

多変数多項式の近似 GCD

Approximate GCD of Multivariate Polynomials

神戸大学 大学院 人間発達環境学研究科 長坂 耕作^{*1}

KOSAKU NAGASAKA

GRADUATE SCHOOL OF HUMAN DEVELOPMENT AND ENVIRONMENT, KOBE UNIVERSITY

Abstract

In this talk, we briefly introduced our new results from numerical experiments for finding the approximate GCD of multivariate polynomials with large degrees. These results have been obtained using our recent algorithm, which is based on SLRA interpolation. This algorithm has been slightly improved and implemented in C.

1 はじめに

本報告では、昨年度に提案したアルゴリズム [長 23] (主に密な二変数多項式を対象にその近似 GCD を求めるもの) を、主に疎な多変数多項式を対象として再検討を行ったアルゴリズム [Nag23] について、いくつかの微改訂と新しい実験を行った結果について取り扱います。

1.1 近似 GCD の定義と記法

K を実数体 $\mathbb{R}$ ないしは複素数体 C とし, $K[\vec{x}]$ を K 上の多項式環とします。つまり, その多項式 $f(\vec{x})$ は, $f(\vec{x}) = \sum_{i=0}^t c_i x_1^{e_{i,1}} \cdots x_\ell^{e_{i,\ell}}$ ($c_i \neq 0$) のように書き下せることになります。このとき, その全次数を $\text{tdeg}(f) = \max_i (e_{i,1} + \cdots + e_{i,\ell}) \in \mathbb{N}$ で表し, 変数毎の次数を $\text{rdeg}(f) = (\max_i e_{i,1}, \dots, \max_i e_{i,\ell}) \in \mathbb{N}^\ell$ で表します。特に区別する必要がない場合や両方それぞれを示したい場合は, まとめて $\text{deg}(f)$ と記載することとします。多項式のノルムは 2 ノルムを用い, $\|f\|_2 = \sqrt{\sum |c_i|^2}$ と表すこととします。本報告では疎な多項式も扱うため, 多項式 $f(\vec{x})$ の台 (非零係数をもつ $f(\vec{x})$ の項の集合 (係数 1)) を $\text{supp}(f)$ で表すことにもしておきます。以上の準備の元, 本稿で扱う近似 GCD の定義 (2 種類) を与えます。

定義 1 (近似 GCD (指定次数に対する摂動量最小化問題, 密な場合))

多項式集合 $F = \{f_1, \dots, f_m\} \subset K[\vec{x}]$ ($m \in \mathbb{N}$) と次数 $k \in \mathbb{N}$ (変数毎次数の場合 $\vec{k} \in \mathbb{N}^\ell$, 以下同様) に対して, 以下を満たす次数 k ($\vec{k}$) の多項式 $g(\vec{x}) \in K[\vec{x}]$ を, F の次数 k ($\vec{k}$) の “近似 GCD” と定義する。

$$f_i(\vec{x}) + \Delta_i(\vec{x}) = g(\vec{x})h_i(\vec{x}), \text{tdeg}(g) = k \text{ (または } \text{rdeg}(g) = \vec{k})$$

ここで, $\Delta_i(\vec{x}), h_i(\vec{x}) \in K[\vec{x}]$ は, $\sum_{i=1}^m \|\Delta_i\|_2^2$ (摂動 (perturbation) と呼ぶ) を最小化し, $\text{deg}(\Delta_i) \leq \text{deg}(f_i)$ を満たすものとする (変数毎次数の場合の不等式条件は変数毎)。

◁

^{*1} E-mail: nagasaka@main.h.kobe-u.ac.jp

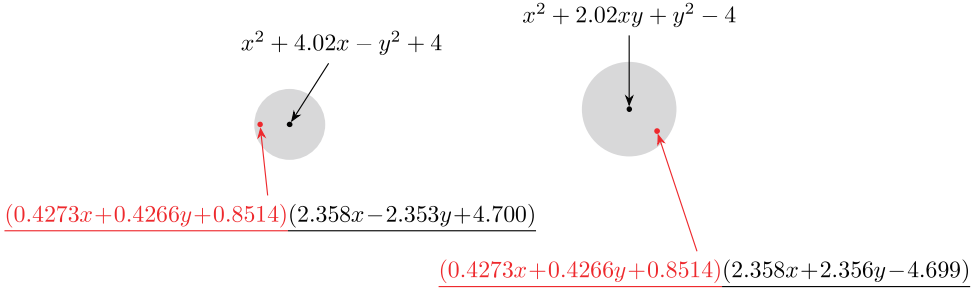


図 1: 近似 GCD の定義に基づく図解

定義 2 (近似 GCD (指定次数に対する摂動量最小化問題, 疎な場合))

