RAIRO. INFORMATIQUE THÉORIQUE

GUY VIRY

Factorisation des polynômes à plusieurs variables à coefficients entiers

RAIRO. Informatique théorique, tome 12, n° 4 (1978), p. 305-318

<http://www.numdam.org/item?id=ITA_1978__12_4_305_0>

R.A.I.R.O. Informatique théorique/Theoretical Computer Science
(vol. 12, n° 4, p. 305 à 318)

FACTORISATION DES POLYNÔMES A PLUSIEURS VARIABLES A COEFFICIENTS ENTIERS (*)

par Guy VIRY (1)

Communiqué par J. Berstel

Résumé. — *Soit P un polynôme à plusieurs variables à coefficients entiers. On donne un algorithme de recherche d'un diviseur de P à coefficients entiers, irréductible et de degré* $\leqq d^0 P/2$. *L'algorithme substitue toutes les variables de P sauf une, par des entiers de la forme C,* C^δ, C^{δ^2}, C^{δ^3}, ... *où C majore deux fois les coefficients des diviseurs de P et où* $\delta = E(d^0 P/2) + 1$, *puis le polynôme à une variable ainsi obtenu est factorisé à l'aide de l'algorithme de Berlekamp.*

PRÉSENTATION

Ces dernières années, la « complexité concrète », c'est-à-dire la conception et l'analyse d'algorithmes efficaces, a fait l'objet d'études poussées. Un des principaux domaines d'investigation concerne le traitement des polynômes, notamment dans le cadre des grands systèmes de manipulations formelles comme le SAC de Collins ou le REDUCE de Hearn.

L'article expose une méthode de factorisation des polynômes de $\mathbf{Z}[x_1, \ldots, x_n]$ en un produit $Q_1 \ldots Q_i \ldots$ où Q_i est irréductible sur $\mathbf{Z}[x_1, \ldots, x_n]$. On appelle hauteur $H(Q_i)$ de Q_i le plus grand coefficient de Q_i en valeur absolue. D'après Gel'fond [3] on a

$$H(Q_1) \ldots H(Q_i) \ldots \leqq e^D H(P) \qquad \text{où} \quad D = \partial_1 + \ldots + \partial_j + \ldots, \tag{1}$$

∂_j étant le degré de P suivant x_j. Ainsi tout diviseur Q_i de P vérifie :

$$H(Q_i) \leqq e^D H(P). \tag{2}$$

Comme les degrés $\partial_j^0 Q_i$ suivant x_j des diviseurs Q_i de P sont majorés par ∂_j, une première méthode de factorisation pourrait consister en la recherche de tous les polynômes Q (en nombre fini) tels que $H(Q) \leqq e^D H(P)$ et $\partial_j^0 Q \leqq \partial_j$ et qui divisent P. Mais les calculs seraient longs.

(*) Reçu le 7 février 1978, et dans sa version définitive le 22 mai 1978.
(1) I.U.T. Informatique, Nancy.

La méthode habituelle de factorisation est due à Musser [7] et à Wang et Rothschild [8]; elle suppose que le polynôme donné $P(x_1, \ldots, x_n)$ n'a pas de facteur au carré. Si c'était le cas un tel facteur serait un diviseur commun à P et à $\partial P/\partial x_1$ et on l'obtiendrait en calculant P.G.C.D. $(P, \partial P/\partial x_1)$ qui exige seulement $(d+1)^n\, n \log^2 d$ opérations élémentaires où $d^0\, P = d$, d'après Moenck [5]. La factorisation consiste à substituer à $n-1$ variables, $n-1$ entiers petits, puis à factoriser le polynôme à une variable ainsi obtenu $P(x_1, a_2, \ldots, a_n)$ sous la forme $Q_1^0(x_1) \ldots Q_i^0(x_1) \ldots$. On écrit que

$$P(x_1, x_2, \ldots, x_n) \equiv Q_1^0(x_1) \ldots Q_i^0(x_1) \ldots$$

modulo $y_2 = x_2 - a_2, \ldots, y_n = x_n - a_n$. Une généralisation du lemme de Hensel donnée par Musser et Wang permet d'en déduire

$$R_1(x_1, y_2, \ldots, y_n), \ldots, R_i(x_1, y_2, \ldots, y_n), \ldots$$

tels que

$$P(x_1, x_2, \ldots, x_n) \equiv R_1(x_1, y_2, \ldots) \ldots R_i(x_1, y_2, \ldots) \ldots$$
$$\text{modulo } (y_2)^h, \ldots, (y_n)^h. \tag{3}$$

Lorsque h majore le degré de $x_2, \ldots, x_n$ dans P la relation (3) donne une factorisation de P sur $\mathbf{Z}[x_1, y_2, \ldots, y_n]$ [tout diviseur de P étant un produit de facteurs R_i obtenus dans (3)]. L'ensemble des opérations à effectuer est dominé par le calcul des coefficients de $R_i(x_1, x_2 - a_2, \ldots, x_n - a_n)$ qui exige d'effectuer les produits $x_1^{\alpha_1}(x_2 - a_2)^{\alpha_2} \ldots (x_n - a_n)^{\alpha_n}$ tels que

$$\alpha_1 + \alpha_2 + \ldots + \alpha_n \leqq d.$$

Ce travail demande d'effectuer $(d+1)^{2n}/(2n)!$ opérations pour chacun des diviseurs Q_i.

