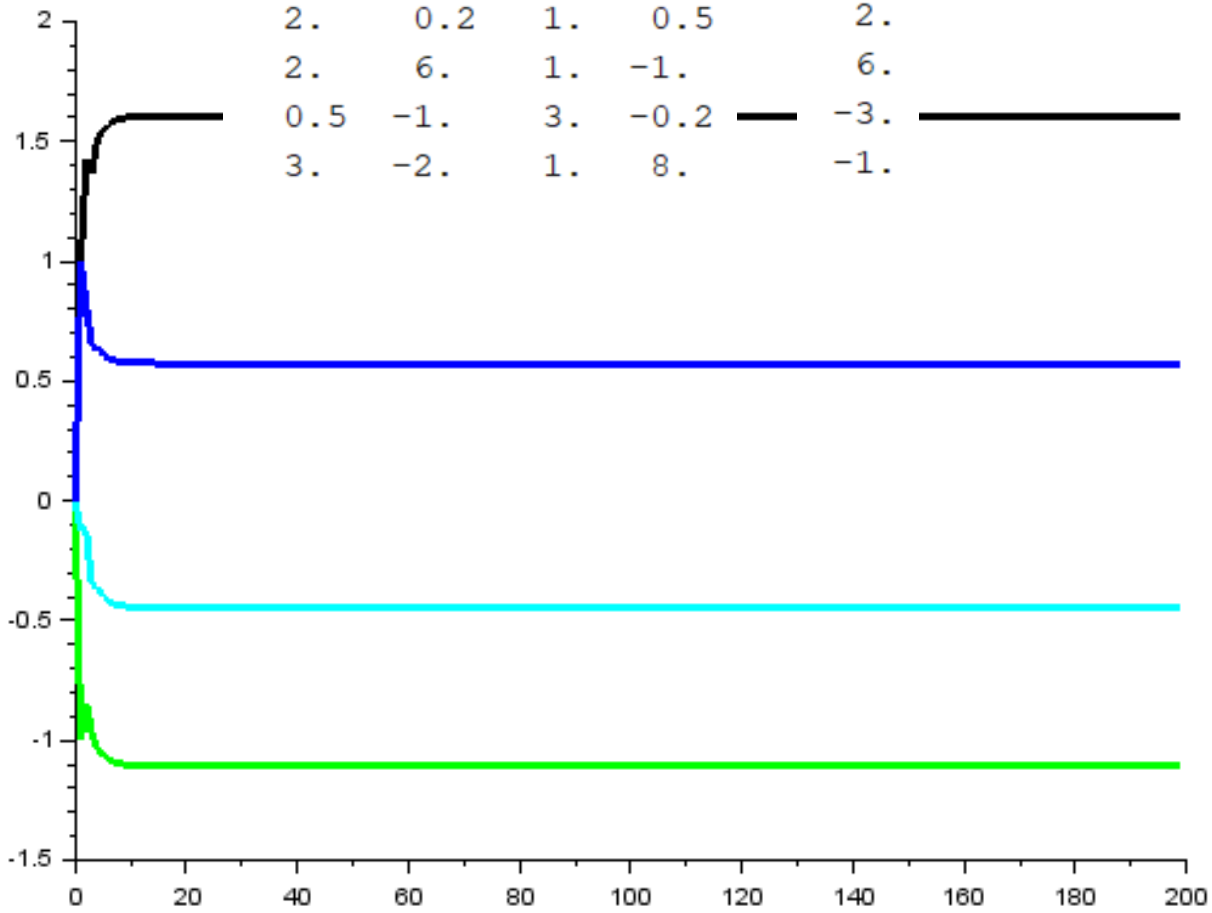
b =

```
   1.
   2.
  -2.
  -1.
```

Przykład 2

3.

```
A  =

   2.    0.2   1.    0.5
   2.    6.    1.   -1.
   0.5  -1.    3.   -0.2
   3.   -2.    1.    8.

b  =

   2.
   6.
  -3.
  -1.
```




```
 1.6070676
 0.5743631
-1.1061099
-0.4457958

 1.6070676
 0.5743631
-1.1061099
-0.4457958
```

Metoda Jacobiego - zbieżność

Rozwiązanie danego równania $Ax=b$ metodą Jacobiego jest zbieżne jeżeli macierz A jest nieredukowalna i diagonalnie dominująca.

Metoda Jacobiego

Macierz A jest nieredukowalna, jeżeli poprzez przestawienie wierszy i kolumn nie można jej sprowadzić do postaci blokowej górnej trójkątnej.

Macierz A o wymiarze n x n nazywamy diagonalnie dominującą, jeśli dla i=1,2,...,n zachodzi nierówność:

$$|a_{ii}| \geq \sum_{j=1\ (j\neq i)}^{n} |a_{ij}|$$

Metoda Gaussa-Seidela

Jest modyfikacją metody Jacobiego.

Wychodząc z równania:

$$x^* = (L+D+U)x+b,$$

można zależność tą zapisać w postaci:

$$(L+D)x^{(i+1)} = Ux^{(i)} + b,$$

a stąd:

$$x^{(i+1)} = -(L+D)^{-1}Ux^{(i)} + (L+D)^{-1}b$$

$$x^{(i+1)} = Bx^{(i)} + c$$

Aby macierz B było dobrze określona $a_{ii} \neq 0$ 33

Metoda Gaussa-Seidela

Ciąg rozwiązań można zapisać:

$$Dx^{(i+1)} = -Lx^{(i+1)} - Ux^{(i)} + b$$

$$\begin{bmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \end{bmatrix} \begin{bmatrix} x_1^{i+1} \\ x_2^{i+1} \\ x_3^{i+1} \end{bmatrix} = - \begin{bmatrix} 0 & 0 & 0 \\ a_{21} & 0 & 0 \\ a_{31} & a_{21} & 0 \end{bmatrix} \begin{bmatrix} x_1^{i+1} \\ x_2^{i+1} \\ x_3^{i+1} \end{bmatrix} - \begin{bmatrix} 0 & a_{12} & a_{13} \\ 0 & 0 & a_{23} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1^{i} \\ x_2^{i} \\ x_3^{i} \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$a_{11}x_1^{i+1} = -a_{12}x_2^{i} - a_{13}x_3^{i} + b_1$$

$$a_{22}x_2^{i+1} = -a_{21}x_1^{i+1} - a_{23}x_3^{i} + b_2$$

$$a_{33}x_3^{i+1} = -a_{31}x_1^{i+1} - a_{21}x_2^{i+1} + b_3$$

Metoda Gaussa-Seidela

Iteracyjną postać wyznaczania poszczególnych przybliżeń rozwiązania układu równań liniowych można po przekształceniach zapisać:

$$x_i^{(k+1)} = \frac{-\sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} + b_i}{a_{ii}}$$

$$gdzie\ i = 1,2, \dots, n$$

Algorytm metody Gaussa-Seidela

Dane wejściowe $x^{(0)}$, A, b, n – liczba niewiadomych, M – liczba iteracji

$$for\ k = 1\ to\ M$$
$$for\ i = 1\ to\ n$$
$$for\ j = 1\ to\ (i-1)$$
$$u_i = u_i + a_{ij}x_j^{(k)}$$
$$for\ j = (i+1)\ to\ n$$
$$v_i = v_i + a_{ij}x_j^{(k-1)}$$
$$x_i^{(k)} = \frac{b_i - u_i - v_i}{a_{ii}}$$

Przykład 3

Metodą Gaussa-Seidela rozwiązać układ równań liniowych.

```
for k=2:M
    for i=1:n
            u=0;
            v=0;
        for j=1:n
            if j>=i then continue; end;
            u=u+A(i,j)*x(j,k);
        end
        for j=1:n
            if j<=i then continue; end;
            v=v+A(i,j)*x(j,k-1);
        end
        x(i,k)=(b(i)-v-u)/A(i,i);
    end
end
```

```
for i=1:M
    x=(-(L+D)^-1)*U*xm+((L+D)^-1)*b;
    xm=x;
end
```

Przykład 3

1.

