数理解析研究所講究録
第1572巻 2007年 50-58

整数係数多項式の近似GCD II

長坂耕作
KOSAKU NAGASAKA
神戸大学人間発達環境学研究科
GRADUATE SCHOOL OF HUMAN DEVELOPMENT AND ENVIRONMENT, KOBE UNIVERSITY*

1 はじめに

本発表では，第16回日本数式処理学会大会で発表し，その後証明等を行い雑誌に投稿した「整数係数多項式の近似GCD」を多変数多項式に拡張する予備実験を行った結果を取り扱っています．従って，整数係数多項式の近似GCDについて一変数多項式の結果を簡単に紹介した上で，多変数多項式に対する整数上の近似GCDについての実験の報告を行います．

2 一変数多項式の整数係数の近似GCD

一変数多項式や多変数多項式の複素数体上での近似GCDを求める研究は非常多くの成果が発表 [9, 4, 2, 13, 26, 25, 3, 27, 19, 29, 28, 21, 10, 18, 6, 20, 5, 15, 17, 22, 24] されています．しかしながら，係数を整数に限定した近似GCD，即ち係数部が離散的に変化する場合の近似GCDを求める問題に取り組んだ例はありませんでした．一見すると，浮動小数点数であるゆえに誤差が問題になるための近似GCDであり，係数部が整数で近似GCDを求める意義はないように思えますが，整数計画問題などに代表されるように，実際の社会の問題の多くは離散的な数を扱わなければなりません．本発表では，係数部の変化が整数上に制限される近似GCDを次のように定義します．

定義 1 (一変数の整数係数多項式の整数上の近似 GCD)
$f(x)$ と $g(x)$ を $\mathbf{Z}$ 上の一変数多項式とし，ε を小さな正整数とする．このとき，ある一変数多項式 $\bar{f}, \bar{g}, h$, $\Delta_f, \Delta_g \in \mathbf{Z}[x]$ が存在して，$f(x)$ と $g(x)$ が次式を満たすならば，次式を満たす多項式 $h(x)$ を**整数係数多項式の整数上の近似 *GCD*** という．

$$f(x) = \bar{f}(x)h(x) + \Delta_f(x),\ g(x) = \bar{g}(x)h(x) + \Delta_g(x),\ \varepsilon = \max\{\|\Delta_f\|, \|\Delta_g\|\}. \tag{1}$$

また，多項式 $\bar{f}(x)$ と $\bar{g}(x)$ を**整数係数多項式の整数上の近似余因子**と，ε をその**許容度**と呼ぶ．なお，$\|p\|$ は多項式 $p(x)$ の適当なノルムとする． ◁

この定義から整数上の近似GCDも通常の近似GCDと同じく，同じ許容度でも一意に定まらないことがわかりますが，本稿では許容度の最小化などは扱いません．例えば次の2つの問題は今後の課題とします．即ち，本稿での許容度は後退誤差になります．

*nagasaka@main.h.kobe-u.ac.jp

問題 1 (一変数の整数係数多項式の整数上の最近近似 GCD)
一変数の整数係数多項式 $f(x)$ と $g(x)$ に対し，$\varepsilon = \max\{\|\Delta_f\|, \|\Delta_g\|\}$ を最小化する (1) 式を満たす定数でない整数係数多項式 $h(x)$ を求めよ．そのような $h(x)$ を整数係数多項式の整数上の最近近似 *GCD* と呼ぶ．◁

問題 2 (一変数の整数係数多項式の整数上の最良近似 GCD)
一変数の整数係数多項式 $f(x)$ と $g(x)$，閾値 $\varepsilon \in \mathbf{N}$ に対し，$\max\{\|\Delta_f\|, \|\Delta_g\|\} \leq \varepsilon$ かつ (1) 式を満たす定数でない整数係数多項式 $h(x)$ を求めよ．そのような $h(x)$ を整数係数多項式の整数上の最良近似 *GCD* と呼ぶ．この定義では，$h(x)$ の次数を最大化していないことに注意されたい．◁

例 1 (互いに素な整数係数多項式の近似 GCD)
$f(x)$ と $g(x)$ を次の互いに素な整数係数多項式とします．

$$\begin{aligned} f(x) &= 54x^6 - 36x^5 - 192x^4 + 42x^3 + 76x^2 - 62x + 15, \\ g(x) &= 73x^5 + 36x^4 - 103x^3 - 70x^2 - 48x + 35. \end{aligned}$$