On propose ici une méthode différente où P peut avoir des facteurs au carré. Remarquons que si on substitue à x_2 un entier $a_2 > 2H(P)$, alors $P(x_1, \ldots, x_n)$ est entièrement défini par $P(x_1, a_2, x_3, \ldots, x_n)$ et si $a_2 > 2e^D(H(P))$ alors les diviseurs Q_i de $P(x_1, \ldots, x_n)$ sont définis par ceux de $P(x_1, a_2, x_3, \ldots)$. Plus généralement si on substitue à $x_2, \ldots, x_n$ des entiers $a_2, \ldots, a_n$ suffisamment grands, alors les diviseurs Q_i de $P(x_1, \ldots, x_n)$ sont définis par ceux de $P(x_1, a_2, \ldots, a_n)$. On recherche plus précisément un diviseur de P de degré $\leqq d^0\, P/2$ et on définit l'application :

$$f : \quad P(x_1, x_2, \ldots, x_n) \to P(x_1, C, C^\delta, \ldots, C^{\delta^{n-2}}),$$

où $\delta = E(d^0\, P/2) + 1$ et où $C/2$ majore $H(Q_i)$.

Il est clair que $f(Q.R) = f(Q).f(R)$ et que f est une injection de $\mathbf{Z}_C[x_1, \ldots, x_n]$ dans $\mathbf{Z}[x_1]$ où $\mathbf{Z}_C[x_1, \ldots, x_n]$ est formé des polynômes à n variables ayant leurs coefficients majorés en valeur absolue par $C/2$ et un degré majoré par d (le fait que f est une injection résulte de l'unicité de l'écriture des entiers m en base C sous la forme

$$m = \sum m_i C^i \qquad \text{où} \quad m_i \in \left]-\frac{C}{2}, \frac{C}{2}\right]\Big).$$

Si Q est un diviseur de P appartenant à $\mathbf{Z}_C[x_1, \ldots, x_n]$ alors Q est l'image réciproque d'un diviseur de $P(x_1, a_2, \ldots, a_n)$. On est ainsi ramené à la factorisation des polynômes à une variable. Rappelons que les méthodes habituelles de factorisation des polynômes à une variable sont basées sur les algorithmes de Berlekamp [2] et de Zassenhaus [9]. La méthode de Berlekamp a été améliorée par Moenck dans [6]. Remarquons qu'ici les coefficients du polynôme $P(x_1, a_2, \ldots)$ qu'on doit factoriser sont grands; admettons qu'ils utilisent δ^{n-1} mots machine (si C peut tenir dans un mot machine). Dans ce cas on peut avantageusement combiner l'algorithme de Moenck à la construction quadratique de Hensel proposée par Zassenhaus dans [9] qui fait passer d'une factorisation $P = Q.R$ modulo N à une factorisation

$P = Q^0.R^0$ modulo N^2.

Dans ces conditions on utilise $(n-1)$ (log δ) fois la construction de Zassenhaus et chaque fois les calculs sont dominés par la multiplication de deux polynômes dont les coefficients ont δ^{n-1} chiffres, ce qui exige des calculs de l'ordre de

$$n\,\delta^n \log^2(\delta) \log(n-1) \qquad \text{où} \quad \delta = E\frac{d^0 P}{2} + 1.$$

Comme dans la méthode de Musser et Wang on utilise un homomorphisme $f: \mathbf{Z}[x_1, \ldots, x_n] \to \mathbf{Z}[x_1]$ tel que $f(Q.R) = f(Q).f(R)$. Mais f peut transformer un polynôme irréductible en un polynôme réductible. On doit donc vérifier si les diviseurs trouvés divisent bien P. Si $f(P)$ admet r diviseurs premiers, le nombre total des diviseurs possibles de P de $d^0 \leqq d/2$ est 2^{r-1}. Les calculs exigés par ces divisions sont majorés par

$2^{r-1}((d+2)/2)^n \log (d+2)$.

Notons que tous les logarithmes utilisés dans l'article sont pris dans la base 2.

1. DÉFINITION D'UNE APPLICATION DE $\mathbf{Z}[x, x_1, \ldots, x_n]$ dans $\mathbf{Z}[x]$

On se donne un polynôme P qu'on veut factoriser. Si P s'écrit $Q_1 \ldots Q_i \ldots Q_r$, alors $d^0 Q_1 + \ldots + d^0 Q_r = d^0 P = d$

et l'un des diviseurs Q_i vérifie $d^0 Q_i \leqq (d^0 P)/r$. Plaçons-nous dans le cas où P n'est pas irréductible et notons Q_1 un diviseur de P tels que $d^0 Q_1 \leqq (d^0 P)/2$. Les variables de P et des autres polynômes qui interviennent seront notées $x, x_1, \ldots, x_n$. On notera ∂_i (resp. ∂_{1i}) le degré de P (resp. de Q_1) suivant x_i; alors $\partial_{1i} \leqq E(d/2)$.

