1 Complexité : opérations de base sur les entiers et les polynomes

2 Multiplication rapide

3 DivisionS

3.1 Division rapide

Principe : On se donne $P_1, P_2 \in \mathbb{D}[X]$ de degrés $d_1 \geqslant d_2$ et on cherche $Q, R \in \mathbb{D}[X]$ tels que $P_1 = Q\,P_2 + R$ avec $\deg(R) < \deg(P_2)$ et on suppose P_2 unitaire.

- $X^{d_1} P_1\big(\frac{1}{X}\big) = \big(X^{d_1-d_2} Q\big(\frac{1}{X}\big)\big)\big(X^{d_2} P_2\big(\frac{1}{X}\big)\big) + X^{(d_1-d_2+1)}\big(X^{d_2-1}\, R\big(\frac{1}{X}\big)\big)$
- $\mathrm{Rev}_k(P) = X^k P\big(\frac{1}{X}\big)$ (On note $\mathrm{Rev}(P) = \mathrm{Rev}_{\deg(P)}(P)$)
- $\mathrm{Rev}(P_1) = \mathrm{Rev}(Q)\,\mathrm{Rev}(P_2) + X^{d_1-d_2+1}\mathrm{Rev}_{m-1}(R)$
- $\mathrm{Rev}(P_1) = \mathrm{Rev}(Q)\mathrm{Rev}(P_2) \bmod X^{d_1-d_2+1}$
- $\mathrm{Rev}(P_2)(0) = 1$ donc $\mathrm{Rev}(P_2)$ inversible mod $X^{d_1-d_2+1}$
- $\mathrm{Rev}(Q) = \mathrm{Rev}(P_1)\,(\mathrm{Rev}(P_2))^{-1} \bmod X^{d_1-d_2+1}$
- $Q = \mathrm{Rev}(\mathrm{Rev}(Q))$ puis $R = P_1 - Q\,P_2$

Ainsi, hormis le calcul de $(\mathrm{Rev}(P_2))^{-1}$ mod $X^{d_1-d_2+1}$, le reste des opérations nécessite $\tilde{O}(d)$ opérations arithmétiques dans $\mathbb{D}$ (additions et multiplications de polynômes de degrés d)

Inversion modulaire :

Etant donné $f \in \mathbb{D}[X]$ et $l \in \mathbb{N}$ avec $f(0) = 1$, on cherche $g \in \mathbb{D}[X]$ tel que $f\,g = 1 \bmod X^l$.

Itération de Newton : étant donnée une fonction ϕ dérivable « pas trop méchante » et d'une valeur initiale g_0 « pas trop mauvaise », on approche g tel que $\phi(g) = 0$ par la suite $g_{i+1} = g_i - \frac{\phi(g_i)}{\phi'(g_i)}$.

Dans notre cas, on cherche g tel que $\frac{1}{g} - f = 0$, ce qui revient à considérer la suite $g_{i+1} = g_i - \frac{\frac{1}{g_i} - f}{\frac{-1}{g_i^2}} = 2\, g_i - f\, g_i^2$.

Theorem 1. *$f, g_0, g_1, \ldots \in \mathbb{D}[X]$ avec $f(0) = 1$, $g_0 = 1$ et $g_{i+1} = 2\, g_i - f\, g_i^2 \bmod X^{2^{i+1}}$. Alors $f\, g_i \equiv 1 \bmod X^{2^i}, \forall i \geqslant 0$.*

Proof. Par récurrence. Pour $i = 0$, on a $f\, g_0 \equiv f(0)\, g_0 \equiv 1 \bmod X^{2^0}$
$1 - f\, g_{i+1} \equiv 1 - f\,(2\, g_i - f\, g_{i+1}^2) \equiv (1 - f\, g_i)^2 \equiv 0 \bmod X^{2^{i+1}}$ □

Algorithme InversionModulaire

Input : $f \in \mathbb{D}[X]$ avec $f(0) = 1$, $l \in \mathbb{N}$
Output : $g \in \mathbb{D}[X]$ tq $f\, g = 1 \bmod X^l$
- $g_0 := 1$ et $r := \lceil \log_2(l) \rceil$
- for $i = 1 \ldots r$ do $g_i := (2\, g_{i-1} - f\, g_{i-1}^2) \operatorname{rem} X^{2^i}$

RETURN(g_r)

La complexité de cet algorithme est $\tilde{O}(l)$ opérations dans $\mathbb{D}$ et donc la division par un polynôme unitaire dans $\mathbb{D}[X]$ coute $\tilde{O}(d)$ opérations dans $\mathbb{D}$.

