**Metody numeryczne**
Laboratorium 8
Info

## *Metody iteracyjne rozwiązywania układów równań liniowych*
## *Metoda Jacobiego*

Metoda Jacobiego jest metodą iteracyjną i pozwala nam obliczyć układ równań składający się z *n* równań zawierających *n* niewiadomych, który można zapisać następująco:

$$A \cdot x = b$$

Zakłada się, że wektor $x^{(0)}$ będący początkowym przybliżeniem jest znany (w praktyce zazwyczaj przyjmuje się, że jest to wektor wypełniony zerami; (0) – oznacza 0-wą iterację czyli etap początkowy). Kolejne *t* przybliżenia oblicza się na podstawie następującego wzoru:

$$x^{(t+1)} = M \cdot x^{(t)} + N \cdot b$$

gdzie:

- *t+1* – oznacza numer iteracji (dla zerowej iteracji *t*=0 ),
- $M = I - N \cdot A$,
    - $N$ – jest pewną macierzą kwadratową,
    - $I$ – macierz jednostkowa (na głównej przekątnej znajdują się jedynki a na pozostałych pozycjach zera),
    - $A$ – macierz współczynników, na podstawie której definiowane są macierze $L$, $D$ i $U$ gdzie $A = L + D + U$
        - $L$ – zawiera elementy dolnej macierzy trójkątnej $A$,
        - $D$ – to macierz diagonalna (zawiera elementy niezerowe na głównej przekątnej),
        - $U$ – zawiera elementy górnej macierzy trójkątnej $A$.

Po każdej iteracji należy sprawdzać warunek stopu (o ile został zdefiniowany), na przykład poprzez wyznaczenie sumarycznego błędu *r*:

$$r^{(t+1)} = \sum_{i=1}^{n} (r_i)^{(t+1)}$$

gdzie $(r_i)^{(t+1)} = \left|\sum_{j=1}^{n}\left(a_{ij} \cdot x_j\right) - b_i\right|$.

Oznacza to, że iteracje należy przerwać po z góry założonej liczby iteracji lub po osiągnięciu błędu rozwiązania *r* < *d*, gdzie *d* to z góry założona dokładność.

W praktyce $A = L + D + U$ przedstawia się następująco:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ a_{21} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & 0 \end{bmatrix} + \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix} + \begin{bmatrix} 0 & a_{12} & \cdots & a_{1n} \\ 0 & 0 & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

Przy założeniu, że $N = D^{-1}$ (warunek konieczny $d_{ij} \neq 0$ dla *i*=*j* gdzie *i*=1, 2, …, *n*; *j*=1, 2, …, *n*) mamy:

$$M = I - N \cdot A = I - D^{-1} \cdot A = I - D^{-1}(L + D + U) = -D^{-1}(L + U).$$

Na tym etapie należy sprawdzić zbieżność metody iteracji prostej dla danego zestawu parametrów. Warunkiem wystarczającym uzyskania zbieżności jest aby dowolna z norm macierzy *H* była mniejsza od jedności, gdzie:

$$h_{ij} = \begin{cases} -\dfrac{a_{ij}}{a_{ii}} & dla \quad i \neq j \\ 0 & dla \quad i = j \end{cases}$$

Sprawdzamy czy największa z sum wierszy macierzy *H* jest mniejsza od jedności:

$$\|H\|_w = \max_i \sum_{j=1}^{n} |h_{ij}| < 1$$

lub czy największa z sum kolumn macierzy *H* jest mniejsza od jedności.

$$\|H\|_k = \max_j \sum_{j=1}^{n} |h_{ij}| < 1$$

## Przykład

$$\begin{cases} 10x_1 + 2x_2 + 3x_3 + x_4 = 19 \\ x_1 + 5x_2 + x_3 + 2x_4 = 3 \\ 2x_1 + x_2 + 4x_3 = 13 \\ x_2 + 2x_3 + 8x_4 = 21 \end{cases}$$

$$A = L + D + U = \begin{bmatrix} 10 & 2 & 3 & 1 \\ 1 & 5 & 1 & 2 \\ 2 & 1 & 4 & 0 \\ 0 & 1 & 2 & 8 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 0 & 1 & 2 & 0 \end{bmatrix} + \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 8 \end{bmatrix} + \begin{bmatrix} 0 & 2 & 3 & 1 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$N = D^{-1} = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0.25 & 0 \\ 0 & 0 & 0 & 0.125 \end{bmatrix}$$

$$H = \begin{bmatrix} 0 & -0.2 & -0.3 & -0.1 \\ -0.2 & 0 & -0.2 & -0.4 \\ -0.5 & -0.25 & 0 & 0 \\ 0 & -0.125 & -0.25 & 0 \end{bmatrix}$$

