

Metoda Jacobiego i Gaussa-Seidela

Dariusz Borkowski

P

oznana w poprzednim rozdziale metoda jest określana jako bezpośrednia metoda rozwiązania układu $Ax = b$. Po skończonej liczbie kroków otrzymujemy dokładne rozwiązanie, gdyby nie błędy zaokrągleń.

Metoda iteracyjna działa inaczej: tworzy ciąg wektorów zbieżny do rozwiązania. Obliczenia przerywamy, gdy rozwiązanie przybliżone osiągnęło wymaganą dokładność lub po ustalonej liczbie iteracji. Dla wielkich układów metody iteracyjne są szybsze i mniej zużywają pamięci.

Metoda Jacobiego

Niech będzie dany układ równań postaci:

$$a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,n}x_n = b_1$$

$$a_{2,1}x_1 + a_{2,2}x_2 + \cdots + a_{2,n}x_n = b_2$$

$$\vdots$$

$$a_{n,1}x_1 + a_{n,2}x_2 + \cdots + a_{n,n}x_n = b_n$$

Z każdego równania możemy wyznaczyć jedną zmienną.

$$x_1 = \frac{b_1 - (a_{1,2}x_2 + \cdots + a_{1,n}x_n)}{a_{1,1}}$$

$$x_2 = \frac{b_2 - (a_{2,1}x_1 + a_{2,3}x_3 + \cdots + a_{2,n}x_n)}{a_{2,2}}$$

$$\vdots$$

$$x_n = \frac{b_n - (a_{n,1}x_1 + a_{n,2}x_2 + \cdots + a_{n,n-1}x_{n-1})}{a_{n,n}}$$

Jeżeli przyjmiemy, że wstawiamy dane po prawej stronie układu, a wyznaczamy wartości po lewej to otrzymujemy wzór iteracyjny.

$$\begin{aligned}
 x_1^{(k+1)} &= \frac{b_1 - (a_{1,2}x_2^{(k)} + \cdots + a_{1,n}x_n^{(k)})}{a_{1,1}} \\
 x_2^{(k+1)} &= \frac{b_2 - (a_{2,1}x_1^{(k)} + a_{2,3}x_3^{(k)} + \cdots + a_{2,n}x_n^{(k)})}{a_{2,2}} \\
 &\vdots \\
 x_n^{(k+1)} &= \frac{b_n - (a_{n,1}x_1^{(k)} + \cdots + a_{n,n-1}x_{n-1}^{(k)})}{a_{n,n}}
 \end{aligned}$$

Twierdzenie 1. *Jeśli macierz A jest dominująca przekątniowo $|a_{i,i}| > \sum_{j \neq i} |a_{i,j}|$, to metoda Jacobiego jest zbieżna dla dowolnego wektora startowego $x^{(0)}$.*

Zadanie 1. *Napisz w MATLAB-ie program rozwiązujący układ równań $AX = B$ z wykorzystaniem iteracyjnej metody Jacobiego. Macierz A jest macierzą dominującą przekątniowo ($|a_{i,i}| > \sum_{j \neq i} |a_{i,j}|$), a wór iteracyjny jest zadany poniższym równaniem.*

$$x_i^{(k)} = \frac{1}{a_{i,i}} \left(b_i - \sum_{j \neq i} a_{i,j} x_j^{(k-1)} \right).$$

Do wyznaczenia dokładności rozwiązania wykorzystaj normę $\|\cdot\|_2$. Deklaracja funkcji powinna być następująca:

$$[xn, err, k] = jacob_i(A, B, x0, delta, max),$$

gdzie

A – macierz dominująca przekątniowo,

B – wektor,

x_0 – wektor początkowy,
 δ – dokładność rozwiązania; Żądamy aby był
spełniony warunek ($\|x_n - x^{(n-1)}\|_2 < \delta$),
gdzie x_n jest przybliżonym rozwiązaniem, a $x^{(n-1)}$
przedostatnim elementem ciągu,
 max – ograniczenie na liczbę kroków; program nie
może wykonać więcej niż max iteracji,
 x_n – przybliżone rozwiązanie,
 err – błąd rozwiązania; $err = \|x^{(n)} - x^{(n-1)}\|_2$,
gdzie $x^{(n)}$, $x^{(n-1)}$ są dwoma ostatnimi elementa-
mi ciągu,
 k – liczba kroków po której wyznaczono rozwiąza-
nie.