多項式集合 $F = \{f_1, \dots, f_m\} \subset K[\vec{x}]$ ($m \in \mathbb{N}$) と次数 $k \in \mathbb{N}$ (変数毎次数の場合 $\vec{k} \in \mathbb{N}^\ell$, 以下同様) に対して, 以下を満たす次数 $k(\vec{k})$ の多項式 $g(\vec{x}) \in K[\vec{x}]$ を, F の次数 $k(\vec{k})$ の “**weakly support-preserved approximate GCD**” と定義する。

$$f_i(\vec{x}) + \Delta_i(\vec{x}) + \tilde{\Delta}_i(\vec{x}) = g(\vec{x})h_i(\vec{x}), \deg(g) = k \quad (\text{または } \vec{k})$$

ここで, $\Delta_i(\vec{x}), \tilde{\Delta}_i(\vec{x}), h_i(\vec{x}) \in K[\vec{x}]$ は, $\sum_{i=1}^m \|\Delta_i + \tilde{\Delta}_i\|_2^2$ (摂動 (perturbation)) と $\sum_{i=1}^m \|\tilde{\Delta}_i\|_2^2$ (逸脱摂動 (**deviant perturbation**)) を最小化し, $\text{supp}(\Delta_i) \subseteq \text{supp}(f_i)$, $\deg(\tilde{\Delta}_i) \leq \deg(f_i)$ かつ $\text{supp}(\tilde{\Delta}_i) \cap \text{supp}(f_i) = \phi$ を満たすものとする (変数毎次数の場合の不等式条件は変数毎)。また, $\tilde{\Delta}_i(\vec{x}) = 0$ ($i = 1, \dots, m$) ならば $g(\vec{x})$ を “**support-preserved approximate GCD**” という。◀

図 1 は, 近似 GCD の定義 (定義 1 と定義 2) に基づいて, その趣旨を図解したものです。図において, 黒い点は問題の入力 (係数に誤差を含む多項式) を表しており, 赤い点が問題の出力 (互いに素でない入力の多項式に近接の多項式) を表しています。灰色の円盤は, 誤差を考慮した結果として想定される未知の多項式 (互いに素でない) が存在する範囲を示します。

2 SLRA Interpolation に基づく近似 GCD アルゴリズム

本報告で扱う SLRA Interpolation に基づく近似 GCD アルゴリズムの概要について紹介します (詳細については割愛しますので, 必要に応じて昨年度の講義録 [長 23] ないしは論文 [Nag23] をご参照ください)。

2.1 複数多項式は 2 つの多項式に集約

入力の多項式集合 F に 2 つを超える多項式が含まれている場合, $F = F_a \cup F_b$ かつ $F_a \cap F_b = \phi$ を満たすように, $F_a = \{f_{a,1}, \dots, f_{a,m_a}\}$ と $F_b = \{f_{b,1}, \dots, f_{b,m_b}\}$ に分割します。その上で $f_a(\vec{x}, x_{\ell+1})$ と $f_b(\vec{x}, x_{\ell+1})$ を次式で定義します。

$$f_a(\vec{x}, x_{\ell+1}) = \sum_{i=1}^{m_a} f_{a,i}(\vec{x}) x_{\ell+1}^{i-1}, \quad f_b(\vec{x}, x_{\ell+1}) = \sum_{i=1}^{m_b} f_{b,i}(\vec{x}) x_{\ell+1}^{i-1}$$

ここで, $x_{\ell+1}$ は新たに導入するスラック変数のようなもので, $m_a = 1$ であるか $m_b = 1$ であれば, $\gcd(f_1, \dots, f_m) = \gcd(f_a, f_b)$ が成立しますので, 以後, $F = \{f_a, f_b\} \subset K[\vec{x}]$ を入力の多項式集合として扱っていきます。

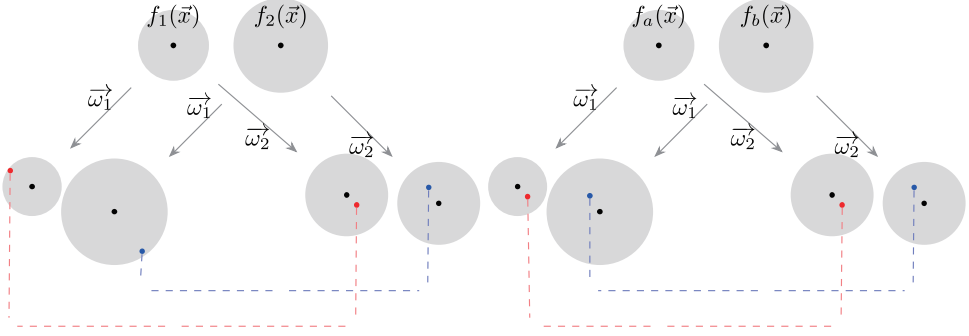


図 2: 多変数多項式の近似 GCD アルゴリズムの枠組み (左図: 通常, 右図: 本稿方式)

