Une solution du TP

La commande suivante charge la librairie « `eigen` » qui contient la fonction `columnvector`. Nous ne servirons pas d'autres fonctions de cette librairie. Elle semble pourtant adaptée à notre problème puisque les fonction `eigenvalues` et `eigenvectors` calculent les valeurs propres et les vecteurs propres d'une matrice. Mais ces fonctions retournent des valeurs exactes et les formules ainsi obtenues sont effrayantes. (Je vous suggère d'essayer.)

```
load(eigen);
```

1 – On saisit la matrice M :

```
M:matrix([10,7,8,7],[7,5,6,5],[8,6,10,9],[7,5,9,10]);
```

On calcule son déterminant :

```
determinant(M);
```

On trouve 1. par la formule $M^{-1} = \frac{1}{\det M}{}^t(\mathrm{com}\, M)$ on en déduit que les coefficients de M^{-1} sont des nombres entiers.

2 – On calcule $N = M^{-1}$, on saisit les vecteurs V et W et on calcule $X = NV$ et $Y = NW$ (la multiplication des matrices est effectuée par l'opérateur « `.` » et non par « `*` ») :

```
N:invert(M);
V:columnvector([32,23,33,31]);
W:columnvector([32.01,22.99,33.01,30.99]);
X:N.V;
Y:N.W;
```

3 – On sait que M est diagonalisable dans une base orthonormée car elle est symétrique réelle. (C'est un théorème classique.)

Soient $\lambda_1 < \lambda_2 < \lambda_3 < \lambda_4$ les valeurs propres de M, et V_1, V_2, V_3 et V_4 des vecteurs propres associés, formant une base orthonormée et soit U un vecteur.

Comme les V_i forment une base orthonormée, on a $U = \sum_{i=1}^{4} \mu_i V_i$ où $\mu_i =< U, V_i >$.

On en déduit que $M^{-1}U = \sum_{i=1}^{4} \mu_i M^{-1} V_i = \sum_{i=1}^{4} \frac{\mu_i}{\lambda_i} V_i$ et donc :

$$||M^{-1}U||^2 = \sum_{i=1}^{4} \left(\frac{\mu_i}{\lambda_i}\right)^2 \geq \left(\frac{\mu_1}{\lambda_1}\right)^2 = \left(\frac{< U, V_1 >}{\lambda_1}\right)^2$$

d'où le résultat cherché : $||M^{-1}U|| \geq \left|\frac{< U, V_1 >}{\lambda_1}\right|$.

4 – On calcule p, le polynôme caractéristique de M, puis le pgcd de p et p' (le polynôme dérivé). Le pgcd vaut 1 donc p et p' n'ont pas de racine commune et donc p n'a pas de racine multiple.

```
charpoly(M,x)$
p:expand(%);
q:diff(p,x);
gcd(p,q);
```

5 – On calcule les racines de p. Ce sont les valeurs propres. Le résultat de la fonction `allroots` est une liste de la forme : `[x=`λ_1`,...,x=`λ_4`]`.

```
vp:allroots(p);
```

6 – Ici ça se corse : soit λ une valeur propre **simple** d'une matrice M. On peut calculer un vecteur propre de M associé à la valeur propre λ de la façon suivante :

- On calcule la matrice $M' = M - \lambda I$. Comme λ est une valeur propre, on en déduit que $\det M' = 0$.
- Écrivons $M' = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}$ et considérons les déterminants suivants :

$$\Delta_1 = \begin{vmatrix} a_{22} & a_{23} & a_{24} \\ a_{32} & a_{33} & a_{34} \\ a_{42} & a_{43} & a_{44} \end{vmatrix} \quad \Delta_2 = \begin{vmatrix} a_{21} & a_{23} & a_{24} \\ a_{31} & a_{33} & a_{34} \\ a_{41} & a_{43} & a_{44} \end{vmatrix} \quad \Delta_3 = \begin{vmatrix} a_{21} & a_{22} & a_{24} \\ a_{31} & a_{32} & a_{34} \\ a_{41} & a_{42} & a_{44} \end{vmatrix} \quad \Delta_4 = \begin{vmatrix} a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{vmatrix}$$

