Compléments sur les polynômes

Bernard.Parisse@ujf-grenoble.fr

16 décembre 2002

1 Le résultant

Il s'agit d'un point de vue d'algèbre linéaire sur le PGCD. Considérons deux polynômes A et B de degrés p et q et de pgcd D et l'identité de Bézout correspondante :

$$AU + BV = D \tag{1}$$

avec degré$(U) < q$ et degré$(V) < p$. Imaginons qu'on cherche U et V en oubliant qu'il s'agit d'une identité de Bézout, en considérant simplement qu'il s'agit d'un problème d'algèbre linéaire de $p+q$ équations (obtenues en développant et en identifiant chaque puissance de X de 0 à $p+q-1$) à $p+q$ inconnues (les p coefficients de V et les q coefficients de U) On sait que A et B sont premiers entre eux si et seulement si ce problème d'algèbre linéaire a une solution pour $D = 1$. Donc si le déterminant du système est non nul, alors A et B sont premiers entre eux. Réciproquement si A et B sont premiers entre eux, le système a une solution unique non seulement avec comme second membre 1 mais avec n'importe quel polynôme de degré inférieur $p+q$, donc le déterminant du système est non nul.

Définition :
On appelle résultant de A et B le déterminant de ce système (1). Il s'annule si et seulement si A et B ne sont pas premiers entre eux (ont au moins une racine commune). On appelle matrice de Sylvester la transposée de la matrice du système (les inconnues étant par ordre décroissant les coefficients de U et V)

$$M(A,B) = \begin{pmatrix} a_a & a_{a-1} & \dots & \dots & a_0 & 0 & \dots & 0 \\ 0 & a_a & \dots & \dots & a_1 & a_0 & \dots & 0 \\ \vdots & & & & & & & \vdots \\ 0 & 0 & \dots & & & & & a_0 \\ b_b & b_{b-1} & \dots & b_0 & 0 & 0 & \dots & 0 \\ \vdots & & & & & & & \vdots \\ 0 & 0 & \dots & & & & & b_0 \end{pmatrix}$$

(cette matrice contient b =degré(B) lignes de coefficients du polynôme A et a =degré(A) lignes de coefficients du polynôme B)

Lien avec l'algorithme du sous-résultant (calcul de PGCD)
On peut calculer le déterminant avec la suite des restes de divisions euclidiennes

de la manière suivante, on part de la pseudo-division de A par B :

$$b_b^{a-b+1}A = BQ + R$$

on effectue alors sur chaque ligne contenant les coefficients de A la manipulation de ligne correspondante, c'est-à-dire multiplier la ligne par b_b^{a-b+1} et soustraire (q_0 fois la ligne de B terminant dans la même colonne$+q_1$ fois la ligne de B terminant une colonne avant$+$...). Toutes les lignes contenant les coefficients de A ont été remplacées par des lignes contenant les coefficients de R. Ces lignes contiennent k zéros initiaux avec $k \geq 1$, ce qui permet de réduire le déterminant à celui de la matrice de Sylvester de R et B (à un coefficient multiplicatif près qui vaut b_b^k par rapport au précédent donc $b_b^{k-b(a-b+1)}$ par rapport au déterminant de départ). On échange ensuite R et B ce qui change éventuellement le signe et on continue en faisant les divisions euclidiennes de l'algorithme du sous-résultant (cf. Knuth où on utilise la matrice de Sylvester pour prouver que l'algorithme du sous-résultant est correct). Rappelons que le sous-résultant définit les suites A_k ($A_0 = A, A_1 = B$), d_k le degré de A_k, $\delta_k = d_k - d_{k+1}$, g_k ($g_0 = 1$, si $k \neq 0$, g_k coefficient dominant de A_k) h_k ($h_0 = 1$, $h_{k+1} = h_k^{1-\delta_k} g_{k+1}^{\delta_k}$) et

$$g_k^{\delta_{k-1}+1}A_{k-1} = A_k Q_{k+1} + g_{k-1}h_{k-1}^{\delta_{k-1}}A_{k+1}$$

Théorème 1 *Le résultant est égal au signe près au coefficient h_k où k correspond au reste A_k constant (en supposant que le résultant soit non nul).*

Preuve

La transcription de l'égalité précédente sur les résultants donne par la méthode ci-dessus :

$$\begin{aligned} g_k^{(\delta_{k-1}+1)d_k}\mathrm{Res}(A_{k-1}, A_k) &= g_k^{d_{k-1}-d_{k+1}}\mathrm{Res}(g_{k-1}h_{k-1}^{\delta_{k-1}}A_{k+1}, A_k) \\ &= g_k^{d_{k-1}-d_{k+1}}(g_{k-1}h_{k-1}^{\delta_{k-1}})^{d_k}\mathrm{Res}(A_{k+1}, A_k) \end{aligned}$$