2.2 アルゴリズムの枠組みと SLRA

本報告で扱う近似 GCD アルゴリズムの基本的な枠組みは補間法になります。すなわち、まず、入力の変数多項式を複数の点で評価（数値を代入）し、複数の点の点での一変数多項式（射影した結果の像）を求めます。それぞれの点での一変数多項式から、それらの近似 GCD（一変数多項式）を求め、結果から多変数多項式としての近似 GCD を補間法で復元します。図 2 において、 $\vec{\omega}_1$ と $\vec{\omega}_2$ は異なる評価点を表しています。

一般的には、図 2 の左側の図にあるように、同一の多項式の像は、一変数多項式の近似 GCD 計算の際に、連動せずバラバラに振動します。これは左側の図のように、赤の点と破線 ($\vec{\omega}_1$ における $f_a(\vec{x})$ の像の振動後の多項式) が、円盤内で必ずしも同じ方向でない方向に動くことを意味します。結果として、昨年度の報告 [長 23] にもありますように、補間法による復元では精度に問題が生じます。一方で、本報告で扱っているアルゴリズムでは、図 2 の右側の図における赤の点と破線のように、一変数多項式における近似 GCD の原像は同一で、補間法により復元することが可能です。

補間法に基づかない直接的な近似 GCD アルゴリズムでは、次式で定義される一般化 Sylvester 部分終結式行列などの最近接階数落ち行列を求めることにより、近似 GCD を求めることが出来ます（本報告で扱うアルゴリズムでは、2 つの多項式に集約していますが、複数多項式向けの式を提示しておきます）。

$$\text{Syl}^k(F) = \begin{pmatrix} C^{d_2-k}(f_1) & & C^{d_1-k}(f_2) \\ & C^{d_3-k}(f_1) & C^{d_1-k}(f_3) \\ & & \ddots & \vdots \\ & & C^{d_m-k}(f_1) & C^{d_1-k}(f_m) \end{pmatrix}$$

ここで、 $C^d(f)$ は畳み込み行列を表しており、 $d = \deg(h)$ なる $f(\vec{x})h(\vec{x})$ に相応する計算を $C^d(f)\vec{h}$ で表せます。このとき、よく知られている事実として、「 $\deg(\gcd(f_1, \dots, f_m))$ が $k \in \mathbb{N}$ であること」の必要十分条件は、「 $\text{Syl}^k(F)$ が 1 のみ階数落ちすること」というのがあります。本報告で扱うアルゴリズムも、この事実に基づいています。最近接階数落ち行列を求める方法はいくつか知られていますが、本報告で扱うアルゴリズムでは、以下で定義する SLRA を活用しています。

定義 3 (SLRA(Structured Low Rank Approximation))

$S \subset K^{p \times q}$ を、 $p \times q$ のサイズを持つ行列の集合 $K^{p \times q}$ の affine 部分空間とする（この部分空間は、行列の構造を定義する）。このとき、 $M \in S$ と $r \in \mathbb{N}$ に対し、 $\|M - M^*\|_F$ を“小さく”する $M^* \in S \cap K_r^{p \times q}$ を求める問題を **Structured Low Rank Approximation** という。ここで、 $K_r^{p \times q}$ は階数 r を持つ行列の部分集合 $\subset K^{p \times q}$ を表し、 $\|M\|_F$ は $M \in K^{p \times q}$ の Frobenius ノルムを表すものとする。