この多項式の組は次のような近似 *GCD* を持ちます．

$$\begin{aligned} f(x) &\approx (6x^4 - 10x^3 - 8x^2 + 7x - 3)(9x^2 + 9x - 5) \\ &= 54x^6 - 36x^5 - 192x^4 + 41\underline{x}^3 + 76x^2 - 62x + 15, \\ g(x) &\approx (8x^3 - 4x^2 - 3x - 7)(9x^2 + 9x - 5) \\ &= 7\underline{2}x^5 + 36x^4 - 103x^3 - 70x^2 - 48x + 35. \end{aligned}$$

この例では，$\Delta_f = -x^3$, $\Delta_g = -x^5$, $\varepsilon = 1$ になっており，最近近似 *GCD* であることがわかります．互いに素な多項式の許容度 $\varepsilon = 1$ の近似 *GCD* は常に最近近似 *GCD* になります．◁

整数は複素数に含まれるので，一見すると従来の近似 GCD アルゴリズムで，整数上の近似 GCD を求めることが出来るように思えます．実際，係数部に含まれる誤差が小さい（簡単な実験では，相対的に 10^{-10} 程度であれば十分）場合，従来の方法でも計算できることが多いです．しかしながら，前述の例 1 における多項式では，相対的な誤差の大きさは 10^{-3} 程度になっており，非常に大きいことがわかります．更に，従来のアルゴリズムを使った場合には，どのようにして複素数を整数に丸めれば良いかという新たな問題もあります．具体的に問題を確認するために，前述の例 1 の多項式を Kaltofen らのアルゴリズム [8] を用いて近似 GCD を計算したのが次の式になります（Kaltofen らによるアルゴリズムの実装を用いて実験を行っています）．

$$\begin{aligned} f(x) &\approx (1.00x^2 + 0.99x - 0.55)(54.34x^4 - 90.20x^3 - 72.20x^2 + 63.65x - 27.07), \\ g(x) &\approx (1.00x^2 + 0.99x - 0.55)(72.84x^3 - 36.20x^2 - 26.83x - 63.33). \end{aligned}$$

この例では後退誤差が 10^{-8} という非常に小さい結果が得られていますが，どのようにして整数に丸めれば良いかが問題となります．単純な四捨五入では例 1 の結果は得られないことが確認できます．そのため，本発表やその先行研究 [16] では複素数体に係数環を拡大することなく，整数の範囲で直接計算する方法を提案しています．一連の方法は非常に簡単で，格子算法 [23, 12, 1] により部分終結式の写像の零空間を計算するだけです．

2.1 格子算法による GCD 計算

近似を考えない厳密な GCD 計算では，多項式剰余列による Euclid の互除法が使われますし，近似 GCD 計算においても同じ性質を QR 分解や特異値分解を行って求めます（もちろん，最近のアルゴリズムはよ

り複雑になっていますので，あくまでも概要です）．これらのアルゴリズムは言い替えれば，ほとんどが部分終結式写像に基づいています．本発表とその先行研究においても，部分終結式写像の性質を格子算法で求めることで，整数係数多項式の近似 GCD を求めています．そこで，まずは部分終結式写像と格子算法による厳密な多項式 GCD の計算方法について簡単に説明します．

2.1.1 部分終結式写像

$f(x)$ と $g(x)$ を次の整数係数多項式とします．

$$f(x)=f_nx^n+f_{n-1}x^{n-1}+\cdots+f_0,\ g(x)=g_mx^m+g_{m-1}x^{m-1}+\cdots+g_0,$$

また，$f(x)$ の k 次の畳み込み行列を $C_k(f)$ とします．

$$C_k(f)=\begin{pmatrix} f_0 & 0 & \cdots & 0 \\ \vdots & f_0 & \ddots & \vdots \\ f_n & \vdots & \ddots & 0 \\ 0 & f_n & \vdots & f_0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & f_n \end{pmatrix} \in \mathbf{Z}^{(n+k)\times k}.$$

次の写像 $Syl_r(f,g)$ のことを，$f(x)$ と $g(x)$ の r 次の部分終結式写像といいます．

$$\begin{array}{rccccc} Syl_r(f,g): & \mathcal{P}_{m-r-1} & \times & \mathcal{P}_{n-r-1} & \to & \mathcal{P}_{n+m-r-1}, \\ & (s(x) & , & t(x)) & \mapsto & s(x)f(x)+t(x)g(x), \end{array}$$

