数理解析研究所講究録
第1759巻 2011年 1-10

種々の行列を利用した整数係数近似GCD計算法
Computing Approximate Polynomial GCD over Integers via Various Matrices

讃岐 勝
MASARU SANUKI
筑波大学 教育開発国際協力研究センター (CRICED)
CENTER FOR RESEARCH ON INTERNATIONAL COOPERATION IN EDUCATIONAL DEVELOPMENT,
UNIVERSITY OF TSUKUBA *

Abstract

We study the computation of approximate polynomial GCD over integers, which is based on the lattice methods on several matrices such as the Bezout and the Hankel matrices. Additionally, we propose an optimization of "monic" approximate GCD over integers. We show that our methods are efficient.

1 はじめに

1980年代末に近似代数の概念が提唱されて以来，実係数・複素係数多項式の近似GCDおよび近似因数分解の研究が今に至るまで，そして今後も盛んに行われると思われる．一方，整数係数多項式についても同時期に考えていたという話は聞くが，その研究記録はほとんど残っていない．

近年，整数係数多項式に関する近似代数に関する研究もされれつつある．実験を行うと，浮動小数係数多項式の算法をそのまま利用することは不可能であることがすぐにかわる．LLL法などの組み合わせ問題 (格子問題) を解く (近似) 解法の利用が有効であることが報告されている (近似因数分解 [Hoeji02, SO09]，近似GCD[Nagasaka08])．

RIMS共同研究「数式処理研究の新たな発展」と研究集会の題目にちなみ，本稿では「浮動小数係数多項式の近似GCD計算法」を「整数係数多項式の近似GCDの計算」に適応した場合，どのように算法が振舞うのかを検証し，整数係数多項式の近似GCD計算法について考察を行った．

本稿で話題にするのは，Sylvester行列以外の行列を用いた近似GCD算法の振る舞いである．整数係数多項式の近似GCD計算する際にはLLL法などの格子算法が利用される．格子算法の計算量は，基底の数に対応する行列のサイズ (数) に比例して計算量が極端に増大する．Sylvester行列のサイズは，与えられた多項式の次数の和で与えられるが，Bezout行列やHankel行列のサイズは次数の最大値 (Sylvester行列の場合のおよそ半分) で抑えられる．そのため，行列によっては計算量を著しく減ることが期待できる．しかしながら，基底の数を減るが，近似算法を用いるため精度に関する問題が生じる．本稿では，問題点に関する考察と解決法に関しても考察を行い，既存の方法との比較を行う．

*sanuki@criced.tsukuba.ac.jp

本稿で扱う近似 GCD はモニックであると仮定する．主係数があらかじめわかっている，または，主係数を近似 GCD および余因子に振り分けられる、ことを仮定すれば先のように仮定できる．主係数の振り分けが難しいか否かについて問題にしない．

以下では次の記号を使う．$\mathbb{K}$ を標数 0 の数体，$\mathbb{Z}[x]$ を主変数 x に関する整数係数の多項式環，$\mathbb{Z}[\boldsymbol{u}][x]$ を従変数 $\boldsymbol{u} = (u_1, \ldots, u_\ell)$ に関する整数係数多項式 $\mathbb{Z}[\boldsymbol{u}]$ を係数とする多項式環とする．$f(x), g(x) \in \mathbb{K}[x]$ について，$\deg(f)$ を主変数 x に関する次数，$||f||_p$ を多項式の p-ノルムとする (本稿では，$p = 2$ および $p = \infty$ を扱う)．$\gcd(f, g)$ および $\mathrm{appgcd}(f, g)$ それぞれを多項式 f と g の GCD(gratest common divisor：最大公約子) および近似 GCD とする (近似 GCD の定義は §1.1 でする)．

基底 $\mathrm{B} = (\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n) \in \mathbb{K}^{m \times n}$ の線形和について，次の 2 つを定義する．span(B) および $\mathcal{L}$(B) をそれぞれ基底 B の $\mathbb{K}$-線形結合および $\mathbb{Z}$-線形結合とする：

$$\begin{aligned} \mathrm{span}(\mathrm{B}) &= \{\boldsymbol{b} \mid \boldsymbol{b} = \sum_{i=1}^{n} a_i \boldsymbol{b}_i \ with\ a_i \in \mathbb{K}\}, \\ \mathcal{L}(\mathrm{B}) &= \{\boldsymbol{b} \mid \boldsymbol{b} = \sum_{i=1}^{n} a_i \boldsymbol{b}_i \ with\ a_i \in \mathbb{Z}\}. \end{aligned}$$