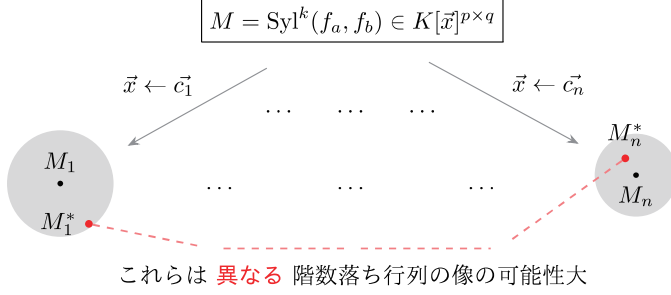


図 3: 非同期的に射影先での SLRA を複数独立に解く場合

2.3 SLRA Interpolation とそれに基づく摂動

前節で導入した一般化 Sylvester 部分終結式行列などを用いて、射影先（一変数多項式）の近似 GCD を SLRA により解いたとしても、それらは非同期的に摂動するため、精度の良い近似 GCD を復元することが出来ません。本節では、同期的に摂動させる SLRA Interpolation について説明します。以下では、入力が多項式 $f_a(x, \vec{x}), f_b(x, \vec{x}) \in K[\vec{x}][x] = K[x_1, x_2, \dots, x_{\ell+1}]$ に対して、その評価点（代入点）を、 $\vec{c}_1, \dots, \vec{c}_n \in \mathbb{C}^\ell$ （ある $n \in \mathbb{N}$ ）とします。これらは、FFT や修正数値的 Ben-Or/Tiwari に基づくアルゴリズム [LYZ10, GLL09, GLL06] などと同様に取れば、1 の p 乗根の累乗となります。

射影先（各評価点）での一変数多項式に対する Sylvester 部分終結式行列を $M_1, \dots, M_n \in K^{p \times q}$ とします。すなわち、 $M_i = \text{Syl}^k(f_a(x, \vec{c}_i), f_b(x, \vec{c}_i))$ です。このとき、それぞれの SLRA において非同期的に摂動を行いますと、図 3 のように、1) $M_1^* \in \mathcal{S} \cap K_r^{p \times q}$ に対して、 $\|M_1 - M_1^*\|_F$ を“小さく”，2) $\dots$, n) $M_n^* \in \mathcal{S} \cap K_r^{p \times q}$ に対して、 $\|M_n - M_n^*\|_F$ を“小さく”，と独立に解くことになり、同一の原像を持つような摂動とはなりません。

本報告で扱っている新しい枠組みでは、基本的な制約 ($M_i^* \in \mathcal{S} \cap K_r^{p \times q}$ に対し、 $\|M_i - M_i^*\|_F$ を“小さく”)に加えて、「 $\text{blkdiag}(M_1^*, \dots, M_n^*) \in \text{blkdiag}(\mathcal{S}, \dots, \mathcal{S}) \cap K_{nr}^{np \times nq}$ に対して、 $\|\text{blkdiag}(M_1, \dots, M_n) - \text{blkdiag}(M_1^*, \dots, M_n^*)\|_F$ を“小さく”する」という付加的な制約を課した大きな SLRA を 1 つだけ解きます (図 4)。ここで、 $\text{blkdiag}(M_1, \dots, M_n)$ は次式で定義されるブロック対角行列です。

$$\text{blkdiag}(M_1, \dots, M_n) = \begin{pmatrix} M_1 & & \\ & \ddots & \\ & & M_n \end{pmatrix}$$

なお、一連の議論に現れる行列のサイズを述べておきます。射影を行わない多変数多項式のままで一般化 Sylvester 部分終結式行列 $\text{Syl}^k(f_a, f_b)$ の大きさは、変数の個数に指数的なサイズとなります。一方、射影した先での M_i の大きさは、

$$(\deg(f_a) + \deg(f_b) - k + 1) \times (\deg(f_a) + \deg(f_b) - 2k + 2)$$

となり、上記のブロック対角行列 $\text{blkdiag}(M_1, \dots, M_n)$ の大きさは、入力の多項式が密な場合では、変数の個数に指数的なサイズとなります。このままでは、ブロック対角行列を扱うことにより計算量の激増が不可避となりますが、 M_i ($i = 1, \dots, n$) それぞれの SVD (特異値分解) を $M_i = U_i \Sigma_i V_i^H$ とすれば、 $\text{blkdiag}(M_1, \dots, M_n)$ の SVD は、次式で与えられる (残念ながら特異値は小さい順にはならない) ことから、計算量はある程度抑えることが可能です。

$$\text{blkdiag}(U_1, \dots, U_n) \text{blkdiag}(\Sigma_1, \dots, \Sigma_n) \text{blkdiag}(V_1, \dots, V_n)^H$$

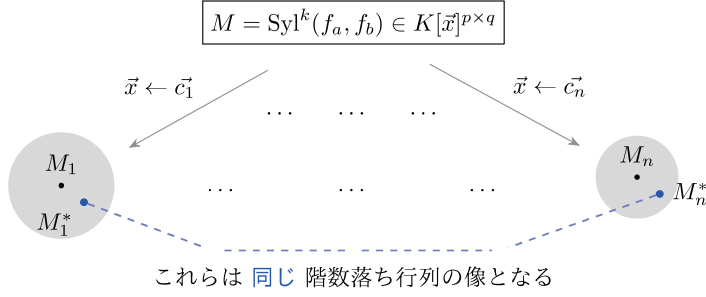


図 4: SLRA Interpolation による同期的な射影先で SLRA を解く場合

3 微改訂と新しい実験

一般化 Sylvester 部分終結式行列から近似 GCD による余因子を求める際や、最近接階数落ち行列を求める手法によっては最近接階数落ち行列を求める際に、次式を満たすベクトル $\vec{v}$ を計算します。