ここで，$r=0,\ldots,\min\{n,m\}-1$ であり，$\mathcal{P}_d$ は d 次以下の多項式全体の集合とします．GCD 計算に用いられる性質としては，r を写像が単射とならない最大整数としたとき，$f(x)/t(x)$ と $g(x)/s(x)$ が $f(x)$ と $g(x)$ の GCD になるというのがあります．

畳み込み行列を用いることで，この部分集結式写像の行列表現である $Syl_r(f,g)=(C_{m-r}(f)\ \ C_{n-r}(g))$ を得られます．そして，$Syl_r(f,g)$ の零空間を求めることで GCD 計算に必要な $s(x)$ と $t(x)$ を求めることができます．実際，近似 GCD のアルゴリズムのいくつか [3, 25] では，$Syl_0(f,g)^t$ の QR 分解を用いています（厳密な GCD 計算の場合など，詳細な情報については [11, 9, 21] などを参照のこと）．

2.1.2 格子算法による GCD 算法

近似 GCD アルゴリズムでは QR 分解で零ベクトルを計算しますが，本発表と先行研究では良く知られている LLL アルゴリズム [12] を使って求めます．格子算法と呼ばれるもので，与えられた整数格子 $L=\{r_1\vec{v}_1+\cdots+r_d\vec{v}_d|r_i\in\mathbf{Z}\}\subseteq\mathbf{Z}^k$ に含まれるベクトルの中から，次を満たす短いベクトル $\vec{u}$ を捜し出すことができます．

$$\|\vec{u}\|\leq 2^{(d-1)/2}\min\{\|\vec{v}\|\mid\vec{0}\neq\vec{v}\in L\},\ \vec{u}\in L.$$

$2^{(d-1)/2}$ という上限はかなり大きいように思えますが，LLL アルゴリズムはほとんどの場合，この上限に比べてはるかに短いベクトルを発見することができます（格子算法については [23, 12, 1] を参照のこと）．

E_i を $i\times i$ の単位行列，c_B を整数とし，$(n+m-2r)\times(2n+2m-3r)$ の大きさの行列 $Syl_r^E(f,g)$ を $(E_{n+m-2r}\mid c_B\times Syl_r^t(f,g))$ と定義します．このとき次の補題が成り立ちます（証明については [16] を参照のこと）．

補題 2
B を $f(x)$ と $g(x)$ の Landau-Mignotte の上限 [14] のうち大きい方とし，$c_B = 2^{(n+m-2r-1)/2}\sqrt{n+m-2r}B$ とする．r を部分集結式写像が単射とならない最大整数とすれば，LLL アルゴリズムにより $Syl_r^E(f,g)$ の短いベクトルを求めることができ，その最初の $(n+m-2r)$ 個の要素は $f(x)$ と $g(x)$ の GCD の余因子の係数ベクトルのスカラー倍になっている． ◁

この補題での c_B は，Landau-Mignotte の上限を使っていることもあり非常に大きくなっていますが，ほとんどの場合，LLL は小さな c_B に対しても必要となる短いベクトルを計算してくれます．

例 2 (格子算法による GCD 計算の例)
次の互いに素でない非常に簡単な多項式の GCD を求めてみます．

$$\begin{aligned} f(x) &= 49x^2 - 25 &&= (7x-5)(7x+5), \\ g(x) &= 49x^2 + 70x + 25 &&= (7x+5)(7x+5). \end{aligned}$$

まず，次のように行列 $Syl_0^E(f,g)$ を作ります．補題は非常に大きな c_B に対してのみ保証されていますが，計算効率の面から $c_B = 1$ としています．そして，この行列に LLL アルゴリズムを適用すると，右側の行列が得られます．

$$\left(\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & -25 & 0 & 49 & 0 \\ 0 & 1 & 0 & 0 & 0 & -25 & 0 & 49 \\ 0 & 0 & 1 & 0 & 25 & 70 & 49 & 0 \\ 0 & 0 & 0 & 1 & 0 & 25 & 70 & 49 \end{array}\right) \rightarrow \left(\begin{array}{cccc|cccc} -5 & -7 & -5 & 7 & 0 & 0 & 0 & 0 \\ -2 & -3 & -2 & 3 & 0 & 10 & 14 & 0 \\ 0 & -1 & -1 & 1 & -25 & -20 & 21 & 0 \\ -2 & -2 & -2 & 3 & 0 & -15 & 14 & 49 \end{array}\right).$$