On notera P_i le polynôme $P(x, a_1, \ldots, a_{i-1}, x_i, \ldots, x_n)$ et Q_{1i} le polynôme $Q_1(x, a_1, \ldots, a_{i-1}, x_i, \ldots, x_n)$. On se propose de définir une substitution de $x_1, \ldots, x_n$ par des entiers $a_1, \ldots, a_n$ de façon que pour $i = 1, \ldots, n$, a_i majore deux fois les coefficients de Q_{1i}, soit $a_i \geqq 2H(Q_{1i})$. Si c'est le cas

$$2H(Q_{1i+1}) \leqq 2H(Q_{1i})(1 + a_i + \ldots + (a_i)^{\partial_{1i}})$$

$$\leqq 2H(Q_{1i}) \frac{a_i^{\partial_{1i}+1} - 1}{a_i - 1} \sim 2H(Q_{1i}).a_i^{\partial_{1i}}$$

$$\leqq a_i^{\delta} \qquad \text{où} \quad \delta = E\frac{d}{2} + 1.$$

On peut donc prendre $a_{i+1} = a_i^{\delta}$ et pour a_1 un majorant de $2H(Q_1)$, soit $2e^D H(P)$. En fait on a souvent $H(Q_1)$ très inférieur à $e^D H(P)$: on sait même d'après (1) que pour l'un des diviseurs Q_i de P (qui peut être Q_1) on a $H(Q_i) \leqq e^{D/2}(H(P))^{1/2}$. Pour cette raison on considérera les n-uples $a = (a_1, \ldots, a_n)$ où $a_1 = 2^k (H(P))^{1/2}$ et où $a_i = a_{i-1}^{\delta}$ pour $i \geqq 2$. Alors pour k assez grand, on a bien $a_i \geqq 2H(Q_{1i})$.

On notera $\mathbf{Z}_a[x, x_1, \ldots, x_n]$ le sous-ensemble de $\mathbf{Z}[x, x_1, \ldots, x_n]$ formé des polynômes R tels que pour $i = 1, 2, \ldots, n$ on ait $a_i \geqq 2\,\mathrm{H}(R_i)$.

THÉORÈME : *Soit $a = (a_1, \ldots, a_n)$ la suite définie ci-dessus.*
On considère l'application

$$f_a : \quad Q(x, x_1, \ldots, x_n) \to Q(x, a_1, \ldots, a_n)$$

de $\mathbf{Z}_a[x, x_1, \ldots, x_n]$ dans $\mathbf{Z}[x]$.

L'application f_a est une bijection et de plus si $Q, R, Q.R \in \mathbf{Z}_a[x, x_1, \ldots, x_n]$ alors $f_a(Q.R) = f_a(Q).f_a(R)$.

Démonstration : Il est clair que $f_a(Q.R) = f_a(Q).f_a(R)$.

Vérifions que f_a est injective. Soit

$$Q = \sum_{i,j,\ldots,r\geqq 0} b_{ij\ldots r} x^i x_1^j \ldots x_n^r$$

et

$$R = \sum_{i,j,\ldots,r\geqq 0} c_{ij\ldots r} x^i x_1^j \ldots x_n^r \qquad \text{tels que} \quad f_a(Q) = f_a(R).$$

Alors on a pour tout i :

$$\sum_{j,\ldots,r\geqq 0} b_{ij\ldots r} a_1^j \ldots a_n^r = \sum_{j,\ldots,r\geqq 0} c_{ij\ldots r} a_1^j \ldots a_n^r,$$

soit

$$\sum_{r\geqq 0} B_r a_n^r = \sum_{r\geqq 0} C_r a_n^r,$$

où

$$B_r = \sum_{j,\ldots,s\geqq 0} b_{ij\ldots r} a_1^j \ldots a_{n-1}^s$$

et où

$$C_r = \sum_{j,\ldots,s\geqq 0} c_{ij\ldots r} a_1^j \ldots a_{n-1}^s.$$

Soit r_0 le plus petit entier tel que $B_{r_0} \neq C_{r_0}$. Alors

$B_{r_0} \equiv C_{r_0}$ (modulo a_n).

Or Q, $R \in \mathbf{Z}_a[x, x_1, \ldots, x_n]$, donc

$$B_{r_0}, C_{r_0} \in \left]-\frac{a_n}{2}, \frac{a_n}{2}\right],$$

donc on doit avoir $B_{r_0} = C_{r_0}$.

Par suite $B_r = C_r$ pour tout $r \geqq 0$. En poursuivant on obtient $b_{ij\ldots r} = c_{ij\ldots r}$ pour tout $i, j, \ldots, r \geqq 0$. Donc $Q = R$.

Vérifions que f_a est surjective. Soit Q un polynôme de $\mathbf{Z}[x]$ qui s'écrit $\sum_{j=0} \varepsilon_j b_j x^j$ où $\varepsilon_j = 1$ ou -1 et où $b_j \in \mathbf{N}$. Remarquons que b_j s'écrit de façon unique

$$b_j = \sum_{i_n\geqq 0} b_{ji_n} (a_n)^{i_n} \qquad \text{où} \quad -\frac{a_n}{2} < b_{ji_n} \leqq \frac{a_n}{2}.$$

On dit alors que b_j est exprimé dans la base a_n symétrisée. De même b_{ji_n} s'écrit de façon unique

$$b_{ji_n} = \sum_{i_{n-1}\geqq 0} b_{ji_ni_{n-1}} (a_{n-1})^{i_{n-1}} \qquad \text{où} \quad -\frac{a_{n-1}}{2} < b_{ji_ni_{n-1}} \leqq \frac{a_{n-1}}{2}.$$