1.1 近似 GCD

まず，整数係数多項式の近似 GCD を次のように定義する．

定義 1 (整数係数多項式の近似 GCD)
整数係数多項式 F と G が次のように整数係数の多項式の積と和で分解できたとする．

$$\begin{cases} F &= C\tilde{F} + \Delta_F \\ G &= C\tilde{G} + \Delta_G \end{cases} \quad with\ \varepsilon = d(\Delta_F, \Delta_G) < ||F||, ||G||. \tag{1}$$

ここで $d(\Delta_F, \Delta_G)$ は，摂動項の大きさを測る関数である ($\Delta_F + \Delta_G$ や $\max\{\frac{||\Delta_F||}{||F||}, \frac{||\Delta_G||}{||G||}\}$ など；摂動項 Δ_F, Δ_G と多項式 F, G が区別できるような関数であればよいとする)．このとき，C を F と G の許容度 $d(\Delta_F, \Delta_G)$ の近似共通因子と呼び，次数最大の C を許容度 $d(\Delta_F, \Delta_G)$ の近似 *GCD* と呼ぶ．

注意 1
本稿では，$||F||, ||G|| \approx 10\, d(\Delta_F, \Delta_G)$ であることを仮定する．*Gathen-Shaparlinski[GS08]* は，$||F||, ||G|| \lll d(\Delta_F, \Delta_G)$ の場合を扱っている．計算法の観点か言えば浮動小数係数近似 *GCD* の方が優れている (計算方法よりも，性質について議論をしている)．長坂 *[長坂 10]* では，

$$\begin{aligned} f(x) &= 54x^6 - 36x^5 - 192x^4 + 42x^3 + 76x^2 - 62x + 15 \\ &= (6x^4 - 10x^3 - 8x^3 + 7x - 3)\underline{(9x^2 + 9x - 5)} - x^3, \\ g(x) &= 73x^5 + 36x^4 - 103x^3 - 70x^2 - 48x + 35 \\ &= (8x^3 - 4x^2 - 3x - 7)\underline{(9x^2 + 9x - 5)} - x^5, \end{aligned}$$

のように $d(\Delta_F, \Delta_G) = \max\{||\Delta_F||, ||\Delta_G||\}$ が小さい場合を扱い，許容度が最小のものを**最近近似 GCD** と呼んでいる．

1.2 SVP と CVP

SVP (Smallest Vector Problem：最小ベクトル問題) および CVP (Closest Vector Problem：最短ベクトル問題) とは，基底 $\mathrm{B} = (\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_n) \in \mathbb{K}^{m \times n}$ の和に関する組み合わせ問題である．

定義 2 (SVP)
基底 B が与えられている．SVP とは，基底 B で張られる格子 $\mathcal{L}(\mathrm{B})$ の中で最小の長さを持つベクトル $\boldsymbol{x} \in \mathcal{L}(\mathrm{B})$ を求める問題である．

定義 3 (CVP)
基底 B とベクトル $\boldsymbol{t} \in \mathbb{K}^m$ が与えられている．SVP とは，基底 B で張られる格子 $\mathcal{L}(\mathrm{B})$ の中でベクトル $\boldsymbol{t}$ に近いベクトル $\boldsymbol{x} \in \mathcal{L}(\mathrm{B})$ を求める問題である．$\boldsymbol{t} = \boldsymbol{0} \in \mathbb{K}^m$ とすることで *CVP* は *SVP* に帰着できる．

SVP および CVP ともに厳密に解く場合，事実上全ての組み合わせを調べる必要があるので，実際に計算する際には厳密解法ではなく近似解法が使われる．本書でも LLL 法などの近似解法を利用する．

定義 4 (LLL 基底)
基底 $\boldsymbol{b}_1, \cdots, \boldsymbol{b}_n$ が次の条件をみたすとき，*LLL* 簡約された基底 *(*単に *LLL* 基底*)* であるという．

$$||\boldsymbol{b}_1|| \leq \frac{3}{4}||\boldsymbol{b}_2|| \leq \cdots \leq \frac{3}{4}||\boldsymbol{b}_n||. \tag{2}$$

上の LLL 基底は，Gram-Schimidt の直交基底法に類似した方法で求めることができる．

アルゴリズム 1 (LLL 法 (改良 Gram-Schmidt 法))
基底 $\mathrm{B} = (\boldsymbol{b}_1, \cdots, \boldsymbol{b}_n)$ について，G を B の *Gram-schmidt* 行列とするとき，次をみたすように軸交換をしつつ *Gram-schmidt* 直交基底法を適応する．

$$\boxed{\mathrm{G} = (\mu_{i,j}) \text{ について，} |\mu_{i,j}| \leq \frac{1}{2} \quad (i > j).}$$

2 1変数多項式近似 GCD

与えられた多項式 $f, g \in \mathbb{Z}[x]$ について，この多項式の近似 GCD を $c(x) = c_k x^k + c_{k-1} x^{k-1} + \cdots + c_0$ と表すことにする．

2.1 Sylvester 行列を利用

Sylvester 行列を利用した整数係数多項式の近似 GCD 計算法は [長坂 07, GS08] など，いくつかのアプローチが研究されている．本稿では，以下で提案する方法と関係させて議論できるように，長坂による方法を紹介する [長坂 07, Nagasaka08].