Rozwiązanie: `jacobi.m`



Metoda Gaussa-Seidela

Analizując metodę Jacobiego można zauważyć, że niektóre wartości można wyznaczyć korzystając z danych nie z kroku poprzedniego ale bieżącego. Wprowadzając takie ulepszenie otrzymujemy wzór rekurencyjny zwany metodą Gaussa-Seidela.

$$\begin{aligned}x_1^{(k+1)} &= \frac{b_1 - (a_{1,2}x_2^{(k)} + \cdots + a_{1,n}x_n^{(k)})}{a_{1,1}} \\x_2^{(k+1)} &= \frac{b_2 - (a_{2,1}x_1^{(k+1)} + a_{2,3}x_3^{(k)} + \cdots + a_{2,n}x_n^{(k)})}{a_{2,2}} \\&\vdots \\x_n^{(k+1)} &= \frac{b_n - (a_{n,1}x_1^{(k+1)} + \cdots + a_{n,n-1}x_{n-1}^{(k+1)})}{a_{n,n}}\end{aligned}$$

Twierdzenie 2. *Jeśli macierz A jest dominująca przekątniowo $|a_{i,i}| > \sum_{j \neq i} |a_{i,j}|$, to metoda*

Gaussa-Seidela jest zbieżna dla dowolnego wektora startowego $x^{(0)}$.

Zadanie 2. Napisz w MATLAB-ie program rozwiązujący układ równań $AX = B$ z wykorzystaniem iteracyjnej metody Gaussa-Seidela. Macierz A jest macierzą dominującą przekątniowo ($|a_{i,i}| > \sum_{j \neq i} |a_{i,j}|$) a wór iteracyjny jest zadany poniższym równaniem.

$$x_i^{(k)} = \frac{1}{a_{i,i}} \left(b_i - \sum_{j < i} a_{i,j} x_j^{(k)} - \sum_{j > i} a_{i,j} x_j^{(k-1)} \right).$$

Do wyznaczenia dokładności rozwiązania wykorzystaj normę $\|\cdot\|_2$. Deklaracja funkcji powinna być następująca:

$[xn, err, k] = gseidel(A, B, x0, delta, max),$

gdzie

A – macierz dominująca przekątniowo,

B – wektor,
 x_0 – wektor początkowy,
 δ – dokładność rozwiązania; Żądamy aby był spełniony warunek ($\|x_n - x^{(n-1)}\|_2 < \delta$),
gdzie x_n jest przybliżonym rozwiązaniem, a $x^{(n-1)}$ przedostatnim elementem ciągu,
 $\max$ – ograniczenie na liczbę kroków; program nie może wykonać więcej niż $\max$ iteracji,
 x_n – przybliżone rozwiązanie,
 err – błąd rozwiązania; $\text{err} = \|x^{(n)} - x^{(n-1)}\|_2$,
gdzie $x^{(n)}$, $x^{(n-1)}$ są dwoma ostatnimi elementami ciągu,
 k – liczba kroków po której wyznaczono rozwiązanie.

Rozwiązanie: `gseidel.m`



Zadanie 3. Sprawdź programy *jacobi* oraz *gseidel* rozwiązując poniższy układ równań.

$$3x_1 + x_2 + x_3 = 5$$

$$x_1 + 3x_2 - x_3 = 3$$

$$3x_1 + x_2 - 5x_3 = -1$$

Uwaga 1. Powyższy przykład pokazuje, że metoda Gaussa-Seidela daje rozwiązanie z taką samą dokładnością jak metoda Jacobiego, ale po mniejszej liczbie kroków. Z kolei, zaletą metody Jacobiego jest to, że jest łatwa do zrównoleglenia i w takim przypadku, będzie szybsza od metody Gaussa-Seidela.