

Corrigé du Partiel de M.A.O. Calcul Formel

21 mai 2021
Master 1 M.F., Orsay

Toutes les fonctions programmées en `Sage` sont disponibles dans le Jupyter Notebook qui accompagne ce pdf.

Partie 1 : Résolution d'une congruence

1. Supposons $F(0) \neq 0$ et notons $f : \mathbb{K}_{n-1}[X] \to \mathbb{K}_{n-1}[X]$ l'application qui à un polynôme $G \in \mathbb{K}_{n-1}[X]$ associe le reste dans la division euclidienne de $F(X)G(X)$ par X^n. Cette application est clairement $\mathbb{K}$-linéaire : c'est un endomorphisme de l'espace vectoriel $\mathbb{K}_{n-1}[X]$, qui est de dimension finie. En outre elle est injective : si $f(G) = 0$ alors X^n divise $F(X)G(X)$, donc X^n divise G (car $F(0) \neq 0$), donc $G = 0$. Donc elle est bijective, et en particulier il existe un et un seul polynôme $G \in \mathbb{K}_{n-1}[X]$ tel que $f(G) = 1$.
   Une méthode alternative peut être de raisonner à la main. L'existence d'un tel polynôme est garantie en prenant une relation de Bézout. Si maintenant, on suppose qu'il existe deux tels polynômes G_1 et G_2 de degré inférieur à $n-1$. On a alors que $X^n \mid F(X)(G_1(X) - G_2(X))$ et donc 0 est une racine de multiplicité au moins n de $G_1 - G_2$ (car $F(0) \neq 0$). Cela implique immédiatement que $G_1 = G_2$ au vu des degrés de G_1 et G_2.
   Réciproquement, si un tel polynôme existe car on a $F(X)G(X) = 1 + Q(X)X^n$ pour un certain polynôme Q, ce qui donne $F(0)G(0) = 1$ donc $F(0) \neq 0$.
2. La relation (1) signifie qu'on a $c_0 = 1$ et $c_k = 0$ pour tout $k \in \{1, \dots, n-1\}$, en notant c_k les coefficients du produit $F(X)G(X)$. En notant $F(X) = \sum_{i=0}^{n-1} f_i X^i$ et $G(X) = \sum_{i=0}^{n-1} g_i X^i$, cette relation équivaut donc au système

$$\begin{cases} f_0 g_0 = 1 \\ \sum_{i=0}^{k} f_{k-i} g_i = 0 \text{ pour tout } k \in \{1, \dots, n-1\} \end{cases}$$

   La matrice de ce système est triangulaire inférieure, et tous ses coefficients diagonaux sont égaux à f_0 qui est non nul par hypothèse. Elle est donc inversible (ce qui donne une autre preuve de l'implication $(i) \Rightarrow (ii)$ dans la question 1). Pour résoudre ce système on peut appliquer sans réfléchir l'algorithme du pivot de Gauss, ce qui

coûte $O(n^3)$ opérations arithmétiques dans $\mathbb{K}$. Mais étant donné que la matrice est triangulaire, c'est vraiment dommage : il suffit de déterminer successivement g_0, g_1, …, g_{n-1} puisque chaque g_k est donné en fonction des précédents par

$$g_k = \frac{-1}{f_0} \sum_{i=0}^{k-1} f_{k-i} g_i.$$

Le calcul de chaque g_k coûte donc $O(k)$ opérations, ce qui au total revient à $O(n^2)$ opérations arithmétiques dans $\mathbb{K}$.