Finalement Q s'écrit de façon unique

$$\sum_{j, i_n, i_{n-1}, \ldots \geqq 0} \varepsilon_j b_{j i_n i_{n-1} \ldots i_1} x^j a_n^{i_n} \ldots a_1^{i_1}.$$

Considérons le polynôme

$$D_a(Q) = \sum_{j, i_n, \ldots, i_1 \geqq 0} \varepsilon_j b_{j i_n \ldots i_1} x^j x_n^{i_n} \ldots x_1^{i_1}$$

appelé développement de Q suivant $x_1, \ldots, x_n$ dans la base $a = (a_1, \ldots, a_n)$. Il est clair que

$$D_a(Q) \in \mathbf{Z}_a[x, x_1, \ldots, x_n] \quad \text{et que} \quad f_a(D_a(Q)) = Q.$$

Le théorème est donc vérifié.

2. APPLICATION A LA RECHERCHE D'UN DIVISEUR IRRÉDUCTIBLE DE P

Si $P = Q_1 . Q_2 . \ldots . Q_r$ et si $P, Q_1, \ldots, Q_r \in \mathbf{Z}_a[x, x_1, \ldots]$ alors $f_a(P) = f_a(Q_1) . \ldots . f_a(Q_r)$. Si on connaît une méthode factorisation des polynômes à une variable, on peut factoriser $f_a(P)$ en $Q_1^0 . \ldots . Q_s^0$ où $Q_1^0, \ldots, Q_s^0$ sont irréductibles, et d'après l'unicité de la factorisation sur $\mathbf{Z}[x]$ les diviseurs $f_a(Q_1), \ldots, f_a(Q_r)$ de $f_a(P)$ sont les produits de un ou plusieurs des facteurs $Q_1^0, \ldots, Q_s^0$. Si par exemple $f_a(Q_1) = Q_1^0 . \ldots . Q_i^0$ alors $D_a(Q_1^0 \ldots . Q_i^0)$ est égal à Q_1 et c'est un diviseur de P, à condition toutefois que $Q_1 \in \mathbf{Z}_a[x, x_1, \ldots, x_n]$; il faut en particulier que $2 H(Q_1) \leqq 2^k (H(P))^{1/2}$.

Supposons que :

$$k \geqq E\left(D \log(e) + \frac{\log H(P)}{2} + 1 \right), \tag{3}$$

où $D = \partial_1 + \ldots + \partial_n$. Dans la suite on notera

$$k_0 = E\left(D \log(e) + \frac{\log H(P)}{2} + 1 \right).$$

D'après la majoration (2) donnée page 1, on a $H(Q_1) < e^D H(P)$, donc (3) implique que $H(Q_1) \leqq 2^{k-1} (H(P))^{1/2}$ et on a vu page 5 que

$$2 H(Q_{1i}) \leqq a_{i-1}^{\delta} = a_i;$$

donc $Q_1 \in \mathbf{Z}_a[x, x_1, \ldots, x_n]$.

On considère d'abord la substitution de $x_1, \ldots, x_n$ par $a_1, \ldots, a_n$ avec $k = 0$ (c'est-à-dire $a_1 = H(P)^{1/2}$). Si $f_a(P)$ est irréductible, alors il en est

de même de P sinon pour tous les diviseurs Q^0 de $f_a(P)$ on calcule $D_a(Q^0)$, mais $D_a(Q^0)$ n'est pas nécessairement un diviseur de P. Si c'est le cas les diviseurs de $f_a(P)$ étant pris par ordre de degrés croissants, $D_a(Q^0)$ est un diviseur irréductible de P; sinon si aucun des diviseurs Q^0 de $f_a(P)$ ne correspond à un diviseur $D_a(Q^0)$ de P on remplace k par $k+1$ puis on recommence le processus pour cette nouvelle valeur de k. On ne s'arrête que lorsque k vaut k_0; alors si aucun diviseur de P n'a été obtenu on conclut à l'irréductibilité de P.

Remarquons que pour représenter les coefficients négatifs à l'aide de la base symétrisée a_1, il faut $a_1 \geqq 3$. On impose donc dans l'algorithme que $a_1 \geqq 3$.

Pour la factorisation de $P(x, a_1, \ldots, a_n)$ on utilise l'algorithme DDF de Moenck [6] et on factorise $P(x, a_1, \ldots)$ modulo N, puis on utilise l'algorithme de Zassenhaus [9] pour obtenir la factorisation de

$P(x, a_1, \ldots, a_n)$ modulo N^{2^p}.

On choisit pour N un nombre premier qu'on suppose contenu dans un mot machine. On suppose aussi que a_1 peut être représenté par un mot machine; alors on peut écrire les coefficients de $P_n = P(x, a_1, \ldots, (a_1)^{\delta^n})$ dans la base symétrisée a_1; par suite les diviseurs $Q(x)$ de P_n déterminés par l'algorithme ont aussi leurs coefficients écrits dans la base a_1 symétrisée. Le calcul de $D_a(Q)$ est donc très simple.

