Metody Numeryczne - Projekt 2

Mikołaj Mróz, Paweł Swiderski

17 stycznia 2023

(2 osoby) Rozwiązywanie układu równań liniowych $Ax = b$ z wykorzystaniem blokowej metody BSOR (Backward SOR), gdzie $A(n \times n)$ jest macierzą postaci

$$A = \begin{pmatrix} A_{11} & A_{12} & 0 \\ A_{12}^T & A_{11} & A_{23} \\ 0 & A_{23}^T & A_{11} \end{pmatrix},$$

gdzie $A_{ij}(p \times p)$ i $n = 3p$. Zakładamy, że A_{11} jest macierzą ortogonalną.

Wyznacz promień spektralny macierzy iteracji dla tej blokowej metody BSOR.

Spis treści

1 Opis metod **3**
1.1 Metoda Backward SOR . . . 3
1.2 Co to znaczy, że macierz jest ortogonalna? . . . 3
1.3 Promień spektralny . . . 3

2 Opis funkcjonalności metody w Matlabie **4**
2.1 Cel zadania i sposób jego rozwiązania . . . 4
2.2 Funkcje zawarte w programie . . . 5
2.2.1 Funkcja BSOR . . . 5

3 Ciekawe przykłady, wizualizacja i analiza wyników **6**
3.1 1. przykład . . . 6
3.2 2. przykład . . . 6
3.3 3. przykład . . . 7
3.4 4. przykład . . . 7
3.5 5. przykład . . . 8
3.6 6. przykład . . . 8

4 Wnioski i analiza **9**

5 Literatura **9**

1 Opis metod

1.1 Metoda Backward SOR

Backward SOR (Successive Over-Relaxation) to metoda iteracyjna służąca do rozwiązywania układów równań liniowych. Polega ona na iteracyjnym znajdowaniu rozwiązania poprzez zastosowanie wzoru:

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

gdzie:
$x_i^{(k)}$ to i-ty element wektora x w k-tej iteracji

a_{ij} to element macierzy A w wierszu i i kolumnie j

b_i to i-ty element wektora b

ω to parametr relaksacji, który może być dostosowywany w celu zwiększenia szybkości zbieżności metody

Metoda backward SOR różni się od innych metod SOR tym, że sumy w wzorze są obliczane od końca do początku, zamiast od początku do końca. Dzięki temu metoda ta jest szczególnie skuteczna w przypadku układów równań z macierzami trójdiagonalnymi, ponieważ zapewnia szybszą zbieżność w porównaniu z innymi metodami iteracyjnymi.

Metoda Backward Successive Over Relaxation (BSOR) to wariant metody Jacobiego, która jest numeryczną metodą rozwiązywania układów równań liniowych. BSOR różni się od metody Jacobiego tym, że używa dodatkowego parametru relaksacji, który pozwala na uzyskanie szybszych i dokładniejszych rozwiązań w niektórych przypadkach. W metodzie BSOR, nowe przybliżenie rozwiązania układu równań jest obliczane za pomocą poprzedniego przybliżenia i parametru relaksacji, który jest wykorzystywany do "rozluźnienia" obliczeń. Dzięki temu, metoda BSOR może działać szybciej niż metoda Jacobiego, ale może być mniej dokładna w niektórych przypadkach.

1.2 Co to znaczy, że macierz jest ortogonalna?

Macierz A jest ortogonalna, jeśli spełnia równanie:

$$A^T A = I$$

gdzie A^T oznacza transponowaną macierz A, a I to macierz jednostkowa o rozmiarze takim samym jak macierz A.

1.3 Promień spektralny

Promień spektralny macierzy to największy wielkościowo z wartości bezwzględnych jej wartości własnych. Jest również znany jako norma spektralna lub norma operatora macierzy. Promień spektralny macierzy odgrywa ważną rolę w badaniach teorii macierzy i ma zastosowania w różnych dziedzinach, takich jak inżynieria, fizyka i matematyka. W szczególności promień spektralny macierzy jest związany z tempem zbieżności iteracyjnych algorytmów oraz stabilnością systemów opisywanych równaniami różniczkowymi.