最初の行ベクトルが余因子に対応しており，実際，$7x-5$ と $7x+5$ の係数ベクトルが表れているのが確認できます．その結果，元の多項式を除することで多項式 GCD である $7x+5$ を得られます．この例は補題 2 における大きな c_B でなくとも，必要な短いベクトルが小さな c_B でも見付かることも示唆しています． ◁

2.1.3 格子算法による近似 GCD 算法

整数上の近似 GCD を求める場合，実際には互いに素な多項式同士になるため，近似余因子の係数ベクトルが部分集結式写像の行列表現の零空間に含まれなくなります．しかし，係数部の摂動により互いに素でなくなることから，近似余因子の係数ベクトルの零ベクトルからの差の大きさは小さいと考えられます．即ち，補題 2 と同様に，近似余因子の係数ベクトルを格子算法による $Syl_r(f,g)$ からの短いベクトルの検出に置き換えることができます．これにより，$s(x)f(x)+t(x)g(x) \approx 0$ を満たす近似余因子候補である $s(x), t(x) \in \mathbf{Z}[x]$ も求められます．近似 GCD を $h(x)$ とすれば，$f(x) \approx t(x)h(x)$ と $g(x) \approx -s(x)h(x)$ なる関係が成り立っています．

厳密な場合，余因子候補から GCD である $h(x)$ を求めるには，単純に $f(x)$ を $t(x)$ で除すれば良いだけですが，近似 GCD では $f(x) \approx t(x)h(x)$ と $g(x) \approx -s(x)h(x)$ なる関係しかありませんので，単純に除するだけでは近似 GCD を求めることはできません．そこで，先行研究 [16] では，c_H を整数として，大きさが $(r+2) \times (n+m+r+4)$ なる次の行列 $H(f,g,t,s)$ を使って計算しています（$\vec{f}$ と $\vec{g}$ はそれぞれ $f(x)$ と $g(x)$ の係数ベクトルです）．

$$H(f,g,t,s) = \left(\begin{array}{c|cc} E_{r+2} & \begin{array}{c} c_H \times \vec{f} \\ c_H \times C_{r+1}(-t)^t \end{array} & \begin{array}{c} c_H \times \vec{g} \\ c_H \times C_{r+1}(s)^t \end{array} \end{array}\right).$$

補題 3

B を $f(x)$ と $g(x)$ の *Landau-Mignotte* の上限のうち大きい方とし，$c_H = 2^{(r+1)/2}\sqrt{r+2}B$ とする．r を部分集結式写像が単射とならない最大整数とすれば，*LLL* アルゴリズムにより $H(f,g,t,s)$ の短いベクトルを求めることができ，その 2 番目から $(r+2)$ 番目までの要素は $f(x)$ と $g(x)$ の係数ベクトルのスカラー倍になっている． ◁

この補題は厳密な場合の話ですが，近似 GCD の計算についても十分機能することを次の例で確認します．

例 3 (格子算法による近似 GCD 計算の例)

前出の例の多項式を少しだけ変動させた次の互いに素な多項式の近似 *GCD* を求めてみます．

$$\begin{aligned} f(x) &= 49x^2 - 24 &&= (7x-5)(7x+5) - 1, \\ g(x) &= 49x^2 + 70x + 25 &&= (7x+5)(7x+5). \end{aligned}$$

まず，次のように行列 $Syl_0^E(f,g)$ を作ります（$c_B = 1$）．そして，この行列に *LLL* アルゴリズムを適用すると，右側の行列が得られます．

$$\left(\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & -24 & 0 & 49 & 0 \\ 0 & 1 & 0 & 0 & 0 & -24 & 0 & 49 \\ 0 & 0 & 1 & 0 & 25 & 70 & 49 & 0 \\ 0 & 0 & 0 & 1 & 0 & 25 & 70 & 49 \end{array}\right) \rightarrow \left(\begin{array}{cccc|cccc} -2 & -3 & -2 & 3 & -2 & 7 & 14 & 0 \\ -5 & -7 & -5 & 7 & -5 & -7 & 0 & 0 \\ 7 & 9 & 6 & -9 & -18 & -21 & 7 & 0 \\ 3 & 5 & 3 & -4 & 3 & -10 & 14 & 49 \end{array}\right).$$