Exercice : Et si on suppose $f(0) = a \in \mathbb{D} \setminus \{0\}$? ou, autrement dit P_2 non unitaire ?

$P_1 = Q\, P_2 + R$ ou encore $P_1 = a\, Q\left(\frac{P_2}{a}\right) + R = Q_a\left(\frac{P_2}{a}\right) + R$.

Il suffit de calculer Q_a en divisant P_1 par $\frac{P_2}{a}$ puis de déduire $Q = \frac{Q_a}{a}$ et enfin $R = P_1 - Q\, P_2$.

En fait on peut directement calculer $a\, Q$ en remplaçant g_i par $a\, g_i$ grace à l'identité $a\, g_{i+1} = 2\,(a\, g_i) - \left(\frac{1}{a} f\right)(a\, g_i)^2$.

On écrit $g_0 = a$ puis $a\, g_1 = 2\, a - \frac{1}{a}\, f\, a^2 = a\,(2 - f)$ et on remarque d'ailleurs que si $f \in \mathbb{D}[X]$, alors $g_1 \in \mathbb{D}[X]$.

Ceci donne d'ailleurs une pseudo-division rapide n'effectuant que des calculs dans $\mathbb{D}$.

4 Algorithmes d'Euclide

On suppose que $\mathbb{D}[X]$ est un anneau Euclidien pour éviter les ennuis et on se donne 2 polynômes $f, g \in \mathbb{D}[X]$.

On forme la suite de triplets (t_l, s_l, r_l) tels que :

$r_0 = f$, $s_0 = 1$ $t_0 = 0$, $r_1 = g$, $s_1 = 0$, $t_1 = 1$ et vérifiant $s_i\, f + t_i\, g = r_i$ avec $d(r_{i+1}) < d(r_i), i \geqslant 1$.

Si elle existe, elle est finie à cause de la condition sur les degrés des r_i et on montre qu'elle se construit (donc existe) grace à l'opération de division :

Traditional Extended Euclidean Algorithm

$r_0 = f,\ s_0 = 1\ t_0 = 0;$
$r_1 = g, s_1 = 0, t_1 = 1:$
i:=1
tant que $r_i \neq 0$
  $q_i, r_{i+1} := \text{Division}(r_{i-1}, r_i)$
  $s_{i+1} := s_{i-1} - q_i\, s_i$
  $t_{i+1} := t_{i-1} - q_i\, t_i$
  i:=i+1
l:=i-1
RETURN$(l, (s_i, t_i, r_i)_{i=0\ldots l+1}, (q_i)_{i=1\ldots l})$

On parle alors de "gcd" pour r_l dans le sens que r_l est un diviseur commun de f et g de degré maximum (tous ces diviseurs sont proportionnels) puisque l'on a $s_l\, f + t_l\, g = r_l$ et qu'il suffit de remarquer que $\gcd(f, g) \sim \gcd(r_i, r_{i+1}) \sim r_l$.

Si $\mathbb{D}$ est un corps, on peut définir "le" pgcd de 2 polynômes en une variable en le supposant unitaire.

Extended Euclidean Algorithm
$\rho_0 = \text{lc}(f),\ \rho_1 = \text{lc}(g)$
$r_0 = \text{normal}(f),\ s_0 = {\rho_0}^{-1}\ t_0 = 0;$
$r_1 = \text{normal}(g), s_1 = 0, t_1 = {\rho_1}^{-1}:$
i:=1
tant que $r_i \neq 0$
  $q_i, r_{i+1} := \text{Division}(r_{i-1}, r_i)$
  $\rho_{i+1} := \text{lc}(r_{i-1} - q_i\, r_i)$
  $s_{i+1} := (s_{i-1} - q_i\, s_i) / \rho_{i+1}$
  $t_{i+1} := (t_{i-1} - q_i\, t_i) / \rho_{i+1}$
  i:=i+1
l:=i-1
RETURN$(l, (\rho_i, s_i, t_i, r_i)_{i=0\ldots l+1}, (q_i)_{i=1\ldots l})$

Exercice - Forme matricielle :
On pose $R_0 = \begin{pmatrix} s_0 & t_0 \\ s_1 & t_1 \end{pmatrix}$ et $Q_i = \begin{pmatrix} 0 & 1 \\ 1 & -q_i \end{pmatrix}$ et $R_i = Q_i \ldots Q_1\, R_0$

Lemma 2.