Algorithme de recherche d'un diviseur irréductible de P

A. Répéter pour $k = 0, 1, \ldots, k_0$

(1) $a_1 \leftarrow \sup(3, E(2^k (H(P))^{1/2})+1)$; $\delta \leftarrow E(d^0 P/2)+1$;

(2) pour $i = 2, \ldots, n$ faire $a_i \leftarrow (a_{i-1})^{\delta}$;

(3) calculer $P_n = P(x, a_1, a_1^{\delta}, \ldots, a_1^{\delta^{n-1}})$ en exprimant les coefficients dans la base a_1 symétrisée; soit N un nombre premier $\geqq a_1$;

(4) factoriser P_n sur $\mathbf{Z}/(N)[x]$ à l'aide de l'algorithme DDF de Moenck, puis appliquer la construction quadratique de Hensel donnant la factorisation de P_n sur $\mathbf{Z}/(N^{2^p})[x]$ avec $N^{2^p} \geqq 2a_n$;

(5) répéter pour chacun des diviseurs Q de P_n pris par ordre des degrés croissants :

Pour $i = n-1, \ldots, 2, 1$ écrire les coefficients de Q dans la base symétrisée $a_1^{\delta^i}$ en découpant les coefficients par tranches de δ^i chiffres, puis substituer x_i à $a_1^{\delta^i}$ dans Q;

Soit $D(Q)$ le polynôme ainsi obtenu.

Si $D(Q)$ divise P alors c'est un diviseur irréductible de P et l'algorithme est fini.

B. Si aucun diviseur $D(Q)$ de P n'a été obtenu pour $k = 0, 1, \ldots, k_0$, alors P est irréductible

3. ÉTUDE D'UN EXEMPLE

Prenons comme exemple l'un des polynômes factorisés par Wang dans [8] où les calculs sont faits par le système Mac Syma (projet MAC M.I.T.) et prennent 41 076 millisecondes (près d'une minute).

$$P(x, y, z) = x^6 y^2 + x^4(y^3 z + z^4 + 1) + x^3(y^4 + y^2 z^3 + z) + x^2 y(z^5 + z) + x((z^4 + 1) y^2 + z^2 y + z^7 + z^3) + z y^2 + z^4.$$

$H(P) = 1$; calculons

$$P_1 = P(x, y, 3) = x^6 y^2 + x^4(3 y^3 + 82) + x^3(y^4 + 27 y^2 + 3) + 246 y x^2 + (82 y^2 + 9 y + 2\,214) x + 3 y^2 + 81.$$

$\delta = E(d^0 P/2) + 1 = 5$; calculons

$$P_2 = P(x, 3^5, 3) = 59\,049 x^6 + 43\,046\,803 x^4 + 3\,488\,378\,727 x^3 + 59\,778 x^2 + 4\,846\,419 x + 177\,228.$$

Factorisons $P_2 = P(x, 3^5, 3)$ comme dans l'algorithme de Wang relatif à la factorisation des polynômes à une variable [8] :

1° On factorise P_2 modulo 5 :

$$P_2 \equiv 4x^6 + 3x^4 + 2x^3 + 3x^2 + 4x + 3 \quad \text{modulo } 5$$
$$\equiv (x^3 + 4x + 1)(4x^3 + 2x + 3) \quad \text{modulo } 5.$$

2° La construction quadratique de Hensel donne

$$P_2 \equiv 4(x^3 + 4x + 1)(x^3 + 8x + 7) \quad \text{modulo } 5^2$$
$$\equiv 299(x^3 + 104x + 351)(x^3 + 318x - 278) \quad \text{modulo } 5^4$$
$$\equiv 59\,049(x^3 + 729x + 5\,976)(x^3 + 74\,693x - 87\,778) \quad \text{modulo } 5^8.$$

3° On recherche les diviseurs réels de P_2 sur $\mathbf{Z}[x]$; on obtient :

$$P_2 = (x^3 + 729x + 5\,976)(59\,049 x^3 + 82x + 3).$$

En supposant que les calculs aient été faits dans la base 3 les diviseurs trouvés s'écrivent :

$$Q = x^3 + 3^6 x + 3^{10} + 3^3 \quad \text{et} \quad R = 3^{10} x^3 + (3^4 + 1) x + 3.$$

Les diviseurs de P_2 s'écrivent :

$$Q = x^3+3.3^5x+3^{5.2}+3^3 \qquad \text{et} \qquad R = 3^{5.2}x^3+(3^4+1)x+3;$$

leurs polynômes associés $Q_1 = x^3+zyx+z^3+y^2$ et $R_1 = y^2x^3+(z^4+1)x+z$ sont bien des diviseurs de P (on le vérifie en divisant P par Q_1 et R_1).

Les calculs sont dominés par la factorisation de P_2, et plus précisément par la construction quadratique de Hensel qui nécessite de faire des multiplications de très grands nombres,

Traitons le même exemple avec l'algorithme de Wang,
On substitue y à 1 et z à 0, puis on factorise :

$$\begin{aligned} P(x, 1, 0) &= x^6+x^4+x^3+x \\ &= x(x^2+1)(x^3+1). \end{aligned}$$

On calcule un majorant des coefficients des diviseurs de $P(x, 1+y, z)$ $B = \sup(6.\sup(6,1)\,2^{6/2},\ 7.\sup(7,6))\,2^{3/2} = 1\,152$. On fait les calculs qui suivent modulo un nombre majorant $2B$ de la forme q^j qui est ici 7^{22} (pour éviter d'avoir à faire des calculs avec des nombres rationnels).

