

近似 GCD アルゴリズムにおける枢軸選択の影響*

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

本報告では、3つの近似 GCD アルゴリズム (QRGCD[3], UVGCD[5], ExQRGCD[4]) で使用される行列分解を枢軸選択に対応させた場合、結果が改善されるかについて扱います。まず、枢軸選択を可能とする修正をアルゴリズムに行う方法を提案した後、近似代数演算を行う C 言語向けライブラリである LIBSNAP の実装を用いて、枢軸選択による改善効果があるかについて実験した結果を報告しています。

1.1 LIBSNAP について

LIBSNAP は、著者が開発している C 言語向けライブラリで、近似代数演算を気軽に使えるようにすることと、同じフレームワークでの近似代数アルゴリズムの比較を容易にすることを目的としており、2 条項 BSD ライセンスで配布しています。LIBSNAP の実装方針としては、比較的汎用のものを積極的に使用することにしており、数値計算部分は、BLAS と LAPACK に依存し、多倍長計算は、GMP, MPFR, MPC に依存するようになっています。また、C 言語 (に加えて C++) での利用を想定しており、C 言語用インターフェイスが存在しないライブラリについては、使用しないようにしています (結果として、多倍長数値計算の MPACK は使えていません)。報告時点での LIBSNAP の現状 (公開部分) としては、低位 API として、密な一変数多項式の演算 (supd.t, supz.t, supfr.t), 密な行列の基本操作 (sdmd.t, sdmz.t, sdmfr.t), BLAS と LAPACK の一部関数の多倍長版が実装され、高位 API として、QRGCD (double と mpfr 版), ExQRGCD (double と mpfr 版), UVGCD (double と mpfr 版) が実装されている段階です。

1.2 近似 GCD アルゴリズムの復習

本報告で扱う近似 GCD の問題設定とアルゴリズムを確認しておきます。

命題 1 (近似 GCD)

$f(x), g(x) \in \mathbb{R}[x]^1$ と $\varepsilon \in \mathbb{R}_{\geq 0}$ が与えられた (ただし, $\|f\|_2 = \|g\|_2 = 1$ とする) とき、次式を満たす $u(x), v(x), d(x) \in \mathbb{R}[x]$ を求めよ。このとき、 $d(x)$ を許容度 ε の近似 GCD という。

$$\max\{\|\Delta_f\|_2, \|\Delta_g\|_2\} < \varepsilon \text{ where } f + \Delta_f = d \cdot f_1, g + \Delta_g = d \cdot g_1 \text{ and } \|d\|_2 = 1$$

*本研究の一部は科研費 (22700011) の支援で行われています。

[†]nagasaka@main.h.kobe-u.ac.jp

¹⁾係数は、複素数体 $\mathbb{C}$ でも同じ議論となるため、簡単のため、実数体 $\mathbb{R}$ としています。

ここで、多項式の次数は、 $m = \deg(f)$, $n = \deg(g)$, $k = \deg(d)$ とします。 $f(x)$ と $g(x)$ の Sylvester 行列 (各行が係数ベクトルに対応) を, $\text{Syl}_{\rightarrow}(f, g)$ で表し, 各列が係数ベクトルに対応しているものを, $\text{Syl}_{\downarrow}(f, g)$ で表すことにします。また, 行列 S の QR 分解を $S = QR$ と書きます (Q は直交行列²⁾で R は上三角行列)。なお, 表記を簡易にするため, $m \geq n$ と仮定します。

アルゴリズム 1 (QRGCD by R.M.Corless, S.M.Watt and L.Zhi)

1. $\text{Syl}_{\rightarrow}(f, g)$ の QR 分解: $\text{Syl}_{\rightarrow}(f, g) = QR$ を計算
2. $\|R_{22}^{(k)}\|_2 > \varepsilon$ かつ $\|R_{22}^{(k-1)}\|_2 < \varepsilon$ を満たす k について
 - Case1:** $\|R_{22}^{(0)}\|_2 > \varepsilon$ ならば, 近似的にも互いに素と判定
 - Case2:** $\|R_{22}^{(k-1)}\|_2 / \|R_{22}^{(k)}\|_2 < 10\varepsilon$ ならば, $d(x) :=$ “the last k -th row of R ” と構成
 - Case3:** $\exists k_1, \|R_{22}^{(k_1-1)}\|_2 / \|R_{22}^{(k_1)}\|_2 < 10\varepsilon$ ならば, $d(x) :=$ “last k_1 -th row” と構成
 - Case4:** それ以外の場合はギャップの検出なしと考え, Split などの方法へ
3. 余因子の Reversal 多項式 ($h(x) \mapsto x^{\deg(h)} h(1/x)$) にも同じ処理