On en déduit que :

$$\frac{\mathrm{Res}(A_{k-1}, A_k)}{g_{k-1}^{d_k}h_{k-1}^{d_{k-1}-1}} = g_k^{d_{k-1}-d_{k+1}-(\delta_{k-1}+1)d_k}h_{k-1}^{\delta_{k-1}d_k+1-d_{k-1}}\mathrm{Res}(A_{k+1}, A_k)$$

On observe que :

$$h_{k-1}^{\delta_{k-1}d_k+1-d_{k-1}} = h_{k-1}^{(\delta_{k-1}-1)(d_k-1)} = \left(h_{k-1}^{\delta_{k-1}-1}\right)^{d_k-1} = \left(\frac{g_k^{\delta_{k-1}}}{h_k}\right)^{d_k-1}$$

donc :

$$\begin{aligned} \frac{\mathrm{Res}(A_{k-1}, A_k)}{g_{k-1}^{d_k}h_{k-1}^{d_{k-1}-1}} &= g_k^{d_{k-1}-d_{k+1}-(\delta_{k-1}+1)d_k}\left(\frac{g_k^{\delta_{k-1}}}{h_k}\right)^{d_k-1}\mathrm{Res}(A_{k+1}, A_k) \\ &= g_k^{d_{k-1}-d_{k+1}-d_k-\delta_{k-1}}\left(\frac{1}{h_k}\right)^{d_k-1}\mathrm{Res}(A_{k+1}, A_k) \\ &= \frac{\mathrm{Res}(A_{k+1}, A_k)}{g_k^{d_{k+1}}h_k^{d_k-1}} \end{aligned}$$

Donc en valeur absolue

$$|\frac{\text{Res}(A_0, A_1)}{g_0^{d_1} h_0^{d_0-1}}| = |\frac{\text{Res}(A_{k-1}, A_k)}{g_{k-1}^{d_k} h_{k-1}^{d_{k-1}-1}}|$$

En prenant le rang k tel que A_k est constant, on a $d_k = 0$ et le résultant est égal à $g_k^{d_{k-1}}$, on obtient donc :

$$|\text{Res}(A_0, A_1)| = |\frac{g_k^{d_{k-1}}}{h_{k-1}^{d_{k-1}-1}}|$$

Comme ici $\delta_{k-1} = d_{k-1}$, le terme de droite est $|h_k|$.

Remarque

On peut calculer au fur et à mesure le signe du résultant en tenant compte des degrés de A_k pour inverser l'ordre de A_{k-1} et A_k dans le résultant.

Utilisation

La valeur du résultant est très utile pour savoir si 2 polynômes dépendant de paramètres sont premiers entre eux en fonction de la valeur des paramètres. En effet, la fonction `gcd` d'un logiciel de calcul formel calculera le PGCD par rapport à toutes les variables en incluant les paramètres. En cherchant quand le résultant s'annule en fonction des paramètres on obtient un autre type d'information.

Exemple :

Chercher quand le polynône $P = x^3 + px + q$ possède une racine multiple en fonction de p et q. On calcule le résultant de P et P' et on trouve $4p^3 + 27q^2$, donc P a une racine multiple si et seulement si $4p^3 + 27q^2 = 0$.

Remarque :

On peut montrer que le résultant de P et P' est divisible par le coefficient dominant de P, on appelle le quotient discriminant.

2 Les suites de Sturm

L'algorithme du sous-résultant appliqué à un polynôme sans racine multiple P et à sa dérivée permet, à condition de changer les signes dans la suite des restes, de connaitre le nombre de racines réelles d'un polynôme dans un intervalle. Ceci est trè utile pour par exemple simplifier des valeurs absolues de polynômes dans un intervalle.

On définit donc la suite de polynômes $A_0 = P, A_1 = P', ..., A_k, 0$ par :

$$A_i = A_{i+1}Q_{i+2} - A_{i+2} \tag{2}$$

avec A_k, le dernier reste non nul, un polynôme constant puisque P n'a pas de racine multiple. On utilise plutot l'algorithme du sous-résultant que l'algorithme d'Euclide, il faut alors s'assurer que les signes de A_i et A_{i+2} sont opposés lorsque A_{i+1} s'annule quitte à changer le signe de A_{i+2} en fonction du signe du coefficient dominant de A_{i+1}, de la parité de la différence des degrés et du signe du coefficient $gh^{1-\delta}$.

On définit $s(a)$ comme étant le nombre de changements de signes de la suite $A_i(a)$ en ignorant les 0. Alors le nombre de racines réelles de $A_0 = P$ sur l'intervalle $]a, b]$ est égal à $s(a) - s(b)$.