$\mathrm{S}_k(f, g)$ を f と g で構成される Sylvester 行列の部分行列とする；

$$\mathrm{S}_k(f,g) = \begin{pmatrix} f_n & f_{n-1} & \cdots & f_0 & & \\ & f_n & f_{n-1} & \cdots & f_0 & \\ & & \ddots & \ddots & \cdots & \ddots \\ g_m & g_{m-1} & \cdots & g_0 & & \\ & g_m & g_{m-1} & \cdots & g_0 & \\ & & \ddots & \ddots & \cdots & \ddots \end{pmatrix} \in \mathbb{K}^{(n+m-2k)\times(n+m-2k)}. \tag{3}$$

$k = \deg(\gcd(f,g))$ のとき，S_k の核は f と $-g$ の余因子の係数を表す．ゆえに、S_k の列ベクトルによる SVP を解くことによって理論的には f と $-g$ の余因子を計算できる．しかし，実際の計算はうまくいかず次のような c_B による工夫 (重み付け) をしなければ，計算がうまくいかないことが多い．

$$\left(E_{n+m-2k} \mid c_B \times \mathrm{S}_k(f,g) \right). \tag{4}$$

ここで，E_k を k 次単位行列である．

命題 5 ([長坂 07])
$\boldsymbol{f}$ と $\boldsymbol{g}$ をそれぞれ $f(x)$ と $g(x)$ の係数行列とする．$C_k(f) \in \mathbb{Z}^{(n+k)\times k}$ を多項式 f の k 次畳み込み行列，行列 $H(f,g,t,s)$ を次で定義される行列とする．

$$H(f,g,\tilde{f},\tilde{g}) = \left(E_{k+2} \;\middle|\; \begin{array}{cc} c_H \times \boldsymbol{f} & c_H \times \boldsymbol{g} \\ c_H \times C_{k+1}(-\tilde{f})^T & c_H \times C_{k+1}(\tilde{g})^T \end{array} \right) \in \mathbb{Z}^{(k+2)\times(n+m+k+4)}. \tag{5}$$

$H(f,g,\tilde{f},\tilde{g})$ の *LLL* 簡約された基底について，*2* 番目から $k+2$ 番目までの要素は，$\tilde{f}$ および $\tilde{g}$ からなる係数ベクトルのスカラー倍となる． ∎

以上より，Sylvester 行列による近似 GCD 計算法は次のように書ける．

アルゴリズム 2 ([長坂 07])
次の流れで計算を行う *(詳細は [長坂 07] を参照)*.

1. 近似 *GCD* の次数 k を決める．
2. c_B を決める *(はじめは小さく、うまくいかなければ徐々に大きくする)*.
3. $\tilde{f}$ と $\tilde{g}$ を $S_k(f,g)$ の核 *(行列 4 を利用)* から計算.
4. $H(f,g,\tilde{f},\tilde{g})$ を *LLL* 簡約する．
5. 結果が妥当か判断する．妥当であれば結果を返し，妥当でなければ，k または c_B を設定し直して再計算する．

2.2 Bezout 行列を利用

多項式 f と g の Beout 多項式 $\mathrm{Bpol}(f,g)$ を次の式で定義する。

$$\mathrm{Bpol}(f,g) = \frac{f(x)g(y) - f(y)g(x)}{x-y} = \sum_{0\le i,j\le n-1} b_{i,j}x^i y^j \in \mathbb{K}[x,y]. \tag{6}$$

このとき，Bezout 行列 $\mathrm{B}(f,g)$ を Bezout 多項式の係数によって，次のように定義する．

$$\mathrm{B}(f,g) = (b_{i,j})_{0\le i,j\le n-1} = (\boldsymbol{b}_0, \boldsymbol{b}_1, \cdots, \boldsymbol{b}_{n-1}) \in \mathbb{K}^{n\times n}. \tag{7}$$