On note $F_0 = x^3+x$; $G_0 = x^3+1$, puis pour $k = 1, 2, \ldots, 8$ on calcule :
R_k = somme des monômes de degré k de $P(x, 1+y, z)-F_{k-1}.G_{k-1}$;
α_i et β_i tels que $\alpha_i F_0+\beta_i G_0 = x^i$ pour $i = 0, 1, \ldots, 6$;
$F_k = F_{k-1}+R'_k$ où R'_k est obtenu en remplaçant x^i par β_i dans R_k;
$G_k = G_{k-1}+R''_k$ où R''_k est obtenu en remplaçant x^i par α_i dans R_k.

La recherche de α_i et β_i tels que $\alpha_i F_0+\beta_i G_0 = x^i$ se fait par des algorithmes du type de celui donné par Knuth [4] qui se ramène à des divisions de polynômes. On obtient ainsi :

$\alpha_0 = 1\,200(x^2+x+1)$ et $\beta_0 = 1\,201(x^2+x+2)$ modulo 7^4;

$\alpha_1 = 1\,200(x^2+x-1)$ et $\beta_1 = 1\,201(x^2+x)$ modulo 7^4;

$\alpha_2 = 1\,200(x^2-x-1)$ et $\beta_2 = 1\,201(x^2-x)$ modulo 7^4;

$\alpha_3 = \ldots$

$R_1 = 2yx^6+zx^4+4yx^3+zx^3+zx^2+2yx+z;$

$F_1 = x^3+x+1\,201z(x^2+x+2)+2.1\,201y(x^2+x)+1\,201z(x^2-x)+\ldots;$

$G_1 = x^3+1+1\,200z(x^2+x+1)+2\,400y(x^2+x-1)+1\,200z(x^2-x-1)+\ldots;$

$R_2 = P(x, 1+y, z)-F_1.G_1,$

où on ne garde que les monômes de degré 2.

On obtient :

$$R_2 = y^2x^6+3yzx^4+7y^2x^3+yzx^2+y^2x+z^2x+2yz-F_1.G_1.$$

La partie la plus coûteuse de ces calculs est visiblement celle des produits $F_1.G_1$, $F_2.G_2$, ...

On trouve finalement :

$$F_6 = x^3+x+z-2xy-2yz+3y^2x+3y^2z-4y^3x-4y^3z$$
$$+xz^4+5y^4x+5y^4z-2xyz^4+3xy^2z^4-4xy^3z^4+5xy^4z^4,$$
$$G_6 = x^3+xyz+xz+y^2+2y+1+z^3.$$

Les diviseurs réels de $P(x, 1+y, z)$ sont G_6 et

$$(1+2y+y^2)F_6 = x^3(1+2y+y^2)+x+z+xz^4 \quad \text{modulo } y^4, z^7.$$

On trouve les diviseurs de $P(x, y, z)$ en remplaçant y par $1-y$, soit

$$G = x^3+xyz+y^2+z^3 \qquad \text{et} \qquad F = x^3y^2+x+xz^4+z.$$

4. ÉTUDE DE LA LONGUEUR DES CALCULS

A. Calculs de l'algorithme proposé

Rappelons les différentes étapes de l'algorithme proposé :

1° On détermine des nombres $a_1, a_2, \ldots, a_n \in \mathbf{N}$ qu'on substitue aux variables $x_1, \ldots, x_n$. On calcule

$$P_n = \sum_{i \geqq 0} A_i x^i \qquad \text{où} \quad A_i \in \mathbf{Z}[a_1, \ldots, a_n];$$

$$A_i(a_1, \ldots, a_n) = A_i(a_1, a_1^{d_2}, \ldots, a_1^{d_n})$$

a au plus $(d+1)^n$ monômes où $d = d^0P \geqq d^0A_i$.
Les calculs sont donc majorés par $d(d+1)^n$ *additions.*

2° On factorise $P(x, a_1, \ldots, a_n) = P_n$ sur $\mathbf{Z}[x]$. On doit d'abord déterminer les facteurs de P_n dont le carré divise P_n. Pour cela on calcule le P.G.C.D. de P_n et de dP_n/dx. Ces calculs exigent d'après Moenck [5] : $(d+1)S\log^2 S$ opérations où S est le nombre de chiffres des coefficients. Or d'après le 3° ci-dessous, $S \sim \delta^n$ où $\delta = E(d^0P/2)+1$; donc les calculs sont de l'ordre de $\delta^{n+1}n^2\log^2\delta$.

Supposons à présent que P_n n'ait plus de facteurs au carré. On factorise P_n modulo un nombre premier N à l'aide de l'algorithme de Moenck [6] qui exige seulement $\partial^3+\partial^2\log N$ opérations, ∂ étant le degré de P suivant x.

3° On utilise l'algorithme de Zassenhaus pour obtenir la factorisation modulo N^{2^p} où N^{2^p} majore deux fois les coefficients des diviseurs de P_n.

Évaluons p en se plaçant dans le cas le plus défavorable où P est irréductible; alors l'algorithme factorise P_n pour $k = 0, 1, \ldots, k_0$, soit $k_0 + 1$ fois. Mais en fait les calculs pour les premières valeurs de k sont négligeables par rapport à ceux de la dernière factorisation. Alors

$a_1 = 2\,e^D\,H(P), \ldots, a_i = (a_{i-1})^\delta = (a_1)^{\delta^{(i-1)}}, \ldots, a_n = (a_1)^{\delta^{n-1}}$.