アルゴリズム 2 (ExQRGCD by K.Nagasaka and T.Masui)

1. $\text{Syl}_{\rightarrow}(f, g)$ の QR 分解: $\text{Syl}_{\rightarrow}(f, g) = QR$ を計算
2. $\|R_{22}^{(k)}\|_2 \leq \varepsilon\sqrt{m+n}$ を満たす k について
 - Case1:** $\|R_{22}^{(0)}\|_2 > \varepsilon\sqrt{m+n}$ ならば, 近似的にも互いに素と判定
 - Case2:** $\varepsilon_r = \|R_{22}^{(k-1)}\|_2 / \|r\|_2$ が最小となる $R_{22}^{(k)}$ の第 1 行ベクトル r から $d(x) := r(x)$ と構成
 - Case3:** ε_r の小さい候補がないならば, Split または 次のステップへ (繰り返し数に依存)
3. 余因子の Reversal 多項式にも同じ処理を繰り返す (二度連続互いに素となるまで)

アルゴリズム 3 (UVGCD by Z.Zeng)

1. $\text{Syl}_{n\downarrow}(f, g)$ の QR 分解: $\text{Syl}_{n\downarrow}(f, g)P_n = Q_nR_n, P_n = I$ を計算
2. $j = n, n-1, \dots, 1$ に対して
 - (a) R_j から最小特異値 σ_{-1} と特異ベクトル $\vec{y}$ を計算 ($R^H \vec{z} = \vec{y}_i, R\vec{z} = \vec{z}, \vec{y}_{i+1} = \vec{z} / \|\vec{z}\|$)
 - (b) $\sigma_{-1} < \varepsilon\sqrt{m-j+1}$ ならば
 - i. 特異ベクトル $\vec{y}$ を余因子 $u(x)$ と $v(x)$ の初期値と設定
 - ii. 最小二乗法により近似 GCD ($d(x)$) の初期値を計算 p
 - iii. Gauss-Newton 法により残差が改善されなくなるまで反復計算
 - iv. 残差が ε 未満ならば, 近似 GCD を出力する
 - (c) $\text{Syl}_{j-1\downarrow}(f, g)P_{j-1} = Q_{j-1}R_{j-1}$ を P_j, Q_j, R_j から計算
3. 互いに素を出力

上記のアルゴリズムでは, Split アルゴリズムや Gauss-Newton 法による精度の改善などが必要とされていますが, 本報告の目的は行列分解部分に対する枢軸選択 (ピボット選択) の導入となりますので, これらのアルゴリズムの詳細については割愛させていただきます (それぞれのアルゴリズムの論文を参照ください)。

²⁾複素数体上の場合, ユニタリ行列となります。

2 3つの近似 GCD アルゴリズムと QR 分解

行列の QR 分解は、与えられた行列 A を直交行列（ユニタリ行列） Q と上三角行列 R との積に分解するものです（つまり、 $A = QR$ ）。基本的な計算方法としては、上三角行列を構成するよう直交変換を繰り返します。次式における H_i が直交変換を表しています（ $*^H$ は、共役転置を表しています）。

$$A = QR = (H_1^H H_2^H \cdots H_r^H) \begin{pmatrix} * & * & * & \cdots & * & * & * \\ 0 & * & * & \cdots & * & * & * \\ 0 & 0 & * & \cdots & * & * & * \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 0 & * & * \\ 0 & 0 & 0 & \cdots & 0 & 0 & * \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 0 \end{pmatrix}$$

よく知られている QR 分解の性質として、基本的に後退誤差（ $\|A - \tilde{Q}\tilde{R}\|$ ）が小さいというのがあります。ただし、基本的に前進誤差（ $\|R - \tilde{R}\|$ ）の小ささは保証されません。

