Propriétés : (Algorithme de Faddeev)

Soit $A \in \mathcal{M}_n(\mathbb{R})$. On pose $B_0 = I_n$ et pour $k \in [\![1; n-1]\!]$ on définit les matrices B_k par récurrence : $B_k = AB_{k-1} - \frac{tr(AB_{k-1})}{k} I_n$. On a alors, au signe près, le polynôme caractéristique de A : $\chi_A(X) = \det(XI_n - A) = X^n + a_1 X^{n-1} + \cdots + a_{n-1}X + a_n$ avec $a_k = \frac{-tr(AB_{k-1})}{k}$ pour tout $k \in [\![1; n]\!]$.

preuve :

1) $\chi'_A(X) = tr\left({}^t com(XI_n - A)\right)$

On note $(a_{i,j}(X))_{1 \leqslant i,j \leqslant n}$ les coefficients de $(XI_n - A)$. On a : $\chi_A(X) = \sum_{\sigma \in \mathcal{S}_n} \varepsilon(\sigma) a_{\sigma(1),1}(X) \dots a_{\sigma(n),n}(X)$ donc en dérivant :

$$\chi'_A(X) = \sum_{\sigma \in \mathcal{S}_n} \left(\sum_{k=1}^{n} \varepsilon(\sigma) a_{\sigma(1),1}(X) \dots a_{\sigma(k-1),k-1}(X) a'_{\sigma(k),k}(X) a_{\sigma(k+1),k+1}(X) \dots a_{\sigma(n),n}(X) \right)$$

soit en échangeant l'ordre de sommation :

$$\chi'_A(X) = \sum_{k=1}^{n} \det\left(C_1(X), \dots, C_{k-1}(X), C'_k(X), C_{k+1}(X), \dots, C_n(X)\right)$$

où $C_k(X)$ représente la $k^{ème}$ colonne de $(XI_n - A)$. Or les coefficients de $C_k(X)$ ne dépendent de X qu'à la $k^{ème}$ ligne, donc $C'_k(X)$ est le $k^{ème}$ vecteur colonne de la base canonique ${}^t(0, \dots 0, 1, 0, \dots 0)$. Si donc on développe $\det\left(C_1(X), \dots, C_{k-1}(X), C'_k(X), C_{k+1}(X), \dots, C_n(X)\right)$ par rapport à la $k^{ème}$ colonne on obtient alors le $k^{ème}$ cofacteur de la diagonale de $(XI_n - A)$. Finalement, $\chi'_A(X)$ est égal à la somme des cofacteurs des éléments diagonaux de $(XI_n - A)$, donc $\chi'_A(X) = tr\left(com(XI_n - A)\right) = tr\left({}^t com(XI_n - A)\right)$.

2) Identification des B_k et a_k

On pose $\chi_A(X) = X^n + a_1 X^{n-1} + \cdots + a_k X^{n-k} + \cdots + a_{n-1}X + a_n$, on a donc :

$$\chi'_A(X) = nX^{n-1} + \cdots + (n-k)a_k X^{n-k-1} + \cdots + a_{n-1}$$

Or chaque cofacteur de $(XI_n - A)$ est un polynôme en X de degré au plus $n-1$, on peut donc décomposer ${}^t com(XI_n - A)$ en matrices $(B_k)_{0 \leqslant k \leqslant n-1}$ dont les coefficients ont le même degré en X :

$${}^t com(XI_n - A) = B_0 X^{n-1} + \cdots + B_k X^{n-k-1} + \cdots + B_{n-1}$$

Soit en prenant la trace et en identifiant au polynôme $\chi'_A(X)$:

$$tr(B_0) = n;\ \dots;\ tr(B_k) = (n-k)a_k;\ \dots;\ tr(B_{n-1}) = a_{n-1} \qquad (*)$$

On se sert maintenant de l'égalité, toujours vraie $(XI_n - A)\,{}^t com(XI_n - A) = \det(XI_n - A)I_n$ soit :

$$(XI_n - A)\left(B_0 X^{n-1} + \cdots + B_k X^{n-k-1} + \cdots + B_{n-1}\right) = \chi_A(X) I_n$$

En développant le membre de gauche et en identifiant à nouveau avec le degré, on a alors :

$$B_0 = I_n;\ \dots;\ B_k - AB_{k-1} = a_k I_n;\ \dots;\ -AB_{n-1} = a_n I_n \qquad (**)$$

soit en prenant à nouveau la trace :

$$tr(B_0) = n;\ \dots;\ tr(B_k) = tr(AB_{k-1}) + na_k;\ \dots;\ 0 = tr(AB_{n-1}) + na_n \qquad (***)$$

Maintenant, en retranchant $(*)$ à $(***)$, on trouve $a_k = \frac{-tr(AB_{k-1})}{k}$ pour tout $k \in [\![1; n]\!]$.
Reste alors à reprendre $(**)$ en remplaçant a_k par $\frac{-tr(AB_{k-1})}{k}$ pour trouver finalement :

$$B_k = AB_{k-1} - \frac{tr(AB_{k-1})}{k} I_n \qquad (\forall k \in [\![1; n-1]\!])$$

José Gregorio : http://agregorio.net

Algorithme en Scilab :

Cet algorithme prend en entrée une matrice carrée A et retourne un vecteur p dont les composantes sont les coefficients $(a_k)_{1\leqslant k\leqslant n}$ du polynôme caractéristique de A. Il retourne aussi $B = A^{-1}$ si A est inversible.

```
function [B,p]=faddeev(A)
    n=size(A,1) ;
    B=eye(n,n) ;
    p=ones(1,n+1) ;

    for i=2:n+1
        p(i)=-trace(A*B)/(i-1) ;
        if(i<>n+1) then B=A*B+p(i)*eye(n,n) ;end ;
    end
    if(p(n+1)<>0) then B=-B/p(n+1) ;
    else B=("la matrice n''est pas inversible") ; end ;
endfunction
```

Remarques :

1. Cette méthode est assez performante puisqu'elle permet, sans coût supplémentaire, de trouver A^{-1} quand A est inversible et $\det(A)$, on le voit à la fin de la ligne $(**)$ où l'on a $-AB_{n-1} = a_n I_n$.

2. Une question du jury pourrait être : quelle est le coût de cet algorithme ? Il faut y avoir un peu réfléchi avant (combien de multiplications, d'additions ?). On peut aussi comparer cet algorithme avec la commande poly(A,'x') de Scilab sur des exemples.

Thèmes abordés :

* Algorithme
* Matrice
* Determinant
* Polynôme caractéristique

Bibliographie :

"Les maths en tête, Algèbre" de Xavier Gourdon (ELLIPSES)

On peut aussi trouver une preuve différente de cet algorithme (*avec les formules de Newton*) dans "Oraux X-ENS, algèbre 2" de S. Francinou, H. Gianella, S. Nicolas (CASSINI)

José Gregorio : http://agregorio.net