Preuve

On considére la suite des signes en un point : elle ne peut contenir deux 0 successifs (sinon toute la suite vaudrait 0 en ce point en appliquant (2), or A_k est constant non nul). Elle ne peut pas non plus contenir +,0,+ ni -,0,- à cause de la convention de signe sur les restes de (2). Donc une racine b de A_i pour $0 < i < k$, n'influe pas sur la valeur de s au voisinage de b (il y a toujours un changement de signe entre les positions $i-1$ et $i+1$). Comme A_k est constant, seules les racines de $A_0 = P$ sont susceptibles de faire varier s. Comme $A_1 = P'$, le sens de variations de A_0 au voisinage d'une racine de A_0 est déterminé par le signe de A_1, donc les possibilités sont -,+ vers +,+ ou +,- vers -,-, ce qui diminue s d'une unité.

3 Exercices sur les polynômes

3.1 Quelques instructions utiles

Les instructions arithmétiques de `maple` sont dans la librairie standard. Celles de `MuPAD` sont dans la librairie standard ou dans la librairie `numlib` (utilisez `?numlib` pour avoir la liste des fonctions de théorie des nombres), on peut lancer la commande `export(numlib);` pour éviter de taper `numlib::` devant les commandes correspondantes.

3.1.1 Entiers

- (MuPAD) `numlib::divisors` / (maple) `numtheory::divisors` : liste des diviseurs d'un entier
- `gcd` : PGCD
- (MuPAD) `numlib::ichrem`/ (maple) `chrem` : restes chinois (entier)
- `igcdex` : Bézout pour des entiers
- (MuPAD) `numlib::genpoly`/ (maple) `genpoly` : crée un polynôme à partir de la représentation z-adique d'un entier (utile pour le PGCD heuristique)
- `iquo` et `irem` quotient et reste de la division euclidienne de deux entiers
- `isprime` test de (pseudo-)primalité
- `modp` : reste euclidien positif,
- `mods` : reste euclidien symétrique
- `nextprime` et `prevprime` (`numlib::prevprime` en mupad) : nombre premier suivant ou précédent

3.1.2 Polynômes

Pour travailler efficacement en MuPAD avec les polynômes, on utilisera `poly`, en précisant le type de coefficients si on travaille dans $\mathbb{Z}/n\mathbb{Z}$, par exemple `poly(x^2+1,[x],IntMod(13))`. Pour travailler dans $\mathbb{Z}/n\mathbb{Z}[X]$ en Maple, on utilise les formes inertes des instructions (qui renvoient l'instruction non évaluée), dont le nom est le même que le nom de commande habituel mais précédé par une majuscule, puis on indique `mod n`, par exemple `Gcd(a,b) mod 11`.

- `coeff` coefficient(s) d'un polynôme,
- (Maple) `coeffs` liste des coefficients d'un polynôme (à développer auparavant, en mupad on utilise `coeff`)
- `content` contenu (pgcd des coefficients)
- `degree` degré
- `divide` division euclidienne,
- `gcd` PGCD,
- `gcdex` Bézout,
- `icontent` : contenu entier pour un polynôme à plusieurs variables
- `indets` : liste des noms de variables d'une expression
- `lcoeff` : coefficient dominant d'un polynôme
- `ldegree` : valuation
- `multcoeffs` multiplie les coefficients d'un polynômes
- (MuPAD) `pdivide` pseudo-division
- (MuPAD) `poly(expr,[var],coeff)` crée un polynôme à partir de l'expression symbolique `expr` par rapport une variable ou à une liste de variables `var`, on peut indiquer dans quel anneau vivent les coefficients (par exemple dans $\mathbb{Z}/13\mathbb{Z}$ avec comme 3ème argument `IntMod(13)`)
- `primpart` : partie primitive d'un polynôme
- `quo`, `rem` (Maple) quotient et reste euclidien
- `tcoeff` : coefficient de plus bas degré d'un polynôme

3.2 Exercices (PGCD et applications)

1. Déterminez la factorisation dans $\mathbb{Z}[X]$ de $P = 72x^6 + 276x^5 - 106x^4 - 217x^3 + 72x^2 + 44x - 16$ en utilisant `gcd` pour déterminer la multiplicité des facteurs.
2. On veut comprendre comment un logiciel de calcul formel calcule

$$\int \frac{x^6+2}{(x^3+1)^2}\, dx$$

On se ramène d'abord à une fraction propre (numérateur N de degré inférieur au dénominateur), Soit $P = X^3 + 1$, calculez le PGCD de P et P', puis deux polynômes U et V tels que :

$$N = UP + VP'$$

On décompose alors l'intégrale en deux morceaux :

$$\int \frac{N}{P^2} = \int \frac{U}{P} + \int V \frac{P'}{P^2}$$

Faites une intégration par parties sur le deuxième terme et en déduire la valeur de l'intégrale du départ.