2.1 3つの近似 GCD アルゴリズムにおける QR 分解の役割

QRGCD と ExQRGCD のアルゴリズムにおいて、QR 分解は近似 GCD の係数ベクトルを直接計算するために使用されています。具体的には、Sylvester 行列を QR 分解し、分解後の上三角行列の行ベクトルから係数ベクトルを復元しています。QRGCD アルゴリズムにおいては、直接近似 GCD を計算しているため、基本的に、QR 分解は計算過程において二度のみ計算されます（与式に対して一度と、反転多項式に対しての一度の計二度）。ExQRGC アルゴリズムにおいては、保守的な検出を行っているため、QR 分解は計算過程において複数回計算されます。

一方、UVGCD において、QR 分解は近似 GCD の次数の推定と近似余因子（近似 GCD による多項式の余因子）の係数ベクトルの計算のために使用されています。そのため、QRGCD や ExQRGCD と異なり、Sylvester 行列ではなく部分終結式写像の表現行列を QR 分解しています。近似余因子の係数ベクトルは、QR 分解の上三角行列 R を用いて、反復法により特異ベクトルを求めて行っています。ただし、部分終結式写像の性質を用いているため、近似 GCD の次数が確定するまで、複数回の QR 分解が必要となります。UVGCD アルゴリズムでは、その計算負荷を低減するため、直前の QR 分解を再利用して次の QR 分解を構成しています（詳しくは、次章を参照のこと）。

2.2 QR 分解における枢軸選択

枢軸選択を行わない QR 分解では、結果は $A = QR$ として得られます。LAPACK では、xGEQRF で計算することが出来ます（ATLAS の拡張³⁾にも含まれています）。スケーラビリティ（複数コア対応）が良いとの報告複数ありますが、本報告の後述の実験では、Reference LAPACK を使用しているため、その効果は出ていないものと考えられます。

³⁾行優先の ATLAS の QR 分解には、3.11.25 まで τ の計算にバグがあり、複素共役が格納されているため、注意が必要です。

列選択ありの QR 分解では、結果は $AP_c = QR$ として得られます (P_c は、列選択に対応する置換行列となります)。LAPACK では、`xGEQP3` で計算することが出来ます (この関数⁴⁾は、以前のものに比べて、Level 3 BLAS を使うよう書き直されています)。列選択ありの QR 分解は、特異値計算に有効 (未消去の列ノルム最大を特異値の見積りに利用するなど) とされています⁵⁾。

完全ピボット選択ありの QR 分解では、結果は $P_r AP_c = QR$ として得られます (P_r が行選択, P_c が列選択に対応する置換行列となります)。完全ピボット選択を行う分解を実現する関数は、LAPACK に単独の関数としては含まれていませんが、事前に行ソート (∞ -ノルム) をすることで似た効果を得られることが報告されています (Cox&Higham, 1997)。また、行ノルムでの後退誤差が改善されるとの報告もあります (Powell&Reid, 1969)。

3 枢軸選択ありの QR 分解への拡張

3.1 QRGCD と ExQRGCD

QRGCD アルゴリズムでは、Sylvester 行列を QR 分解した結果の上三角行列 R の右下からのブロック $R_{22}^{(k-1)}$ と $R_{22}^{(k)}$ の大きさについて調べ、その比が $\frac{\|R_{22}^{(k-1)}\|_2}{\|R_{22}^{(k)}\|_2} < 10\epsilon$ を満たす場合に、近似 GCD の候補として、 $d(x) := r_{\ell-k, \ell-k} x^k + \dots + r_{\ell-k, \ell-1} x + r_{\ell-k, \ell}$ を取り出します。図 1 は、これを図示したものです。

ここで重要なのは、GCD の係数ベクトルが R の行ベクトルに現れる理由です。 $f(x)$ と $g(x)$ の Sylvester 行列の行空間は、イデアル $\langle f, g \rangle$ のある次数以下の要素集合と同一視出来るため、GCD の係数ベクトルが行空間に含まれることになります。QR 分解による近似 GCD 検出は、この性質を使用しているため、一般には、列交換を行うことが出来ません。

