

Projekt nr 16: rozwiązywanie równania $AX = B$ metodą Cholesky'ego-Banachiewicza (rozkład $A = LDL^T$)

Michał Kukła
Grupa 4

5 kwietnia 2023

1 Opis matematyczny

1.1 Rozkład $A = LDL^T$

Zajmiemy się najpierw opisem rozkładu $A = LDL^T$. Taki rozkład możemy zastosować dla macierzy A , która jest kwadratowa, symetryczna i dodatnio określona. Macierz L to macierz dolnotrójkątna z jedynkami na przekątnej, L^T to jej transpozycja (zatem jest macierzą górnortrójkątną z jedynkami na przekątnej), natomiast macierz D to macierz diagonalna.

Cały proces rozkładu opisany jest w poniższym algorytmie. Oznaczone a_{ij} to element macierzy rozkładanej A na i -tym wierszu i j -tej kolumnie, d_{ij} to element macierzy D , c_{ij} to element macierzy pomocniczej C , a l_{ij} oznacza element macierzy L .

$$d_{11} = a_{11}$$

$$\text{Dla } i = 2, 3, \dots, n$$

$$\text{Dla } j = 1, 2, \dots, i-1$$

$$l_{ij} = (a_{ij} - \sum_{k=1}^{j-1} c_{ik} * l_{jk}) / d_{jj}$$

$$c_{ij} = d_{jj} * l_{ij}$$

$$d_{ii} = a_{ii} - \sum_{k=1}^{i-1} c_{ik} * l_{ik}$$

W celu wyznaczenia tego rozkładu dla macierzy A o wymiarach $n \times n$ trzeba wykonać $\frac{1}{6}n^3 + n^2 - \frac{7}{6}n$ działań, a dla D $\frac{1}{6}n^3 - \frac{1}{6}n$ działań.

1.2 Rozwiązywanie układu równań $AX = B$

Niech $A \in R^{n \times n}$ jest macierzą symetryczną dodatnio określoną oraz $B \in R^{n \times m}$, $m \geq 1$. Naszym celem jest znalezienie macierzy X takiej, że $AX = B$. Rozłóżmy A w ten sposób, że $A = LDL^T$. Wtedy otrzymujemy równanie $LDL^T X = B$. Zaczniemy od rozwiązania układu $LZ = B$ i wyznaczamy Z (czyli wyznaczamy, ile wynosi $DL^T X$), potem rozwiązujemy układ $DY = Z$ i wyznaczamy Y , a pod koniec rozwiązujemy układ $L^T X = Y$ i wyznaczamy X . Ogólny sposób znalezienia X to obliczenie $Z = L^{-1}B$, potem $Y = D^{-1}Z$ i na koniec $X = L^{T^{-1}}Y$. Inaczej można to zrobić przez eliminację Gaussa macierzy $[L \ B]$ i otrzymujemy macierz Z (jest to macierz szerokości macierzy B i odpowiada ostatnim kolumnom (jest ich tyle, ile wynosi liczba kolumn B) macierzy $[L \ B]$), $[D \ Z]$ i otrzymujemy macierz Y , a następnie Gausssem $[L^T \ Y]$ i otrzymamy macierz X . Metody te są podobne, ponieważ przy odwracaniu macierzy także można stosować metodę eliminacji Gaussa.

2 Opis implementacji

Zadanie rozpocząłem od napisania funkcji, rozkładającej macierz A na macierze $LDLT$. Kod rozkładu znajduje się poniżej:

```
D(1,1) = A(1,1);
for i = 2:n
    for j = 1:i-1
        suma1 = 0;
        for k = 1:j-1
            if j == 1
                continue
            else
                suma1 = suma1 + C(i,k)*L(j,k);
            end
        end
        L(i,j) = (A(i,j) - suma1)/D(j,j);
        C(i,j) = D(j,j)*L(i,j);
    end
    suma2=0;
    for k = 1:i-1
        suma2 = suma2 + C(i,k)*L(i,k);
    end
    D(i,i) = A(i,i) - suma2;
end
```

Szczegółowy kod znajduje się w pliku matlabowym. Na początku funkcji jeszcze sprawdzam, czy macierz A jest kwadratowa, symetryczna i dodatnio określona. Funkcja na wejściu otrzymuje macierz A , a na wyjściu macierze L , D i LT oraz wartość `info`. Jeżeli `info = 0`, to znaczy, że rozkład taki nie istnieje.