Bezout 行列の GCD について次の定理が成り立つ．

定理 6 (Barnett の定理：その 1)
$k = \deg(\gcd(f,g))$ とする．このとき，$n-k$ 個のベクトル $\boldsymbol{b}_k, \cdots, \boldsymbol{b}_{n-1}$ は一次独立であり，かつ k 個のベクトル $\boldsymbol{b}_0, \cdots, \boldsymbol{b}_{k-1}$ は $n-k$ 個のベクトル $\boldsymbol{b}_k, \cdots, \boldsymbol{b}_{n-1}$ で張ることができる；

$$\boldsymbol{b}_i = c_0^{(i)}\boldsymbol{b}_k + \sum_{j=1}^{n-k-1} c_j^{(i)}\boldsymbol{b}_{k+j} \tag{8}$$

$$= (\boldsymbol{b}_k, \cdots, \boldsymbol{b}_{n-k})\boldsymbol{c}_i. \tag{9}$$

このとき，$0 \le i \le k-1$ に対して，$c_0^{(i)} = c_i/c_k$ である． ∎

近似 GCD の許容度について，次が成立する．

補題 7 (近似 GCD の許容度と Bezout 行列の摂動の関係 [讃岐 09])
式 (2) で表現される多項式 F と G が与えられている．このとき，次が成立する．

$$||\mathrm{B}(F,G) - \mathrm{B}(F-\Delta_F, G-\Delta_G)||/||\mathrm{B}(F,G)|| = O(\varepsilon). \tag{10}$$

∎

浮動小数係数多項式の場合には $||\mathrm{B}(f,g)|| \approx 1$ と仮定できる．整数係数多項式の場合には仮定できないが，$||\mathrm{B}(f,g)|| \approx 1$ の場合には Bezout 行列を用いて整数係数多項式の近似 GCD が計算できる．式 (10) をみればわかるが相対的に許容度を見積もらなければいけない．例えば，

$$f(x) = 100(x^3+x+1)(x^3-x+1)+x, \quad g(x) = 100(x^3+x+1)(x^3-x^2+1)-x^2,$$

の Bezout 行列は

$$\mathrm{B}(f,g) = \begin{pmatrix} 10000 & -10000 & 10000 & 100 & -9900 & 0 \\ -10000 & 10000 & -19900 & 100 & -100 & -10000 \\ 10000 & -19900 & 20100 & -10100 & -9900 & 10000 \\ 100 & 100 & -10100 & 300 & -9900 & -10000 \\ -9900 & -100 & -9900 & -9900 & -101 & -100 \\ 0 & -10000 & 10000 & -10000 & -100 & 9900 \end{pmatrix}$$

であり、$f(x)-x$ と $g(x)+x^2$ の Bezout 行列は

$$\mathrm{B}(f-x, g+x^2) = \begin{pmatrix} 10000 & -10000 & 10000 & 200 & -10000 & 0 \\ -10000 & 10000 & -19800 & 0 & 0 & -10000 \\ 10000 & -19800 & 20000 & -10000 & -10000 & 10000 \\ 200 & 0 & -10000 & 400 & -10000 & -10000 \\ -10000 & 0 & -10000 & -10000 & 0 & -200 \\ 0 & -10000 & 10000 & -10000 & -200 & 10000 \end{pmatrix}$$

となる．そのため，Bezout 行列を用いた近似 GCD 計算の場合，許容度は浮動小数係数多項式の近似 GCD の許容度のように，

$$\max\{\frac{||\Delta_F||}{||F||}, \frac{||\Delta_G||}{||G||}\} \tag{11}$$

または

$$\max\left\{\frac{||\Delta_C||}{\max\{||F||, ||G||\}}\right\} \tag{12}$$

と定義するべきである．このため，許容度が最小というものは考えない (注意 1)．

補題 8 (GCD がモニックの場合)
各 i について，$c_j^{(i)} \in \mathbb{K}$. 特に，$f(x), g(x) \in \mathbb{Z}[x]$ ならば，$c_j^{(i)} \in \mathbb{Z}$. ∎

近似 GCD がモニックであれば，CVP を解くことによって理論的には近似 GCD を計算することができる．しかし，実際には計算がうまくいかないことが多い．なぜならば，厳密な CVP を解くわけではなく，SVP および CVP の近似解法を用いて解くからである．

例 1 (Bezout 行列を利用した厳密 GCD 計算)
次の多項式 f と g は厳密に *GCD* をもつ.

$$\begin{aligned} f &= (x^3+x+1)(x^3-x+1), \\ g &= (x^3+x+1)(x^3-x^2+1). \end{aligned}$$

GCD を *Bezout* 行列の列から構成される *CVP* に帰着させて計算させたとき，次の値を得た．*CVP* は *2* つの方法によって解いた *(1* つは *LLL* 法を利用し，もう *1* つは *LLL* 法を利用しない *Babai* の方法 *(*丸めアルゴリズム*)* である*)*.