しかしながら、GCD の最大次数は $\min(\deg(f), \deg(g))$ で押さえられるため、Sylvester 行列の右側から $\min(\deg(f), \deg(g)) + 1$ 列についてのみ順序を保持すれば、残りの左側については列交換を行っても問題なく、行空間の基底ベクトルとして GCD を検出可能となります。

$$\text{Syl}_{\rightarrow}(f, g) = Q \left(\begin{array}{ccccccc} & & & & \leftarrow \min(\deg(f), \deg(g)) + 1 \rightarrow & & \\ r_{1,1} & r_{1,2} & \cdots & \cdots & \cdots & \cdots & r_{1,\ell} \\ & r_{2,2} & r_{2,3} & \cdots & \cdots & \cdots & r_{2,\ell} \\ & & \ddots & \ddots & \ddots & \ddots & \vdots \\ & & & r_{\ell-k, \ell-k} & r_{\ell-k, \ell-k+1} & \cdots & r_{\ell-k, \ell} \\ & & & \boxed{R_{22}^{(k)}} & \boxed{R_{22}^{(k-1)}} & \cdots & r_{\ell-(k-1), \ell} \\ & & & & & \ddots & \vdots \\ & & & & & & r_{\ell, \ell} \end{array} \right)$$

図 1: QRGCD の検出の仕組み

一方、QRGCD を拡張した ExQRGCD についても、同じ性質が成立します。違いは、QRGCD で比を用いて決定される抽出対象の行ベクトルが、 $\epsilon_r = \frac{\|R_{22}^{(k-1)}\|_2}{\|r_k\|_2}$ を最小とする行ベクトルになるだけです (図 2 参照)。そのため、Sylvester 行列の右側から $\min(\deg(f), \deg(g)) + 1$ 列についてのみ順序を保持すれば、残りの左側については列交換を行っても問題なく、行空間の基底ベクトルとして GCD を検出可能となります。

⁴⁾ 枢軸選択の結果は、 $1, 2, \dots, n$ に戻る ($0, 1, \dots, n-1$ でない) ことに注意が必要です。

⁵⁾ 発表時のスライドは間違っており、正しくは「特異値の見積り」です。ご指摘頂いた京都大学の木村先生に感謝の意を表します。

$$\text{Syl}_{\rightarrow}(f, g) = Q \left(\begin{array}{c|cccccc|c} & & & & & & \leftarrow \min(\deg(f), \deg(g)) + 1 \rightarrow & \\ \hline r_{1,1} & r_{1,2} & \cdots & \cdots & \cdots & \cdots & r_{1,\ell} \\ r_{2,2} & r_{2,3} & \cdots & \cdots & \cdots & \cdots & r_{2,\ell} \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \hline \boxed{\vec{r}_k} & r_{\ell-k, \ell-k} & r_{\ell-k, \ell-k+1} & \cdots & r_{\ell-k, \ell} \\ \hline & r_{\ell-(k-1), \ell-(k-1)} & \cdots & r_{\ell-(k-1), \ell} \\ & \vdots & \ddots & \vdots \\ \hline & \boxed{R_{22}^{(k-1)}} & & & r_{\ell, \ell} \end{array} \right)$$

図 2: ExQRGCD の検出の仕組み

3.1.1 行選択

QRGCD と ExQRGCD では、前述のとおり、Sylvester 行列の行空間の性質から近似 GCD を求めており、その行ベクトルの順序に依存していません。結果として、完全ピボット選択における行選択と同等の効果を
得るために、QR 分解の前処理として、行ベクトルの入れ替えを行うことが可能です。特に、これらのアル
ゴリズムで使用される Sylvester 行列は次の構造を持っているため、アルゴリズムの冒頭で、 $\|f\|_{\infty} \geq \|g\|_{\infty}$
を満たすように多項式の順序を入れ替えるだけで、行選択と同じ効果が期待できます。

$$f(x) = f_m x^m + f_{m-1} x^{m-1} + \cdots + f_1 x + f_0, \quad g(x) = g_n x^n + g_{n-1} x^{n-1} + \cdots + g_1 x + g_0,$$

$$\text{Syl}_{\rightarrow}(f, g) = \begin{pmatrix} f_m & f_{m-1} & \cdots & f_1 & f_0 & & \\ & f_m & f_{m-1} & \cdots & f_1 & f_0 & \\ & & \ddots & \ddots & \cdots & \ddots & \ddots \\ & & & f_m & f_{m-1} & \cdots & f_1 & f_0 \\ g_n & g_{n-1} & \cdots & g_1 & g_0 & & & \\ & g_n & g_{n-1} & \cdots & g_1 & g_0 & & \\ & & \ddots & \ddots & \cdots & \ddots & \ddots & \\ & & & g_n & g_{n-1} & \cdots & g_1 & g_0 \end{pmatrix} \in K^{(m+n) \times (m+n)}$$