$$\min_{\|\vec{v}\|_2=1} \|\text{Syl}^k(F)\vec{v}\|_2$$

近似 GCD に関する論文等では、特異値分解などに基づく特異ベクトルを用いてベクトル $\vec{v}$ を計算することが多いです（逆に、この方法しか紹介されていないことが多い）。しかしながら、このベクトルは固有ベクトルでも代替することが可能です。すなわち、次の対称行列（エルミート行列かつ半正定値）

$$\text{Syl}^k(F)^H \text{Syl}^k(F)$$

の最小固有値に対応する固有ベクトルも一般に解と知られています（固有値は非負）。全ての固有値や固有ベクトルを計算する必要がないことから、特異値分解に基づく方法よりも効率的な計算が期待されます。

3.1 実験環境などの概要

全ての実験は、Ubuntu 22.04.3 LTS (256GB memory, Intel Xeon E5-2687W v4 (CPU: 12 cores, 24 threads, 3.0–3.5GHz)) を用いて実施しました。使用した言語は、Mathematica 12.0 および C 言語 (BLAS, LAPACK, FFTW) で、C 言語版は GCC 11.4.0 (-march=native -O2) でビルドし、各種ライブラリは、Ubuntu (22.04.3 LTS) の公式パッケージ版の OpenBLAS 0.3.20 (pthread version), FFTW 3.3.8, LAPACK/LAPACKe 3.10.0 (*syevr, *gesdd(*gesvdx), *gelsy) を用いています。

比較対象のアルゴリズム (SVD に基づくもの [GKM⁺04], FFT に基づくもの [LYZ10], GaussNewton に基づくもの [ZD04], STLN に基づくもの [KYZ06]) も含めて、Mathematica での実装における各種パラメータなどの設定は次のとおりです。反復を行うアルゴリズムでは、入出力時に多項式の正規化を行っています。ペナルティ法に準ずる手法での Penalty は、提案手法・SVD・FFT では $1.0\text{e}+2$ を、その他の方法では $1.0\text{e}+8$ を採用しています。反復を行うアルゴリズムでの停止条件は、Mathematica の組み込み関数の慣例に倣う形で、 $1.49\text{e}-8$ としています。C 言語版でも、入出力時に多項式の正規化を行っています。ペナルティ法に準ずる手法での Penalty は一律 $1.0\text{e}+2$ に、停止条件は一律 $1.0\text{e}-8$ としています。

実験に用いた多変数多項式は、疎な多項式の組 (Ex.A, Ex.B, Ex.C, Ex.D) です。これらの組は、1) GCD と余因子が各最大 6 個 (Ex.D は 12 個) の単項式を持つようにランダムに生成し、2) それぞれの係数は $[-1, 1] \subset \mathbb{R}$ の範囲からランダムに生成し、3) GCD と余因子の積を取った後の各項に約 $1.0\text{e}-4$ の絶対誤差

	Ex.A	Ex.B	Ex.C	Ex.D
$ F , \vec{x} , d, k, t_s$	3, 3, 10, 3, 34.7	4, 3, 10, 3, 34.3	3, 3, 20, 6, 36	4, 3, 30, 9, 142.5
全次数でのサイズ	$2, 280 \times 360$	$3, 420 \times 480$	$15, 540 \times 2, 040$	$74, 412 \times 8, 096$
変数毎次数でのサイズ	$11, 664 \times 1, 536$	$17, 496 \times 2, 048$	$85, 750 \times 10, 125$	$421, 824 \times 42, 592$

表 1: 実験に用いた行列サイズの目安

を付与したものです。表 1 に、これらの疎な多項式の組に対する一般化 Sylvester 部分終結式行列のサイズを掲載してあります。例えば、3 変数 4 多項式 30 次となる Ex.D でのサイズは「421, 824 × 42, 592」ですが、この密な行列としての表現には「133.9GB」のメモリが必要となります。

3.2 実験結果と考察

近似 GCD の次数を全次数で指定する場合と、変数毎の次数で指定する場合の 2 種類それぞれに対して、Mathematica を用いたものと、C 言語を用いたものそれぞれで近似 GCD の計算を行った結果が、表 2–表 5 です。各セル上段には、定義 2 の摂動の大きさを左に、逸脱摂動の大きさを右に記載しています。各セル下段には、計算時間（経過時間）を左に、収束までの反復回数を右に記載しています（下線を引いている場合、その回数を最大反復回数として実行し、その上限値に達したことを意味します）。C 言語版の場合は、それに加えて、下段に `time` 命令で計測したピークメモリ使用量も掲載しています。