i. $R_i \begin{pmatrix} f \\ g \end{pmatrix} = \begin{pmatrix} r_i \\ r_{i-1} \end{pmatrix}$

ii. $R_i = \begin{pmatrix} s_i & t_i \\ s_{i+1} & t_{i+1} \end{pmatrix}$

iii. $\gcd(f,g) \sim \gcd(r_i, r_{i+1}) \sim r_l$

iv. $s_i f + t_i g = r_i$

v. $\deg(s_i) = \deg(r_1) - \deg(r_{i-1})\,, \deg(t_i) = \deg(r_0) - \deg(r_{i-1})$

vi. $s_i t_{i+1} - t_i s_{i+1} = (-1)^i$ *(En particulier* s_i *et* t_i *premiers entre eux)*

vii. $\gcd(r_i, t_i) \sim \gcd(f, t_i)$

viii. $f = (-1)^i(t_{i+1} r_i - t_i r_{i+1})\ ,\ g = (-1)^i(s_{i+1} r_i - s_i r_{i+1})$

Exercice - Complexité arithmétique de l'EEA ?

Lemma 3. *Unicité des polynômes de l'EEA*
$f, g, r, s, t \in \mathbb{F}[X]$ *avec* $\deg(f) = n, r = s\,f + t\,g$ *tq* $\deg(r) + \deg(t) < n = \deg(f)$.
On suppose $\deg(r_j) \leqslant \deg(r) < \deg(r_{j-1})$.
Alors, $\exists \alpha \in \mathbb{F}[X]$ *tq* $r = \alpha r_j, s = \alpha s_j, t = \alpha t_j$.

Proof. On pose $\begin{pmatrix} s_j & t_j \\ s & t \end{pmatrix}\begin{pmatrix} f \\ g \end{pmatrix} = \begin{pmatrix} r_j \\ r \end{pmatrix}$. Supposons $s_j\,t \neq s\,t_j$. Alors $f = \frac{\det\begin{pmatrix} r_j & t_j \\ r & t \end{pmatrix}}{\det\begin{pmatrix} s_j & t_j \\ s & t \end{pmatrix}}$.
Mais $\deg(r_j\,t - t_j\,r) \leqslant \max(\deg(r_j) + \deg(t), \deg(r) + \deg(t_j)) \leqslant \max(\deg(r) + \deg(t), \deg(r) + n - \deg(r_{j-1})) < n$, ce qui est impossible.

Ainsi $s_j\ t = s\ t_j$ et comme s_j et t_j premiers entre eux alors t_j divise t, cad $\exists \alpha \in \mathbb{F}[X]$ tq $\alpha t_j = t$ et donc $\alpha s_j\ t = s\ \alpha t_j = s\ \alpha t$, cad $\alpha s_j = s$ et finalement $\alpha r_j = \alpha(s_j f + t_j g) = s\,f + t\,g = r$.

□

4.1 Le "half" gcd

Les remarques :

Euclide → Calcul de $O(n^2)$ scalaires, besoin de $O(n)$ scalaires pour le GCD (ou 1 ligne) : $\begin{pmatrix} r_{j-1} \\ r_j \end{pmatrix} := R_j \begin{pmatrix} r_0 \\ r_1 \end{pmatrix}$.
Les quotients ne dépendent que des premiers termes des polynômes.

Le principe

On définit HGCD(r_0, r_1) : LA matrice M_ν telle que

- $\begin{bmatrix} r_\nu \\ r_{\nu+1} \end{bmatrix} = M_\nu \begin{bmatrix} r_0 \\ r_1 \end{bmatrix}$ avec $\deg(r_\nu) \geqslant \frac{n}{2} > \deg(r_{\nu+1})$

Application au PGCD : calcul de M tq $\begin{bmatrix} \boldsymbol{\gcd(r_0, r_1)} \\ \boldsymbol{0} \end{bmatrix} = M \begin{bmatrix} \boldsymbol{r_0} \\ \boldsymbol{r_1} \end{bmatrix}$

Fonction MatGCD(r_0, r_1)

- $M_\nu = \mathrm{HGCD}(r_0, r_1)$
- $\begin{bmatrix} r_\nu \\ r_{\nu+1} \end{bmatrix} = M_\nu \begin{bmatrix} r_0 \\ r_1 \end{bmatrix}$ (degrés autour de $\frac{n}{2}$)
- IF $r_{\nu+1} = 0$, RETURN(M_ν) ($r_\nu \neq 0$ par définition)
- Division Euclidienne : $\begin{bmatrix} r_{\nu+1} \\ r_{\nu+2} \end{bmatrix} = Q_{\nu+1} \begin{bmatrix} r_\nu \\ r_{\nu+1} \end{bmatrix}$ (degrés $< \frac{n}{2}$ par construction)
- IF $r_{\nu+2} = 0$, RETURN($Q_{\nu+1} M_\nu$) ($r_\nu \neq 0$ par construction)
- $M_\mu = \mathrm{MatGCD}(r_{\nu+1}, r_{\nu+2})$ (polynômes de degrés $< \frac{n}{2}$ par contruction)
- RETURN($M_\mu Q_{\nu+1} M_\nu$)