3. Le polynôme G est l'unique polynôme de $\mathbb{K}_{n-1}[X]$ pour lequel il existe un polynôme Q vérifiant $F(X)G(X) - X^n Q(X) = 1$. Cette relation est une relation de Bezout : l'algorithme d'Euclide étendu permet de la trouver, et donc de déterminer $G(X)$ (rappelons que cet algorithme fournit bien une relation de Bezout "minimale", c'est-à-dire dans notre exemple telle que $\deg G < n$ et $\deg Q < \deg F$). Son coût est $O(n^2)$ opérations arithmétiques dans $\mathbb{K}$, comme celui de l'approche vue à la question 2.
4. ($\star$) On obtient un temps d'exécution de `0.0010838508605957031` et on obtient

$$G = -390x^9 - 208x^8 - 111x^7 - 59x^6 - 32x^5 - 16x^4 - 8x^3 - 4x^2 - 2x - 1.$$

5. Pour $k = 0$, on a bien $F(X)G_0(X) = \frac{1}{F(0)} F(X) \equiv 1 \bmod X$. Supposons que $FG_{k-1} = 1 + X^{2^{k-1}} Q(X)$ pour un certain k tel que $1 \leq k \leq L-1$, avec $Q \in \mathbb{K}[X]$. Alors on a :

$$FG_k = FG_{k-1}(2 - FG_{k-1}) = (1 + X^{2^{k-1}} Q(X))(1 - X^{2^{k-1}} Q(X)) = 1 - X^{2^k} Q(X)^2$$

donc $FG_k \equiv 1 \bmod X^{2^k}$, ce qui termine la preuve. Avec $k = L$, cette relation donne bien $FG_L \equiv 1 \bmod X^n$.
6. ($\star$) On obtient

$$\left[1, 1, -x^3 - x^2 + 1, -12x^7 - 4x^6 + 2x^5 + 3x^4 - x^3 - x^2 + 1\right].$$

7. L'algorithme naïf donne $\alpha = 2$, et l'algorithme de multiplication rapide de Karatsuba done $\alpha = \log_2(3) \simeq 1.58$. On renvoie au cours pour la description du principe de l'algorithme, l'idée principale étant de se contenter de trois multiplications au lieu de quatre.
8. La question 5 montre qu'en calculant successivement G_0, G_1, …, G_L on trouve le polynôme G cherché. Si on le fait naïvement, le calcul de chaque G_k en fonction de G_{k-1} coûte à lui seul $O(n^\alpha)$ opérations arithmétiques dans $\mathbb{K}$, car il faut calculer le produit FG_{k-1} et le degré de F peut valoir jusqu'à $n-1$. L'observation cruciale est que calculer ce produit en entier est inutile, puisque G_k est défini par une troncature modulo X^{2^k}. Notons F_k le polynôme F tronqué, dans lequel on ne garde que les puissances de X strictement inférieures à 2^k ; autrement dit, $F_k \in \mathbb{K}_{2^k-1}[X]$ est défini par $F_k \equiv F \bmod X^{2^k}$. Alors le calcul de $G_{k-1}(2 - F_k G_{k-1})$ se fait en $O((2^k)^\alpha)$

opérations : on calcule le produit $F_k G_{k-1}$, puis on soustrait ce polynôme à 2, et on le multiplie par G_{k-1}. Notons que $F_k G_{k-1}$ peut être de degré compris entre 2^k et $2^k + 2^{k-1} - 2$: on peut le tronquer en ne gardant que les termes de degré $< 2^k$, ce qui peut gagner un peu de temps de calcul, mais ne pas le faire change simplement la constante implicite dans le symbole O. Rappelons aussi que tronquer un polynôme ne coûte aucune opération arithmétique. En notant λ une constante telle que le coût du calcul de G_k à partir de G_{k-1} soit majoré par $\lambda 2^{\alpha k}$, on obtient un coût du calcul de G majoré par

$$\sum_{k=1}^{L} \lambda 2^{\alpha k} = \lambda \cdot 2^{\alpha} \cdot \frac{2^{\alpha L} - 1}{2^{\alpha} - 1} = O(2^{\alpha L}) = O(n^{\alpha}).$$