Sprawdzamy warunek wystarczający uzyskania zbieżności:

$$\|H\|_k = \max_j \sum_{j=1}^{n} |h_{ij}| = 0.75 < 1$$

Przygotowania do właściwych obliczeń.

$$x^{(t+1)} = M \cdot x^{(t)} + N \cdot b$$
$$x^{(1)} = -N \cdot (L + U) \cdot x^{(0)} + N \cdot b$$

$$\begin{bmatrix} x_1^{(1)} \\ x_2^{(1)} \\ x_3^{(1)} \\ x_4^{(1)} \end{bmatrix} = -\begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0.25 & 0 \\ 0 & 0 & 0 & 0.125 \end{bmatrix} \cdot \left( \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 0 & 1 & 2 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 2 & 3 & 1 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \right) \cdot \begin{bmatrix} x_1^{(0)} \\ x_2^{(0)} \\ x_3^{(0)} \\ x_4^{(0)} \end{bmatrix} + \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0.25 & 0 \\ 0 & 0 & 0 & 0.125 \end{bmatrix} \cdot \begin{bmatrix} 19 \\ 3 \\ 13 \\ 21 \end{bmatrix}$$

Upraszczając:

$$\begin{bmatrix} x_1^{(1)} \\ x_2^{(1)} \\ x_3^{(1)} \\ x_4^{(1)} \end{bmatrix} = \begin{bmatrix} 0 & -0.2 & -0.3 & -0.1 \\ -0.2 & 0 & -0.2 & -0.4 \\ -0.5 & -0.25 & 0 & 0 \\ 0 & -0.125 & -0.25 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1^{(0)} \\ x_2^{(0)} \\ x_3^{(0)} \\ x_4^{(0)} \end{bmatrix} + \begin{bmatrix} 1.9 \\ 0.6 \\ 3.25 \\ 2.625 \end{bmatrix}$$

Przechodzimy do właściwych obliczeń.
Przyjmujemy, że wektor *x* będący początkowym przybliżeniem zawiera same 0.
Dla *t*=0 otrzymujemy:

$$\begin{bmatrix} x_1^{(1)} \\ x_2^{(1)} \\ x_3^{(1)} \\ x_4^{(1)} \end{bmatrix} = \begin{bmatrix} 0 & -0.2 & -0.3 & -0.1 \\ -0.2 & 0 & -0.2 & -0.4 \\ -0.5 & -0.25 & 0 & 0 \\ 0 & -0.125 & -0.25 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1.9 \\ 0.6 \\ 3.25 \\ 2.625 \end{bmatrix} = \begin{bmatrix} 1.9 \\ 0.6 \\ 3.25 \\ 2.625 \end{bmatrix}$$

Dla *t*=1

$$\begin{bmatrix} x_1^{(2)} \\ x_2^{(2)} \\ x_3^{(2)} \\ x_4^{(2)} \end{bmatrix} = \begin{bmatrix} 0 & -0.2 & -0.3 & -0.1 \\ -0.2 & 0 & -0.2 & -0.4 \\ -0.5 & -0.25 & 0 & 0 \\ 0 & -0.125 & -0.25 & 0 \end{bmatrix} \cdot \begin{bmatrix} 1.9 \\ 0.6 \\ 3.25 \\ 2.625 \end{bmatrix} + \begin{bmatrix} 1.9 \\ 0.6 \\ 3.25 \\ 2.625 \end{bmatrix} = \begin{bmatrix} 0.5425 \\ -1.48 \\ 2.15 \\ 1.7375 \end{bmatrix}$$

