Université Bordeaux 1
Mathématiques

Préparation à l'agrégation
Année 2010–2011

Algorithme d'Euclide modulaire sur les polynômes

Il est rappelé que le jury n'exige pas une compréhension exhaustive du texte. Vous êtes laissé(e) libre d'organiser votre discussion comme vous l'entendez. Il vous est conseillé de mettre en lumière vos connaissances à partir du fil conducteur constitué par le texte. Le jury demande que la discussion soit accompagnée d'exemples traités sur ordinateur. Il est souhaitable que vous organisiez votre présentation comme si le jury n'avait pas connaissance du texte. Le jury aura néanmoins le texte sous les yeux pendant votre exposé.

1. Introduction

Soient les polynômes

$$f(x) = 824x^5 - 65x^4 - 814x^3 - 741x^2 - 979x - 764$$

$$g(x) = 216x^4 + 663x^3 + 880x^2 - 916x + 617.$$

L'algorithme d'Euclide classique appliqué à f et g, fait apparaître des polynômes aux coefficients relativement grands. Ce fait n'est pas rare, et il y a lieu de se demander dans quelle mesure la taille des coefficients pénalise, en temps et en place, les performances de l'algorithme. Une étude fine montre que l'algorithme se fait tout de même en temps polynomial.

On peut quand même chercher des alternatives qui réduisent, ou même évitent l'explosion des coefficients. On peut tout d'abord chercher à utiliser le fait que le pgcd n'est défini qu'à une unité multiplicative près, et n'utiliser par exemple que des polynômes unitaires. C'est **l'algorithme d'Euclide unitaire**. De fait, avec cette variante, la taille des coefficients est plus petite.

On peut aussi s'arranger pour n'avoir que des polynômes dans $\mathbb{Z}[x]$, et donc faire à chaque étape une pseudo-division, c'est-à-dire multiplier le dividende par une puissance suffisante du coefficient dominant du diviseur. cela donne un algorithme (**algorithme d'Euclide primitif**) au moins aussi performant que l'algorithme unitaire.

Une autre possibilité est la *réduction modulo quelque chose*. Si comme dans l'exemple proposé on travaille dans $\mathbb{Z}[x]$, on peut réduire modulo un nombre premier bien choisi, calculer le pgcd des polynômes réduits dans $\mathbb{F}_p[x]$ et relever le résultat dans $\mathbb{Z}[x]$, si c'est possible.

Nous allons décrire l'algorithme unitaire, puis explorer la piste modulaire.

2. Variante « unitaire » de l'algorithme d'Euclide

Dans cette variante de l'algorithme d'Euclide, on divise à chaque étape le dernier reste par son coefficient dominant.

Une analyse de cet algorithme montre qu'il n'est pas plus coûteux en nombre d'opérations sur le corps de base que les algorithmes classiques.

Si on essaie le nouvel algorithme d'Euclide sur l'exemple de l'introduction, on s'aperçoit que les coefficients qui aparaissent sont moins gros que quand on utilise l'algorithme classique, et c'est ce qui se passe généralement, comme peut l'indiquer l'exécution des deux algorithmes sur plusieurs couples de polynômes choisis de façon aléatoire.

A titre de comparaison avec les algorithmes modulaires, indiquons le coût de cet algorithme unitaire, sur $\mathbb{Q}[x]$.

Théorème 2.1. *Soient f et g deux polynômes de $\mathbb{Z}[x]$, de degré inférieur à n, dont tous les coefficients sont bornés par A en valeur absolue, et soit δ l'écart maximum de degrés entre deux restes successifs dans l'algorithme d'Euclide unitaire appliqué à ces polynômes. Alors l'algorithme est exécuté en au plus $O(n^3 m\delta^2 \log^2(nA))$ opérations sur les mots.*

3. Borne de Mignotte

Si l'on veut retrouver le pgcd de deux polynômes f et g de $\mathbb{Z}[x]$, connaissant le pgcd de leur réduction modulo un nombre premier p, il nous faut bien choisir le nombre premier p. Pour cela, nous allons établir une borne pour les coefficients de ce pgcd.

On définit la norme L_2 d'un polynôme $f = \sum_{i=0}^n f_i x^i$ de $\mathbb{C}[x] : ||f||_2 = (\sum_{i=0}^n |f_i|^2)^{1/2}$. On veut trouver une borne B pour la norme d'un facteur de f en fonction de $||f||_2$, c'est-à-dire, une borne B telle que pour tout facteur h de f, $||h||_2 \leqslant B$.

