TD 6 - Algèbre linéraire : éléments propres d'une matrice symétrique réelle

On considère une matrice symétrique réelle A d'ordre n.
On cherche une approximation de toutes les valeurs propres de la matrice A par la méthode de Jacobi.
Principe :
La matrice A étant symétrique, elle est diagonalisable. Il existe alors une matrice orthogonale O telle que $O^T AO = diag\,(\lambda_1, \lambda_2, \ldots, \lambda_n)$; les réels λ_i sont les valeurs propres de la matrice A, comptées avec leur multiplicité.
La méthode de Jacobi consiste à appliquer une suite de similitudes qui amènent progressivement la matrice A à la forme diagonale.
Chacune des transformations successives s'écrivent $\Omega^T M\Omega$ où Ω est une matrice orthogonale de la forme :

$$\begin{pmatrix}
1 & 0 & & \cdots & & & \cdots & & 0 \\
0 & \ddots & & & & & & & \\
 & & 1 & & & & & & \\
 & & & c & \ddots & s & & & \\
\vdots & & & \ddots & 1 & \ddots & & & \vdots \\
 & & & -s & \ddots & c & & & \\
 & & & & & & 1 & & \\
 & & & & & & & \ddots & 0 \\
0 & & & \cdots & & \cdots & & 0 & 1
\end{pmatrix}$$

avec

$$c = \cos\theta \text{ et } s = \sin\theta$$

On construit la suite de matrice (A_k) telle que : $A_0 = A$ et, pour tout k entier naturel, $A_{k+1} = \Omega_k^T A_k \Omega_k$.
Les matrices Ω_k sont choisies de manière à annuler un coefficient non diagonal de A_k, de plus grande valeur absolue :
Notons $A_k = [a_{ij}]$ et $a_{pq} = \max_{i\neq j} a_{ij}$.
Alors $\Omega_k = I_n + c\,(E_{pp} + E_{qq}) + s\,(E_{pq} - E_{qp})$ où $c = \cos\theta_k$ et $s = \sin\theta_k$.
On remarque que le produit par Ω_k ou Ω_k^T n'affecte que les indices de ligne et de colonne p et q.
On note $A_{k+1} = \left[a'_{ij}\right]$ et on obtient les égalités :

$$\begin{cases}
a'_{pj} = ca_{pj} - sa_{qj} \\
a'_{iq} = sa_{ip} + ca_{iq} \\
a'_{pp} = c^2 a_{pp} + s^2 a_{qq} - 2csa_{pq} \\
a'_{qq} = s^2 a_{pp} + c^2 a_{qq} + 2csa_{pq} \\
a'_{pq} = cs\,(a_{pp} - a_{qq}) + \left(c^2 - s^2\right) a_{pq}
\end{cases}$$

L'annulation du coefficient a'_{pq} conduit à l'égalité : $\dfrac{a_{qq} - a_{pp}}{2a_{pq}} = \cot 2\theta_k$;
si on pose $t = \tan\theta_k$ alors t est solution de l'équation : $t^2 + 2\dfrac{a_{qq} - a_{pp}}{2a_{pq}}t - 1 = 0$.
On en déduit c et s puis la matrice A_{k+1}. On arrête le calcul des matrices A_k lorsque les termes non diagonaux deviennent inférieurs à un certain seuil.
Le produit des matrices Ω_k donne alors une valeur approchée de la matrice orthogonal O telle que $O^T AO = diag\,(\lambda_1, \lambda_2, \ldots, \lambda_n)$. On en déduit une approximation des vecteurs propres associés aux valeurs propres λ_i.

Travail demandé :

1. Programmer la méthode de Jacobi : le programme reçoit la matrice symétrique réelle A et retourne une matrice A_{k_0} qui approche la matrice $diag\,(\lambda_1, \lambda_2, \ldots, \lambda_n)$ pour la précision requise.
   On effectue au fur et à mesure les calculs du produit des matrices Ω_k ; l'algorithme affiche une matrice dont les vecteurs colonnes sont des approximations des vecteurs propres.
2. Comparer les résultats obtenus avec ceux de la commande Scilab : [D,O]=bdiag(A).

Script Scilab

```
// Eléments propres d'une matrice réelle symétrique
// par la méthode de Jacobi
// ------------------------------------------------
M=[1,1,0.5;1,1,0.25;0.5,0.25,2];
A=evstr(x_matrix('entrer une matrice réelle symétrique :',M));
n=size(A);
Q=eye(n(1),n(1));
// ------------------------------------------------
M=tril(A,-1);
[m,lc]=max(abs(M));
p=lc(1);
q=lc(2);
// m contient l'élément max de A
while m>0.0005
// détermination de cos(theta) et sin(theta)
r=(A(q,q)-A(p,p))/(2*A(p,q))
if r==0 then
    t=1;
else
    t=-r+sign(r)*sqrt(1+r^2);
end
c=1/sqrt(1+t^2);
s=t/sqrt(1+t^2);
// ------------------------------------------------
// nouvelles lignes et colonnes
// après similitude
Ligne_p=c.*A(p,:)-s.*A(q,:);
Ligne_q=s.*A(p,:)+c.*A(q,:);
//Col_p=c.*A(:,p)-s*A(:,q)=Ligne_p'
//Col_q=s.*A(:,p)+c.*A(:,q)=Ligne_q'
App=c^2*A(p,p)+s^2*A(q,q)-2*c*s*A(p,q);
Aqq=s^2*A(p,p)+c^2*A(q,q)+2*c*s*A(p,q);
//Apq=c*s*(A(p,p)-A(q,q))+A(p,q)*(c^2-s^2) égal à 0
// ------------------------------------------------
//substitution dans la matrice A
A(p,:)=Ligne_p;
A(q,:)=Ligne_q;
A(:,p)=Ligne_p';
A(:,q)=Ligne_q';
A(p,p)=App;
A(q,q)=Aqq;
A(p,q)=0;//A(p,q)=Apq;
A(q,p)=0;//A(q,p)=Apq;
// On calcule la matrice orthogonale du changement de base
Col_p=c.*Q(:,p)-s*Q(:,q);
Col_q=s.*Q(:,p)+c.*Q(:,q);
Q(:,p)=Col_p;
Q(:,q)=Col_q;
// ------------------------------------------------------
M=tril(A,-1);
[m,lc]=max(abs(M));
p=lc(1);
q=lc(2);
end
// lecture des valeurs propres sur la diagonale de A
disp('valeurs propres:');
disp(diag(A));
// affichage de la matrice de vecteurs propres Q
disp('vecteurs propres (vecteurs colonnes):');
disp(Q);
```