Dla *t*=2

$$\begin{bmatrix} x_1^{(3)} \\ x_2^{(3)} \\ x_3^{(3)} \\ x_4^{(3)} \end{bmatrix} = \begin{bmatrix} 0 & -0.2 & -0.3 & -0.1 \\ -0.2 & 0 & -0.2 & -0.4 \\ -0.5 & -0.25 & 0 & 0 \\ 0 & -0.125 & -0.25 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0.5425 \\ -1.48 \\ 2.15 \\ 1.7375 \end{bmatrix} + \begin{bmatrix} 1.9 \\ 0.6 \\ 3.25 \\ 2.625 \end{bmatrix} = \begin{bmatrix} 1.37725 \\ -0.6335 \\ 3.34875 \\ 2.2725 \end{bmatrix}$$

…

Dla *t*=9

$$\begin{bmatrix} x_1^{(10)} \\ x_2^{(10)} \\ x_3^{(10)} \\ x_4^{(10)} \end{bmatrix} = \begin{bmatrix} 0 & -0.2 & -0.3 & -0.1 \\ -0.2 & 0 & -0.2 & -0.4 \\ -0.5 & -0.25 & 0 & 0 \\ 0 & -0.125 & -0.25 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0.5425 \\ -1.48 \\ 2.15 \\ 1.7375 \end{bmatrix} + \begin{bmatrix} 1.02497 \\ -0.97278 \\ 3.02978 \\ 2.01689 \end{bmatrix} = \begin{bmatrix} 0.98393 \\ -1.0177 \\ 2.98071 \\ 1.98915 \end{bmatrix}$$

Przy obliczeniach sumarycznego błędu *r* wykorzystamy następujący zapis macierzowy:

$$A \cdot (x)^{(10)} - b$$

$$\begin{bmatrix} 10 & 2 & 3 & 1 \\ 1 & 5 & 1 & 2 \\ 2 & 1 & 4 & 0 \\ 0 & 1 & 2 & 8 \end{bmatrix} \cdot \begin{bmatrix} 0.98393 \\ -1.0177 \\ 2.98071 \\ 1.98915 \end{bmatrix} - \begin{bmatrix} 19 \\ 3 \\ 13 \\ 21 \end{bmatrix} = \begin{bmatrix} 0.2648 \\ 0.14558 \\ 0.127 \\ 0.14307 \end{bmatrix}$$

$$r^{(10)} = \sum_{i=1}^{4} (r_i)^{(10)} = 0.2648 + 0.14558 + 0.127 + 0.14307 = 0.68045$$

Sprawdzenie.

Porównanie rozwiązania dokładnego *xx* z rozwiązaniem przybliżonym *x*.

$$xx = A^{-1} \cdot b = \begin{bmatrix} 1 \\ -1 \\ 3 \\ 2 \end{bmatrix} \qquad x^{(10)} = \begin{bmatrix} 0.98393 \\ -1.0177 \\ 2.98071 \\ 1.98915 \end{bmatrix}$$

**Plik Lab8_1.m**

```
% założenie: elementy głównej przekątnej macierzy D są niezerowe
% rozmiar wektora b jest adekwatny do rozmiaru macierzy A
% obliczenia realizowane są dla 10-ciu iteracji

A=[10,2,3,1; 1,5,1,2; 2,1,4,0; 0,1,2,8]
b=[19; 3; 13; 21]

n=length(b);
L=zeros(n);
D=zeros(n);
U=zeros(n);

for i=1:n
    for j=1:n
        if i>j
            L(i,j)=A(i,j);
        elseif  i<j
              U(i,j)=A(i,j);
          else
              D(i,j)=A(i,j);
          end;
    end
end

N=inv(D);
H=zeros(n);
for i=1:n
    for j=1:n
        if i!=j
           H(i,j)=-A(i,j)/A(i,i);
        end
    end
end;

if (max(sum(abs(H)))<1) || (max(sum(abs(H')))<1)
   x=zeros(4,1);
   for t=0:9
        disp(['Wynik dla iteracji nr ',num2str(t)]);
        x=-N*(L+U)*x+N*b;
       disp(x);
   end
   disp("Sumaryczny blad rozwiazania");
   disp(sum(abs((A*x)-b)));
   disp("Sprawdzenie - wynik dokladny");
   disp(inv(A)*b);
else
   disp("Warunek wystarczający uzyskania zbieżności nie został spełniony");
end;
```