Lemme 3.1. *Pour $f \in \mathbb{C}[x]$ et $z \in \mathbb{C}$, on a $||(x-z)f||_2 = ||(\overline{z}x - 1)f||_2$.*

Pour $f \neq 0$, écrivons

$$f = \sum_{i=0}^{n} f_i x^i = f_n \prod_{i=1}^{n} (x - z_i).$$

On définit $M(f) = |f_n| \prod_{i=1}^n \max\{1, |z_i|\}$. Alors $M(f) \geqslant |\operatorname{cd}(f)|$ et $M(f) = M(g)M(h)$ si $f = gh$. Le théorème suivant compare $M(f)$ et $||f||_2$.

Théorème 3.2 (Landau)**.** *Pour tout $f \in \mathbb{C}[x]$, $f \neq 0$, on a $M(f) \leqslant ||f||_2$.*

Pour montrer cela, on peut ordonner les racines de f de façon que $|z_1|, \ldots, |z_k| > 1$ et $|z_{k+1}|, \ldots, |z_n| \leqslant 1$. Alors $M(f) = |f_n z_1 \ldots z_k|$. Soit

$$g := f_n \prod_{i=1}^{k} (\overline{z}_i x - 1) \prod_{i=k+1}^{n} (x - z_i) =: \sum_{i=1}^{k} g_i x^i.$$

Alors

$$M(f)^2 = |f_n \overline{z}_1 \ldots z_k|^2 = |g_n|^2 \leqslant ||g||_2^2.$$

Pour finir, par utilisations successives du lemme 3.1, on trouve que $||g||_2 = ||f||_2$.

Pour la suite, on va également utiliser la norme $L_1 : ||f||_1 = \sum_{i=0}^n |f_i|$. Alors

$$||f||_\infty \leqslant ||f||_2 \leqslant ||f||_1 \leqslant (n+1)||f||_\infty.$$

Théorème 3.3. *Soit $h = \sum_{i=0}^{m} h_i x^i \in \mathbb{C}[x]$ de degré m, qui divise $f = \sum_{i=0}^{n} f_i x^i \in \mathbb{C}[x]$ de degré n. Alors*

$$||h||_2 \leqslant ||h||_1 \leqslant 2^m M(h) \leqslant \left|\frac{h_m}{f_n}\right| 2^m ||f||_2.$$

La seconde inégalité vient de la règle de Viète décrivant les coefficients d'un polynôme en fonction de ses racines. La troisième utilise le théorème 3.2.

Corollaire 3.4 (Mignotte)**.** *Soient f, g et h trois polynômes de $\mathbb{Z}[x]$ de degrés respectifs n, m et k, tels que gh divise f (dans $\mathbb{Z}[x]$). Alors*

$$||g||_\infty ||h||_\infty \leqslant ||g||_1 ||h||_1 \leqslant 2^{m+k} ||f||_2 \leqslant (n+1)^{1/2} 2^{m+k} ||f||_\infty. \tag{1}$$

$$||h||_\infty \leqslant ||h||_2 \leqslant 2^k ||f||_2 \leqslant 2^k ||f||_1 \quad \textit{et} \quad ||h||_\infty \leqslant ||h||_2 \leqslant (n+1)^{1/2} 2^k ||f||_\infty. \tag{2}$$

4. Algorithme modulaire : version « grand nombre premier »

On considère deux polynômes primitifs f et g de $\mathbb{Z}[x]$. On va calculer leur pgcd h dans $\mathbb{Z}[x]$, en utilisant un algorithme *modulaire*, c'est-à-dire qu'on va réduire modulo un ou plusieurs nombres premiers.

Pour choisir ces nombres premiers, il vaut mieux savoir borner les coefficients du pgcd h. Soit A un majorant de $||f||_\infty$ et $||g||_\infty$. On suppose que le degré n de f est supérieur ou égal au degré de g. Alors $||h||_\infty \leqslant (n+1)^{1/2} 2^n A$.

On peut procéder de la façon suivante. Soit b le pgcd de $\mathrm{cd}(f)$ et $\mathrm{cd}(g)$. Soit $B = (n+1)^{1/2} 2^n A b$. On choisit un nombre premier p au hasard entre $2B$ et $4B$. Soient $\overline{f}$ et $\overline{g}$ les réductions de f et g modulo p. On utilise l'algorithme d'Euclide pour calculer le polynôme unitaire v de $\mathbb{Z}[x]$ tel que $||v||_\infty \leqslant p/2$ et tel que sa réduction podulo p est le pgcd de $\overline{f}$ et $\overline{g}$. On calcule ensuite w, f^* et g^* dans $\mathbb{Z}[x]$ de norme infinie bornée par $p/2$ tels que

$$w \equiv bv \pmod{p}, \quad f^* w \equiv bf \pmod{p}, \quad g^* w \equiv bg \pmod{p}. \tag{3}$$

Alors, si

$$||f^*||_1 ||w||_1 \leqslant B \quad \text{et} \quad ||g^*||_1 ||w||_1 \leqslant B, \tag{4}$$

le pgcd de f et g est égal à $pp(w)$. Sinon, il faut recommencer avec un autre nombre premier.

