

Partiel de M.A.O. Calcul Formel Durée : 3 heures

21 mai 2021
Master 1 M.F., Orsay

Dans tout le problème, on note $\mathbb{K}$ un corps sur lequel les opérations arithmétiques (addition, soustraction, multiplication, division) sont implémentées. On note n un entier strictement positif (dont on imagine qu'il est susceptible d'être très grand) et $\mathbb{K}_{n-1}[X]$ l'ensemble des polynômes à coefficients dans $\mathbb{K}$ de degré inférieur ou égal à $n-1$. Lorsqu'on évoquera le coût d'un algorithme, il s'agira toujours de son coût en nombre d'opérations arithmétiques dans $\mathbb{K}$ et on attendra toujours une réponse de la forme $O(\ldots)$: déterminer la constante implicite dans le symbole $O(\ldots)$ n'est pas demandé.

Il est autorisé d'admettre le résultat de certaines questions pour traiter les suivantes.

Partie 1 : Résolution d'une congruence

Fixons un entier $n \geq 1$ et un polynôme $F \in \mathbb{K}_{n-1}[X]$. Le but de cette partie est de construire, de plusieurs façons, un polynôme $G \in \mathbb{K}_{n-1}[X]$ tel que

$$F(X)G(X) \equiv 1 \bmod X^n. \tag{1}$$

1. Démontrer que les assertions suivantes sont équivalentes :
   (i) $F(0) \neq 0$.
   (ii) Il existe un et un seul polynôme $G \in \mathbb{K}_{n-1}[X]$ vérifiant (1).

Dans toute la suite on suppose $F(0) \neq 0$, si bien qu'il existe un et un seul polynôme $G \in \mathbb{K}_{n-1}[X]$ vérifiant (1).

2. Écrire un système linéaire, dont les inconnues sont les coefficients de G, qui soit équivalent à la relation (1). En utilisant ce point de vue, comment s'appelle l'algorithme général qui permet de déterminer G, et quel est son coût en nombre d'opérations arithmétiques dans $\mathbb{K}$? Étant donné la forme particulière de la matrice de ce système, comment peut-on améliorer cette approche et quel est alors le coût de ce calcul ?
3. Proposer une approche différente de celle de la question 2, fondée sur une application directe d'un algorithme vu en cours, qui permet de déterminer G. Donner son coût en nombre d'opérations arithmétiques dans $\mathbb{K}$ et le comparer à celui obtenu à la question 2.
4. ($\star$) Implémenter une des méthodes précédentes (au choix) en une fonction `Sage` qui prend en entrée un entier n et un polynôme F de degré inférieur ou égal à $n-1$, et renvoie le polynôme G. En utilisant cette fonction, calculer G lorsque

   $$\mathbb{K} = \mathbb{Q}, \quad n = 10, \quad F(X) = 2X^9 + 3X^7 - 5X^6 + 2X - 1.$$

   On fera afficher le temps d'exécution et on vérifiera que $F(X)G(X) \equiv 1 \bmod X^n$.

Les questions qui suivent sont indépendantes des questions 2 et 3, et visent à construire G de façon plus efficace. On suppose toujours $F(0) \neq 0$, et en outre on suppose que n est une puissance de 2. On écrit $n = 2^L$ et on suppose $L \geq 1$. On pose $G_0 = \frac{1}{F(0)}$ et pour tout entier k tel que $1 \leq k \leq L$ on note G_k l'unique polynôme de degré strictement inférieur à 2^k tel que

$$G_k \equiv G_{k-1}(2 - FG_{k-1}) \bmod X^{2^k}.$$

Autrement dit, G_k est obtenu à partir du produit $G_{k-1}(2 - FG_{k-1})$ en ne gardant que les termes de degré strictement inférieur à 2^k.

5. Démontrer par récurrence que pour tout $k \in \{0, \ldots, L\}$ on a

$$FG_k \equiv 1 \bmod X^{2^k}$$

et en déduire que G_L est le polynôme G qu'on cherche à construire ; on rappelle que $n = 2^L$.

6. ($\star$) Programmer en `Sage` une fonction qui prend en entrée un entier L et un polynôme F de degré inférieur ou égal à $2^L - 1$, et renvoie une liste contenant les polynômes G_0, G_1, …, G_L. En utilisant cette fonction, déterminer cette liste de polynômes lorsque

$$\mathbb{K} = \mathbb{Q}, \quad L = 3, \quad F(X) = 5X^7 - 2X^4 + X^3 + X^2 + 1.$$

On vérifiera que $F(X)G_L(X) \equiv 1 \bmod X^n$.

Dans toute la suite du problème on note $\alpha \geq 1$ un réel pour lequel on sait, quel que soit $d \geq 1$, calculer le produit de deux polynômes de $\mathbb{K}[X]$ de degré inférieur ou égal à d en $O(d^\alpha)$ opérations arithmétiques dans $\mathbb{K}$.