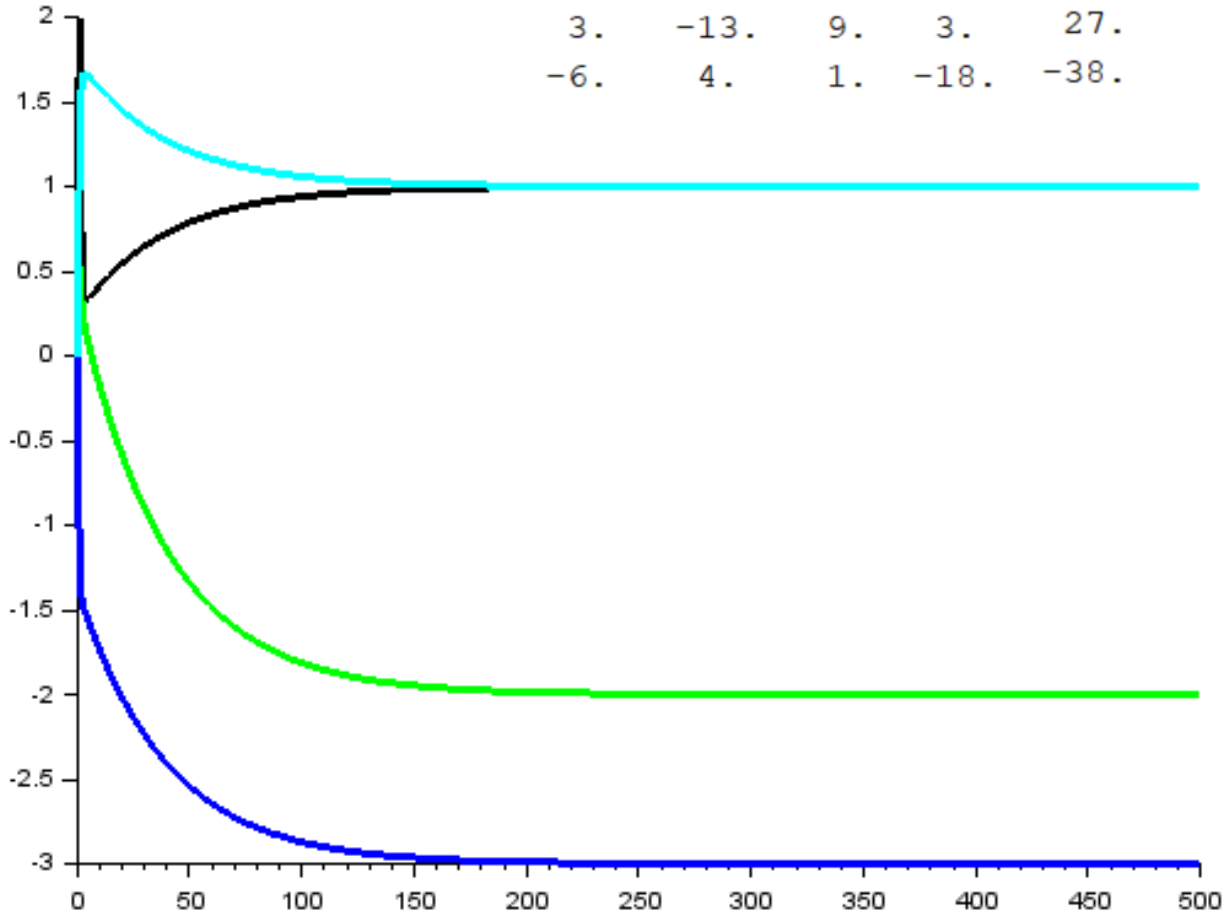
A =

6.	-2.	2.	4.	12.
12.	-8.	6.	10.	34.
3.	-13.	9.	3.	27.
-6.	4.	1.	-18.	-38.

b =

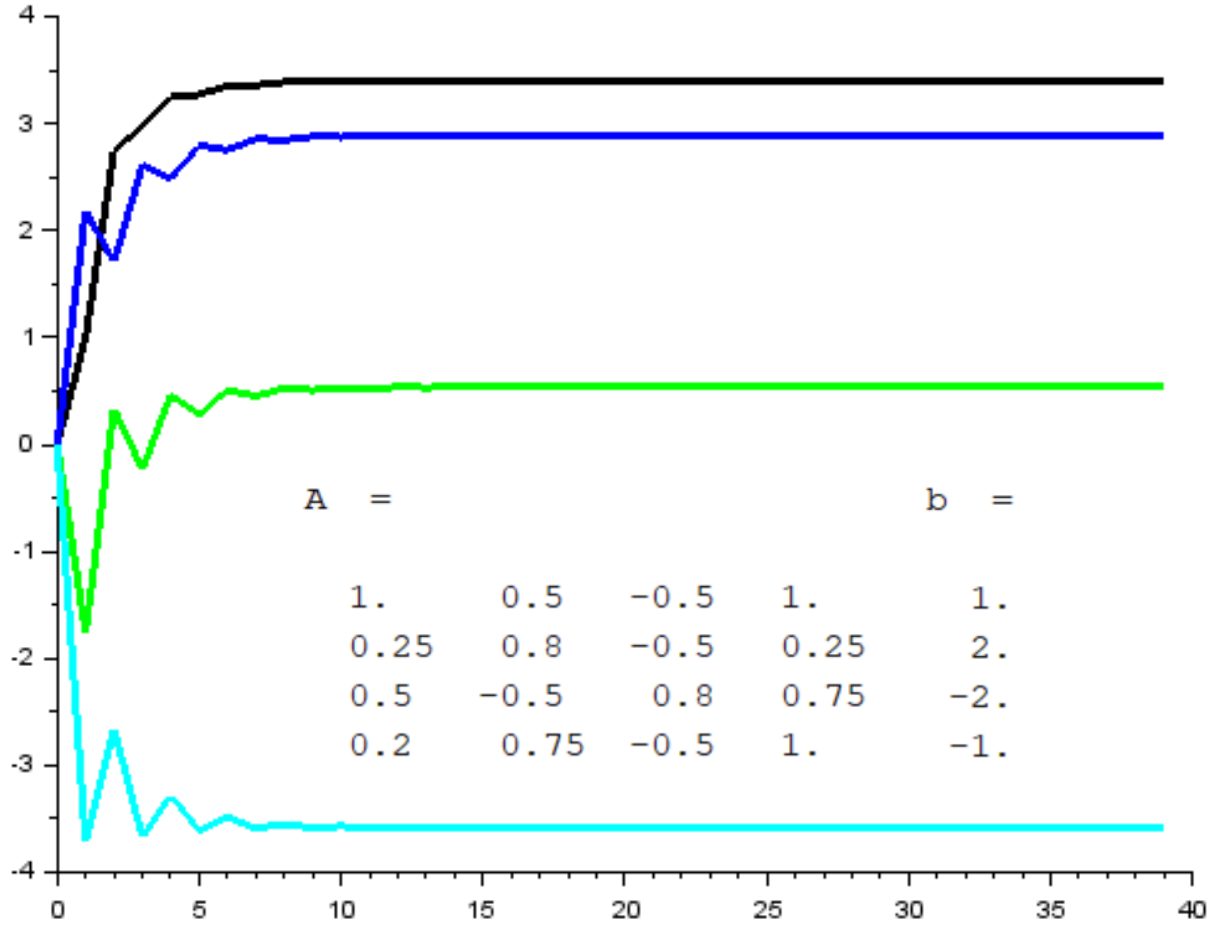
0.9999976
-2.9999948
-1.9999925
1.0000024

1.
-3.
-2.
1.