表 2 を含め全ての実験において、明示的に「w/o ev」と記載がなければ、余因子ベクトルの計算は特異値分解ではなく、固有ベクトル計算を用いて行っています。ただし、表 3 では、STLN (Ex.A) は固有ベクトル計算を用いていません（原因不明ですが、その先の計算で Mathematica が躓くためです）。主に Mathematica と C 言語版との違いになりますが、表 4 と表 5 では、所感として、色々と計算精度が足りてない感じを受けます（`cexp` を `cexpl` に変更することで微改善するも、補間点を増やしても Mathematica に劣る結果を確認しています）。なお、評価点の累乗計算は、Mathematica では累乗、C 言語では `cexp` を用いて行っています。

参 考 文 献

[GKM⁺04] Shuhong Gao, Erich Kaltofen, John May, Zhengfeng Yang, and Lihong Zhi. Approximate factorization of multivariate polynomials via differential equations. In *ISSAC 2004*, pages 167–174. ACM, New York, 2004.

[GLL06] Mark Giesbrecht, George Labahn, and Wen-shin Lee. Symbolic-numeric sparse interpolation of multivariate polynomials. In *ISSAC 2006*, pages 116–123. ACM, New York, 2006.

[GLL09] Mark Giesbrecht, George Labahn, and Wen-shin Lee. Symbolic-numeric sparse interpolation of multivariate polynomials. *J. Symbolic Comput.*, 44(8):943–959, 2009.

[KYZ06] Erich Kaltofen, Zhengfeng Yang, and Lihong Zhi. Approximate greatest common divisors of several polynomials with linearly constrained coefficients and singular polynomials. In *ISSAC 2006*, pages 169–176. ACM, New York, 2006.

	Ex.A	Ex.B	Ex.C	Ex.D
$ F , \vec{x} , d, k, t_s$	3, 3, 10, 3, 34.7	4, 3, 10, 3, 34.3	3, 3, 20, 6, 36	4, 3, 30, 9, 142.5
SVD (w/o ev)	1.620e-7, 5.400e-8 (1.6222e+0, -)	3.425e-7, 1.008e-7 (2.4523e+0, -)	1.866e-7, 6.852e-8 (5.2814e+1, -)	1.185e-6, 4.454e-7 (1.2462e+3, -)
SVD (GKMYZ'04)	1.620e-7, 5.400e-8 (1.6426e+0, -)	3.425e-7, 1.008e-7 (2.4887e+0, -)	1.866e-7, 6.852e-8 (4.9684e+1, -)	1.185e-6, 4.454e-7 (7.0947e+2, -)
FFT (LYZ'10)	5.885e-7, 2.271e-7 (1.1601e+0, -)	3.919e+0, 5.445e-3 (1.3371e+0, -)	5.055e-7, 2.074e-7 (2.0793e+1, -)	7.546e-1, 4.150e-3 (1.8134e+2, -)
GaussNewton (ZD'04)	2.284e-7, 2.78e-39 (4.0446e+0, 5)	4.427e-7, 5.23e-39 (6.1371e+0, 6)	2.374e-7, 4.56e-36 (1.0909e+2, 5)	1.632e-6, 1.22e-38 (1.9615e+3, 6)
STLN (KYZ'06)	2.284e-7, 9.10e-31 (3.8682e+1, 5)	4.427e-7, 2.23e-31 (72.392e+0, 6)	2.374e-7, 2.08e-30 (2.0035e+3, 6)	- (out of memory, -)
New (SVD)	2.281e-7, 1.207e-9 (2.2999e+0, 4)	3.897e-7, 3.111e-8 (4.0504e+0, 4)	2.676e-7, 7.518e-9 (5.2709e+1, 3)	1.539e-6, 8.143e-8 (7.3698e+2, 4)
New (FFT)	2.282e-7, 1.081e-9 (1.8378e+0, 4)	8.190e-1, 1.024e-3 (2.8838e+0, 4)	3.149e-7, 2.691e-9 (2.2566e+1, 3)	2.658e-6, 7.843e-7 (2.2122e+2, 4)
New (x3, SVD)	2.217e-7, 3.419e-9 (3.7240e+0, 4)	3.769e-7, 3.435e-8 (8.2748e+0, 4)	2.557e-7, 3.46e-17 (5.4106e+1, 3)	1.509e-6, 1.049e-7 (8.0803e+2, 4)
New (x3, FFT)	2.340e-7, 3.062e-9 (3.2528e+0, 4)	1.228e-2, 7.864e-5 (7.0767e+0, 4)	3.849e-7, 3.07e-15 (2.2676e+1, 3)	2.948e-6, 9.640e-7 (2.8640e+2, 4)