3.1.2 SPLIT

SPLIT は、QRGCD と ExQRGCD で場合により使用される多項式の分離を行うアルゴリズムです。入
力として、 $f(x) \in K[x]$ を受け取り、 $f(x) \approx f_{in}(x)f_{out}(x)$ を満たす $f_{in}(x), f_{out}(x) \in K[x]$ を返します。こ
こで、 $f_{in}(x)$ の根は絶対値 1 未満、 $f_{out}(x)$ の根は絶対値 1 超となります。詳細は、QRGCD の論文 [3] を
参照して頂くとして、ここではその概要のみを掲載しておきます ($p_0(x) = f(x)$ とした記述です)。

1. Graeffe の Root Squaring: $p_{i+1}(x) = (-1)^m p_i(-\sqrt{x})(p_i(\sqrt{x}))$ ($0 \leq i \leq k-1$)
2. 偏角の原理 (数値積分) で、 $p_k(x) \approx F_k(x)G_k(x)$ に分離
3. Newton 法 ($u \cdot F_k + v \cdot G_k = 1$) で、 $F_k(x), G_k(x)$ の精度を改善
4. Lifting で復元: $G_{k-i} = \gcd(p_{k-i}, G_{k-i+1}(x^2))$, $F_{k-i} = p_{k-i}/G_{k-i}$

Newton 法の部分では, Bezout 係数 ($u \cdot F_k + v \cdot G_k = 1$ を満たす u, v) の計算が必要になります。QRGCD の論文による方法では, QR 分解で Bezout 係数を計算することになっています。図 3 にあるように, 直交行列 Q の最後の列が u, v に対応し, Q^H は分解対象の行列の列入れ替えで不変なため, Bezout 係数の計算においては, $\text{Syl}_{\rightarrow}(F_k, G_k)$ の最後の列が入れ替わらなければ列選択が可能となります。

$$\text{Syl}_{\rightarrow}(F_k, G_k) = QR = \begin{pmatrix} q_{1,1} & \cdots & q_{1,\ell-1} & \boxed{q_{1,\ell}} \\ q_{2,1} & \cdots & q_{2,\ell-1} & \boxed{q_{2,\ell}} \\ \vdots & \ddots & \vdots & \vdots \\ q_{\ell,1} & \cdots & q_{\ell,\ell-1} & \boxed{q_{\ell,\ell}} \end{pmatrix} \begin{pmatrix} r_{1,1} & \cdots & r_{1,\ell-1} & \left| r_{1,\ell} \right| \\ 0 & \cdots & r_{2,\ell-1} & \left| r_{2,\ell} \right| \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & \left| r_{\ell,\ell} \right| \end{pmatrix}$$

図 3: SPLIT における Bezout 係数の計算

3.2 UVGCD

UVGCD では, $\text{Syl}_{k\downarrow}(f, g)$ の最小特異値 σ_{-1} と対応する特異ベクトル $\vec{y}$ の計算に QR 分解が使用されます。 $\text{Syl}_{k\downarrow}(f, g) = QR$ と分解が得られた後に, $\vec{y}_0$ をランダムに設定し, 以下の漸化式で特異ベクトルを求めてから, 特異値を $\sigma_{-1} = \|R\vec{y}\|$ として計算します。

$$R^H \vec{z}_i = \vec{y}_i \Rightarrow R \vec{y}_{i+1} = \vec{z}_i \Rightarrow \vec{y}_{i+1} = \vec{y}_i / \|\vec{y}_i\|$$