Zadanie wykonałem na 3 sposoby, jakimi są rozwiązanie układu równań korzystając z funkcji `\`, korzystając z funkcji `rref`, a trzecim sposobem rozwiązywanie układu równań tylko za pomocą pętli i podstawowych operacji arytmetycznych.

2.1 Korzystając z funkcji `\`

Pierwszy sposób jest krótki, kod wygląda tak:

```
Z = L\B;
Y = D\Z;
X = LT\Y;
```

Funkcja `\` to tak naprawdę wynik funkcji `inv` i funkcji mnożenia macierzy. Szczegóły zostały opisane w opisie matematycznym. Na wejściu funkcja przyjmuje macierze A i B . Następnie macierz A rozkłada na LDL^T (jeżeli to możliwe), na wyjściu dostajemy macierz X .

2.2 Korzystając z funkcji rref

Kod z rozwiązaniem znajduje się poniżej:

```
A1 = [L B];
Z = rref(A1);
Z = Z(:, size(A1,2) - size(B,2)+1:size(A1,2));

A2 = [D Z];
Y = rref(A2);
Y = Y(:, size(A2,2) - size(Z,2)+1: size(A2,2));

A3 = [LT Y];
X = rref(A3);
X = X(:, size(A3,2)- size(Y,2)+1: size(A3,2));
```

Początek funkcji to oczywiście zastosowanie funkcji, która rozkłada macierz A na LDL^T . Następnie łączymy poziomo macierze L i B , wykonujemy eliminację Gaussa i z otrzymanej macierzy bierzemy ostatnie kolumny (tyle samo kolumn, ile miała macierz B). Dostaliśmy macierz Z . To samo wykonujemy dla macierzy D i Z , otrzymując macierz Y i na koniec tak samo postępujemy dla macierzy L^T i Y , otrzymując ostatecznie szukaną macierz X . Na wejściu funkcja przyjmuje macierze A i B . Następnie macierz A rozkłada na LDL^T (jeżeli to możliwe), na wyjściu dostajemy macierz X .

2.3 Korzystając z pętli, wektorów i podstawowych operacji arytmetycznych

Kod do tego sposobu znajduje się poniżej:

```
n = size(L,1);
m = size(B,2);
x = zeros(n,m);

y = zeros(n,m);
d = zeros(n,1);
for i = 1:n
    d(i,1) = D(i,i);
end

for i = 1:n
    sum = L(i, 1:i-1)*y(1:i-1,:);
    y(i,:) = B(i,:)-sum;
end
L = L';
y = y./d;
for i = flip(1:n)
    sum = L(i, i+1:n)*x(i+1:n,:);
    x(i,:) = y(i,:)-sum;
end
L = L';
x = zeros(n,m);
```

Nie wgłębiając się w szczegóły kodu, metoda ta polega krótko na napisaniu ponownie metody eliminacji Gaussa. Robimy to w pętli. Najpierw redukujemy macierz do postaci trójkątnej górnej i rozwiązujemy układ równań, otrzymując ostatecznie szukaną macierz X . Na wejściu funkcja przyjmuje macierze A i B . Następnie macierz A rozkłada na LDL^T (jeżeli to możliwe), na wyjściu dostajemy macierz X .

3 Skrypt testujący

W tym punkcie przedstawię wyniki funkcji przeze mnie napisanych, porównam je do wzorcowego rozwiązania (funkcji `inv`), porównam czasy wykonania i obliczę różne błędy i współczynniki.