  et le vecteur $V = \begin{pmatrix} \Delta_1 \\ -\Delta_2 \\ \Delta_3 \\ -\Delta_4 \end{pmatrix}$.
  On a alors : $\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \end{pmatrix} \times V = a_{11}\Delta_1 - a_{12}\Delta_2 + a_{13}\Delta_3 - a_{14}\Delta_4 = \det M' = 0$.
  On a aussi pour $i > 1$:

$$\begin{pmatrix} a_{i1} & a_{i2} & a_{i3} & a_{i4} \end{pmatrix} \times V = a_{i1}\Delta_1 - a_{i2}\Delta_2 + a_{i3}\Delta_3 - a_{i4}\Delta_4 = \det \begin{pmatrix} a_{i1} & a_{i2} & a_{i3} & a_{i4} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}$$

  qui est nul car la première ligne est égale à l'une des autres.
  Ce qui précède signifie donc que $M' \times V = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} \begin{pmatrix} \Delta_1 \\ -\Delta_2 \\ \Delta_3 \\ -\Delta_4 \end{pmatrix} = 0$ donc que V est une vecteur propre de M associé à la valeur propre λ.

Dans la procédure suivante λ est représenté par *vp*, M' par *MM*, et le vecteur propre V par *Vp* :

```
vecteur_propre(M,vp):=block
(
   [MM,n,I,Vp],
   n:length(M),
   I:ident(n),
   MM:M-vp.I,
   Vp:makelist((-1)**(i-1)*determinant(submatrix(1,MM,i)),i,1,n),
   Vp:columnvector(Vp),
   uvect(Vp)
)$
```

V est calculé par l'expression suivante (voir la documentation pour la syntaxe de la fonction `submatrix`.) :

```
Vp:makelist((-1)**(i-1)*determinant(submatrix(1,MM,i)),i,1,n),
```

Il est ensuite normalisé. (fonction `uvect`.)

Attention : si la valeur propre est multiple, alors la matrice $M' = M - \lambda I$ est de rang < 3 (ou $< n-1$ dans le cas général) et donc les déterminants $\Delta_1, \ldots, \Delta_4$ sont nuls (ou presque nuls en calcul approché). Le résultat de la procédure et alors erroné.

On peut maintenant calculer les 4 vecteurs propres, les matrices P, Q et D, et faire les vérifications demandées :

```
vp1:rhs(vp[1]);
Vp1:vecteur_propre(M,vp1);

vp2:rhs(vp[2]);
Vp2:vecteur_propre(M,vp2);

vp3:rhs(vp[3]);
Vp3:vecteur_propre(M,vp3);

vp4:rhs(vp[4]);
Vp4:vecteur_propre(M,vp4);

P:addcol(Vp1,Vp2,Vp3,Vp4);
Q:transpose(P);
D:matrix([vp1,0,0,0],[0,vp2,0,0],[0,0,vp3,0],[0,0,0,vp4]);
determinant(P);
P.Q-ident(4);
M-P.D.Q;
```

7 – Enfin :

```
U:columnvector([1,-1,1,-1]);
abs(innerproduct(U,Vp1)/vp1);
```

On trouve $||M^{-1}U|| \geq 163,96\ldots$

8 – On a $W = V + \dfrac{1}{100}U$ donc $M^{-1}(W-V) = \dfrac{1}{100}M^{-1}U$ et donc $||M^{-1}(W-V)|| \geq 1,6396\ldots$ ce qui explique l'écart important entre $M^{-1}V$ et $M^{-1}W$. Le phénomène d'instabilité observé est donc lié au fait qu'une des valeurs propres de M est très petite.