Przykład 3

2.




```
 3.4032634
 2.8904429
 0.5361305
-3.5804196

 3.4032634
 2.8904429
 0.5361305
-3.5804196
```

Przykład 3

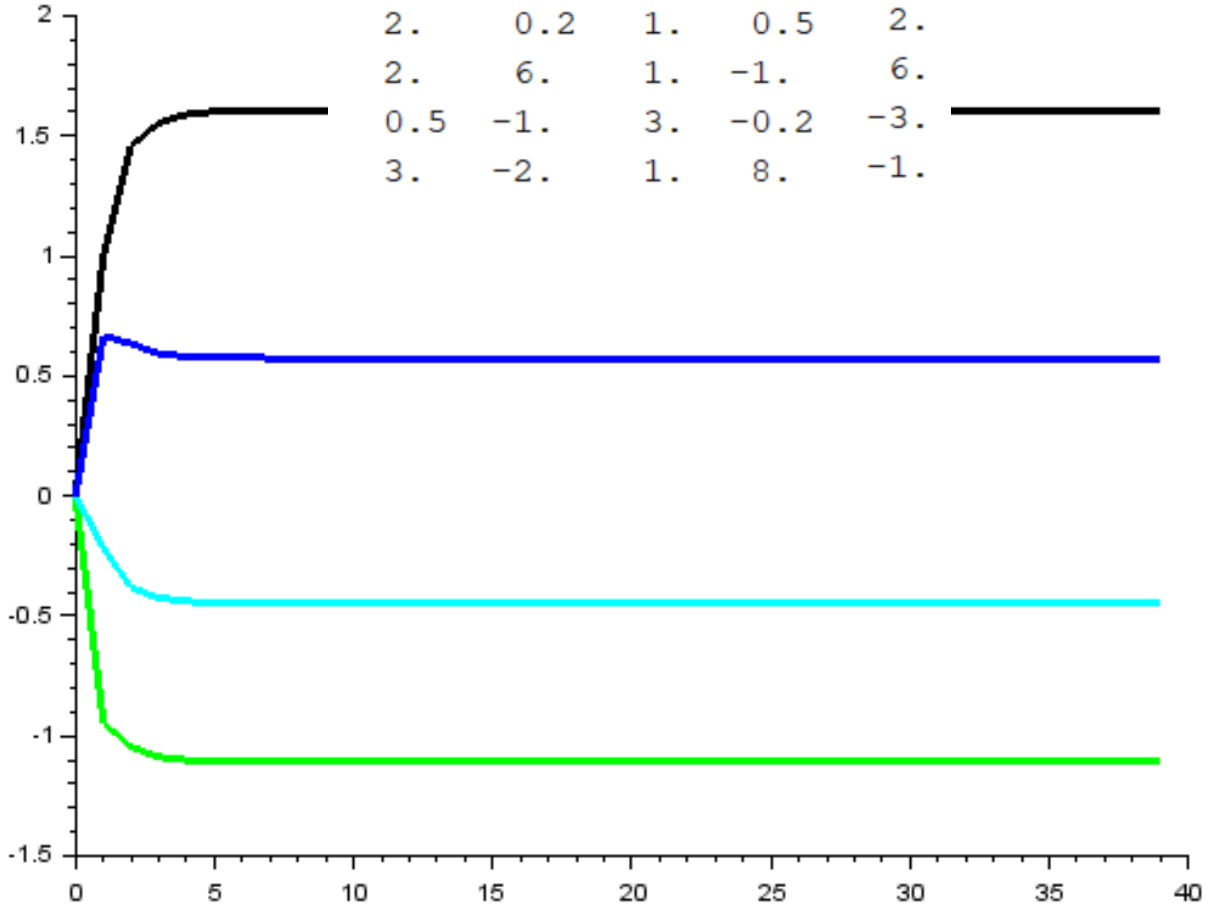
3.

A =

```
 2.    0.2   1.    0.5   2.
 2.    6.    1.   -1.    6.
 0.5  -1.    3.   -0.2  -3.
 3.   -2.    1.    8.   -1.
```

b =

```
 1.6070676
 0.5743631
-1.1061099
-0.4457958
```




```
 1.6070676
 0.5743631
-1.1061099
-0.4457958
```

Metoda nadrelaksacji

- Jest to zmodyfikowana metoda Gaussa-Seidela.
- Zakłada ona przyśpieszenie zbieżności osiągnięcia rozwiązania.
- Realizuje się to poprzez przemnożenie poprawki obliczeniowej przez odpowiednio dobraną liczbę ω.

Metoda nadrelaksacji

- Dla $\omega = 1$ metodę nadrelaksacji określa się mianem SOR - **Successive Over Relaxation**.
- Zwiększając ω można przyśpieszyć osiągnięcie zbieżności. Dopuszczalne wartości są z zakresu (0,2) .
- Dla innych wartości ω metoda może nie być zbieżna.

Metoda nadrelaksacji

Wychodząc z równania analogicznego do metody Gaussa-Seidela:

$$\omega(L+U+D)x=\omega b$$

$$Dx^{*}=Dx^{*}-\omega[(L+D+U)x-b]$$

$$(D+\omega L)x^{(i+1)}=(1-\omega)Dx^{(i)}-\omega Ux^{(i)}+\omega b$$

$$Dx^{(i+1)}=(1-\omega)Dx^{(i+1)}-\omega(Lx^{(i)}+Ux^{(i)}-b)$$

Metoda nadrelaksacji

W efekcie można zapisać:

$$x^{(i+1)} = B_\omega x^{(i)} + c$$

gdzie:

$$B_\omega = (D+\omega L)^{-1}((1-\omega)D-\omega U)$$

$$c = \omega(D+\omega L)^{-1} b$$

Metoda nadrelaksacji

Iteracyjnie równanie można zapisać:

$$a_{ii}x_i^{(k+1)} = (1-\omega)a_{ii}x_i^{(k)} +$$

$$+\omega\left(-\sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)} + b_i\right)$$

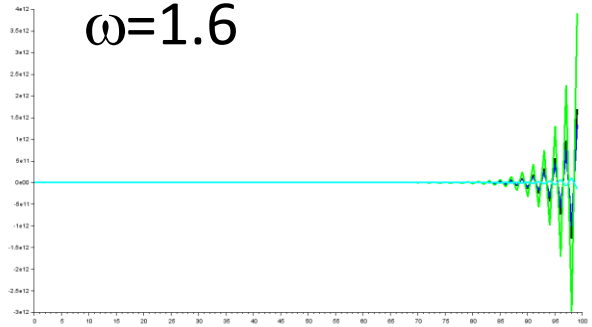
Przykład 4

Metodą nadrelaksacji rozwiązać układ równań.

```
for k=2:M
    for i=1:n
            u=0;
            v=0;
        for j=1:n
            if j>=i then continue; end;
            u=u+A(i,j)*x(j,k);
        end
        for j=1:n
            if j<=i then continue; end;
            v=v+A(i,j)*x(j,k-1);
        end
        x(i,k)=((1-w)*A(i,i)*x(i,k-1)+w*(b(i)-u-v))/A(i,i);
    end
end
```

```
for k=1:M
    x=((D+w*L)^-1)*((1-w)*D-w*U)*xm+w*((D+w*L)^-1)*b;
    xm=x;
```