Complexité : $G(n) \leqslant H(n) + G\left(\frac{n}{2}\right) + \tilde{O}(n)$ et donc $G(n) \in \tilde{O}(H(n) + n)$

Fonction HGCD(r_0, r_1)

- $m := \lceil \frac{n}{2} \rceil$
- IF $\deg(r_1) < m$ RETURN($\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$)
- $q_0 := \mathrm{quo}(r_0, X^m)$; $q_1 := \mathrm{quo}(r_1, X^m)$
- $M = \mathrm{HGCD}(q_0, q_1)$; (restes de degrés autour de $\frac{n}{4}$)
- $\begin{bmatrix} r_\nu \\ r_{\nu+1} \end{bmatrix} = M \begin{bmatrix} r_0 \\ r_1 \end{bmatrix}$ ($r_{\nu+1}$ de degré autour de $\frac{n}{4}$+m $\sim \frac{3n}{4}$)
- IF $\deg(r_{\nu+1}) < m$ RETURN(M)
- Division : $\begin{bmatrix} r_{\nu+1} \\ r_{\nu+2} \end{bmatrix} = Q \begin{bmatrix} r_\nu \\ r_{\nu+1} \end{bmatrix}$ ($r_{\nu+2}$ de degré $\sim \frac{3n}{4}$)
- $l := 2m - \deg(r_{\nu+1})$ ($l \sim \frac{n}{4}$)
- $q_{\nu+1} := \mathrm{quo}(r_{\nu+1}, X^l)$; $q_{\nu+2} := \mathrm{quo}(r_{\nu+2}, X^l)$ (de degrés $\sim \frac{n}{2}$)
- $M'' := \mathrm{HGCD}(q_{\nu+1}, q_{\nu+2})$ (restes de degrés autour de $\frac{n}{4}$)
- $\begin{bmatrix} r_\mu \\ r_{\mu+1} \end{bmatrix} = M'' \begin{bmatrix} r_{\nu+1} \\ r_{\nu+2} \end{bmatrix}$ (de degrés autour de $\frac{n}{4} + l \sim \frac{n}{2}$)
- RETURN($M'' Q M$)

Complexité : $H(n) \leqslant 2\, H\left(\frac{n}{2}\right) + \tilde{O}(n)$ et donc $H(n) \in \tilde{O}(n)$

On note $f \uparrow k = f \bmod X^{n-k}$ et on dit que 2 polynômes $f, f^\star$ coïncident jusqu'à k si $f \uparrow k = f^\star \uparrow k$.

On remarque alors que si $(f,\ g)$ et $(f^\star,\ g^\star)$ coïncident jusqu'à 2 k avec $k \geqslant \deg(f) - \deg(g) \geqslant 0$ et que l'on applique la division avec reste : $f = q\ g + r$, $f = q^\star g^\star + r^\star$ alors $q = q^\star$ et soit (g, r) et $(g^\star, r^\star)$ coïncident jusqu'à $2\,(k - \deg(q))$ ou $r = 0$ ou $k - \deg(q) < \deg(g) - \deg(r)$.

Pour faire les choses proprement, on définit une fonction $\nu\colon \mathbb{N} \to \mathbb{N}$ telle que $\nu(k) = \max\{0 \leqslant j \leqslant l : \sum_{1 \leqslant j \leqslant l} \deg(q_i) \leqslant k\}$ et son équivalente $\nu^\star(k) = \max\{0 \leqslant j \leqslant l : \sum_{1 \leqslant j \leqslant l} \deg(q_i^\star) \leqslant k\}$, les q_i et $q_i^\star$ étant les quotients apparaissant dans l'algorithme d'Euclide.

Ceci donne un lemme technique que l'on prouve par récurrence :

Lemma 4. *Si (r_0, r_1) et $(r_0^\star, r_1^\star)$ coïncident jusqu'à $2k$, alors $\nu(k) = \nu^\star(k)$, $q_i = q_i^\star$ et $\rho_i = \rho_i^\star$ pour $1 \leqslant i \leqslant \nu(k)$.*

Et un algorithme de type "divide and conquer" :

FastEEA

Input : r_0, r_1, k
Output : $\nu(h), M_{\nu(k)} = Q_{\nu(k)} \dots Q_1 = \begin{pmatrix} s_{\nu(k)} & t_{\nu(k)} \\ s_{\nu(k)}+1 & t_{\nu(k)+1} \end{pmatrix}$
If $r_1 = 0$ or $k < n_0 - n_1$, then return 0 and $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
$d := \lfloor k/2 \rfloor$
$j-1, R$:=FastEEA $(r_0 \uparrow 2\,d = r_1 \uparrow (2\,d - (n_0 - n_1))$
$(j - 1 = \nu(d), R = Q_{j-1} \dots Q_1)$