D'après la définition de $a_1, \ldots, a_n$ donnée page 4, $(a_1)^{\delta^n}$ majore deux fois les coefficients de $Q_{1n} = Q_1(x, a_1, \ldots, a_n)$.

La factorisation de P_n modulo N a été faite au 2° avec $N = 2\,e^D\,H(P)$ et l'algorithme de Zassenhaus fait passer de cette factorisation à la factorisation modulo N^{2^p} tel que $N^{2^p} \geqq (a_1)^{\delta^n}$. Il suffit donc que $2^p \geqq \delta^n$, soit que $p \geqq n \log \delta$. Le nombre d'étapes de l'algorithme de Zassenhaus est donc majoré par $p = n \log \delta$. Chaque étape a des calculs dominés par le produit de polynômes de degré d. Le produit de deux polynômes à une variable peut se réduire à effectuer $d \log d$ multiplications de coefficients en utilisant la transformation de Fourier, ce qui donne $d \log d.\, S \log S \log \log S$ multiplications élémentaires, où S est le nombre de chiffre des coefficients (d'après [1], p. 273). Si on écrit les coefficients dans la base N [où $N \sim a_1 \leqq 2\,e^D\,H(P)$ peut être écrit dans un mot machine], alors pour la dernière étape de l'algorithme de Zassenhaus on a $S \leqq \delta^n$ et les calculs son majorés par

$d \log d.\, \delta^n.n.\log \delta\,(\log n + \log \log \delta)$

donc par $n\,(\delta^{n+1} \log^2 \delta.\sup(\log n, \log \log \delta)$.

Au cours des étapes précédentes on a $S \leqq (\delta)^{n-i}$. La complexité des différentes étapes suit grossièrement une loi géométrique, donc la complexité de la totalité de l'algorithme de Zassenhaus est du même ordre que celle de la dernière étape. Rappelons que l'algorithme de Zassenhaus doit être appliqué à chacun des r diviseurs premiers modulo N de P_n. L'algorithme exige donc de faire au total $rn\,((\delta)^{n+1} \log n \log^2(\delta)$ opérations où $r \leqq d$.

4° Chaque diviseur $Q_1, \ldots, Q_r$ de P_n est développé suivant $x_1, \ldots, x_n$ dans la base $(a_1, a_1^{d_2}, \ldots, a_1^{d_n})$. Les calculs de factorisation de P_n sont supposés faits dans la base a_1, donc ce développement des coefficients des diviseurs s'obtient en séparant les coefficients par tranches de d_n chiffres, la i-ième tranche de d_n chiffres correspondant au coefficient de x_n^i. Chacune de ces tranches est elle-même séparée en tranches de d_{n-1} chiffres et ainsi de suite. L'exécution de ce travail est de l'ordre de grandeur du nombre des coefficients qu'on a à calculer, soit $(\delta)^n$ opérations,

5° On vérifie que les diviseurs présumés obtenus ci-dessus divisent bien P. Si P_n a r diviseurs premiers, on doit faire 2^{r-1} divisions, ce qui exige $2^{r-1}\,(\delta)^{n+1} \log(\delta)$ opérations d'après Aho ([8], p. 314, exercices 8.22 et 8.23).

B. Calculs de l'algorithme de Wang

On détermine les nombres $a_1, \ldots, a_n \in \mathbf{Z}$ qu'on substitue aux variables $x_1, \ldots, x_n$. Puis on étudie le polynôme $P^1 = P(x, a_1+x_1, \ldots, a_n+x_n)$, Il faut choisir $a_1, \ldots, a_n$ tels que $d^0 P^1 = d^0 P = d$ et que

$$P_n = P(x, a_1, \ldots, a_n)$$

n'ait pas de diviseur au carré.

Il arrive qu'on puisse prendre plusieurs valeurs a_i nulles ce qui simplifie beaucoup les calculs.

Mais en général, il n'est pas possible de choisir certaines valeurs a_i nulles et le calcul des monômes de $P^1 = P(x, a_1+x_1, \ldots, a_n+x_n)$ exige d'effectuer les produits $(a_1+x_1)^{k_1} \ldots (a_n+x_n)^{k_n}$ où $k_1+\ldots+k_n = d$, ce qui donne $(1+k_1) \ldots (1+k_n)$ monômes.

On obtient ainsi un polynôme de $(d+1)^{n+1}$ monômes.

Les principales étapes de l'algorithme de Wang sont les suivantes :

1° On factorise $P(x, a_1, \ldots, a_n)$ à l'aide d'une méthode du type Berlekamp, suivie de l'algorithme de Zassenhaus. Il me paraît plus intéressant d'utiliser l'algorithme de Moenck; on a seulement $\partial^3+\partial^2 \log N$ opérations.

2° On détermine les diviseurs de P^1 modulo $x_1^{d_1}, \ldots, x_n^{d_n}$ et modulo un entier B majorant deux fois les coefficients des diviseurs de P^1. On utilise une généralisation de l'algorithme de Zassenhaus.