Przykład 4



11:25

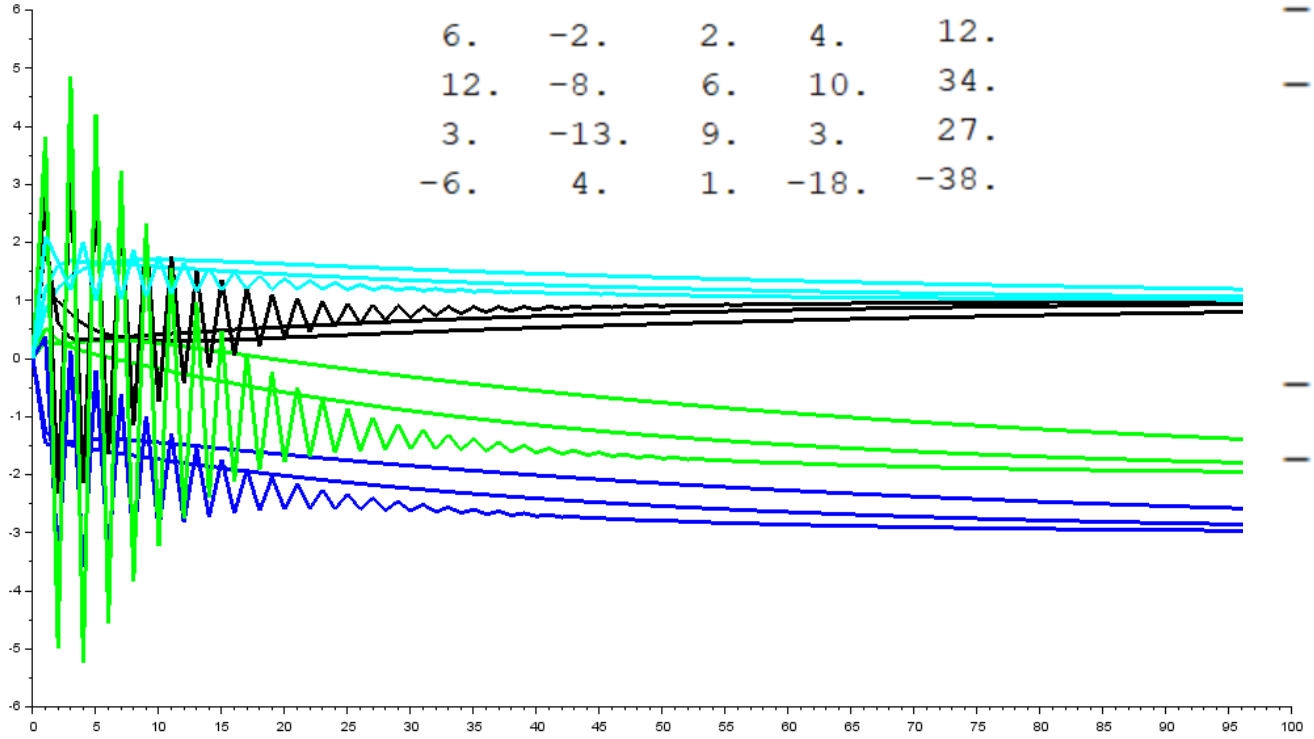
1. ω=0.6;1;1.5

A =

6.	-2.	2.	4.	12.
12.	-8.	6.	10.	34.
3.	-13.	9.	3.	27.
-6.	4.	1.	-18.	-38.

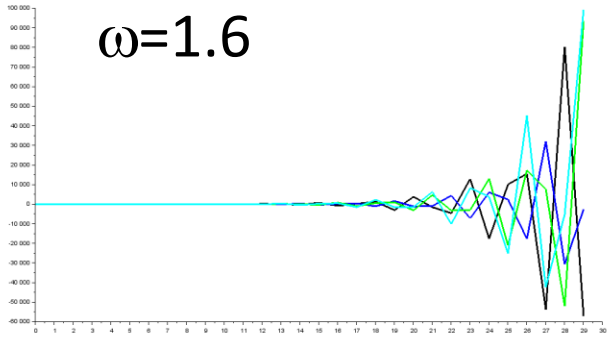
b =

0.9859471
-2.9686063
-1.9553679
1.0139561



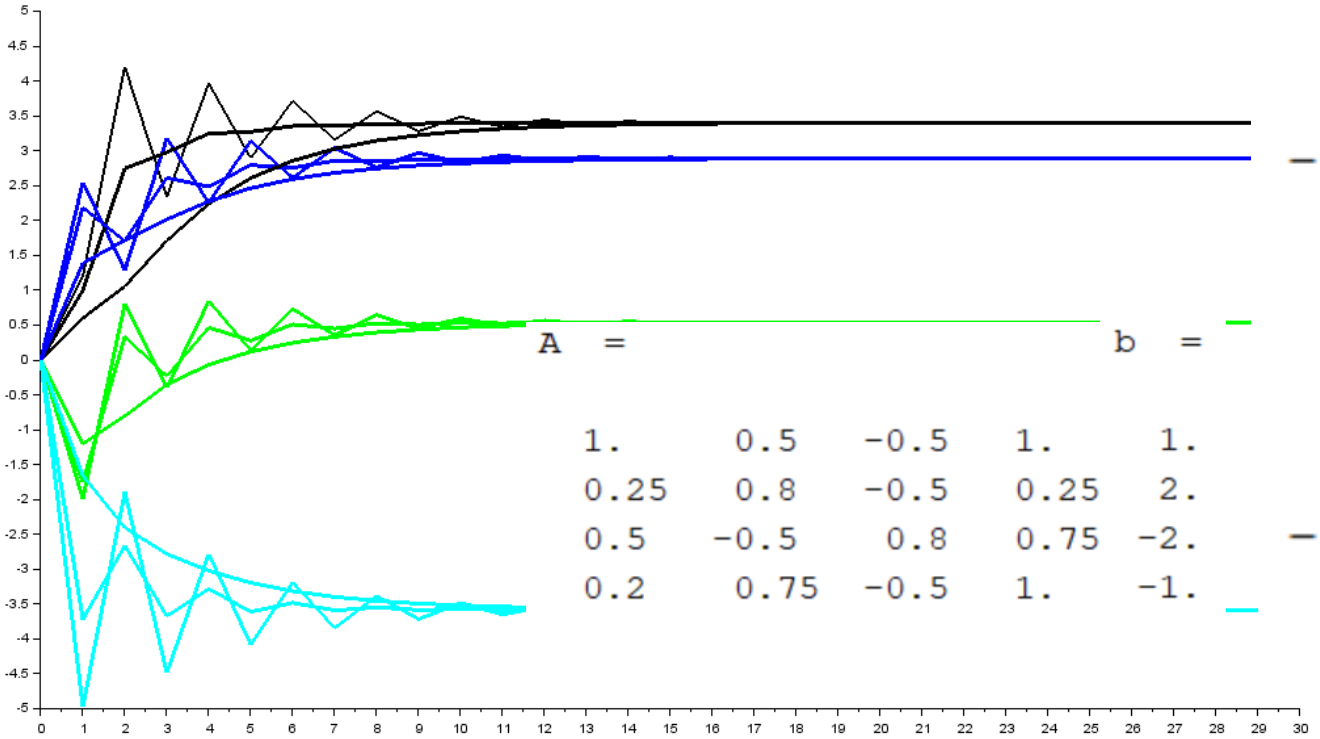
1.
-3.
-2.
1.

