

modular 算法による多項式 GCD 計算について

K. Yokoyama, M. Nara, T. Takeshima
富士通(株)国際研 横山和弘 野呂正行 竹島卓

多項式の GCD 計算の効率的アルゴリズムの研究は数多く行われており、特に 1 変数多項式の場合には Euclid の互除法を改良した PRS 方式によるアルゴリズムが示され、高速な 1 変数多項式の GCD 計算が既存の数式処理システムの上に実現されている。一方、多変数多項式の場合には、PRS 方式以外に modular 算法を適用した EZ 法 (Extended Zassenhaus 法, J. Moses, D. Y. Y. Yun による) や、Groebner 基底を利用した方法が開発された。本論文では modular 算法について考察する。

EZ 法では「偽の GCD 因子」の発生問題があるが、これに対して (1) EZ 法の一部に格子算法を適用し、「偽の GCD 因子」からでも真の GCD 因子を作り出すこと、(2) EZ 法において選択すべき素数 p と評価ベクトル a として、「偽の GCD 因子」の発生しないものを選ぶ際の検索件数およびそのような p, a の実例について報告する。

簡単のため、monic な (従って原始的な) 2 変数多項式 $F(x, y)$ と $G(x, y)$ の GCD 計算における EZ 法について議論する。

(1) EZ 法

ここでは簡単のために、EZ 法が無平方多項式に適用される場合を考察する。無平方でない多項式の無平方化にも EZ 法は適用できることを断っておく。(主変数を x とする。)

①ある整数 a を取り、

$$\text{degree}_x F(x, y) = \text{degree}_x F(x, a) \text{ かつ}$$

$$\text{degree}_x G(x, y) = \text{degree}_x G(x, a) \text{ かつ}$$

$$F(x, a) \neq G(x, a)$$

となるようにする。(この a を代入することを evaluation と呼ぶ)。

② $D(x) = \text{GCD}(F(x, a), G(x, a))$ を計算する。

$H(x) = D(x)$ の F (または G) に関する余因子) と置く。

ここで、 H と D が互いに素になるように F か G を選ぶ。できないときには a を取り直す。

③素数 p を $\text{mod } p$ でも H と G とが互いに素であるように選び、多変数多項式の因子の係数の上限の評価 (Gel'fond 等) を越えるように $q = p^e$ をとる。さらに、 $q, S = \langle y - a \rangle$ に関し

て、Hensel構成を用いて D と H とを S^k まで持ち上げる。ここで、

$$k = \max \{ \text{degree}_y \text{lc}(F) + \text{degree}_{x,y} F + 1, \\ \text{degree}_y \text{lc}(G) + \text{degree}_{x,y} G + 1 \} .$$

④持ち上げたGCDの候補 $D(x,y)$ についてそれが真に F および G の因子であるかどうかを判定する (trial division). 因子ならば、 D と H とが互いに素であることより、 D が求めるGCDである。そうでないときには、①に戻り a を取り直して②, ③, ④を再度試みる。

このEZ法ではevaluation value a と素数 p (特に a) の取り方によっては、偽の因子の発生のために何度か全体のループを繰り返さねばならないことになる。以下(2), (3) では格子算法によって偽の因子からでも真の因子を作り出す方式を示し、(4) では無効 (invalid) な a がどの位存在するか、また有効 (valid) な a はどう決めればよいか考察する。

(2) 有限体上の因数分解+格子算法

①素数 p および整数 a をつぎのように選ぶ。

$$\text{degree}_x F(x,y) = \text{degree}_x F(x,a) \text{ かつ}$$

$$\text{degree}_x G(x,y) = \text{degree}_x G(x,a) \text{ かつ}$$

$$F(x,a) \neq G(x,a) \pmod{p} \quad \text{かつ}$$

$$\text{resultant}_x (F, F')(a) \neq 0 \pmod{p} \quad \text{かつ}$$

$$\text{resultant}_x (G, G')(a) \neq 0 \pmod{p} .$$

②イデアルとして $S = \langle p, y-a \rangle$ を考える。 $\text{mod } S$ において F と G とのGCDを求める。

$$\text{すなわち } h^*(x) = \text{GCD}(F(x,a), G(x,a)) \pmod{p} .$$

③ Berlekampのアルゴリズムにより $h^*(x)$ を因数分解し、既約因子 $h^*_1, h^*_2, \dots, h^*_t$ を取り出す。

④ S 上の各既約因子 h^*_i に対し、多項式の組 (F, h^*_i) および (G, h^*_i) への格子算法が存在する。この算法により得られた因子を $H_{F,i}$ および $H_{G,i}$ とおくと、それらが同一の多項式ならばそれらはGCDの(既約)因子である。さもなければGCDの因子ではない。またGCDの因子はこれらのもので尽くされる。

この方法は F, G のGCDだけでなく、その既約分解を求める必要がある場合には (EZ GCD+多変数の因数分解よりも) 有効と考えられる。

(3) 余因子+拡張格子算法

① (2)の①と同様に p , a を求めておく. そして(2)の②と同様, $\text{mod } S$ において F と G とのGCDをもとめそれを $h^*(x)$ とおく.