Au départ :

$$P^1 \equiv P(x, a_1, \ldots, a_n) = F_0 . G_0 \qquad \text{modulo } x_1, \ldots, x_n, 2B$$

$$R_1 \equiv P^1 - F_0 . G_0.$$

Pour $k = 1, 2, \ldots, d$ on calcule

R_k = monômes de degré k de $P^1 - F_{k-1} . G_{k-1}$;

α_i et β_i tels que $\alpha_i G_0 + \beta_i F_0 = x^i$ pour $i = 0, 1, \ldots, d$;

$F_k = F_{k-1} + R'_k$ et $G_k = G_{k-1} + R''_k$ (R'_k et R''_k étant obtenus en remplaçant x^i par α_i ou β_i dans R_k).

Les polynômes F_k et G_k comportent beaucoup de termes, notamment à la dernière étape : ils sont de degré d par rapport à chaque variable. Ils sont donc formés de $(d+1)^{n+1}$ monômes. Le calcul de $F_k . G_k$ exige donc de faire $(n+1)(d+1)^{n+1} \log (d+1)$ opérations élémentaires en utilisant la transformation de Fourier. Soit au total $r(n+1)(d+1)^{n+1} \log (d+1)$ opérations pour les r diviseurs de P_n. Ce calcul est à peu près du même ordre que celui du produit de polynômes à une variable à grands coefficients utilisé dans l'algorithme présenté ci-dessus.

3° Pour chacun des r diviseurs premiers Q^1 de P^1 obtenu au 2°, on calcule $Q^1(x, x_1-a_1, \ldots, x_n-a_n)$ où $d^0 Q^1 \leqq d$. Chaque monôme $x^k x_1^{k_1} \ldots x_n^{k_n}$ de Q^1 est transformé en $x^k (x_1-a_1)^{k_1} \ldots (x_n-a_n)^{k_n}$, et pour chacun d'eux on doit effectuer $(1+k_1) \ldots (1+k_n)$ multiplications où $k_1+\ldots+k_n \leqq d$. On a donc au total

$$\Pi = \sum_{\substack{k_1, \ldots, k_n=0 \\ k_1+\ldots+k_n \leqq d}} (1+k_1) \ldots (1+k_n)$$

multiplications à effectuer.

On peut montrer que

$$\Pi = \sum_{0 \leqq k_1 \leqq d} (1+k_1) \sum_{k_2=0}^{d-k_1} (1+k_2) \ldots \sum_{k_n=0}^{d-k_1-\ldots-k_{n-1}} (1+k_n)$$
$$= \frac{(d+1)^{2n}}{(2n)!}.$$

Comme il y a r diviseurs de P_0 ces calculs sont de l'ordre de $r(d+1)^{2n}/(2n)!$

4° On calcule les diviseurs réels de P et notamment les coefficients ($\in \mathbf{Z}[x_1, \ldots, x_n]$) du monôme de degré maximum en x, les diviseurs obtenus au 2° étant unitaires. Les calculs sont un peu plus longs, mais du même ordre que ceux de l'algorithme proposé, soit $2^{r-1}((d+2)/2)^n \log(d+2)$.

C. CONCLUSION

La méthode de Wang et la méthode proposée ici ont des calculs du même ordre, au départ pour factoriser $P(x, a_1, \ldots)$ modulo N, et à la fin pour tester si les diviseurs obtenus conviennent. La différence essentielle est dans le fait que la méthode de Wang exige un changement de variable $x_i \to a_i+x_i$ dont l'exécution donne des calculs importants.

Remarquons que si le polynôme donné P a de petits coefficients et un degré faible et si les coefficients de $P(x, a_1, \ldots, a_n)$ sont contenus dans un mot machine, alors la méthode proposée est très rapide : ses calculs sont de l'ordre de grandeur de la factorisation d'un polynôme à une variable.

BIBLIOGRAPHIE

1. A. V. AHO, J. E. HOPCROFT and J. D. ULLMANN, *The Design and Analysis of Computer Algorithms*, Addison-Wesly, Reading, Massachusetts, 1974.
2. E. R. BERLEKAMP, *Algebraic Coding Theory*, MacGraw-Hill, New York, 1968, p. 146-150.
3. A. O. GEL'FOND, *Transcendental and Algebraic Numbers*, GITTL, Moscow, 1952, trad. anglaise, Dover, New York, 1960, p. 135 et 138.

4. D. E. KNUTH, *The Art of Computer Programming*, vol. II : *Seminumerical* algorithms, Addison-Wesley, Reading, Massachusetts, 1969, p. 302.
5. R. T. MOENCK, *Fast Computation of GCD's; ACM Symposium of Computing*, 30 avril-2 mai 1973, p. 142-151.
6. R. T. MOENCK, *On the efficiency of algorithms for Polynomial factoring*, Math. of Comp., vol. 31, n° 137, 1977, p. 235-250.
7. D. R. MUSSER, *Multivariate Polynomial Factorisation*, J. Ass. Comp. Mach., vol. 22, n° 2, 1975, p. 291-308.
8. P. S. WANG and L. P. ROTHSCHILD, *Factoring Multivariate Polynomials over the Integers*, Math. of Comp., vol. 29, n° 131, 1975, p. 935-950.
9. H. ZASSENHAUS, *On Hensel factorisation*, J. Number Theory, vol. 1, 1969, p. 291-311.