$\begin{pmatrix} r_{j-1} \\ r_j \end{pmatrix} := R \begin{pmatrix} r_0 \\ r_1 \end{pmatrix}$, $\begin{pmatrix} n_{j-1} \\ n_j \end{pmatrix} = \begin{pmatrix} \deg(r_{j-1}) \\ \deg(r_j) \end{pmatrix}$

If $r_{j=0}$ or $k < n_0 - n_j$ then return $j-1$ and R

$q_j := r_{j-1} \operatorname{quo} r_j$; $\rho_{j+1} := \operatorname{lc}(r_{j-1} \operatorname{rem} r_j)$
$r_{j+1:=(r_{j-1} \operatorname{rem} r_j)\rho_{j+1}^{-1}}$, $n_{j+1:=\deg(r_{j+1})}$
$d^\star := k - (n_0 - n_j)$

$h - j, S$:=FastEEA $(r_j \uparrow 2\,d^\star = r_{j+1} \uparrow (2\,d^\star - (n_j - n_{j+1}))$
$(h - j = \nu(d^\star), S = Q_h \dots Q_{j+1})$
$Q_j := \begin{pmatrix} 0 & 1 \\ \rho_{j+1}^{-1} & -q_j \rho_{j+1}^{-1} \end{pmatrix}$

Return h and $S\,Q_j\,R$

5 Resultants - Sous-Resultants

Buts :

- Comprendre les problèmes de spécialisation de l'algorithme d'Euclide
- Avoir des bornes de complexité précises

Une famille de morphismes d'espaces vectoriels sur $\mathbb{D}$ pour f, g fixés dans $\mathbb{D}[X]$ de degrés n, m:

$$\begin{array}{rccc} \varphi_k : & P_{m-k} \times P_{n-k} & \to & P_{n+m-2k} \\ & (s,t) & \to & (s\,f+t\,g) \text{ quo } X^k \end{array}$$

Theorem 5. *$0 \leqslant k \leqslant m < n$. k n'apparait pas dans $(n_i = \deg(r_i))_{i=m\ldots l}$ si et seulement si $\exists (s,t)$ tq $t \neq 0, \deg(s) < m-k, \deg(t) < n-k, \deg(s\,f+t\,g) < k$.*

Proof. $\Rightarrow$ Choisit i, tq $2 \leqslant i \leqslant l+1$ et $n_i < k < n_{i+1}$ alors $s = s_i$ et $t = t_i$ conviennent
et on a bien $\deg(s) = m - n_{i-1} < m-k$, $\deg(t) = n - n_i$ et $\deg(s\,f+t\,g) = \deg(r_i) = n_i < k$.

$\Leftarrow \exists i \in \mathbb{N}, \alpha \in \mathbb{F}[X] \setminus \{0\}$ tq $\alpha s_i\,f + \alpha t_i\,g = \alpha r_i$ et $t = \alpha\,t_i, s = \alpha\,s_i, r = \alpha\,r_i$
Alors $n - n_i \leqslant \deg(\alpha) + n - n_{i-1} = \deg(\alpha\,t_i) = \deg(t) < n-k$
et $n_i \leqslant \deg(\alpha) + n_i = \deg(\alpha r_i) = \deg(r) < k$
et donc $n_{i-1} < k < n_i$ ce qui montre que k n'apparait pas dans la suite des restes. □

Proposition 6. *$0 \leqslant k \leqslant m < n$*

- *k apparait dans $(n_i = \deg(r_i))_{i=m\ldots s} \Leftrightarrow \varphi_k$ est un isomorphisme*
- *Si $k = n_i < n$ alors $(s_i, t_i) =$ unique solution de $\varphi_k(s_i, t_i) = 1$ (restes normalisés)*

Proof.