右側半分の係数ベクトルに対応していた部分の大きさが最も小さい 2 行目を近似余因子候補として取り出し，次のように行列 $H(f,g,t,s)$ を作ります（$c_H = 1$）．そして，この行列に *LLL* アルゴリズムを適用すると，右側の行列が得られます．

$$\left(\begin{array}{ccc|cccccc} 1 & 0 & 0 & -24 & 0 & 49 & 25 & 70 & 49 \\ 0 & 1 & 0 & 5 & -7 & 0 & -5 & -7 & 0 \\ 0 & 0 & 1 & 0 & 5 & -7 & 0 & -5 & -7 \end{array}\right) \rightarrow \left(\begin{array}{ccc|cccccc} 1 & 5 & 7 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 5 & -7 & 0 & -5 & -7 & 0 \\ 0 & 0 & 1 & 0 & 5 & -7 & 0 & -5 & -7 \end{array}\right).$$

1 行目のベクトルに近似 *GCD* である $7x+5$ と変動に対応する係数ベクトルが出現しているのがわかると思います．この例では，$7x-5$ と $7x+5$ が整数上の近似余因子，許容度が 1 になっています．

ここまでの議論をまとめたものが次のアルゴリズムになります．

アルゴリズム 1 (1 変数版の整数上の近似 GCD アルゴリズム)

Input: $f(x), g(x) \in \mathbf{Z}[x]$, *of degrees* n *and* m, *respectively.*

Output: $s(x), t(x), h(x) \in \mathbf{Z}[x]$ *satisfying* $f(x) \approx s(x)h(x)$ *and* $g(x) \approx t(x)h(x)$.

1. $\varepsilon \leftarrow 1$; *while* $\varepsilon < \min\{\|f\|, \|g\|\}$ *do* **2–8**

2. $r \leftarrow \min\{n, m\} - 1$; *while* $r \geq 0$ *do* **3–7**

3. $c \leftarrow 1$; *while* $c \leq c_B$ *do* **4–6**

4. *construct a matrix* $Syl_r^E(f,g)$ *with* c, *apply the LLL algorithm and for each short vector do* **5**

5. *construct a matrix* $H(f,g,t,s)$, *apply the LLL algorithm, let* $h(x), t(x), s(x)$ *be candidate approximate GCD and cofactors, and output* $s(x), t(x), h(x)$ *if* $\max\{\|f(x) - t(x)h(x)\|, \|g(x) - s(x)h(x)\|\} \leq \varepsilon$

6. $c \leftarrow c \times 10$

7. $r \leftarrow r - 1$

8. $\varepsilon \leftarrow \varepsilon \times 10$

9. *output "not found".* ◁

このアルゴリズムの詳細については触れませんが，ステップ1での$\varepsilon = 1$，ステップ2での$r = \min\{n, m\} - 1$，ステップ3での$c = 1$など，近似GCD計算ゆえのパラメータ設定やLLLアルゴリズムの特性を考えた計算効率上の配慮などを行っています．従って，これらの値を変化させることでアルゴリズムの効率や結果は大きく変わることになります．

3 多変数多項式の整数係数の近似GCD

多変数多項式の整数上の近似GCDについても，基本的な考え方や方法は同じです．単純に，部分終結式写像とその行列表現を多変数の場合のものを用いて，格子算法で近似余因子候補を求め，再度格子算法により近似GCDを求めることになります．多変数多項式の近似GCDや厳密なGCD計算にも複数のアルゴリズムが知られていますが，本研究で取り上げる部分終結式写像の行列はGaoらによる近似GCD[7]で用いられているものと同じです．まずは，多変数多項式に関する近似GCDの定義を行っておきます．

定義 4 (多変数の整数係数多項式の整数上の近似 GCD)
$f(\vec{x})$ と $g(\vec{x})$ を $\mathbf{Z}[\vec{x}] = \mathbf{Z}[x_1, \ldots, x_\ell]$ 上の多変数多項式とし，ε を小さな正整数とする．このとき，ある多変数多項式 $\bar{f}, \bar{g}, h, \Delta_f, \Delta_g \in \mathbf{Z}[\vec{x}]$ が存在して，$f(\vec{x})$ と $g(\vec{x})$ が次式を満たすならば，次式を満たす多項式 $h(\vec{x})$ を**整数係数多項式の整数上の近似 *GCD*** という．

$$f(\vec{x}) = \bar{f}(\vec{x})h(\vec{x}) + \Delta_f(\vec{x}),\ g(\vec{x}) = \bar{g}(\vec{x})h(\vec{x}) + \Delta_g(\vec{x}),\ \varepsilon = \max\{\|\Delta_f\|, \|\Delta_g\|\}. \tag{2}$$