- *LLL* 法を利用
  $\boldsymbol{c}_0 = (1,-1,1)^T, \boldsymbol{c}_1 = (1,1,0)^T, \boldsymbol{c}_2 = (0,-1,1)^T$.
- *Babai* の方法を利用
  $\boldsymbol{c}_0 = (0,0,1)^T, \boldsymbol{c}_1 = (0,0,1)^T, \boldsymbol{c}_2 = (0,-1,2)^T$.

厳密に解くと次の結果を得る *(*この場合は，*LU* 分解で解くことが可能 *[Sanuki09])*.

- *LU* 分解を用いた厳密解法
  $\boldsymbol{c}_0 = (1,0,-1)^T, \boldsymbol{c}_1 = (1,1,-1)^T, \boldsymbol{c}_2 = (0,1,1)^T$.

以上から，近似算法は正確な *GCD* 計算には向かないことがわかる．しかし，近似 *GCD* の許容度の観点からみると，*LLL* 法を利用して得られた結果は決して悪いものではない.

2.3 Hankel 行列を利用

$\frac{g}{f}$ $(\deg(g) \leq \deg(f) = n)$ の級数展開

$$\frac{g}{f} = \sum_{i=0} h_i x^{-i}$$

の係数から構成される Hankel 行列 $\mathrm{H}(f,g)$ を次のように定義する.

$$\mathrm{H}(f,g) = \begin{pmatrix} h_0 & h_1 & \cdots & h_{n-1} \\ h_1 & h_2 & \cdots & h_n \\ \vdots & \vdots & \ddots & \vdots \\ h_{n-1} & h_n & \cdots & h_{2n-2} \end{pmatrix} = (\boldsymbol{h}_0, \boldsymbol{h}_1, \cdots, \boldsymbol{h}_{n-1}) \in \mathbb{Q}^{n\times n} \tag{13}$$

このとき，Hankel 行列の各列について Bezout 行列の場合と似た関係式が得られる.

定理 9 (Barnett の定理：その 2 [DG02])
$k = \deg(\gcd(f,g))$ とする．このとき，$n-k$ 個のベクトル $\boldsymbol{h}_0, \cdots, \boldsymbol{h}_{n-k-1}$ は一次独立であり，かつ k 個のベクトル $\boldsymbol{h}_{n-k}, \cdots, \boldsymbol{h}_{n-1}$ は $n-k$ 個のベクトル $\boldsymbol{h}_0, \cdots, \boldsymbol{h}_{n-k-1}$ で張ることができる；

$$\boldsymbol{h}_{n-k+i} = \sum_{j=0}^{n-k-2} d_j^{(i)} \boldsymbol{h}_j + d_{n-k-1}^{(i)} \boldsymbol{h}_{n-k}$$

このとき，

$$\begin{pmatrix} 1 \\ c_{k-1} \\ \vdots \\ c_0 \end{pmatrix} = \begin{pmatrix} f_m & & & \\ f_{m-1} & f_m & & \\ \vdots & \vdots & \ddots & \\ f_{m-k-1} & f_{m-k} & \cdots & f_m \end{pmatrix} \begin{pmatrix} 1 \\ d_{n-k-1}^{(k-1)} \\ \vdots \\ d_{n-k-1}^{(0)} \end{pmatrix}$$

をみたす． ∎

補題 10 (近似 GCD の許容度と Hankel 行列の摂動の関係)
式 (2) で表現される多項式 F と G が与えられている．このとき，次が成立する．

$$||\mathrm{H}(F,G) - \mathrm{H}(F-\Delta_F, G-\Delta_G)||/||\mathrm{H}(F,G)|| = O(\varepsilon). \tag{14}$$

ここでも，Bezout 行列の場合と同様に GCD はモニックであることを仮定する．このとき，$f = f(1/x)\cdot x^n$ および $g = g(1/x)\cdot x^m$ なる変換によって多項式の定数項が 1 になり，級数展開で得られる各項は $h_i \in \mathbb{Z}$ となる．さらに，$d_i^j \in \mathbb{Z}$ をみたすので格子算法によって計算することができる．

例 2 (Hankel 行列を利用)
例 *1* で与えられた多項式について，*GCD* を計算した．

- *LLL* 法を利用
  $\boldsymbol{c}_0 = (-1,2,1)^T, \boldsymbol{c}_1 = (1,1,-1)^T, \boldsymbol{c}_2 = (1,0,-1)^T$.