- k n'apparait pas $\Leftrightarrow \exists (s,\,t) \neq (0,\,0)$, $\varphi_k(s,\,t) = 0 \Leftrightarrow \varphi_k$ pas injective (thm précédent)
- Si $k = n_i$ alors $(s_i,\,t_i)$ est solution et comme φ_k est un isomorphisme, $(s_i,\,t_i) =$ unique solution de $\varphi_k(s_i, t_i) = 1$

□

On Introduit :

$$S_k = \underbrace{\begin{pmatrix} f_n & & & & g_m & & & & \\ f_{n-1} & f_n & & & g_{m-1} & g_m & & & \\ \vdots & & \ddots & & \vdots & & \ddots & & \\ f_{n-m+k+1} & \cdots & \cdots & f_n & g_{k+1} & \cdots & \cdots & g_m & \\ \vdots & & & \vdots & \vdots & & & & \ddots \\ f_{k+1} & \cdots & \cdots & f_m & g_{m-n+k+1} & \cdots & \cdots & \cdots & g_m \\ \vdots & & & \vdots & \vdots & & & & \vdots \\ \vdots & & & \vdots & \vdots & & & & \vdots \\ f_{2k-m+1} & \cdots & \cdots & f_k & g_{2k-n+1} & \cdots & \cdots & \cdots & g_k \end{pmatrix}}_{m-k \qquad\qquad n-k}$$

Corollary 7. *$0 \leqslant k \leqslant m < n$*

- *k apparait dans $(n_i = \deg(r_i))_{i=m\ldots l} \Leftrightarrow \det(S_k) \neq 0$*
- *Si $k = n_i < n$ et si $(y_0, \ldots, y_{m-k-1}, z_{n-k-1}, \ldots z_0)$ est l'unique solution de*

$S_k\,[y_0, ..., y_{m-k-1}, z_{n-k-1}, ...z_0]^T = [0, ..., 0, 1]^T$ *alors :*

- $s_{i=\sum_{0\leqslant j<m-k} y_j X^j}$ *et* $t_{i=\sum_{0\leqslant j<n-k} z_j X^j}$

Definition 8. $\sigma_k = \det(S_k)$ *est le* $k-$ieme *coefficient sous-résultant associé à* f *et* g.

Definition 9. $f, g \in \mathbb{D}[X]$ *ou* $\mathbb{D}$ *est un anneau factoriel.*

$\sigma_0 = \det(S_0)$ *est le Résultant de* f *et* g *et* S_0 *est parfois appelée Matrice de Sylvester associée à* f *et* g.

Theorem 10. $\mathbb{D}$ *anneau factoriel.*

Resultant$(f, g, X) \in \mathbb{D}$ *et est nul si et seulement si :*

- *Soit* $\mathrm{lc}_X(f) = 0$ *et* $\mathrm{lc}_X(g) = 0$
- *Soit* f, g *ont un facteur commun dans* $\mathbb{D}[X]$.

Proof. PGCD non trivial $\Leftrightarrow \exists U, V \in \mathbb{D}[X], U f + V g = 0$ avec $\deg(U) < \deg(g)$ et $\deg(V) < \deg(f)$. □

Remark 11. f, $g \in \mathbb{Q}[X, Y]$, alors Resultant$(f, g, Y) \in \mathbb{Q}[X]$ a pour zéros les projections des points d'intersection des 2 courbes $f = 0$, $g = 0$ ainsi que des asymptotes communes

Si $g = \frac{\partial f}{\partial Y}$ alors Resultant$(f, g, Y) \in \mathbb{Q}[X]$ a pour zéros la projection des points critiques de la projection sur l'axe des X ainsi que celle des points singuliers et des asymptotes de $f = 0$

5.1 Complexité

Theorem 12. *Soient* $f, g \in \mathbb{D}[X]$. *Si* $r_i, s_i, t_i \in \mathbb{F}[X]$ *sont les polynômes apparaissant à la* $i-$eme *étape de l'EEA normalisé, alors* $\sigma_{n_i} r_i, \sigma_{n_i} s_i, \sigma_{n_i} t_i \in \mathbb{D}[X]$.

Proof. Les coefficients de s_i et t_i sont solution de $S_k\, Z = [0, ..., 0, 1]^T$ et s'écrivent donc $\frac{\det(S_k^{(i)})}{\sigma_k}$ ou $S_k^{(i)}$ s'obtient à partir de S_k en remplaçant la i-eme colonne par $[0, ..., 0, 1]^T$ (méthode de Cramer). □

Definition 13. $\sigma_{n_i} r_i = \mathrm{Sres}_{n_i}(f, g, X), n_i = \deg(r_i, X)$

est le n_i-eme *polynôme sous-résultant (ou sous-résultant de degré* n_i *de* f, g.

Si il n'existe pas de reste de degré k, *on pose* $\mathrm{Sres}_k(f, g, X) = 0$.

Theorem 14. *Soient* f, $g \in \mathbb{Z}[X]$ *de degrés au plus* d *avec coefficients de taille binaire au plus* τ. *Alors les coefficients apparaissant dans les polynômes sous résultants sont de taille* $\tilde{O}(d\,\tau)$.