Przykład 4



11:25

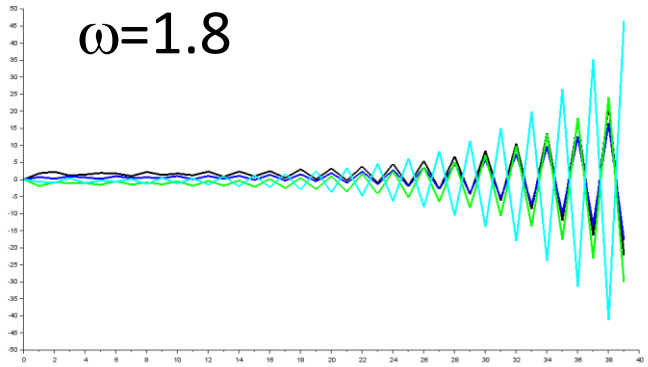
2. ω=0.6;1;1.2



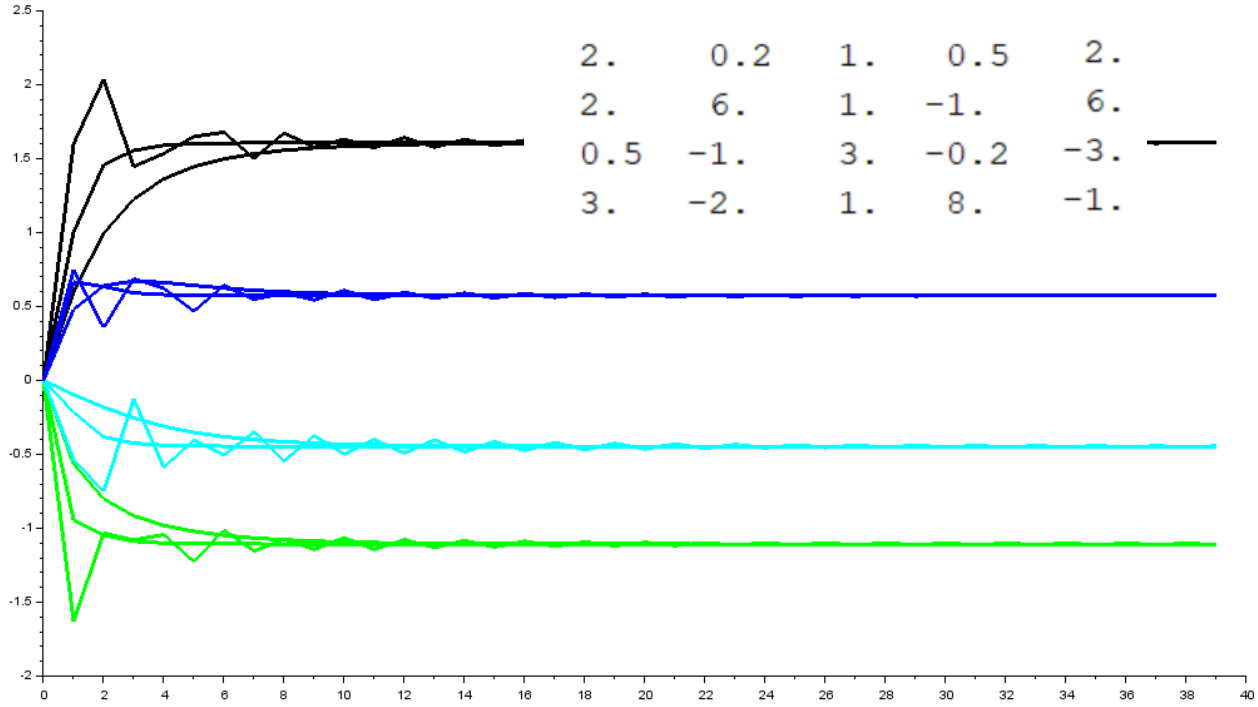
```
A    =                                   b    =

 1.     0.5    -0.5    1.      1.
 0.25   0.8    -0.5    0.25    2.
 0.5   -0.5     0.8    0.75   -2.
 0.2    0.75   -0.5    1.     -1.
```

```
 3.4031235
 2.8905418
 0.536034
-3.5805771

 3.4032634
 2.8904429
 0.5361305
-3.5804196
```

Przykład 4



3. ω=0.6;1;1.6



A =

2.	0.2	1.	0.5	2.
2.	6.	1.	-1.	6.
0.5	-1.	3.	-0.2	-3.
3.	-2.	1.	8.	-1.

b =

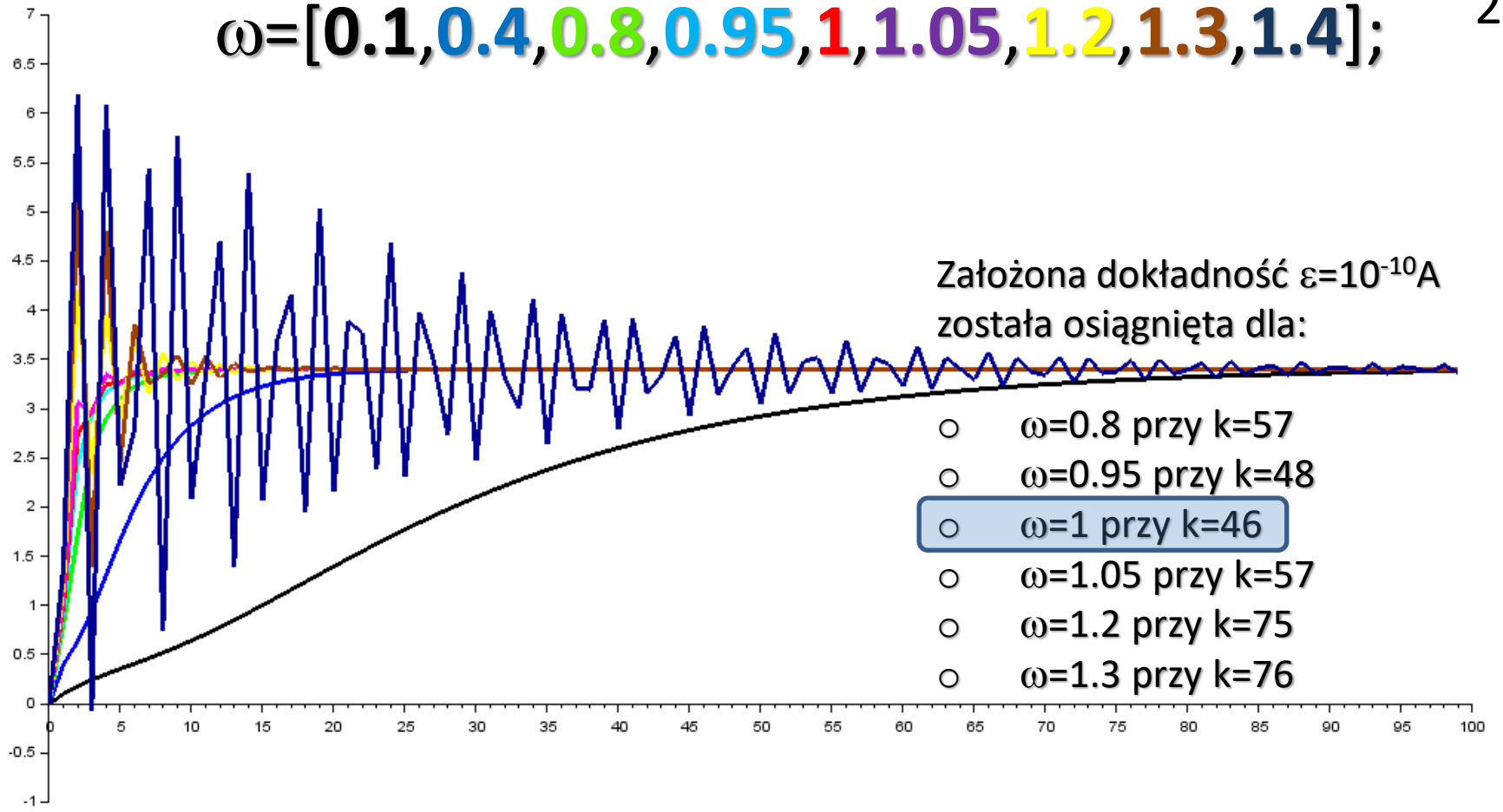
1.6060323
0.5735971
-1.107133
-0.4441763

1.6070676
0.5743631
-1.1061099
-0.4457958

Wpływ ω na zbieżność

2.