Babai の方法 *(*丸めアルゴリズム*)* は，*LLL* 法を利用する方法より精度がでないことが知られているので、実験に使用しなかった．

許容度の観点からみると，上の結果も決して悪いものではない．しかし，整数係数多項式の近似 *GCD* としてみると受け入れられ難い．

2.4 Bezout 行列，Hankel 行列の単純な拡張

Bezout 行列，Hankel 行列ともに計算はうまくいかなかった．[長坂 07] では，行列の係数に重みをつけて LLL 法がうまく適用できる形に変換している．Bezout 行列，Hankel 行列において，上の方法を適応させるため，GCD の係数 c_i が同時に求まるような次の行列を考える．

$$\begin{pmatrix} \tilde{\mathrm{H}} & & & \\ & \tilde{\mathrm{H}} & & \\ & & \ddots & \\ & & & \tilde{\mathrm{H}} \end{pmatrix} \begin{pmatrix} \boldsymbol{d}_0 \\ \boldsymbol{d}_1 \\ \vdots \\ \boldsymbol{d}_{k-1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{h}_{n-k} \\ \boldsymbol{h}_{n-k+1} \\ \vdots \\ \boldsymbol{h}_{n-1} \end{pmatrix}. \tag{15}$$

各 $\tilde{\mathrm{H}}$ に対して，定数倍だけし行列の要素の大きさを操作することをすぐに考えつくが，直交してしまっているので，この行列の作り方に対しては要素の大きさを変化させても計算結果に変化はない．

2.5 Bezout 行列+Hankel 行列を利用

Bezout 行列および Hankel 行列をそれぞれ利用する場合，Sylvester 行列の場合にみられた工夫をすることは難しい．そこで，Bezout 行列と Hankel 行列を利用して計算できないかを考える．

次の関係式から近似 GCD を計算する．

$$\left(\begin{array}{ccc} \tilde{\mathrm{H}} & & \\ & \tilde{\mathrm{H}} & \\ & & \ddots \\ & & \tilde{\mathrm{H}} \\ \hline \tilde{\mathrm{B}} & & \\ & \tilde{\mathrm{B}} & \\ & & \ddots \\ & & \tilde{\mathrm{B}} \\ \hline f_m & \cdots & f_{m-k+1} \quad -1 \end{array}\right) \left(\begin{array}{c} d_0 \\ d_1 \\ \vdots \\ d_{k-1} \\ \hline c_0 \\ c_1 \\ \vdots \\ c_{k-1} \\ \hline 0 \end{array}\right) = \left(\begin{array}{c} h_{n-k} \\ h_{n-k+1} \\ \vdots \\ h_{n-1} \\ \hline b_0 \\ b_1 \\ \vdots \\ b_{k-1} \\ \hline 0 \end{array}\right). \tag{16}$$

さらに，次のように重みをつける．

$$\left(\begin{array}{ccc} w_H \times \left(\begin{array}{ccc} \tilde{\mathrm{H}} & & \\ & \tilde{\mathrm{H}} & \\ & & \ddots \\ & & \tilde{\mathrm{H}} \end{array}\right) \\ \hline w_B \times \left(\begin{array}{ccc} \tilde{\mathrm{B}} & & \\ & \tilde{\mathrm{B}} & \\ & & \ddots \\ & & \tilde{\mathrm{B}} \end{array}\right) \\ \hline f_m \quad \cdots \quad f_{m-k+1} \quad -1 \end{array}\right) \left(\begin{array}{c} d_0 \\ d_1 \\ \vdots \\ d_{k-1} \\ \hline c_0 \\ c_1 \\ \vdots \\ c_{k-1} \\ \hline 0 \end{array}\right) = \left(\begin{array}{c} w_H \times \left(\begin{array}{c} h_{n-k} \\ h_{n-k+1} \\ \vdots \\ h_{n-1} \end{array}\right) \\ \hline c_B \times \left(\begin{array}{c} b_0 \\ b_1 \\ \vdots \\ b_{k-1} \end{array}\right) \\ \hline 0 \end{array}\right). \tag{17}$$

ここで，c_H, c_B は整数である．一方は Sylvester 行列の場合と同様に大きくすればよいがもう一方は 1 としても問題はない．

3 多変数多項式近似 GCD

3.1 Bezout 構成

与えられた多変数多項式 $F(x, \boldsymbol{u})$ と $G(x, \boldsymbol{u})$ について，この与多項式の (変数 x に関する)Bezout 多項式を 1 変数の場合と同様に次のように定義する．

$$\mathrm{Bpol}(F(x,\boldsymbol{u}), G(x,\boldsymbol{u})) = \frac{F(x,\boldsymbol{u})G(y,\boldsymbol{u}) - F(y,\boldsymbol{u})G(x,\boldsymbol{u})}{x-y} = \sum_{0 \le i,j \le n-1} b_{i,j}(\boldsymbol{u}) x^i y^j \in \mathbb{K}[x,y,\boldsymbol{u}]. \tag{18}$$