Proof. $|\det(S_k)|, |\det(S_k^{(i)})| \leqslant \|f\|_2^{m-k} \|g\|_2^{n-k} \leqslant (n+1)^{n-k} \|f\|_\infty^{m-k} \|g\|_\infty^{n-k}$ (Borne d'Hadamard) □

Theorem 15. *Soient f, $g \in \mathbb{Z}[X, Y]$ de degrés au plus d avec coefficients de taille binaire au plus τ. Alors les coefficients apparaissant dans les polynômes sous résultants sont de degrés $O(d^2)$ avec des coefficients de taille $\tilde{O}(d\tau)$.*

Remark 16. Les coefficients dans l'algorithme "normalisé" restent petits.

Exercice : Complexité Euclide "normalisé"

5.2 Spécialisation

Theorem 17. *$\phi: \mathbb{D} \to \mathbb{D}'$ morphisme d'anneaux factoriels tel que $\phi(\mathrm{lc}_X(f)) \neq 0$ et $\phi(\mathrm{lc}_X(g)) \neq 0$. Alors $\phi(\mathrm{Sres}_j(f, g, X)) = \mathrm{Sres}_j(\phi(f), \phi(g), X)$*

Proof. Ils "suffit" de composer ϕ et φ_k. □

Applications :

- Calculs avec paramètres puis évaluation
- Calculs par Evaluation / Interpolation

6 Evaluation et interpolation rapides

On considère les 2 problèmes suivants :

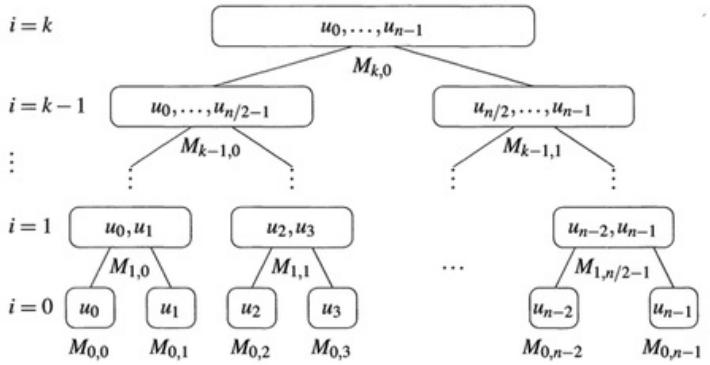
1 - evaluer un polynôme en plusieurs points
2 - étant donné n couples de valeurs $(u_i, f(u_i))$, calculer le polynôme f (de degré strictement inférieur à n)

On forme les polynomes $m_i = x - u_i$, $m = \prod_{i=0}^{n-1} m_i$ et on considère le morphisme $\chi: \begin{array}{ccc} \frac{\mathbb{D}[X]}{\langle m \rangle} & \to & \mathbb{D} \\ f & \longmapsto & (f(u_0), ..., f(u_{n-1})) \end{array}$ qui est un isomorphisme de $\mathbb{D}$-algèbres dès lors que $\mathbb{D}$ est un corps et que les u_i sont distincts (cas particulier du théorème Chinois).

Arbre des Produits

Pose $M_{i,j} = \prod_{l=0}^{2^i-1} m_{j2^i+l}$ pour $0 \leqslant i \leqslant k = \log_2(n)$ et $0 < j \leqslant 2^{k-i}$

- $m = \prod_{i=0}^{n-1} m_i = M_{k,0}$
- $M_{0,j} = m_j$
- $M_{i+1,j} = M_{i,2j} M_{i,2j+1}$



Evaluation Rapide

On pose $m_i = x - u_i$

- $q_0 = f \text{ quo } M_{k-1,0}$ et $q_1 = f \text{ quo } M_{k-1,1}$
- $f(u_i) = \left\{ \begin{array}{l} q_0(u_i)M_{k-1,0}(u_i) + r_0(u_i) = r_0(u_i) \text{ si } 0 \leqslant i < \frac{n}{2} \\ q_1(u_i)M_{k-1,1}(u_i) + r_1(u_i) = r_1(u_i) \text{ si } \frac{n}{2} \leqslant i < n \end{array} \right\}$

Ainsi évaluer f en $u_0, ..., u_{n-1}$ c'est évaluer r_0 en $u_0, ..., u_{\frac{n}{2}-1}$ et r_1 en $u_{\frac{n}{2}}, ..., u_{n-1}$

Interpolation Rapide

(unique polynôme f de degré $<n$ tel que $f(u_i) = v_i$, $i = 0...n-1$)

- $f = \sum_{i=0}^{n} (v_i\, s_i) \frac{m}{(x-u_i)}$ avec $s_i = \prod_{j \neq i} \frac{1}{u_i - u_j}$ (Formule de Lagrange)
- $m'(u_i) = \frac{m}{x-u_i}|_{x=u_i} = \frac{1}{s_i}$