La condition (4) est en fait nécessaire et suffisante. En effet, si cette condition est vérifiée, alors $||f^* w||_\infty \leqslant p/2$. Comme $||bf||_\infty \leqslant p/2$ et $f^* w \equiv bf \bmod p$, on en déduit que $f^* w = bf$. De même, $g^* w = bg$. Donc w divise $\mathrm{pgcd}(bf, bg)$. Alors, un raisonnement sur les degrés, et le fait que $\overline{h}$ divise $\mathrm{pgcd}(\overline{f}, \overline{g})$ montrent que $\mathrm{pp}(w) = \mathrm{pgcd}(f, g)$.

Réciproquement, si $\mathrm{pp}(w) = \mathrm{pgcd}(f, g)$, alors w divise bf. D'après la borne de Mignotte, $||bf/w||_\infty \leqslant B \leqslant p/2$. Donc $f^* = bf/w$ et de même, $g^* = bg/w$. Donc, le corollaire 3.4 montre que la condition (4) est vérifiée.

Théorème 4.1. *Le coût pour une exécution de l'algorithme précédent est au plus de* $O(n^2(n^2+\log^2 A))$ *opérations sur les mots.*

Il reste le problème de trouver le nombre premier par lequel on va réduire :

Théorème 4.2. *Soit* $r = \mathrm{Res}(f/h, g/h)$.

(1) *La condition* (4) *est vraie si et seulement si* p *ne divise pas* r.
(2) r *est un entier non nul majoré en valeur absolue par* $(n+1)^n A^{2n}$.
(3) *un premier* p *tiré uniformément au hasard entre* $2B$ *et* $4B$ *vérifie* $p \nmid r$ *avec probabilité au moins* $1/2$.

5. Algorithme modulaire : version « petits nombres premiers »

On peut modifier cet algorithme en réduisant modulo plusieurs petits nombres premiers, et en relevant les résultats par l'utilisation du théorème chinois. On doit alors à chaque pas effectuer plusieurs fois l'algorithme d'Euclide, mais les coefficients qui interviennent alors sont plus petits, ce qui en pratique diminue considérablement le temps d'exécution pour de grandes valeurs des degrés et des normes infinies des polynômes f et g de départ.

Soient deux polynômes primitifs f et g de $\mathbb{Z}[x]$, tels que $\deg(f) = n \geqslant \deg(g)$ et de norme infinie inférieure ou égale à A. On pose $b = \mathrm{pgcd}(\mathrm{cd}(f), \mathrm{cd}(g))$, $k = \lceil 2\log_2((n+1)^n b A^{2n})\rceil$, $B = (n+1)^{1/2}2^n Ab$ et $l = \lceil \log_2(2B+1)\rceil$.

On choisit un ensemble S de $2l$ nombres premiers inférieurs à $2k\log k$, puis de S on retire les nombres premiers qui divisent b : $S \leftarrow \{p \in S \ : p \nmid b\}$.

Pour chaque p dans S, on calcule le polynôme unitaire v_p à coefficients dans $\{0, \ldots, p-1\}$ dont la réduction modulo p est égale au pgcd des réductions $\overline{f}$ et $\overline{g}$ modulo p.

Soit $e = \min\{\deg(v_p : p \in S\}$. On fait : $S \leftarrow \{p \in S : \deg(v_p) = e\}$. Si $\#S \geqslant l$, alors on enlève $|S| - l$ éléments de S ; sinon, on recommence avec un nouveau choix de S.

On calcule, grâce au théorème chinois, les polynômes w, f^* et g^* de $\mathbb{Z}[x]$ de norme infinie inférieure à $(\prod_{p\in S} p)/2$ tels que pour tout $p \in S$

$$w \equiv bv \bmod p \ , \ f^* w \equiv bf \bmod p \ , \ g^* w \equiv bg \bmod p. \tag{5}$$

Alors si

$$||f^*||_1 ||w||_1 \leqslant B \quad \text{et} \quad ||g^*||_1 ||w||_1 \leqslant B, \tag{6}$$

on a $pp(w) = \mathrm{pgcd}(f, g)$. Sinon, il faut recommencer.

Théorème 5.1. *Cet algorithme peut être exécuté en* $\widetilde{O}(n(n^2+\log^2 A))$ *opérations sur les mots.*

Référence :

Modern computer algebra, Joachim von zur Gathen, Jürgen Gerhard.