

Ćwiczenie 8 – problem własny (metoda Jacobiego)

Treść zajęć

wartości własne macierzy, metoda Jacobiego

Cel zajęć

zapoznanie się z metoda poprzez jej zaprogramowanie w matlabie

Wzory, algorytm

Jak wiadomo, wartości na przekątnej głównej dla macierzy diagonalnej są wartościami własnymi tej macierzy. Idea metody Jacobiego polega na sprowadzeniu zadanej macierzy do postaci diagonalnej poprzez przekształcenia równoważne (transformacje przez obrót).

Macierz obrotu:

$$\mathbf{R} = \begin{bmatrix} 1 & & & & \\ & \dots & & & \\ & & c & s & \\ & & & 1 & \\ & & -s & c & \\ & & & & \dots \\ & & & & & 1 \end{bmatrix}$$

gdzie poza przekątną jedynie na pozycji (p, q) znajdują się wielkości niezerowe, realizuje obrót w płaszczyźnie p - q o kąt θ , taki że $c = \cos \theta$, $s = \sin \theta$. Można tak dobrać kąt obrotu, aby element macierzy a_{pq} wyzerował się.

algorytm dla elementów p, q

$$\theta = \frac{a_{qq} - a_{pp}}{2a_{pq}}, \quad t = \frac{\text{sgn}(\theta)}{|\theta| + \sqrt{\theta^2 + 1}}, \quad c = \frac{1}{\sqrt{t^2 + 1}}, \quad s = t \cdot c$$

$$a'_{ij} = a_{ij}$$

$$a'_{rp} = c \cdot a_{rp} - s \cdot a_{rq}, \quad a'_{pr} = a'_{rp}$$

$$a'_{rq} = c \cdot a_{rq} + s \cdot a_{rp}, \quad a'_{qr} = a'_{rq}$$

$$a'_{pp} = c^2 \cdot a_{pp} + s^2 \cdot a_{qq} - 2s \cdot c \cdot a_{pq}$$

$$a'_{qq} = s^2 \cdot a_{pp} + c^2 \cdot a_{qq} + 2s \cdot c \cdot a_{pq}$$

$$a'_{pq} = (c^2 - s^2) \cdot a_{pq} + s \cdot c \cdot (a_{pp} - a_{qq})$$

$$a'_{qp} = (c^2 - s^2) \cdot a_{qp} + s \cdot c \cdot (a_{pp} - a_{qq})$$

(albo w zapisie macierzowym):

$$\mathbf{A}' = \mathbf{R}^T \mathbf{A} \mathbf{R}$$

Poczynając od największego elementu (cały czas mamy na uwadze wartości bezwzględne) poza przekątniowego, kolejno sprowadzamy następne elementy poza przekątniowe do zera. Nazywa się to „wymiataniem”. Co prawda poprzednio wyzerowane elementy przestają być zerowe, ale – jak się okazuje – pozostają znacznie mniejsze niż były pierwotnie. Iteracyjnie więc zmierzamy do macierzy diagonalnej.

algorytm:

(w pętli iteracyjnej z warunkami stopu $\text{eps} < 1e-4$, liczba iteracji = 100):

1. znalezienie maksymalnego (co do modułu) elementu poza przekątniowego, zanotowanie jego położenia p, q
2. obliczenie kąta obrotu i elementów macierzy obrotu \mathbf{R}
3. wykonanie obrotu
4. aktualizacja macierzy przejścia (wyjściowo jest to macierz jednostkowa), $\mathbf{B}' = \mathbf{B} \cdot \mathbf{R}$ (poza pętlą):
5. weryfikacja rozwiązania ($\mathbf{A}' = \mathbf{B}'^T \mathbf{A} \mathbf{B}$)

Problem

Znaleźć wartości własne macierzy:

$$M = \begin{pmatrix} 2 & -3 & 6 \\ -3 & 4 & 1 \\ 6 & 1 & -3 \end{pmatrix}$$

Rozwiązanie

(kod w matlabie)

```
% Adam Zaborski, ćw. 8 - wartości własne metodą Jacobiego
clc
clear all
format compact
eps = 0.0001;
n = 3;
b = [2, -3, 6; -3, 4, 1; 6, 1, -3];
a = b;
%Znajdywanie maks - największego poza przekątniowego elementu macierzy a
% p, q wskaźniki największego elementu poza przekątniowego
% ponieważ podobne obliczenie będzie powtórzone jeszcze raz można byłoby
% zadeklarować jako funkcję (ale skoro jest taka niewielka i tylko raz
% powtórzona...)
% wypełnianie macierzy jednostkowej x(n x n)
x = eye(n);
k = 0;
maks = 2 * eps; % tylko po to aby choć jedna iteracja
while maks > eps && k < 100
    k = k + 1;
    maks = 0.0;
    for i = 1: n
        for j = 1: n
            if i ~= j && abs(a(i,j)) > maks
                maks = abs(a(i,j));
                p = i; q = j;
            end
        end
    end
    teta = ((a(q,q) - a(p,p)) / (2 * a(p,q)));
    t = sign(teta) / (abs(teta) + sqrt(teta*teta+1));
    c = 1 / sqrt(t * t + 1); s = t * c;
    % wypełnienie macierzy obrotu r
    % eye(n) - tworzenie macierzy jednostkowej (n x n)
    r = eye(n);
    r(p,p) = c;
    r(q,q) = c;
    r(p,q) = s;
    r(q,p) = -s;
    a = r' * a * r; % mnożenie przez macierz ortogonalną r
    x = x * r; % aktualizacja macierzy przejścia
end
a, x, k
% sprawdzenie rozwiązania procedurą MatLaba eig; ww-wartości wł.
```

```
ww = (eig(b))'
```

```
% kolejne sprawdzenie z zastosowaniem macierzy przejścia
```

```
x'*b*x
```

Wynik obliczeń: $(-7.5526, 7.3108, 3.2418)$