9. Soit n un entier quelconque ; il existe une puissance de 2, notée n', telle que $\frac{1}{2}n' < n \leq n'$. Etant donné $F \in \mathbb{K}_{n-1}[X] \subset \mathbb{K}_{n'-1}[X]$ on peut appliquer ce qui précède avec n', et trouver ainsi un polynôme $G \in \mathbb{K}_{n'-1}[X]$ tel que $FG \equiv 1 \bmod X^{n'}$. En tronquant G modulo X^n on trouve le polynôme cherché. Le coût de ce calcul est celui de la question 8, c'est-à-dire $O(n'^{\alpha})$ opérations dans $\mathbb{K}$. Comme $n' < 2n$ il s'agit encore d'un coût en $O(n^{\alpha})$ opérations dans $\mathbb{K}$.
10. ($\star$) On obtient un temps d'exécution de `0.00013208389282226562` bien inférieur pour l'exemple de la question 4. On obtient dans le cas de $\mathbb{Z}/97\mathbb{Z}$ le polynôme

$$G = 91x^{13}+54x^{12}+7x^{11}+54x^{10}+28x^9+60x^8+23x^7+82x^6+81x^5+88x^4+66x^3+80x^2+47x+7.$$

Partie 2 : Application à la division euclidienne

11. L'algorithme de division euclidienne naïf, vu en cours, coûte $O(ab)$ opérations arithmétiques dans $\mathbb{K}$ puisque $a, b > 0$.
12. Notons $P(X) = \sum_{i=0}^{d} p_i X^i$. Alors $\widetilde{P}(X) = \sum_{i=0}^{d} p_i X^{d-i} = \sum_{j=0}^{d} p_{d-j} X^j$ par le changement d'indice $j = d - i$. Il s'agit donc d'un polynôme de degré inférieur ou égal à d, dont les coefficients sont les mêmes que ceux de P mais dans l'ordre inverse. On peut déjà remarquer, même si ce n'est pas demandé dans cette question, que calculer $\widetilde{P}$ à partir de P ne coûte aucune opération arithmétique.
13. ($\star$) On obtient un type
`<class 'sage.rings.fraction_field_element.FractionFieldElement_1poly_field'>`
qui fait que `Sage` considère Q comme une fraction rationnelle et tout un tas de commandes pour les polynômes ne fonctionnent alors plus. Ce n'est donc pas la bonne façon de procéder. On utilise plutôt que $\tilde{P}$ est le polynôme obtenue en renversant l'ordre des coefficients. Ainsi, on obtient ainsi bien $\tilde{P} = 2x^3 - x^2 + x + 1$ et un type
`<class 'sage.rings.polynomial.polynomial_rational_flint.Polynomial_rational_flint'>`
identique à celui de P.

14. On a

$$\widetilde{A}(X) = X^a A(1/X) = X^a B(1/X)Q(1/X) + X^a R(1/X) = \widetilde{B}(X)\widetilde{Q}(X) + X^{a-b+1}\widetilde{R}(X)$$

en posant $\widetilde{R}(X) = X^{b-1}R(1/X)$. Comme $\deg R \leq b-1$ et $\deg Q = a-b$, la question 12 montre que $\widetilde{Q}$ et $\widetilde{R}$ sont des polynômes, ce qui termine la preuve.

15. On applique la question 9 de la partie 1 avec $n = a - b + 1$ et $F(X) = \widetilde{B}(X)$ si $b \leq a - b$ (i.e., si $b \leq a/2$) ; dans le cas général on prend pour F le polynôme obtenu en tronquant $\widetilde{B}$ modulo X^n. Comme B est exactement de degré b, le coefficient constant $F(0) = \widetilde{B}(0)$ de F est non nul, puisque c'est le coefficient de degré b de B. On peut donc calculer un polynôme G tel que $\deg G \leq a - b$ et $\widetilde{B}(X)G(X) \equiv F(X)G(X) \equiv 1 \bmod X^{a-b+1}$. On a alors :

$$\widetilde{A}(X)G(X) \equiv \widetilde{B}(X)G(X)\widetilde{Q}(X) \equiv Q(X) \bmod X^{a-b+1}$$

d'après la question 14. Comme $\deg Q = a - b$ on peut calculer Q en tronquant le polynôme $\widetilde{A}(X)G(X)$ modulo X^{a-b+1}. Estimons maintenant le coût de ce calcul. D'après la question 9 de la partie 1, le calcul de G peut être fait en $O(n^\alpha)$ opérations arithmétiques dans $\mathbb{K}$. Au lieu de calculer $\widetilde{A}(X)G(X)$ on peut préalablement tronquer $\widetilde{A}(X)$ modulo X^{a-b+1} ; le calcul de se produit par l'algorithme de Karatsuba coûte alors $O(n^\alpha)$ opérations. Finalement on obtient donc Q en $O(n^\alpha)$ opérations.

