

Algorithme de Fadeev

Pierre Lissy

May 5, 2010

Lemme 1. *Dérivée en temps de $\det(A(t)) : \sum_1^n \det(C_1(t), \dots, C_i'(t), C_n(t))$.*

Proof. On utilise la forme développée du déterminant. $\det(A(t)) = \sum_{\sigma \in S_n} a_{\sigma(1),1}(t) \dots a_{\sigma(n),n}(t)$. On dérive par rapport au temps, on obtient en utilisant la formule de dérivation d'un produit à n termes que $\det'(A(t)) = \sum_{\sigma \in S_n} \sum_{i=1}^n a_{\sigma(1),1}(t) \dots a_i'(t) a_{\sigma(n),n}(t) = \sum_{i=1}^n \sum_{\sigma \in S_n} a_{\sigma(1),1}(t) \dots a_i'(t) a_{\sigma(n),n}(t) = \sum_1^n \det(C_1(t), \dots, C_i'(t), C_n(t))$. $\square$

Lemme 2. $\chi'_A = \text{tr}(\text{com}(XI_n - A))$

Proof. On utilise la formule précédente. $\det(tI_n - A)' = \sum_1^n \det(C_1(t), \dots, C_i'(t), C_n(t))$ avec $C_i = t\varepsilon_i - A_i$. La dérivée en temps donne donc ε_i . On a donc la somme sur i des déterminants $\det(C_1(t), \dots, \varepsilon_i, C_n(t))$. Or justement chacun de ces déterminants est égal au mineur obtenu en retirant la ligne i et la colonne i , autrement dit c'est le coefficients diagonal numéro i de la matrice des cofacteurs de $XI_n - A$. Comme on fait la somme sur i on obtient bien la trace. $\square$

Décrivons l'algorithme en lui-même. Ecrivons le polynôme caractéristique sous la forme $x^n + a_1x^{n-1} + a_n$. Les cofacteurs étant de degré au plus $n-1$ on peut écrire la transposée de la comatrice sous la forme $B_0X^{n-1} + \dots + B_n$. L'égalité $\chi'_A = \text{tr}({}^t\text{com}(XI_n - A))$ entraîne $(n-1)a_1 = \text{tr}(B_1), \dots, a_{n-1} = \text{Tr}(B_{n-1})$. De plus, on a la relation $(XI_n - A)^t \text{Com}(XI_n - A) = \det(XI_n - A)I_n$, soit encore $B_0X^n + (B_1 - AB_0)X^{n-1} + \dots + (B_{n-1} - AB_{n-2}X - AB_{n-1}) = \chi_A(X)I_n$. En identifiant les coefficients, on obtient $B_0 = I_n, \dots, B_{n-1} - AB_{n-2} = a_{n-1}I_n$, puis en prenant la trace de tout ceci on en déduit que $na_1 = \text{Tr}(B_1) - \text{Tr}(AB_0), \dots, na_{n-1} = \text{tr}(B_{n-1} - AB_{n-2}), na_n = -\text{tr}(AB_{n-1})$.

Mais en retranchant les deux égalités obtenues, on voit que $a_1 = -\text{tr}(AB_0), \dots, (n-1)a_{n-1} = -\text{Tr}(AB_{n-2}), na_n = \text{tr}(AB_{n-1})$. On voit donc qu'en réinjectant dans l'égalité matricielle ci-dessus, on peut obtenir les B_i par récurrence: $B_0 = I_n, B_1 = AB_0 - \text{tr}(AB_0)I_n, \dots, B_{n-2} = AB_{n-2} - \text{Tr}(AB_{n-2})/(n-1)I_n$. Ainsi, on a déjà trouvé le polynôme caractéristique. De plus, si la matrice est inversible, toujours l'égalité matricielle nous donne que $A^{-1} = -B_{n-1}/a_n = n/(\text{Tr}(AB_{n-1}))B_{n-1}$.

Cet algorithme est donc d'intérêt triple. Il permet d'obtenir le polynôme caractéristique, le déterminant et l'inverse de la matrice en coût très raisonnable. En effet, le coup est grosso modo du n^4 (on effectue à chaque fois un produit matriciel ce qui coûte n^3 , on rajoute un calcul de trace qui est négligeable, et on fait tout ceci pour $n-1$ matrices).