また，Bezout 行列も同様に定義する．

$$\begin{aligned} \mathrm{B}(F,G) &= \left(b_{i,j}(\boldsymbol{u})\right)_{i,j} = \left(\boldsymbol{b}_0(\boldsymbol{u}), \boldsymbol{b}_1(\boldsymbol{u}), \ldots, \boldsymbol{b}_{n-1}(\boldsymbol{u})\right) \in \mathbb{K}[\boldsymbol{u}]^{n \times n}, & (19) \\ &= \mathrm{B}^{(0)} + \delta\mathrm{B}^{(1)} + \cdots + \delta\mathrm{B}^{(w)} + \cdots. & (20) \end{aligned}$$

ここで，$\delta \mathrm{B}^{(w)} = (\delta \boldsymbol{b}_0^{(w)}, \ldots, \delta \boldsymbol{b}_{n-1}^{(w)}) \in \mathbb{K}[\boldsymbol{u}]^{n \times n}$ は Bezout 行列の各要素について従変数 $\boldsymbol{u}$ に関する全次数 w の斉次多項式部分の項のみからなる行列である; $\boldsymbol{b}_i(\boldsymbol{u}) = \boldsymbol{b}_i^{(0)} + \delta \boldsymbol{b}_i^{(1)}(\boldsymbol{u}) + \cdots + \delta \boldsymbol{b}_i^{(w)}(\boldsymbol{u}) + \cdots$.

1 変数多項式の場合から，多変数多項式について次の拡張が可能である [Sanuki09].

$$\begin{aligned}(\boldsymbol{b}_k \cdots \boldsymbol{b}_{n-1})\boldsymbol{c}_i(\boldsymbol{u}) &= \boldsymbol{b}_i(\boldsymbol{u}) \\ \tilde{\mathrm{B}}\boldsymbol{c}_i(\boldsymbol{u}) &= \boldsymbol{b}_i(\boldsymbol{u}).\end{aligned}$$

ここで、$\boldsymbol{c}_i(u) = \left(\frac{c_i(\boldsymbol{u})}{c_k(\boldsymbol{u})}, \cdots\right) \in \mathbb{K}(\boldsymbol{u})^{n-k}$ である．ここで，$\boldsymbol{s} \in \mathbb{K}^\ell$ を $c_k(\boldsymbol{s}) \neq 0$ をみたすようにとり，イデアル I を $I = \langle \boldsymbol{u} - \boldsymbol{s} \rangle$ と定める．このとき次が成り立つ $(0 \leq i \leq k-1)$.

$$\tilde{\mathrm{B}}\boldsymbol{c}_i(\boldsymbol{u}) \equiv \boldsymbol{b}_i(\boldsymbol{u}) \pmod{I^{w+1}}, \tag{21}$$

$$\tilde{\mathrm{B}}^{(0)}\delta \boldsymbol{c}_i^{(w)} + \sum_{i=1}^{w} \delta \tilde{\mathrm{B}}^{(i)} \delta \boldsymbol{c}_i^{(w-i)} = \delta \boldsymbol{b}_i^{(w)}(\boldsymbol{u}). \tag{22}$$

$j = 0, \ldots, w-1$ 次まで $\delta \boldsymbol{c}_i^{(j)}$ が計算できたと仮定する．このとき，$\delta \boldsymbol{c}_i^{(w)}$ は次の線形方程式を解くことによって計算ができる．

$$\tilde{\mathrm{B}}^{(0)}\delta \boldsymbol{c}_i^{(w)} = \delta \boldsymbol{b}_i^{(w)}(\boldsymbol{u}) + \sum_{i=1}^{w} \delta \tilde{\mathrm{B}}^{(i)} \delta \boldsymbol{c}_i^{(w-i)}. \tag{23}$$

3.2 整数係数 Bezout 構成

LLL 法は整数を要素とするベクトル (基底) についてのみ適応ができる．LLL 法を Bezout 構成に適応するために，行列を次のように分解する．

$$\delta \tilde{\mathrm{B}}^{(w)} = \sum_{i_1 + \cdots + i_\ell = w} \delta \tilde{\mathrm{B}}_{i_1, \ldots, i_\ell}^{(w)} u_1^{i_1} \cdots u_\ell^{i_\ell}.$$