2 Opis funkcjonalności metody w Matlabie

2.1 Cel zadania i sposób jego rozwiązania

Celem zadania było zaimplementowanie metody backward SOR rozwiązywania układów liniowych macierzy dla macierzy blokowych, oraz obliczenie promienia spektralnego macierzy iteracyjnej.

1. Obliczenie rozwiązania układu równań oraz promienia spektralnego macierzy iteracyjnej

Zrobiliśmy to za pomocą metody BSOR w Matlabie, która oblicza przybliżone rozwiązanie układu równań, oraz promień spektralny macierzy iteracyjnej. Funkcja ta zwraa także wskaźnik uwarunkowania macierzy A, oraz liczbę iteracji k, dla której otrzymaliśmy wynik.

2. Ocena szybkości poprawności działania metody oraz szybkości zbieżności dla poszczególnych przypadków

Na koniec dla ciekawych przypadków sprawdziliśmy działanie naszej metody, oraz jak szybko zbiega ona dla różnych przykładów równań z macierzami blokowymi.

2.2 Funkcje zawarte w programie

2.2.1 Funkcja BSOR

```
function [xk, rho] = BSOR(A, b, w, tol, max_iter, xo)
% Ta funkcja jest numeryczna metoda rozwiazywania ukladow rownan liniowych.
% Jest to implementacja metody BSOR (Backward Successive Over Relaxation).
%
% Wejscia:
% A: macierz nxn, reprezentujaca wspolczynniki ukladow rownan liniowych
% b: wektor nx1, reprezentujacy prawa strone ukladow rownan liniowych
% w: parametr relaksacji, wartosc skalarna pomiedzy 0 a 2
% tol: tolerancja, wartosc skalarna wskazujaca pozadany poziom
% dokladnosci rozwiazania
% max_iter: maksymalna liczba iteracji do wykonania przed zatrzymaniem
% xo: poczatkowe oszacowanie rozwiazania, wektor nx1
%
% Wyjscia:
% xk: przyblizone rozwiazanie ukladow rownan liniowych, wektor nx1
% rho: promien spektralny macierzy iteracji, wartosc skalarna
n = size(A,2);

D = diag(diag(A));
rho = max(abs(eig(inv(D)*(D-A))))
assert(rho < 1, "Metoda nie jest zbiezna")

xk = xo;
for i = 1:n
    xk(i) = xo(i) + 1;
end

k = 0;
while norm(xk-xo) > tol && k <= max_iter
    x = xk;
    for i = n:-1:1
        sum1 = 0;
        sum2 = 0;
        for j = 1:n
            if j < i
                sum1 = sum1 + A(i,j) * xk(j);
            elseif j > i
                sum2 = sum2 + A(i,j) * xo(j);
            end
        end
        xk(i) = (1-w) * xo(i) + w * (b(i) - sum1 - sum2) / A(i,i);
    end
    xo = x;
    k = k+1;
end

D = diag(diag(A));
rho = max(abs(eig(inv(D)*(D-A))));
k
cond = norm(inv(A)) * norm(A)

end
```

3 Ciekawe przykłady, wizualizacja i analiza wyników

Oznaczenia:
a_k - współczynniki liniowej kombinacji wielomianów Czybyszewa II-ego rodzaju
$w_n(x)$ - wielomian którego całkę oznaczoną przybliżamy
a, b - granice całki

3.1 1. przykład

$$\text{A} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

$$\text{b} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

$\omega = 0.5$
$\text{tol} = 10^{-4}$
$\text{maxiter} = 10^3$
$x_0 = [0, 0, 0]$

Otrzymane wartości: [inf, -inf, inf]
rho = 1.4142
Metoda nie jest zbieżna

3.2 2. przykład

$$\text{A} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

$$\text{b} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

$\omega = 1$
$\text{tol} = 10^{-4}$
$\text{maxiter} = 10^3$
$x_0 = [0, 0, 0]$

Otrzymane wartości: [inf, -inf, inf]
rho = 1.4142
Metoda nie jest zbieżna

3.3 3. przykład

$$\mathrm{A} = \begin{bmatrix} 1 & 0.1 & 0 \\ 0.1 & 1 & 0.5 \\ 0 & 0.5 & 1 \end{bmatrix}$$

$$\mathrm{b} = \begin{bmatrix} 2 \\ 4 \\ 2 \end{bmatrix}$$

$\omega = 1$
tol $= 10^{-4}$
maxiter $= 10^3$
$x_0 = [0, 0, 0]$

Otrzymane wartości:
$x_k = [1.6217, 3.7834, 0.1085]$
rho $= 0.5099$
k $= 54$
cond $= 3.0808$
błąd względny $= 10^{-5}$

