ALGORITHME D'EUCLIDE SUR LES POLYNÔMES : TAILLE DES COEFFICIENTS

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.

On peut aussi chercher des alternatives qui réduisent, ou même évitent l'explosion des coefficients. On peut 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. Il n'est pas clair à priori que les calculs en deviennent plus simples, mais on peut toujours essayer.

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.

En fait, on va voir qu'avec ces variantes, la taille des coefficients semble plus petite. En plus, on va borner la taille de ces coefficients. Pour cela, il nous faudra voir qu'ils sont intimement liés à des **sous-résultants**, que nous allons définir.

2. Variante "unitaire" de l'algorithme d'Euclide

Dans l'algorithme d'Euclide classique, on divise à chaque étape le dernier reste par son coefficient dominant. Pour l'algorithme d'Euclide étendu, si $r_0^* = f$, $r_1^* = g$, si (r_i^*) est la suite des restes obtenus, on calcule à chaque étape un s_i^* et un t_i^* tels que $s_i^* f + t_i^* g = r_i^*$. Dans notre nouvel **algorithme d'Euclide étendu unitaire**, il faut alors calculer s_i et t_i, pour avoir $s_i^* f + t_i^* g = r_i^*$, où les r_i sont les restes successifs dans cet algorithme.

Plus précisément, si on note $\mathrm{cd}(h)$ le coefficient dominant d'un polynôme h, on note $\rho_0 = \mathrm{cd}(f)$, $r_0 = f/\rho_0$, $s_0 = \rho_0^{-1}$, $t_0 = 0$, $\rho_1 = \mathrm{cd}(g)$, $r_1 = g/\rho_1$, $s_1 = \rho_1^{-1}$, et ensuite, pour $i \geq 1$, q_i est le quotient de la division de r_{i-1} par r_i, $\rho_{i+1} = \mathrm{cd}(r_{i-1} - q_i r_i)$, $r_{i+1} = (r_{i-1} - q_i r_i)/\rho_{i+1}$, $s_{i+1} = (s_{i-1} - q_i s_i)/\rho_{i+1}$, $t_{i+1} = (t_{i-1} - q_i t_i)/\rho_{i+1}$.

Une analyse de ces algorithmes montre qu'il n'est pas plus couteux 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 apparaissent 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.

On peut bien entendu lier les polynômes intervenant dans l'algorithme d'Euclide étendu unitaire à ceux qui apparaissent dans l'algorithme classique.

Théorème 2.1. *Soient* $\alpha_i = \rho_i\rho_{i-2}\dots\rho_2\rho_0$ *si* $i \geq 0$ *est pair et* $\alpha_i = \rho_i\rho_{i-2}\dots\rho_3\rho_1$ *si* $i \geq 1$ *est impair. Alors pour tout* i,

$$q_i^* = \frac{\alpha_{i-1}}{\alpha_i} q_i \ , \ r_i^* = \alpha_i r_i \ , \ s_i^* = \alpha_i s_i \ , \ t_i^* = \alpha_i t_i.$$

3. Sous-résultants

Nous connaissons le lien entre resultant et pgcd. Maintenant, nous allons étudier le lien entre les résultats qui interviennent dans l'algorithme d'Euclide et les sous-résultants.

Soit K un corps, et soient f et g deux polynômes de $K[x]$, de degrés respectifs n et m, avec $n \geq m$. On note $n_i = \deg(r_i)$ la suite des degrés des restes successifs de l'algorithme d'Euclide appliqué à f et g, avec $r_{l+1} = 0$ est le premier reste nul et $\deg(0) = -\infty$.

Théorème 3.1. *Soit* $0 \leq k \leq m$. *Alors* k *n'apparaît pas dans la suite des degrés si et seulement si il existe* s, t *dans* $K[x]$ *tels que* $t \neq 0$, $\deg(s) < m-k$, $\deg(t) < n-k$, $\deg(sf+tg) < k$.

Notons que pour $k = 0$, on retrouve un résultat connu. Pour montrer le sens direct, on utilise le fait que $\deg(s_i) = m - n_i$ et que $\deg(t_i) = n - n_i$. Pour la réciproque, on utilise encore ces égalités, ainsi que le résultat d'unicité suivant.

Lemme 3.2. *On suppose que* $r = sf + tg$, *avec* $\deg(r)+\deg(t) < n$. *Soit* i *compris entre* 0 *et* $l+1$ *tel que* $\deg(r_i) \leq \deg(r) < \deg(r_{i-1})$. *Alors il existe* α *dans* $K[x]$ *tel que* $r = \alpha r_i$, $s = \alpha s_i$ *et* $t = \alpha t_i$.

A présent, traduisons le théorème 3.1 dans le langage de l'algèbre linéaire. Soit $P_d \subset K[x]$ l'espace vectoriel des polynômes de degré strictement inférieur à d. Pour $0 \leq k \leq m$, considérons l'application

$$\begin{aligned} \varphi_k \ : \ P_{m-k} \times P_{n-k} &\longrightarrow P_{n+m-2k} \\ (s,t) &\longmapsto (sf+tg) \text{ quo } x^k \end{aligned}$$

où h quo q désigne le quotient de h par q. Voilà ce que devient alors le théorème 3.1.