ここで，$\delta \tilde{\mathrm{B}}_{i_1, \ldots, i_\ell}^{(w)} \in \mathbb{K}^{n \times (n-k)}$ である．この分解によって，各 $u_1^{i_1} \cdots u_\ell^{i_\ell}$ の係数から数値ベクトル，数値行列をでき，$\delta \boldsymbol{c}^{(w)}$ の各項を構成することができる．線形方程式 (23) を解くとき，それぞれについて LLL 法を適応する必要はないため，効率の悪い方法ではない．ネックとなるのは，近似 CVP を解くために計算の精度が出ないことである．

4 まとめ

本稿では，Sylvester 行列の代わりに，Bezout 行列および Hankel 行列を用いて整数係数近似 GCD が計算できるのか，これまでのアイデアを組み合わせて数値実験を行った．その結果，

- Sylvester 行列を用いる方法は SVP に帰着できるが，Bezout 行列および Hankel 行列を用いる方法は CVP に帰着される．LLL 法など近似解法を用いて計算する場合，SVP の方が精度よく計算できるので，工夫なしでは Sylvester 行列を用いる方法が期待した計算結果を出力することが多い．
- Bezout 行列および Hankel 行列を用いて計算したい場合，Barnett の定理で得られる関係式より，新たに行列 (基底) を作成する必要がある．基底をなす各ベクトルが長くなるが，基底の数は変わらないので，簡約基底の計算時間は Sylverter 行列の場合に比べて早い．

そのため，Bezout 行列および Henkel 行列を用いる方法の方が優れているように思える．しかし，

- Sylveter 行列による方法は SVP に帰着するが，Bezout 行列および Henkel 行列を用いる方法は CVP に帰着される．近似解法においては CVP より SVP の方が精度良く解ける (同等の精度で解けるか (近似 CVP から近似 SVP への帰着) は未解決)[Cohen93].

それゆえ，LLL 算法を用いる場合に，摂動の小さくない多項式の近似 GCD を計算するときは SVP に帰着する必要がある．また，反復解法などで精度をよくする方法を開発する必要がある．

参 考 文 献

[Barnett70] S. Barnett. *Greatest common divisor of two polynomials.* Linear Algebra Appl., **3**, 1970, 7–9.

[Barnett71] S. Barnett. *Greatest common divisor of several polynomials.* Proc. Camb. Phil. Soc., **70**, 1971, 263–268.

[BP94] D. Bini and V. Pan. *Polynomial and matrix computations: volume 1 fundamental algorithms.* Birkhäuser, 1994.

[Cohen93] H. Cohen. *A course in computational algebraic number theory*, GTM 138, Splinger-Verlag, 1993.

[DG02] G. M. Diaz-Toca and L. Gonzalez-Vega. *Barnett's theorems about the greatest common divisor of several univariate polynomials though Bézout-like matrices. J. Symb. Compu.*, **34**, (2002), 59–81.

[GS08] J. von zur Gathen and I. E. Shparlinski. *Approximate Polynomial gcd: Small Degree and Small Height Perturbations.* Lecture Notes in Computer Science 4957, Springer, 2008, 276–283.

[長坂 07] 長坂耕作，整数係数多項式の近似 GCD II，京都大学数理解析研究所講球録 1572，50–58, 2007.

[Nagasaka08] K. Nagasaka. *Approximate polynomial GCD over integers.* ACM Commu. in Compu. Alge. (ISSAC 2008 poster abstract). **42(3)**, 2008, 124–126.

[長坂 10] 長坂耕作，準同型暗号と整数及び整数多項式の近似 *GCD*，本講球録に収録, 京都大学数理解析研究所講球録，2010.

[SO09] T. Sasaki and Y. Ookura. *Approximate factorization of polynomials over* $\mathbb{Z}$. Proc. of SNC 2009, ACM Press, 2009, 169–176.

[Sanuki09] M. Sanuki. *Computing multivariate approximate GCD based on Barnett's theorem*, Proc. of Symbolic-Numeric Computation 2009 (SNC 2009), H. Sekigawa & H. Kai (Eds.), 2009, 149-157, Kyoto, Japan, 3-5 August 2009.

[讃岐 09] 讃岐勝，ベズー多項式による前処理付き近似 *GCD* 計算, 数式処理, 日本数式処理学会, 16(2) 2009.

[讃岐 10] 讃岐勝，多変数近似 *GCD* の最適化と最小許容度見積もりに向けて，第 19 回日本数式処理大会，名古屋大学，2010.

[Hoeji02] M. van Hoeji. *Factoring polynomials and the Knapsack problem.* J. Number Theory, 95(2), 167–189, 2002.