ω=[0.1,0.4,0.8,0.95,1,1.05,1.2,1.3,1.4];

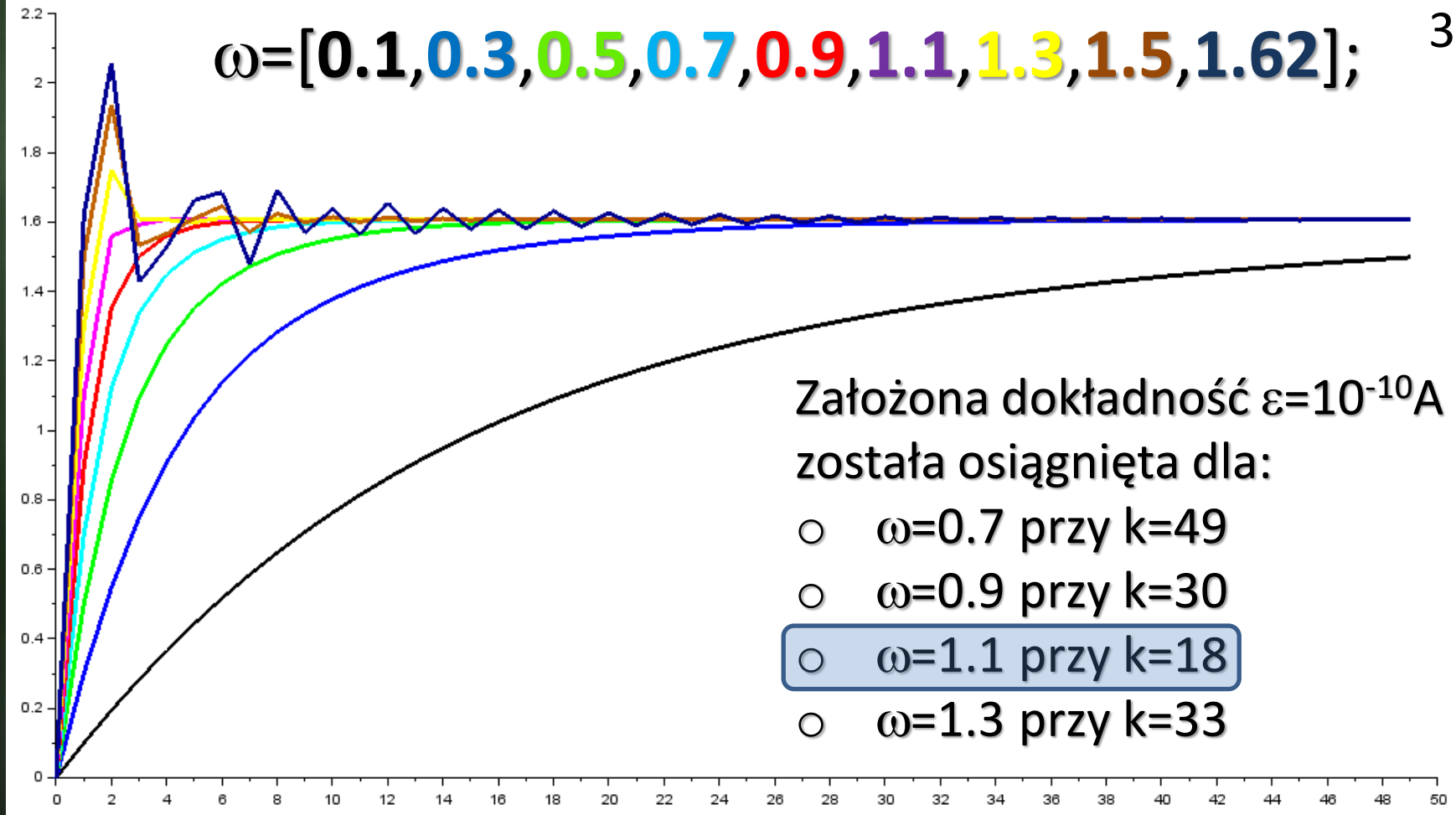


Założona dokładność $\varepsilon=10^{-10}A$ została osiągnięta dla:

- ω=0.8 przy k=57
- ω=0.95 przy k=48
- ω=1 przy k=46
- ω=1.05 przy k=57
- ω=1.2 przy k=75
- ω=1.3 przy k=76

Wpływ ω na zbieżność

3.

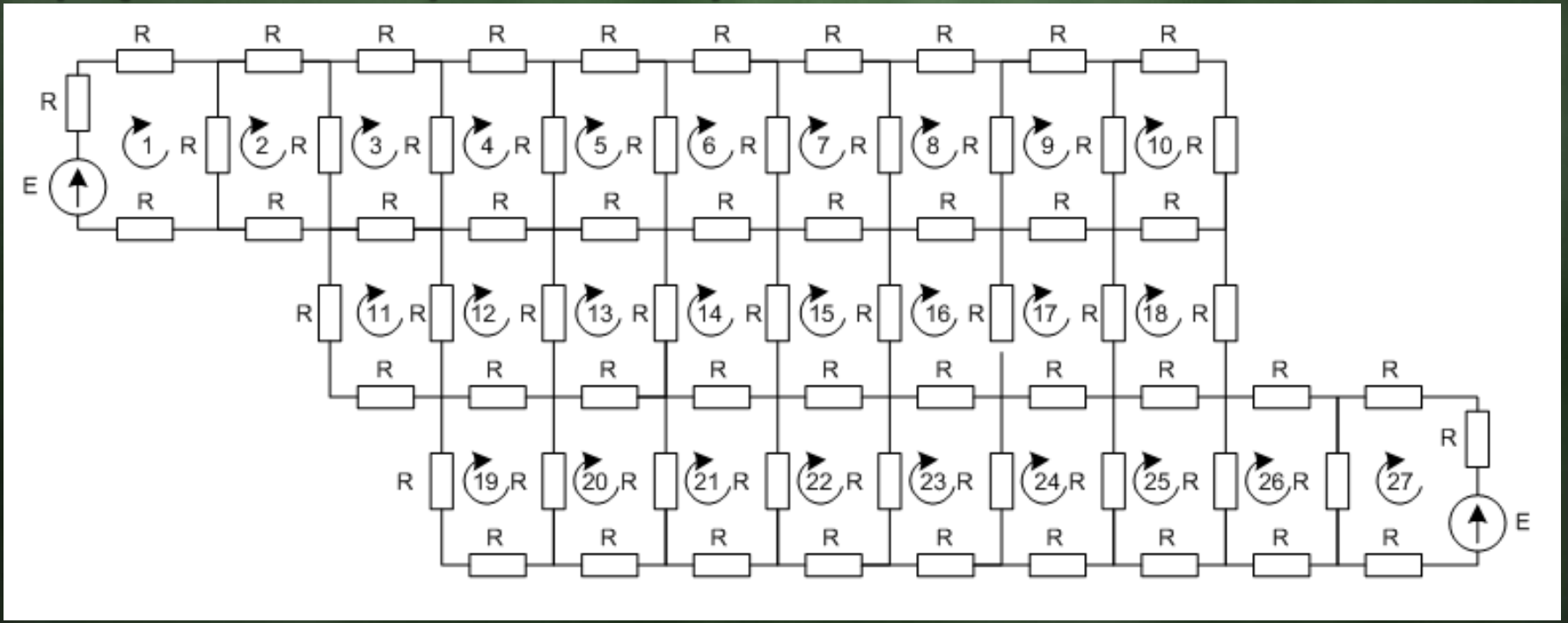


Założona dokładność $\varepsilon=10^{-10}$A została osiągnięta dla:

- $\omega=0.7$ przy k=49
- $\omega=0.9$ przy k=30
- $\omega=1.1$ przy k=18
- $\omega=1.3$ przy k=33

Przykład 5

Z zastosowaniem poznanych metod obliczyć wartości prądów oczkowych w zadanym obwodzie.