Partie 3 : Application aux sommes de Newton

16. On a dans $\mathbb{K}(X)$:

$$P'(X) = c\sum_{i=1}^{d}\prod_{j\neq i}(X - \alpha_j) = P(X)\sum_{i=1}^{d}\frac{1}{X - \alpha_i}$$

donc

$$\widetilde{P'}(X) = X^{d-1}P'(1/X) = X^{d-1}P(1/X)\sum_{i=1}^{d}\frac{X}{1 - \alpha_i X} = \widetilde{P}(X)\sum_{i=1}^{d}\frac{1}{1 - \alpha_i X}.$$

Or on a pour tout i :

$$\frac{1}{1 - \alpha_i X} = \Big(\sum_{k=0}^{n-1}\alpha_i^k X^k\Big) + \frac{\alpha_i^n X^n}{1 - \alpha_i X}$$

ce qui donne en permutant les sommes finies :

$$\widetilde{P'}(X) = \widetilde{P}(X)\sum_{k=0}^{n-1}S_k(P)X^k + X^n\widetilde{P}(X)\sum_{i=1}^{d}\frac{\alpha_i^n}{1 - \alpha_i X}$$

Dans cette expression le dernier terme est de la forme $X^n R(X)$ avec $R \in \mathbb{K}(X)$, et 0 n'est pas un pôle de R. Par ailleurs c'est aussi un polynôme, car il est égal à $\widetilde{P'}(X) - \widetilde{P}(X)\sum_{k=0}^{n-1} S_k(P)X^k$. Donc c'est un polynôme multiple de X^n, ce qui termine la preuve.

17. On applique la question précédente et la partie 1 (question 9) avec $n = d+1$, et $F = \widetilde{P}$. On a bien $\deg F \leq n-1$ et $F(0) \neq 0$ car $\deg P = d$. On calcule donc G en $O(d^\alpha)$ opérations arithmétiques dans $\mathbb{K}$. On multiplie ensuite G par le polynôme $\widetilde{P'}(X)$, ce qui coûte aussi $O(d^\alpha)$ opérations. Les sommes de Newton cherchées sont alors les coefficients de degrés 0 à d de ce polynôme, puisque

$$G(X)\widetilde{P'}(X) \equiv G(X)\widetilde{P}(X)\sum_{k=0}^{d} S_k(P)X^k \equiv \sum_{k=0}^{d} S_k(P)X^k \bmod X^{d+1}.$$

18. ($\star$) On obtient la liste suivante :

$$[S_0, \ldots, S_{13}] = [13, 0, 32, 0, 27, 45, 44, 38, 25, 35, 29, 27, 24, 54]\,.$$

Noter qu'une version naïve en $O(d^2)$ est aussi implémentée à l'aide des identités de Newton qui fournissent que si $P = \sum_{i=0}^{d} a_{d-i}X^i$ avec $a_0 \neq 0$, alors

$$\forall k \in \{0, \ldots, d\}, \quad a_0 S_k = -k a_k - \sum_{i=1}^{k-1} a_{d-i} S_{k-i}.$$

Vous pouvez essayer de démontrer cette identité ou la trouver par exemple dans l'exercice 5.31 de la 3ème édition du livre *Oraux X-ENS, Algèbre 1* de S. Francinou, H. Gianella et S. Nicolas. Pour plus de compléments, vous pouvez aussi regarder l'algorihtme de Fadeev qui utilise ce résultat sur les sommes de Newton pour obtenir un algorithme de calcul du polynôme minimal d'une matrice souvent plus efficace que le calcul d'un déterminant par pivot de Gauß.