3.4 4. przykład

$$\mathrm{A} = \begin{bmatrix} 1 & 0.1 & 0 \\ 0.1 & 1 & 0.5 \\ 0 & 0.5 & 1 \end{bmatrix}$$

$$\mathrm{b} = \begin{bmatrix} 2 \\ 4 \\ 2 \end{bmatrix}$$

$\omega = 0.5$
tol $= 10^{-1}$
maxiter $= 10^4$
$x_0 = [0, 0, 0]$

Otrzymane wartości:
$x_k = [1.6792, 3.3086, 0.5050]$
rho $= 0.5099$
k $= 8$
cond $= 3.0808$
błąd względny $= \approx 0.5$

3.5 5. przykład

$$\mathrm{A} = \begin{bmatrix} 1 & 0 & 0.2 & 0.1 & 0 & 0 \\ 0 & 1 & 0.2 & 0.1 & 0 & 0 \\ 0.2 & 0.2 & 1 & 0 & 0.2 & 0.1 \\ 0.1 & 0.1 & 0 & 1 & 0.2 & 0.1 \\ 0 & 0 & 0.2 & 0.2 & 1 & 0 \\ 0 & 0 & 0.1 & 0.1 & 0 & 1 \end{bmatrix}$$

$$\mathrm{b} = \begin{bmatrix} 0.1 \\ 0.2 \\ 0.1 \\ 0.3 \\ 0.05 \\ 0.1 \end{bmatrix}$$

$\omega = 0.5$
tol $= 10^{-4}$
maxiter $= 10^3$
$x_0 = [0, 0, 0, 0, 0, 0]$

Otrzymane wartości:
$x_k = [0.7039, 2.7039, -0.1271, 3.2137, 5.3826, 3.6913]$
rho $= 0.4414$
k $= 50$
cond $= 2.5806$
błąd względny $= 5 * 10^{-5}$

3.6 6. przykład

$$\mathrm{A} = \begin{bmatrix} 1 & 0 & 0.2 & 0.1 & 0 & 0 \\ 0 & 1 & 0.2 & 0.1 & 0 & 0 \\ 0.2 & 0.2 & 1 & 0 & 0.2 & 0.1 \\ 0.1 & 0.1 & 0 & 1 & 0.2 & 0.1 \\ 0 & 0 & 0.2 & 0.2 & 1 & 0 \\ 0 & 0 & 0.1 & 0.1 & 0 & 1 \end{bmatrix}$$

$$\mathrm{b} = \begin{bmatrix} 0.1 \\ 0.2 \\ 0.1 \\ 0.3 \\ 0.05 \\ 0.1 \end{bmatrix}$$

$\omega = 1.5$
tol $= 10^{-4}$
maxiter $= 10^3$
$x_0 = [0, 0, 0, 0, 0, 0]$

Otrzymane wartości:
$x_k = [0.7039, 2.7039, -0.1271, 3.2137, 5.3826, 3.6913]$
rho $= 0.4414$
k $= 643$
cond $= 2.5806$
błąd względny $= 7 * 10^{-5}$

4 Wnioski i analiza

Metoda ta jest szczególnie skuteczna w przypadku macierzy blokowych o dużym rozmiarze, ponieważ pozwala na liczenie rozwiązań dla poszczególnych bloków niezależnie od siebie. To znacznie zmniejsza złożoność obliczeniową w porównaniu z metodami, które wymagają obliczania całej macierzy jako całości.

Metoda backward SOR to skuteczna i efektywna metoda liczenia rozwiązań układów równań liniowych dla macierzy blokowych, która charakteryzuje się dobrą zbieżnością i niską złożonością obliczeniową.

5 Literatura

-Wiedza własna

-Iterative Methods for Solving Linear Systems

-Block SOR methods for rank-deficient least-squares problems