② $b(x)=F(x,a)/h^*(x)$, (F の余因子)

$c(x)=G(x,a)/h^*(x)$ (G の余因子)とおく.

③多項式の組 (F,b) および (G,c) への拡張格子算法が存在し, $F(x,y)$ の因子である $B(x,y)$ と $G(x,y)$ の因子である $C(x,y)$ とが求まる. (拡張というのは, b や c が $\text{mod } p^k$ で既約でないことによる. KaltofenやLenstraのオリジナル版は既約因子に適用するものである.)

④ $\text{new } F=F(x,y)/B(x,y)$,

$\text{new } G=G(x,y)/C(x,y)$ とおく. このとき,

if $\text{new } G \mid \text{new } F$ then $\text{new } G$ is the GCD;

if $\text{new } F \mid \text{new } G$ then $\text{new } F$ is the GCD;

otherwise try to find $\text{GCD}(\text{new } F, \text{new } G)$

もともと a がvalidなevaluationであったならば, ④でGCDがでる. そうでない場合でも格子算法の御陰でGCDの因子がでる. このため a を取り直しての再試行の際の問題は小さくなっている. (cf. EZ法の場合は問題のサイズは変わらない.)

この方法はさらに改良できる. すなわち, ③において格子算法を適用する前にtrial divisionを行うことにすれば, a がvalidなevaluationであった場合にはその時点でGCDが求まる.

(4) valid evaluation の探索について

W.S.Brownが1971年の2nd Symposium on Symbolic and Algebraic Manipulationにおいて考察を行っているが, それは計算機の1語に収まるmodulus p を先ずとって, その後invalidなevaluation a がどの位あるかを調べたものである. 我々はより厳密な評価を目指している.

①「 a がinvalidである」必要十分条件は「 a が F と G とから定まる多項式 $B(y)$ の零点になっている」ことである. ここに, $B(y)=A^*(y,0)$ であり, $A^*(y,t)$ は t を因子として持たず, 次式で定義される. $A(y,t)=t^e A^*(y,t)=\text{resultant}_x(F(x,y), G(x,y+t))$.

②これから, a の探索件数の上界は $\deg_y B(y)$ の限界として, $(M_F N_G + N_F M_G)$ と見積もら

れる。ここに、 $M_F = \text{degree}_y F$, $N_G = \text{degree}_x G$, $N_F = \text{degree}_x F$, $M_G \geq \text{degree}_y G$ である。これは Brown の評価とほぼ同じ結果である。

③また、 F, G から定まるある bound A を越える任意の整数 a は valid である。

A は $A(y)$ の零点の限界を $B(y)$ の係数の絶対値の最大値として見積もると、 $A = (N_F + N_G)! \cdot M_G^{N_F} M_F^{N_G} C_F^{N_G} C_G^{N_F}$ となる。ここに $C_F = F$ の数係数の絶対値の最大値、 $C_G = G$ の数係数の絶対値の最大値、である。

④さらに、 $a > \max \{A, 2 \times \text{Gel'fond's bound}\}$ ならば、つぎのような対応で Hensel 構成を経なくとも GCD が求まる。Kronecker's trick が変数を主変数の指数に還元して GCD を計算させるのに対して、この方法は変数を係数に還元して GCD を計算する方法となっている。 $\text{GCD}(F(x, a), G(x, a)) = \sum_i \{g_i x^i\}$ とし、係数 g_i をさらに

$$g_i = \sum_j \{g_{i,j} a^j\}, \quad |g_{i,j}| < 1/2 a \text{ と一意に分解すれば,}$$

$$\text{GCD}(F(x, y), G(x, y)) = \sum_{i,j} \{g_{i,j} x^i y^j\} \text{ と求まる.}$$

⑤ a に対して valid な p は、 $p \nmid B(a)$ となるものである。いま③のように A をとり、 L を $B(y)$ の y についての次数とすると、 $|B(a)| < A^{L+2}$ (ただし、 $A \geq 2$ とする) であるので、積が A^{L+1} をこえる相異なる k 個の素数 p_1, p_2, \dots, p_k の少なくともひとつは $B(a)$ を割らないといえる。したがって素数 p の invalid になる個数は高々 $(L+2) \log A$, すなわち、

$$N^2 M \log(NMC) + 2N \log(NMC) \text{ となる。ここに } N = N_F + N_G, M = \max \{M_F, M_G\},$$

$C = \max \{C_F, C_G\}$ である。このほか p から決めていくときには、mod p で $A^\#(y, t)$ が消えなければ良いので、例えば、 $p \nmid \text{lc}_y(\text{resultant}_x(F(x, y), G(x, y+t)))$ を満たせばよい。この lc_y の上界を越える p は valid になる。

以上で述べたことは、まだ研究途上の事柄であって、今後我々の製作している数式処理システムなどで効果を(あるいは役立たなさを)確かめるつものものである。なお、ここで述べた格子算法の応用については、筆者等の論文(横山和弘, 竹島卓, Euclid 環上の因数分解および GCD について 格子算法の応用, コンピュータソフトウェア, Vol.5, No. 1(1988), pp.42-61)を参照されたい。また拡張格子算法については別の機会に述べたい。