Pierwsze 4 przykłady wymyśliłem sam (nie są wygenerowane). Oto fragment kodu:

```
disp("Pierwszy przykład: ")
A1 = [4,-2,-2; -2,5,-1; -2,-1,6];
B1 = eye(3);
[L1, D1, L1T, info] = rozkladChol(A1)

rozwUkladu1(A1, B1)
rozwUkladu2(A1,B1)
solve_chol(A1,B1)
inv(A1)*B1

tic; rozwUkladu1(A1, B1); toc
tic; rozwUkladu2(A1,B1); toc
tic; solve_chol(A1,B1); toc
tic; inv(A1)*B1; toc
```

Kolejne 4 przykłady były wygenerowane, jedynie wybierałem, jaką wielkość ma mieć macierz A i macierz B. Oto fragment kodu:

```
disp("Piąty przykład: ")

[A5, B5] = gen_mac_sym_dod(10, 3);
B5 = eye(10); % użyjemy tu macierzy jednostkowej

A5
B5

[L5,D5,L5T, info] = rozkladChol(A5)

rozwUkladu1(A5,B5)
rozwUkladu2(A5,B5)
solve_chol(A5,B5)
inv(A5)*B5

tic; rozwUkladu1(A5,B5); toc
tic; rozwUkladu2(A5,B5); toc
tic; solve_chol(A5,B5); toc
tic; inv(A5)*B5; toc
```

Jak widać, za każdym razem wykonuję 4 różne wykonania zadania rozwiązania równania $AX = B$ i sprawdzam na terminalu, czy wyszły te same wyniki. Następnie liczę czas wykonania każdej funkcji.

Ponadto, liczę, jak bardzo się różni macierz funkcji `solve_chol` (napisanej przeze mnie ostatniej funkcji) od wyniku funkcji `inv`.

```

disp(" ")
disp("Badanie błędu wynikowego")
A = solve_chol(A1,B1) - inv(A1)*B1
B = solve_chol(A2,B2) - inv(A2)*B2
C = solve_chol(A3,B3) - inv(A3)*B3
D = solve_chol(A4,B4) - inv(A4)*B4
E = solve_chol(A5,B5) - inv(A5)*B5
F = solve_chol(A6,B6) - inv(A6)*B6
G = solve_chol(A7,B7) - inv(A7)*B7
H = solve_chol(A8,B8) - inv(A8)*B8

```

W wyniku otrzymaliśmy, że błędy wartości z wyznaczonej macierzy X są mniejsze niż 10^{-13} w każdej metodzie. Oznacza to, że metoda jest dokładna.

```

sum(A, "all")
sum(B, "all")
sum(C, "all")
sum(D, "all")
sum(E, "all")
sum(F, "all")
sum(G, "all")
sum(H, "all")

```

Tutaj badam sumę różnic wszystkich elementów - w najgorszym przypadku jest to 10^{-12} .

Oprócz tego w pliku matlabowym obliczałem wskaźniki uwarunkowania wygenerowanych macierzy A, błędy rozkładu $A = LDL^T$, błędy względne, współczynniki stabilności oraz współczynniki poprawności. W pierwszym przypadku wyszło, że jedna z najmniejszych macierzy była źle uwarunkowana, błędy rozkładu wynosiły max 10^{-17} , błędy względne max 10^{-14} , współczynnik stabilności 10^{-15} , a współczynnik poprawności także 10^{-15} .

4 Wnioski

Na podstawie przeprowadzonych testów wyszło, że najszybsza była funkcja inv, później była funkcja stosująca "\", następnie moja funkcja, a najwolniej działała funkcja wykorzystująca rref. Później porównując tylko moją funkcję (bo nie korzystała z "\") ani z rref) do funkcji inv wyszło, że moja funkcja zwraca bardzo bliskie wartości - suma błędów dla każdej macierzy A wyniosła max 10^{-12} . Uwarunkowanie macierzy jest bardzo różne i wynosi od ok 0 do 418 - zatem możemy przyjąć, że są one dobrze uwarunkowane. Błędy rozkładu, błędy względne, współczynniki stabilności i poprawności są bardzo małe, nie większe od 10^{-14} .

Zatem wnioskuję, że mimo że moja funkcja była wolniejsza od funkcji inv oraz tej wykorzystującej "\", tak wyniki funkcji były dokładne. Podobnie rozkład $A = LDL^T$ był bardzo dokładny.