Corollaire 3.3. *Soient* $0 \leq k \leq m \leq n$, *et* $1 \leq i \leq l+1$.

- k *apparaît dans la suite des restes si et seulement si* ϕ_k *est un isomorphisme.*

- *Si $k = n_i < n$, alors (s_i, t_i) est l'unique solution de $\varphi_k(s_i, t_i) = 1$.*

Comme pour le résultant, on va écrire la matrice de φ_k. On écrit $f = \sum_{j=0}^{n} f_j x^j$ et $g = \sum_{j=0}^{m} g_j x^j$. Pour $P_{m-k} \times P_{n-k}$, on prend comme base les $(x^i, 0)$, avec $i < m-k$, et les $(0, x^i)$, avec $i < n-k$. Pour P_{n+m-2k}, on prend les x^i, $i < n+m-2k$. On obtient une matrice S_k (l'écrire) telle que si

$$s = \sum_{j=0}^{m-k-1} y_j x^j \ , \ t = \sum_{j=0}^{n-k-1} z_j x^j \ , \ sf + tg = \sum_{j=0}^{n+m-k} u_j x^j,$$

alors

$$S_k(y_{m-k-1}, \ldots, y_0, z_{n-k-1}, \ldots, z_0)^t = (u_{n+m-k-1}, \ldots, u_k)^t.$$

On peut encore écrire le corollaire 3.3 sous la forme suivante.

Corollaire 3.4. *Soient $0 \leq k \leq m \leq n$, et $1 \leq i \leq l+1$.*

- *k apparaît dans la suite des degrés si et seulement si $det(S_k) \neq 0$.*
- *Si $k = n_i < n$, et si $y_0, \ldots, y_{m-k-1}, z_0, \ldots, z_{n-k-1} \in K$ donnent l'unique solution de*

$$S_k(y_{m-k-1}, \ldots, y_0, z_{n-k-1}, \ldots, z_0)^t = (0, \ldots, 0, 1)^t, \tag{1}$$

alors

$$s_i = \sum_{j=0}^{m-k-1} y_j x^j \ \textit{et} \ t_i = \sum_{j=0}^{n-k-1} z_j x^j.$$

Remarquons que S_0 est la matrice de Sylvester de f et g, et que le résultant $\mathrm{res}(f, g)$ est égal à $\det(S_0)$. Les $\sigma_k = \det(S_k)$ sont appelés les **sous-résultants** de f et g.

L'inégalité de Hadamard donne la majoration suivante des sous-résultants, dans le cas où f et g sont des polynômes à coefficients dans $\mathbb{C}$. Alors, si $f = \sum_{i=0}^{n} f_i x^i$, on note $||f||_2 = (\sum_{i=0}^{n} |f_i|^2)^{1/2}$, et $||f||_\infty = \mathrm{Max}\{|f_i| \ : \ i \in \{0, \ldots, n\}\}$.

Théorème 3.5. *Soit $0 \leq k \leq m$. Alors*

$$|\sigma_k| = |det(S_k)| \leq ||f||_2^{m-k} ||g||_2^{n-k} \leq (n+1)^{n-k} ||f||_\infty^{m-k} ||g||_\infty^{n-k}.$$

4. Taille des coefficients

Dans ce paragraphe, on utilise les sous-résultants pour trouver une borne des coefficients dans l'Algorithme d'Euclide étendu unitaire sur $\mathbb{Q}[x]$.

Théorème 4.1. *Soient f et g dans $\mathbb{Z}[x]$ de degrés respectifs n et m, avec $n \geq n$, de norme infinie inférieure ou égale à A, et soit $\delta = max\{n_{i-1} - n_i \ : \ 1 \leq i \leq l\}$ la différence maximale entre les degrés de deux restes consécutifs. Les résultats r_i, s_i, t_i de l'algorithme d'Euclide étendu unitaire pour f et g dans $\mathbb{Q}[x]$ ont des numérateurs et dénominateurs (premiers entre eux) inférieurs en valeur absolue à $B = (n+1)^n A^{n+m}$. Quant aux q_i et ρ_i, ils*

sont bornés en valeur absolue par $C = (2B)^{\delta+2}$. *L'algorithme est exécuté en au plus* $O(n^3 m\delta^2 \log^2(nA))$ *opérations sur les mots.*

Pour montrer cela, on se souvient que s_i et t_i forment l'unique solution du système linéaire (1) avec $k = n_i$. On sait aussi que $\sigma_{n_i} s_i$, $\sigma_{n_i} t_i$ et $\sigma_{n_i} r_i = \sigma_{n_i} s_i f + \sigma_{n_i} t_i g$ sont dans $\mathbb{Z}[x]$. Par l'inégalité de Hadamard et la règle de Cramer, on a les inégalités suivantes.