表 2: 全次数 (疎な場合, Mathematica)

	Ex.A	Ex.B	Ex.C	Ex.D
$ F , \vec{x} , d, k, t_s$	3, 3, 10, 3, 34.7	4, 3, 10, 3, 34.3	3, 3, 20, 6, 36	4, 3, 30, 9, 142.5
SVD (w/o ev)	1.283e-7, 5.609e-8 (4.4859e+0, -)	3.059e-7, 1.409e-7 (1.0219e+1, -)	2.035e-7, 8.892e-8 (9.7887e+2, -)	- (out of memory, -)
SVD (GKMYZ'04)	1.283e-7, 5.609e-8 (3.2163e+0, -)	3.059e-7, 1.409e-7 (5.6467e+0, -)	2.035e-7, 8.892e-8 (2.1378e+2, -)	- (out of memory, -)
FFT (LYZ'10)	1.098e-6, 4.786e-7 (1.7126e+0, -)	4.345e+0, 5.049e-3 (2.0785e+0, -)	8.553e-7, 4.004e-7 (8.2638e+1, -)	2.850e+0, 4.951e-3 (2.1797e+3, -)
GaussNewton (ZD'04)	2.284e-7, 6.38e-39 (1.1433e+1, 6)	4.427e-7, 1.16e-38 (2.0012e+1, 6)	2.374e-7, 3.63e-26 (1.0784e+3, 4)	- (out of memory, -)
STLN (KYZ'06)	2.284e-7, 6.64e-30 (3.5653e+2, 5)	4.427e-7, 8.74e-30 (1.0306e+3, 6)	- (> 3.600e+3, -)	- (out of memory, -)
New (SVD)	2.273e-7, 1.210e-9 (3.8591e+0, 4)	3.751e-7, 3.655e-8 (7.2637e+0, 4)	2.694e-7, 1.003e-8 (2.1572e+2, 3)	- (out of memory, -)
New (FFT)	2.306e-7, 2.688e-9 (2.3883e+0, 4)	2.070e+0, 1.132e-3 (3.5819e+0, 4)	3.177e-7, 6.109e-9 (8.0318e+1, 3)	5.030e-6, 1.779e-6 (2.2420e+3, 4)
New (x3, SVD)	2.184e-7, 3.673e-9 (5.2514e+0, 4)	3.581e-7, 3.902e-8 (1.1393e+1, 4)	2.557e-7, 3.66e-17 (2.1925e+2, 3)	- (out of memory, -)
New (x3, FFT)	2.389e-7, 6.408e-9 (3.7919e+0, 4)	8.634e-2, 1.300e-4 (7.7194e+0, 4)	3.849e-7, 3.22e-15 (8.0576e+1, 3)	6.417e-6, 2.30e-6 (2.3055e+3, 4)

表 3: 変数毎の次数 (疎な場合, Mathematica)

	Ex.A	Ex.B	Ex.C	Ex.D
$ F , \vec{x} , d, k, t_s$	3, 3, 10, 3, 34.7	4, 3, 10, 3, 34.3	3, 3, 20, 6, 36	4, 3, 30, 9, 142.5
SVD (GKMYZ'04)	1.759e-7, 5.388e-8 (3.0426e-2, -)	4.024e-7, 1.007e-7 (5.2238e-2, -)	2.413e-7, 6.837e-8 (8.8446e-1, -)	1.486e-6, 4.452e-7 (4.089e+1, -)
FFT (LYZ'10)	5.793e-7, 1.496e-7 (1.3733e-1, -)	1.054e+1, 2.827e-4 (1.3885e-1, -)	9.862e-7, 1.924e-7 (1.8107e+0, -)	1.007e+1, 1.529e-3 (2.2829e+1, -)
GaussNewton (ZD'04)	2.283e-7, 2.77e-15 (1.296e-1, 4, 25M)	4.426e-7, 5.22e-15 (2.145e-1, 4, 36M)	2.373e-7, 2.13e-15 (8.618e+0, 4, 482M)	1.632e-6, 1.22e-14 (6.271e+2, 4, 8.1G)
New (SVD)	1.746e-7, 5.315e-8 (4.660e-2, 4, 24M)	3.963e-7, 9.315e-8 (5.28e-1, 80, 32M)	2.403e-7, 6.764e-8 (9.510e-1, 6, 357M)	1.483e-6, 4.433e-7 (4.097e+1, 3, 6.1G)
New (FFT)	6.289e-7, 1.642e-7 (1.202e-1, 4, 45M)	1.707e+1, 1.821e-5 (5.87e-1, 80, 45M)	5.007e-7, 1.032e-7 (1.804e+0, 6, 672M)	1.024e+1, 1.534e-3 (2.287e+1, 3, 4.4G)
New (svdx,e-15,SVD)	1.733e-7, 5.130e-8 (1.063e-1, <u>8</u> , 23M)	3.744e-7, 8.559e-8 (1.728e-1, <u>8</u> , 32M)	2.077e-7, 4.203e-8 (1.085e+0, <u>8</u> , 356M)	1.459e-6, 4.272e-7 (4.187e+1, <u>8</u> , 6.1G)
New (svdx,e-15,FFT)	4.274e-7, 9.510e-8 (1.787e-1, <u>8</u> , 45M)	9.018e+0, 3.224e-4 (2.247e-1, <u>8</u> , 45M)	3.353e-7, 5.619e-8 (1.940e+0, <u>8</u> , 671M)	5.568e+0, 1.318e-3 (2.393e+1, <u>8</u> , 4.4G)