E=100V, R=10Ω

Przykład 5

W oparciu o zasady metody oczkowej można zapisać równania opisujące obwód.

1. $4RI_1-RI_2=E$
2. $4RI_2-RI_1-RI_3=0$
3. $4RI_3-RI_4-RI_{11}=0$
4. $4RI_4-RI_3-RI_5-RI_{12}=0$
5. $4RI_5-RI_4-RI_6-RI_{13}=0$
6. $4RI_6-RI_5-RI_7-RI_{14}=0$
7. $4RI_7-RI_6-RI_8-RI_{15}=0$
8. $4RI_8-RI_7-RI_9-RI_{16}=0$
9. $4RI_9-RI_8-RI_{10}-RI_{17}=0$
10. $4RI_{10}-RI_9-RI_{11}-RI_{18}=0$
11. $4RI_{11}-RI_3-RI_{12}=0$
12. $4RI_{12}-RI_{11}-RI_{13}-RI_4-RI_{19}=0$
13. $4RI_{13}-RI_{12}-RI_{14}-RI_5-RI_{20}=0$
14. $4RI_{14}-RI_{13}-RI_{15}-RI_6-RI_{21}=0$
15. $4RI_{15}-RI_{14}-RI_{16}-RI_7-RI_{22}=0$
16. $4RI_{16}-RI_{15}-RI_{17}-RI_8-RI_{23}=0$
17. $4RI_{17}-RI_{16}-RI_{18}-RI_9-RI_{24}=0$
18. $4RI_{18}-RI_{17}-RI_{10}-RI_{25}=0$
19. $4RI_{19}-RI_{20}-RI_{12}=0$
20. $4RI_{20}-RI_{19}-RI_{21}-RI_{13}=0$
21. $4RI_{21}-RI_{20}-RI_{22}-RI_{14}=0$
22. $4RI_{22}-RI_{21}-RI_{23}-RI_{15}=0$
23. $4RI_{23}-RI_{22}-RI_{24}-RI_{16}=0$
24. $4RI_{24}-RI_{23}-RI_{25}-RI_{17}=0$
25. $4RI_{25}-RI_{24}-RI_{26}-RI_{18}=0$
26. $4RI_{26}-RI_{25}-RI_{27}=0$
27. $4RI_{27}-RI_{26}=-E$

Na podstawie pokazanych równań można utworzyć układ równań macierzowych **Ax=b**.

Przykład 5

Na podstawie pokazanych równań można utworzyć układ równań macierzowych **Ax=b**.

$$\begin{bmatrix}
4R & -R & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-R & 4R & -R & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -R & 4R & -R & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -R & 4R & -R & 0 & 0 & 0 & 0 & 0 & 0 & -R & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -R & 4R & -R & 0 & 0 & 0 & 0 & 0 & 0 & -R & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -R & 4R & -R & 0 & 0 & 0 & 0 & 0 & 0 & -R & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -R & 4R & -R & 0 & 0 & 0 & 0 & 0 & 0 & -R & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -R & 4R & -R & 0 & 0 & 0 & 0 & 0 & 0 & -R & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -R & 4R & -R & 0 & 0 & 0 & 0 & 0 & 0 & -R & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -R & 4R & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -R & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -R & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4R & -R & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -R & 0 & 0 & 0 & 0 & 0 & 0 & -R & 4R & -R & 0 & 0 & 0 & 0 & 0 & -R & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -R & 0 & 0 & 0 & 0 & 0 & 0 & -R & 4R & -R & 0 & 0 & 0 & 0 & 0 & -R & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -R & 0 & 0 & 0 & 0 & 0 & 0 & -R & 4R & -R & 0 & 0 & 0 & 0 & 0 & -R & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -R & 0 & 0 & 0 & 0 & 0 & 0 & -R & 4R & -R & 0 & 0 & 0 & 0 & 0 & -R & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -R & 0 & 0 & 0 & 0 & 0 & 0 & -R & 4R & -R & 0 & 0 & 0 & 0 & 0 & -R & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -R & 0 & 0 & 0 & 0 & 0 & 0 & -R & 4R & -R & 0 & 0 & 0 & 0 & 0 & -R & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -R & 0 & 0 & 0 & 0 & 0 & 0 & -R & 4R & 0 & 0 & 0 & 0 & 0 & 0 & -R & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -R & 0 & 0 & 0 & 0 & 0 & 0 & 4R & -R & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -R & 0 & 0 & 0 & 0 & 0 & -R & 4R & -R & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -R & 0 & 0 & 0 & 0 & 0 & -R & 4R & -R & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -R & 0 & 0 & 0 & 0 & 0 & -R & 4R & -R & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -R & 0 & 0 & 0 & 0 & 0 & -R & 4R & -R & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -R & 0 & 0 & 0 & 0 & 0 & -R & 4R & -R & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -R & 0 & 0 & 0 & 0 & 0 & -R & 4R & -R & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -R & 0 & 0 & 0 & 0 & 0 & -R & 4R & -R \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -R & 0 & 0 & 0 & 0 & 0 & -R & 4R
\end{bmatrix}
\begin{bmatrix} I1 \\ I2 \\ I3 \\ I4 \\ I5 \\ I6 \\ I7 \\ I8 \\ I9 \\ I10 \\ I11 \\ I12 \\ I13 \\ I14 \\ I15 \\ I16 \\ I17 \\ I18 \\ I19 \\ I20 \\ I21 \\ I22 \\ I23 \\ I24 \\ I25 \\ I26 \\ I27 \end{bmatrix}
=
\begin{bmatrix} E \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -E \end{bmatrix}$$