7. Rappeler la valeur de α qui correspond à l'algorithme naïf, puis la valeur améliorée de α vue en cours et le nom de l'algorithme qui permet de l'obtenir. On donnera une *brève* description du principe de cet algorithme. *Le paramètre α est utilisé jusqu'à la fin du problème pour calculer les complexités. Pour les questions de programmation en* `Sage`*, on utilisera la multiplication des polynômes telle qu'elle est implémentée en* `Sage`*.*
8. En utilisant la question 5, démontrer qu'en menant bien les calculs, on peut déterminer le polynôme $G \in \mathbb{K}_{n-1}[X]$ vérifiant (1) en $O(n^\alpha)$ opérations arithmétiques dans $\mathbb{K}$.
9. Expliquer comment étendre le résultat de la question 8 au cas où n n'est pas une puissance de 2.
10. ($\star$) En utilisant la méthode de la question précédente et la fonction programmée à la question 6, créer en `Sage` une nouvelle fonction qui prend en entrée un entier n et un polynôme F de degré inférieur ou égal à $n - 1$, et renvoie le polynôme $G \in \mathbb{K}_{n-1}[X]$ vérifiant (1). Tester cette fonction avec

$$\mathbb{K} = \mathbb{Z}/97\mathbb{Z}, \quad n = 14, \quad F(X) = 6X^{13} + X^{11} + 43X^8 + 59X^4 + 3X + 14$$

où les coefficients de F sont des entiers modulo 97, puis avec

$$\mathbb{K} = \mathbb{Q}, \quad n = 10, \quad F(X) = 2X^9 + 3X^7 - 5X^6 + 2X - 1.$$

Dans les deux cas on vérifiera que $F(X)G(X) \equiv 1 \bmod X^n$, et dans le deuxième cas on comparera également le temps d'exécution de cet algorithme avec celui de la question 4.

Partie 2 : Application à la division euclidienne

Dans cette partie on se donne deux polynômes non nuls A et B, à coefficients dans $\mathbb{K}$. On note $a = \deg A$ et $b = \deg B$; on suppose $a > b$. L'objectif est de donner un algorithme, plus efficace que celui vu en cours, pour calculer le quotient Q dans la division euclidienne de A par B, c'est-à-dire le polynôme Q pour lequel il existe un reste $R \in \mathbb{K}[X]$ vérifiant $A = BQ + R$ et $\deg R < b$.

11. Rappeler le coût (en nombre d'opérations arithmétiques dans $\mathbb{K}$) de l'algorithme de division euclidienne naïf, vu en cours.
12. Soit $P \in \mathbb{K}[X]$ un polynôme de degré $d \in \mathbb{N}$. Démontrer que $\widetilde{P}(X) = X^d P(1/X)$ est un polynôme de degré inférieur ou égal à d, appelé *polynôme réciproque de* P, et exprimer les coefficients de $\widetilde{P}$ en fonction de ceux de P.
13. ($\star$) Déterminer le type de `Q` où `Q=x^3P(1/x)` avec `P=x^3+x^2-x+2`. Implémenter une fonction en `Sage` qui prend en entrée un polynôme P et qui renvoie son polynôme réciproque $\widetilde{P}$, la sortie devant être du même type que P.

Dans les questions 14 et 15 on considère les polynômes suivants :

$$\widetilde{A}(X) = X^a A(1/X), \quad \widetilde{B}(X) = X^b B(1/X), \quad \widetilde{Q}(X) = X^{a-b} Q(1/X).$$

14. Démontrer qu'on a
$$\widetilde{A}(X) \equiv \widetilde{B}(X)\widetilde{Q}(X) \bmod X^{a-b+1}.$$
15. Déduire de la question précédente et de la partie 1 (questions 8 et 9) un algorithme qui permet de calculer Q en $O((a-b+1)^\alpha)$ opérations arithmétiques dans $\mathbb{K}$.

Partie 3 : Application aux sommes de Newton

Dans cette partie on se donne un polynôme $P \in \mathbb{K}[X]$ de degré $d \geq 1$, et on suppose que P est scindé sur $\mathbb{K}$. On note $\alpha_1, \ldots, \alpha_d$ ses racines, comptées avec multiplicités. Autrement dit, en notant c le coefficient dominant de P, on a

$$P(X) = c \prod_{i=1}^{d} (X - \alpha_i).$$

Pour tout entier $k \in \mathbb{N}$, on appelle *k-ième somme de Newton* de P l'élément

$$S_k(P) = \alpha_1^k + \ldots + \alpha_d^k \in \mathbb{K}.$$

En notant $P = \sum_{j=0}^{d} a_j X^j$, on rappelle que le polynôme dérivé de P, noté P', est défini par la formule $P'(X) = \sum_{j=0}^{d-1}(j+1)a_{j+1}X^j$. De façon analogue aux notations introduites dans la partie 2 on pose

$$\widetilde{P}(X) = X^d P(1/X) \quad \text{et} \quad \widetilde{P'}(X) = X^{d-1}P'(1/X).$$

16. Démontrer que pour tout entier $n \geq 1$ on a

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

17. Déduire de la question précédente et de la partie 1 (questions 8 et 9) un algorithme qui permet de calculer tous les $S_k(P)$, pour k compris entre 0 et d, en $O(d^\alpha)$ opérations arithmétiques dans $\mathbb{K}$.

18. ($\star$) Implémenter l'algorithme de la question précédente, en programmant en `Sage` une fonction qui prend en entrée un polynôme P de degré d, et renvoie la liste des $S_k(P)$ pour k allant de 0 à d. Tester cette fonction avec le premier polynôme utilisé à la question 10, c'est-à-dire avec

$$\mathbb{K} = \mathbb{Z}/97\mathbb{Z}, \quad n = 14, \quad P(X) = 6X^{13} + X^{11} + 43X^8 + 59X^4 + 3X + 14.$$