また，多項式 $\bar{f}(\vec{x})$ と $\bar{g}(\vec{x})$ を**整数係数多項式の整数上の近似余因子**と，ε をその**許容度**と呼ぶ．なお，$\|p\|$ は多項式 $p(\vec{x})$ の適当なノルムとする． ◁

$f(\vec{x})$ と $g(\vec{x})$ を整数係数多項式とし，それぞれの全次数を $n = \mathrm{tdeg}(f)$ と $m = \mathrm{tdeg}(g)$ とします．本発表では，次の写像 $Syl_r(f, g)$ のことを，$f(\vec{x})$ と $g(\vec{x})$ の r 次の部分終結式写像として1変数多項式と同じように扱います．

$$Syl_r(f,g): \begin{array}{ccccc} \mathcal{P}_{m-r-1} & \times & \mathcal{P}_{n-r-1} & \to & \mathcal{P}_{n+m-r-1}, \\ (s(\vec{x}) & , & t(\vec{x})) & \mapsto & s(\vec{x})f(\vec{x}) + t(\vec{x})g(\vec{x}), \end{array}$$

ここで，$r = 0, \ldots, \min\{n, m\} - 1$ であり，$\mathcal{P}_d$ は全次数が d 次以下の多項式全体の集合とします．1変数多項式の場合と同様に，r を写像が単射とならない最大整数としたとき，$f(\vec{x})/t(\vec{x})$ と $g(\vec{x})/s(\vec{x})$ が $f(\vec{x})$ と $g(\vec{x})$ のGCDになります．

多項式 $p(\vec{x})$ の係数ベクトル $\vec{p}$ は，その多項式の全次数以下の単項式を辞書式順序で並べたものを用います．即ち，$f(\vec{x})$ の係数ベクトル $\vec{f}$ は，全次数 n 次以下の単項式の辞書式順序で並べたもの，$g(\vec{x})$ の係数ベクトル $\vec{g}$ は，全次数 m 次以下の単項式の辞書式順序で並べたものになります．多変数版の k 次の畳み込み行列 $C_k(f)$ は，全次数が k の多項式 $p(\vec{x})$ に対し $C_k(f)\vec{p} = \vec{fp}$ を満たす行列として定義します．この畳み込み行列を用いることで，この部分集結式写像の行列表現である $Syl_r(f, g) = (C_{m-r}(f)\ \ C_{n-r}(g))$ を得られます．そして，1変数と同様に $Syl_r(f, g)$ の零空間を求めることでGCD計算に必要な $s(\vec{x})$ と $t(\vec{x})$ を求めることができます．後の処理は1変数多項式の場合と同じく，c_B 倍を行って単位行列を付加した行列 $Syl_r^E(f, g)$ に対してLLLアルゴリズムを適用し，求まった近似余因子候補から行列 $H(f, g, t, s)$ を作り近似GCDを求めます．

例 4 (多変数多項式の近似 GCD 計算の例)
次の簡単な多項式で，多変数の場合の整数上の近似 GCD を求めてみます．

$$\begin{aligned} f(x,y) &= 49x^2 - 24y^2 - 1 &&= (7x-5y)(7x+5y) - 1 + y^2, \\ g(x,y) &= 49x^2 - 70xy + 25y^2 &&= (7x-5y)(7x-5y). \end{aligned}$$

まず，次のように行列 $Syl_0^E(f,g)$ を作ります（$c_B = 1$）．

$$\left(\begin{array}{cccccc|cccccccccc} 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -24 & 0 & 0 & 0 & 0 & 49 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -24 & 0 & 0 & 0 & 0 & 49 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -24 & 0 & 0 & 49 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 25 & 0 & 0 & -70 & 0 & 49 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 25 & 0 & 0 & -70 & 0 & 49 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 25 & 0 & -70 & 49 \end{array}\right)$$

そして，この行列に LLL アルゴリズムを適用すると，次の行列が得られます．

$$\left(\begin{array}{cccccc|cccccccccc} 0 & -2 & 3 & 0 & -2 & -3 & 0 & 2 & 0 & -2 & -3 & 0 & -7 & 0 & 14 & 0 \\ 0 & 5 & -7 & 0 & 5 & 7 & 0 & -5 & 0 & 5 & 7 & 0 & -7 & 0 & 0 & 0 \\ 0 & 7 & -9 & 0 & 6 & 9 & 0 & -7 & 0 & -18 & 9 & 0 & 21 & 0 & 7 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -24 & 0 & 0 & 0 & 0 & 49 & 0 & 0 \\ 0 & -3 & 5 & 0 & -3 & -4 & 0 & 3 & 0 & -3 & -5 & 0 & -10 & 0 & -14 & 49 \\ -1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 49 & 0 & 0 & -70 & 0 & 0 & 0 & 0 \end{array}\right)$$