Il s'agit donc d'évaluer m' en $u_0, ..., u_{n-1}$ puis de calculer les $(v_i s_i)$ puis
$\sum_{i=0}^{n} \left(\frac{v_i}{m'(u_i)} \right) \frac{m}{(x-u_i)} =$
$M_{k-1,1} \left(\sum_{i=0}^{\frac{n}{2}-1} \left(\frac{v_i}{m'(u_i)} \right) \frac{M_{k-1,0}}{(x-u_i)} \right) + M_{k-1,0} \left(\sum_{i=\frac{n}{2}}^{n-1} \left(\frac{v_i}{m'(u_i)} \right) \frac{M_{k-1,1}}{(x-u_i)} \right)$

Complexité arithmétique : essentiellement linéaire en n

7 Theoreme Chinois et applications

Soit $\mathbb{D}$ un anneau Euclidien (on ne se servira que de $\mathbb{Z}$ et $\mathbb{Q}[X]$) et $m_0, ..., m_{r-1}$ des éléments de $\mathbb{D}$ premiers. On pose $m = m_0...m_{r-1}$ et on définit les morphismes d'anneaux :

π_i: $\begin{array}{ccc} \mathbb{D} & \to & \frac{\mathbb{D}}{m_i} \\ f & \longmapsto & f \bmod m_i \end{array}$ et plus généralement π: $\begin{array}{ccc} \mathbb{D} & \to & \frac{\mathbb{D}}{\langle m_0 \rangle} \times ... \times \frac{\mathbb{D}}{\langle m_{r-1} \rangle} \\ f & \longmapsto & (\pi_0(f), ..., \pi_{r-1}(f)) \end{array}$

π est surjectif de noyau $\langle m \rangle$ et induit donc un isomorphisme $\frac{\mathbb{D}}{\langle m \rangle} \approx \frac{\mathbb{D}}{\langle m_0 \rangle} \times \ldots \times \frac{\mathbb{D}}{\langle m_{r-1} \rangle}$

Ceci donne donc un algorithme pour projeter modulo chaque m_i puis remonter modulo m.

Image modulaire :

- $f \bmod m_i = \left\{ \begin{array}{l} q_0 M_{k-1,0} + r_0 = r_0 \bmod m_i \ \text{ si } 0 \leqslant i < \frac{n}{2} \\ q_1 M_{k-1,1} + r_1 = r_1 \bmod m_i \ \text{ si } \frac{n}{2} \leqslant i < n \end{array} \right\}$

Remontée Chinoise :

Soit $(\nu_0, \ldots, \nu_{r-1}) \in \frac{\mathbb{D}}{\langle m_0 \rangle} \times \ldots \times \frac{\mathbb{D}}{\langle m_{r-1} \rangle}$ et on cherche $f \in \frac{\mathbb{D}}{\langle m \rangle}$

Considérons $n_i = m / m_i$.

Alors n_i et m_i sont premiers entre eux et il existe donc s_i et t_i dans D tels que $s_i n_i + t_i m_i = 1$.

On pose alors $c_i = \nu_i \, s_i \text{ rem } m_i$ et enfin $\nu = \sum_{i=0}^{r-1} c_i \, n_i$ et on vérifie que $\pi(\nu) = (\nu_0, \ldots, \nu_{r-1})$.

D'une part : $\nu_i s_i n_i = \nu_i - \nu_i t_i m_i = \nu_i \bmod m_i = c_i n_i \bmod m_i$

D'autre part $c_i n_i = 0 \bmod m_j$ pour $j \neq i$

Comme une interpolation :

- calculer les s_i tq $s_i \left(\frac{m}{m_i} \bmod m_i \right) = 1 \bmod m_i$ avec $\deg(s_i) < \deg(m_i)$ avec EEA
- $f = \sum_{i=0}^{n} \left((\nu_i \ s_i) \bmod m_i \right) \frac{m}{m_i} = M_{k-1,0} \left(\sum_{i=0}^{\frac{n}{2}-1} \left((\nu_i \ s_i) \bmod m_i \right) \frac{M_{k-1,1}}{m_i} \right) + M_{k-1,1} \left(\sum_{i=\frac{n}{2}}^{n-1} \left((\nu_i \, s_i) \bmod m_i \right) \frac{M_{k-1,0}}{m_i} \right)$

Application : borne+calcul modulaire+ Algorithme rapide

Ainsi : Résultant, pgcd dans $\mathbb{Z}[X]$ en $\tilde{O}(d \, \tau)$ opérations binaires.