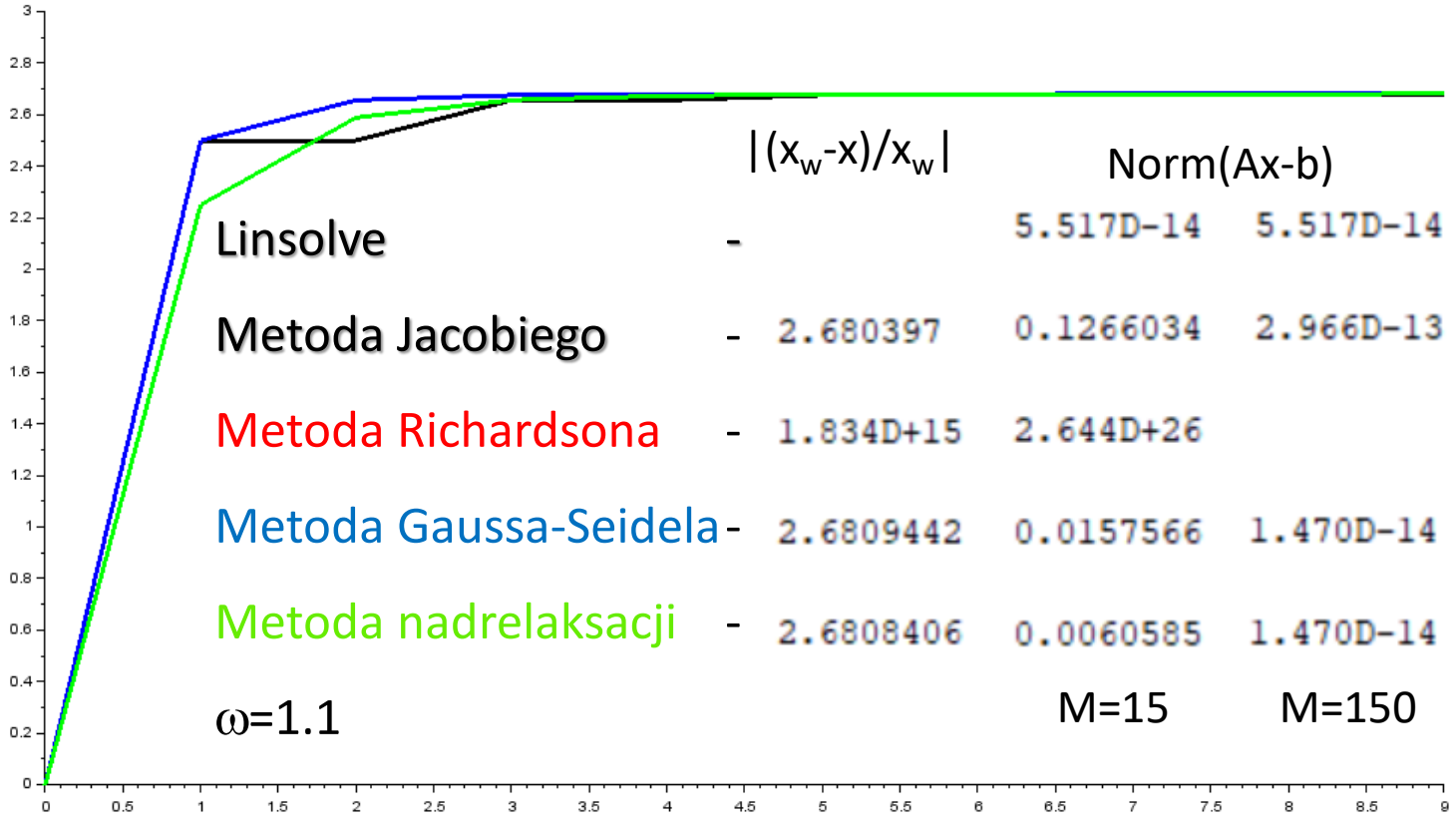
Przykład 5

```
4  R=10; E=100; n=27; M=15;
5  w=1.1;
6  x=zeros(4,n,M);
```

```
7  b=zeros(n,1);
8  b(1)=E; b(27)=-E;
```

```
9  A=zeros(n,n);
10 C=4*R*eye(n,n);
11 A=A+C;
12 for i=1:(n-1)
13     for j=2:n
14         if j==(i+1) then A(i,j)=-R;  A(j,i)=-R; end
15     end
16 end
17 for i=3:10
18     for j=11:18
19         if j==(i+8) then A(i,j)=-R; A(j,i)=-R; end
20     end
21 end
22 for i=12:18
23     for j=19:25
24         if j==(i+7) then A(i,j)=-R; A(j,i)=-R; end
25     end
26 end
```

Przykład 5



		$\lvert(x_w-x)/x_w\rvert$	Norm(Ax-b)	
Linsolve	-		5.517D-14	5.517D-14
Metoda Jacobiego	-	2.680397	0.1266034	2.966D-13
Metoda Richardsona	-	1.834D+15	2.644D+26	
Metoda Gaussa-Seidela	-	2.6809442	0.0157566	1.470D-14
Metoda nadrelaksacji	-	2.6808406	0.0060585	1.470D-14
ω=1.1			M=15	M=150

```
 2.6809656
 0.7238622
 0.2144834
 0.0700496
 0.0241109
 0.0065613
-0.0030362
-0.0115605
-0.0203644
-0.0230352
 0.0640219
 0.0416041
 0.0198326
 0.0051707
-0.0071459
-0.0228411
-0.0468621
-0.0717763
 0.0125122
 0.0084449
 0.0014347
-0.0078768
-0.0257961
-0.0724665
-0.2172079
-0.7245888
-2.6811472
```

Przykład 5

Linsolve(A,-b)	Metoda Jacobiego		Metoda Gaussa-Seidela		Metoda nadrelaksacji	
2.6809656	2.6808493	0.0001162	2.6809824	0.0000168	2.6809821	0.0000165
0.7238622	0.7236444	0.0002178	0.7239248	0.0000625	0.7239105	0.0000483
0.2144834	0.213686	0.0007974	0.2146797	0.0001962	0.2146063	0.0001228
0.0700496	0.0693702	0.0006794	0.0704241	0.0003745	0.0702428	0.0001932
0.0241109	0.0230912	0.0010196	0.0246382	0.0005273	0.0243374	0.0002265
0.0065613	0.0068224	0.0002611	0.0071639	0.0006026	0.0067805	0.0002192
-0.0030362	-0.0025815	0.0004548	-0.0024581	0.0005781	-0.0028542	0.000182
-0.0115605	-0.0097543	0.0018061	-0.0110937	0.0004667	-0.0114302	0.0001303
-0.0203644	-0.0190642	0.0013002	-0.0200585	0.000306	-0.0202869	0.0000775
-0.0230352	-0.0218253	0.0012099	-0.022896	0.0001392	-0.0230025	0.0000327
0.0640219	0.0635196	0.0005023	0.0642158	0.0001939	0.0641284	0.0001066
0.0416041	0.0401676	0.0014365	0.0420436	0.0004395	0.0418088	0.0002047
0.0198326	0.0192913	0.0005413	0.0204582	0.0006255	0.0200801	0.0002475
0.0051707	0.0046591	0.0005116	0.0058773	0.0007066	0.0054106	0.0002399
-0.0071459	-0.0055034	0.0016424	-0.0064758	0.0006701	-0.0069471	0.0001988
-0.0228411	-0.0212188	0.0016223	-0.022304	0.0005371	-0.0226991	0.0001421
-0.0468621	-0.0441621	0.0027	-0.04651	0.0003521	-0.0467773	0.0000848
-0.0717763	-0.0705069	0.0012694	-0.0716139	0.0001624	-0.07174	0.0000363
0.0125122	0.0120608	0.0004515	0.0127505	0.0002382	0.0126124	0.0001002
0.0084449	0.007574	0.0008709	0.0088107	0.0003658	0.0085784	0.0001335
0.0014347	0.0017511	0.0003164	0.0018475	0.0004128	0.0015657	0.000131
-0.0078768	-0.0073785	0.0004983	-0.0074885	0.0003883	-0.0077683	0.0001085
-0.0257961	-0.0239386	0.0018576	-0.0254861	0.00031	-0.0257185	0.0000777
-0.0724665	-0.0710953	0.0013713	-0.0722614	0.0002051	-0.0724196	0.0000469
-0.2172079	-0.21578	0.0014279	-0.2171051	0.0001028	-0.2171861	0.0000218
-0.7245888	-0.7241966	0.0003922	-0.7245603	0.0000285	-0.7245831	0.0000057
-2.6811472	-2.6809641	0.0001831	-2.6811401	0.0000071	-2.6811459	0.0000013