表 4: 全次数 (疎な場合, C 言語)

	Ex.A	Ex.B	Ex.C	Ex.D
$ F , \vec{x} , d, k, t_s$	3, 3, 10, 3, 34.7	4, 3, 10, 3, 34.3	3, 3, 20, 6, 36	4, 3, 30, 9, 142.5
SVD (GKMYZ'04)	1.376e-7, 5.590e-8 (4.6337e-1, -)	4.177e-7, 1.409e-7 (9.3380e-1, -)	3.026e-7, 8.859e-8 (7.5514e+1, -)	6.592e+1, 4.55e-10 (6.6355e+3, -)
FFT (LYZ'10)	1.398e-6, 3.433e-7 (2.8189e-1, -)	1.046e+1, 2.808e-4 (3.2588e-1, -)	2.435e-6, 3.504e-7 (2.8340e+1, -)	1.407e+1, 1.530e-3 (1.2969e+3, -)
GaussNewton (ZD'04)	2.283e-7, 6.37e-15 (3.372e+0, 4, 282M)	4.425e-7, 1.15e-14 (8.510e+0, 4, 511M)	2.373e-7, 2.31e-15 (1.360e+3, 4, 12G)	- (> 1.08e+4, -)
New (SVD)	1.363e-7, 5.480e-8 (4.905e-1, 4, 209M)	4.867e-7, 1.604e-7 (1.43e+0, 80, 387M)	2.963e-7, 8.638e-8 (7.575e+1, 6, 8.9G)	6.592e+1, 6.17e-10 (6.718e+3, 3, 53G)
New (FFT)	1.582e-6, 3.736e-7 (2.680e-1, 4, 53M)	1.695e+1, 2.400e-5 (7.82e-1, 80, 52M)	1.101e-6, 2.098e-7 (2.809e+1, 6, 1.1G)	1.426e+1, 1.533e-3 (1.299e+3, 3, 9.2G)
New (svdx,e-15,SVD)	1.327e-7, 5.172e-8 (5.462e-1, <u>8</u> , 209M)	3.680e-7, 1.146e-7 (1.078e+0, <u>8</u> , 386M)	2.247e-7, 5.006e-8 (7.611e+1, <u>8</u> , 8.9G)	6.592e+1, 3.46e-10 (7.157e+3, <u>8</u> , 53G)
New (svdx,e-15,FFT)	8.223e-7, 2.272e-7 (3.304e-1, <u>8</u> , 52M)	9.012e+0, 3.159e-4 (4.226e-1, <u>8</u> , 52M)	4.969e-7, 1.040e-7 (2.879e+1, <u>8</u> , 1.1G)	8.805e+0, 1.407e-3 (1.365e+3, <u>8</u> , 9.2G)

表 5: 変数毎の次数 (疎な場合, C 言語)

- [LYZ10] Zijia Li, Zhengfeng Yang, and Lihong Zhi. Blind image deconvolution via fast approximate GCD. In *ISSAC 2010—Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation*, pages 155–162. ACM, New York, 2010.
- [Nag23] Kosaku Nagasaka. SLRA interpolation for approximate GCD of several multivariate polynomials. In *Proceedings of the International Symposium on Symbolic & Algebraic Computation (ISSAC 2023)*, pages 470–479. ACM, New York, 2023.
- [ZD04] Zhonggang Zeng and Barry H. Dayton. The approximate GCD of inexact polynomials. II. A multivariate algorithm. In *ISSAC 2004*, pages 320–327. ACM, New York, 2004.
- [長 23] 長坂耕作. 二変数多項式の近似 GCD. In 研究集会 *Computer Algebra – Foundations and Applications*, volume 2255 of 京都大学数理解析研究所講究録, pages 117–125. 京都大学数理解析研究所, 2023.