近似 GCD の場合，右側半分の係数ベクトルに対応していた部分の大きさが最も小さいベクトルが近似余因子とは限らず，小さい順に試行する必要があります．順次試行することで結果として 2 行目が近似余因子に対応していることがわかるので，その場合の処理についてのみ紹介します．2 行目を取り出し，次のように行列 $H(f,g,t,s)$ を作ります（$c_H = 1$）．

$$\left(\begin{array}{cccc|cccccccccc} 1 & 0 & 0 & 0 & -1 & 0 & -24 & 0 & 0 & 49 & 0 & 0 & 25 & 0 & -70 & 49 \\ 0 & 1 & 0 & 0 & 0 & 5 & 0 & 7 & 0 & 0 & 0 & -5 & 0 & 7 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 5 & 0 & 7 & 0 & 0 & 0 & -5 & 0 & 7 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 5 & 7 & 0 & 0 & 0 & 0 & -5 & 7 \end{array}\right)$$

そして，この行列に LLL アルゴリズムを適用すると，次の行列が得られます．

$$\left(\begin{array}{cccc|cccccccccc} 1 & 0 & 5 & -7 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 5 & 0 & 7 & 0 & 0 & 0 & -5 & 0 & 7 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 5 & 0 & 7 & 0 & 0 & 0 & -5 & 0 & 7 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 5 & 7 & 0 & 0 & 0 & 0 & -5 & 7 \end{array}\right)$$

1 行目のベクトルに近似 GCD である $7x-5y$ と変動に対応する係数ベクトルが出現しているのがわかると思います．この例では，$7x+5y$ と $7x-5y$ が整数上の近似余因子，摂動多項式が y^2-1 になっています．

4 まとめ

本発表では，1 変数多項式の場合と同様に多変数多項式の整数上の近似 GCD を計算する方法の取り組みについて取り上げました．1 変数多項式の厳密な場合の補題等，多変数多項式の場合にも自然に拡張できるものの証明を与えていない点など未着手の部分もありますが，同じ方法で計算できる可能性を示せました．

今後の課題としては，本方法における近似GCDの発見がLLLアルゴリズムに直接依存しすぎしていることの改善（ナップザック法による因数分解ではそうではない）と，多変数多項式の場合には計算時間が非常にかかることの改善などを考えています．なお，実験に使用したMathematicaのプログラムは，SNAPパッケージとして公開予定ですが，現在のコードは古いMathematica用になっているため，これも今後の課題となっています．

参 考 文 献

[1] W. Backes and S. Wetzel. Heuristics on lattice basis reduction in practice. *ACM J. Exp. Algorithmics*, 7:21 pp. (electronic), 2002. Fourth Workshop on Algorithm Engineering (Saarbrücken, 2000).

[2] D. Christou and M. Mitrouli. Estimation of the greatest common divisor of many polynomials using hybrid computations performed by the ERES method. *Appl. Numer. Anal. Comput. Math.*, 2(3):293–305, 2005.

[3] R. M. Corless, S. M. Watt, and L. Zhi. *QR* factoring to compute the GCD of univariate approximate polynomials. *IEEE Trans. Signal Process.*, 52(12):3394–3402, 2004.

[4] G. M. Diaz-Toca and L. Gonzalez-Vega. Computing greatest common divisors and squarefree decompositions through matrix methods: the parametric and approximate cases. *Linear Algebra Appl.*, 412(2-3):222–246, 2006.

[5] I. Z. Emiris, A. Galligo, and H. Lombardi. Numerical univariate polynomial GCD. In *The mathematics of numerical analysis (Park City, UT, 1995)*, volume 32 of *Lectures in Appl. Math.*, pages 323–343. Amer. Math. Soc., Providence, RI, 1996.

[6] I. Z. Emiris, A. Galligo, and H. Lombardi. Certified approximate univariate GCDs. *J. Pure Appl. Algebra*, 117/118:229–251, 1997. Algorithms for algebra (Eindhoven, 1996).

[7] S. Gao, E. Kaltofen, J. May, Z. Yang, and L. Zhi. Approximate factorization of multivariate polynomials via differential equations. In *ISSAC 2004*, pages 167–174. ACM, New York, 2004.