$$|\sigma_{n_i}| \leq ||f||_2^{m-n_i} ||g||_2^{n-n_i} \leq (n+1)^{n-n_i} A^{n+m-2n_i} \leq B.$$

$$||\sigma_{n_i} s_i||_\infty \leq ||f||_2^{m-n_i-1} ||g||_2^{n-n_i} \leq (n+1)^{n-n_i-1/2} A^{n+m-2n_i-1} \leq B.$$

$$||\sigma_{n_i} t_i||_\infty \leq ||f||_2^{m-n_i} ||g||_2^{n-n_i-1} \leq (n+1)^{n-n_i-1/2} A^{n+m-2n_i-1} \leq B.$$

En utilisant ces inégalités, on a aussi

$$||\sigma_{n_i} r_i||_\infty \leq 2(n+1)^{1/2} B.$$

En fait, on peut obtenir une meilleure borne pour $||\sigma_{n_i} r_i||_\infty$. En effet, on peut remarquer que si U et V sont les matrices obtenues à partir de S_{n_i} en remplaçant la dernière ligne par

$$(x^{m-n_i-1}, \ldots, x, 1, 0, \ldots, 0) \text{ et } (0, \ldots, 0, x^{n-n_i-1}, \ldots, x, 1),$$

alors $\sigma_{n_i} s_i = \det(U)$ et $\sigma_{n_i} t_i = \det(V)$. Donc, si W est la matrice obtenue à partir de S_{n_i} en remplaçant la dernière ligne par

$$(fx^{m-n_i-1}, \ldots, fx, f, gx^{n-n_i-1}, \ldots, gx, g),$$

alors $\sigma_{n_i} r_i = \det(W)$. En utilisant le fait que les coefficients de x^j dans $\sigma_{n_i} r_i$ est obtenu en ne prenant que des termes contenant x^j, on peut démontrer que

$$||\sigma_{n_i} r_i||_\infty \leq B.$$

Ensuite, on considère la division $r_{i-1} = q_i + \rho_i r_i + \rho_{i+1} r_{i+1}$. On pose $k = \deg(q_i) = n_{i-1} - n_i$. Par multiplication, on trouve la pseudo-division dans $\mathbb{Z}[x]$

$$\sigma_{n_i}^{k+1}(\sigma_{n_{i-1}} r_{i-1}) = (\sigma_{n_i}^k \sigma_{n_{i-1}} q_i)(\sigma_{n_i} r_i) + (\sigma_{n_i}^{k+1} \sigma_{n_{i-1}} \rho_{i+1} r_{i+1}).$$

On peut alors trouver les inégalités

$$||\sigma_{n_i}^k \sigma_{n_{i-1}} q_i||_\infty \leq (2B)^{k+1}$$

et

$$||\sigma_{n_i}^{k+1} \sigma_{n_{i-1}} \rho_{i+1} r_{i+1}||_\infty \leq (2B)^{k+2},$$

en utilisant le lemme (4.2) ci-dessous. Ces inégalités suffisent pour prouver les bornes énoncées par le théorème.

Lemme 4.2. *Soient* a *et* b *deux Íléments de* $\mathbb{Z}[x]$, *de degrés respectifs* n *et* m *avec* $n \geq m > 0$. *Soient* $k = n - m$ *et* $c = cd(b) \in \mathbb{Z}$. *Soient de plus* $q = \sum_{i=0}^k q_i x^i$ *et* r *dans* $\mathbb{Z}[x]$ *tels que* $a^* = c^{k+1} a = qb + r$ *et* $deg(r) < m$. *Soit* $r_i \in \mathbb{Z}[x]$ *le reste dans la* $i^{ème}$ *itération de l'algorithme classique de division des polynômes appliqué à* $r_{k+1} = a^*$ *et* b, *pour* $k \geq i \geq 0$. *Alors si*

$||a||_\infty \leq A$, $||a||_\infty \leq A$ et $|c| \leq C$, on a pour tout i : $|q_i| \leq A(B+C)^{k-i}C^i$ et $||r_i||_\infty \leq A(B+C)^{k+1-i}C^i$. En particulier, $||r||_\infty \leq A(B+C)^{k+1}$.

5. Variante "primitive" de l' algorithme d'Euclide

Une autre variante de l'algorithme d'Euclide est **l'algorithme d'Euclide primitif**. Soient f et g deux polynômes primitifs. Alors on peut implémenter l'algorithme d'Euclide en effectuant à chaque pas la *pseudo-division* évoquée dans le lemme (4.2). On trouve alors que la norme infinie des résultats intermédiaires est au plus $(2(n+1)^n A^{n+m})^{\delta+2}$, et que l'algorithme utilise au plus $O(n^3 m\delta^2 \log^2(nA))$ opérations sur les mots.

L'estimation pour l'algorithme d'Euclide primitif est donc la même que pour l'algorithme d'Euclide unitaire. Son avantage est d'éviter les opérations sur les quotients.

Référence :

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