TM-0418

modular算法による
多項式GCD計算について

横山和弘，野呂正行，竹島　卓

November, 1987

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191～5
Telex ICOT J32964

Institute for New Generation Computer Technology

modular 算法による多項式ＧＣＤ計算について

富士通㈱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，

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

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

$F(x,a) \neq 0 \neq G(x,a)$

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

②D(x)= 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 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) を越えるように巾 $q=p^e$ をとる．さらに,q,S=<y-a>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，

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

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．

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

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

$F(x,a) \neq 0 \neq -G(x,a) \bmod p$　　かつ

$\mathrm{resultant}_x\ (F,F')(a) \neq 0 \neq 0 \bmod p$　かつ

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

②イデアルとしてS=<p,y-a> を考える．mod S において Fと GとのＧＣＤを求める．

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

③ Berlekampのアルゴリズムにより$h^*(x)$ を因数分解し，既約因子$h^*_1,h^*_2,\ldots,h^*_t$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$H_{G,i}$とおくとき，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．さもなくば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（ＥＺＧＣＤ＋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，mod S において FとGとの GCDをもとめそれをh*(x) とおく．

②b(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．）

④new F=F(x,y)/B(x,y),

new G=G(x,y)/C(x,y) とおく．このとき,

if new G | new F then new G is the GCD;

if new F | new G then new F is the GCD;

otherwise try to find GCD(new F, 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．$A(y,t)=t^{e}A^{*}(y,t)=\mathrm{resultant}_{x}(F(x,y),G(x,y+t))$.

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，$M_F=\deg_y F$, $N_G=\deg_x G$, $N_F=\deg_x F$, $M_G \geq \deg_y G$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 ×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．　$\mathrm{GCD}(F(x,a),G(x,a))=\mathrm{sum}_i \{g_i x^i\}$ とし，係数 g_i をさらに

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

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

⑤逆に 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，

$p / | \mathrm{lc}_y (\mathrm{resultant}_x (F(x,y),G(x,y+t)))$

を満たせばよい．このlc_yの上界を求めるとそれを越える pは当然上式を満たすので，その上界がある程度小さく抑えられれば旨い．

ところで,validでない aの個数をI_aとおけば，

$I_a<p$なる素数 pが上式を満たすなら，$0<a<p-1$ の間で validでない aに（不幸にも）当たる確率はI_a/pである．

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）確かめるつもりのものである．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（Euclid環上の因数分解およびＧＣＤについて　格子算法の応用）がソフトウェア科学会誌上に掲載される予定であるので，詳しくはそちらを参照されたい．また拡張格子算法については別の機会に述べたい．