[8] E. Kaltofen, Z. Yang, and L. Zhi. Approximate greatest common divisors of several polynomials with linearly constrained coefficients and singular polynomials. In *ISSAC 2006*, pages 169–176. ACM, 2006.

[9] N. Karcanias, S. Fatouros, M. Mitrouli, and G. H. Halikias. Approximate greatest common divisor of many polynomials, generalised resultants, and strength of approximation. *Comput. Math. Appl.*, 51(12):1817–1830, 2006.

[10] N. K. Karmarkar and Y. N. Lakshman. On approximate GCDs of univariate polynomials. *J. Symbolic Comput.*, 26(6):653–666, 1998. Symbolic numeric algebra for polynomials.

[11] M. A. Laidacker. Another theorem relating Sylvester's matrix and the greatest common divisor. *Math. Mag.*, 42:126–128, 1969.

[12] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261(4):515–534, 1982.

[13] T. Y. Li and Z. Zeng. A rank-revealing method with updating, downdating, and applications. *SIAM J. Matrix Anal. Appl.*, 26(4):918–946 (electronic), 2005.

[14] M. Mignotte. An inequality about factors of polynomials. *Math. Comp.*, 28:1153–1157, 1974.

[15] M. Mitrouli and N. Karcanias. Computation of the GCD of polynomials using Gaussian transformations and shifting. *Internat. J. Control*, 58(1):211–228, 1993.

[16] K. Nagasaka. Approximate gcd of two univariate polynomials over integers. *(submitted)*, 2007.

[17] M.-a. Ochi, M.-T. Noda, and T. Sasaki. Approximate greatest common divisor of multivariate polynomials and its application to ill-conditioned systems of algebraic equations. *J. Inform. Process.*, 14(3):292–300, 1991.

[18] V. Y. Pan. Approximate polynomial gcds, Padé approximation, polynomial zeros and bipartite graphs. In *Proceedings of the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms (San Francisco, CA, 1998)*, pages 68–77, New York, 1998. ACM.

[19] V. Y. Pan. Computation of approximate polynomial GCDs and an extension. *Inform. and Comput.*, 167(2):71–85, 2001.

[20] C. Rössner and J.-P. Seifert. The complexity of approximate optima for greatest common divisor computations. In *Algorithmic number theory (Talence, 1996)*, volume 1122 of *Lecture Notes in Comput. Sci.*, pages 307–322. Springer, Berlin, 1996.

[21] D. Rupprecht. An algorithm for computing certified approximate GCD of n univariate polynomials. *J. Pure Appl. Algebra*, 139(1-3):255–284, 1999. Effective methods in algebraic geometry (Saint-Malo, 1998).

[22] T. Sasaki and M.-T. Noda. Approximate square-free decomposition and root-finding of ill-conditioned algebraic equations. *J. Inform. Process.*, 12(2):159–168, 1989.

[23] C.-P. Schnorr and M. Euchner. Lattice basis reduction: improved practical algorithms and solving subset sum problems. *Math. Programming*, 66(2, Ser. A):181–199, 1994.

[24] A. Schönhage. Quasi-gcd computations. *J. Complexity*, 1(1):118–137, 1985.

[25] C. J. Zarowski, X. Ma, and F. W. Fairman. QR-factorization method for computing the greatest common divisor of polynomials with inexact coefficients. *IEEE Trans. Signal Process.*, 48(11):3042–3051, 2000.

[26] Z. Zeng and B. H. Dayton. The approximate GCD of inexact polynomials. II. A multivariate algorithm. In *ISSAC 2004*, pages 320–327. ACM, New York, 2004.

[27] L. Zhi. Displacement structure in computing approximate GCD of univariate polynomials. In *Computer mathematics*, volume 10 of *Lecture Notes Ser. Comput.*, pages 288–298. World Sci. Publ., River Edge, NJ, 2003.

[28] L. Zhi and M.-T. Noda. Approximate GCD of multivariate polynomials. *Sūrikaisekikenkyūsho Kōkyūroku*, (1138):64–76, 2000. Research on the theory and applications of computer algebra (Japanese) (Kyoto, 1999).

[29] L. H. Zhi and M.-T. Noda. Approximate GCD of multivariate polynomials. In *Computer mathematics (Chiang Mai, 2000)*, volume 8 of *Lecture Notes Ser. Comput.*, pages 9–18. World Sci. Publ., River Edge, NJ, 2000.