3. Calculez le pgcd de $x^{202} + x^{101} + 1$ et sa dérivée modulo 3 et modulo 5. Conclusion ?
4. $P = 51x^3 - 35x^2 + 39x - 115$ et $Q = 17x^4 - 23x^3 + 34x^2 + 39x - 115$. Calculez le pgcd de P et Q modulo 5, 7 et 11. En déduire le pgcd de P et Q par le théorème des restes chinois. Pourquoi ne doit-on pas essayer modulo 17 ?
5. Détaillez l'algorithme du PGCD heuristique pour les polynômes $P = (x+1)^7 - (x-1)^6$ et sa dérivée. Comparez avec l'algorithme d'Euclide naïf.
6. Montrer que si $P(\frac{p}{q}) = 0$ avec p et q premiers entre eux, alors p divise le coefficient constant de P et q le coefficient dominant de P. Déterminer les racines rationnelles du polynôme $2X^3 - X^2 - 1262X - 11339$. Écrire un programme mettant en oeuvre cet algorithme, que pensez-vous de son efficacité ?
   Retrouvez les racines rationnelles de $(2*x-999)*(3*x-1000)*(5*x-1001)$ par une méthode modulaire/p-adique (avec $p = 11$)
7. Écrire un programme qui détermine le degré probable du pgcd de 2 polynômes en une variable en utilisant le pgcd modulaire (on considère le degré probable déterminé lorsqu'on trouve deux nombres premiers réalisant le minimum des degrés trouvés)
8. Même question pour un polynôme à plusieurs variables en utilisant cette fois l'évaluation en toutes les variables sauf une
9. Écrire un programme mettant en oeuvre le pgcd heuristique pour des polynômes à une variable.
10. Que se passe-t-il lorsqu'on exécute l'algorithme de Yun dans $\mathbb{Z}/n\mathbb{Z}$?
11. Calculer le pgcd par une méthode modulaire de $(xy - x + 1)(xy + x^2 + 1)$ et $(xy - x - y)(xy - x + 1)$

3.3 Exercices (factorisation) :

- Déterminer les degrés des facteurs de $(x^3 + x + 1)(x^4 + x + 1)$ modulo 5 et 7 (sans utiliser la commande factor). Peut-on en déduire que $x^3 + x + 1$ et $x^4 + x + 1$ sont irréductibles sur $\mathbb{Z}$?
- Factoriser le polynôme $x^5 + x + 1$ en utilisant la méthode de Berlekamp.
- Utiliser les options "verbose" de votre logiciel de calcul formel pour factoriser $x^{202} + x^{101} + 1$ et vérifiez que vous avez compris la méthode utilisée.
- Montrer que $2x + x^2y + x^3 + 2x^4 + y^3 + x^5$ est irréductible sur $\mathbb{Z}$ sans utiliser l'instruction factor (on pourra factoriser pour quelques valeurs de x ou de y)
- Calculer avec un logiciel les valeurs numériques des racines complexes de $P(x) = x^5 + x + 1$. Trouver les combinaisons de racines dont la somme est entière (aux arrondis près). En déduire la factorisation en facteurs irréductibles sur $\mathbb{Z}$ de P.
- Factorisation numérique sur $\mathbb{C}$. Écrire un programme qui calcule une racine d'un polynôme à coefficients complexes en utilisant une méthode itérative

de type méthode de Newton (avec éventuellement un préfacteur lorsqu'on débute la recherche). Les polynômes seront représentés par la liste de leurs coefficients et l'évaluation faite par la méthode de Horner. Trouver ensuite toutes les racines du polynôme en éliminant la racine trouvée (toujours avec Horner). Trouver les combinaisons de racines correspondant à un facteur à coefficients entiers.

– Même question pour les facteurs de degré 2 d'un polynôme à coefficients réels sans racines réelles en utilisant la méthode de Bairstow décrite ci-dessous.

  On cherche un facteur $F = x^2 + sx + p$ de P, on calcule le quotient et le reste de la division $P = FQ + R$ par une méthode de type Horner, il s'agit de rendre R (vu comme un vecteur à 2 composantes) nul. On calcule donc $\partial_{s,p}R$ (en cherchant le quotient et le reste de xQ et Q par F, pourquoi ?) et on pose :
$$(s,p)_{n+1} = (s,p)_n - \lambda(\partial_{s,p}R)^{-1}R(s,p)_n$$
  où λ est un préfacteur compris entre 0 et 1 et ajusté à 1 lorsqu'on est proche du facteur.