この過程で用いられる QR 分解の上三角行列 R は, 図 4 にあるように, 全ての列ベクトルになりますが, 単なる特異ベクトルの計算が目的のため, 列選択は自由に行うことが可能となります。

$$\text{Syl}_{k\downarrow}(f, g) = QR = \begin{pmatrix} q_{1,1} & \cdots & q_{1,\ell-1} & q_{1,\ell} \\ q_{2,1} & \cdots & q_{2,\ell-1} & q_{2,\ell} \\ \vdots & \ddots & \vdots & \vdots \\ q_{\ell,1} & \cdots & q_{\ell,\ell-1} & q_{\ell,\ell} \end{pmatrix} \begin{pmatrix} \deg(f) + \deg(g) - 2k \\ r_{1,1} & \cdots & r_{1,\ell-1} & r_{1,\ell} \\ 0 & \cdots & r_{2,\ell-1} & r_{2,\ell} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & r_{\ell,\ell} \end{pmatrix}$$

図 4: UVGCD における特異値ベクトルの計算

ところが, UVGCD では最小特異値の大きさから近似 GCD の次数を推定するため, 行列 $\text{Syl}_{k\downarrow}(f, g) \in K^{(m+n-k) \times (m+n-2k)}$ に対する QR 分解で近似 GCD の次数が確定しなかった場合, 1 行 2 列を新たに追加した行列 $\text{Syl}_{k-1\downarrow}(f, g) \in K^{(m+n-k+1) \times (m+n-2k+2)}$ の QR 分解を行う必要があります。この QR 分解を最初から計算するのは計算コストの点で問題があるため, UVGCD では追加部分に既計算分の直交変換を行った後, 追加部分に対応する新たな直交変換のみを計算します。従って, 初期の $k = \min(\deg(f), \deg(g)) - 1$ における QR 分解の際は, 全ての列を対象とした列選択が可能となりますが, アップデート時には, 図 5 における追加部分に対応する 2 列分の列選択のみが可能となります。

行選択については, QRGCD や ExQRGCD と異なり, 単純な多項式の順序では解決できません。部分終結式写像の表現行列を作成してから, 行のソーティングが必要となることと, 各回のアップデート処理 ($k \Rightarrow k-1$) において, 既存の直交変換を適用してからソーティングしなおす必要があります。(その他の実験結果から類推する限り) 手間に比べてメリットが期待されないため, 今回の報告では, この行選択についての実験は行っていません。

$$\text{Syl}_{k-1\downarrow}(f, g) = \left(\begin{array}{cccc|cc} f_m & & & & g_n & & \\ f_{m-1} & f_m & & & g_{n-1} & g_n & \\ \vdots & f_{m-1} & \ddots & & \vdots & g_{n-1} & \ddots \\ f_1 & \vdots & \ddots & & g_1 & \vdots & \ddots \\ f_0 & f_1 & \vdots & & g_0 & g_1 & \vdots \\ & f_0 & \ddots & \vdots & & g_0 & \ddots & \vdots \\ & & \ddots & f_1 & & & \ddots & g_1 \\ & & & f_0 & & & & g_0 \end{array} \right) \begin{array}{c} \boxed{\text{追加部分}} \\ f_m \\ f_{m-1} \quad g_n \\ \vdots \quad g_{n-1} \\ \vdots \quad \vdots \\ f_1 \quad g_1 \\ f_0 \quad g_0 \end{array} \right) \in K^{(m+n-k+1) \times (m+n-2k+2)}$$

図 5: UVGCD におけるアップデート処理

4 枢軸選択導入による比較実験

前章で確認した枢軸選択導入可否について簡単にまとめておきます。

QRGCD と ExQRGCD

- Sylvester 行列のうち，左側 $\max(\deg(f), \deg(g)) - 1$ 列までは枢軸選択可能
- 行選択相当も， $\|f\|_\infty$ と $\|g\|_\infty$ の評価で可能
- SPLIT における Bezout 係数の計算では，最後の列を除いて枢軸選択可能

UVGCD

- 初期の Sylvester 行列は，全域で枢軸選択可能
- 更新時は，最後の 2 列のみ枢軸選択可能（ただし，行列全域への直交変換が必要）
- Gauss-Newton 法による最適化部分も最小二乗法なので適用は可能だが今回は考慮しない

以下の実験は，LIBSNAP の 2013 年 12 月下旬時点の未公開最新版を用いて，倍精度（double）と倍精度複素数（double complex）で行っています。実験環境は，Ubuntu 12.04 LTS（Intel i7-3960X，48GB メモリ）で，当該環境でコンパイルした ATLAS 3.10.0 と LAPACK 3.4.2 を用いています。

4.1 数値実験の内容（次数差大，低次 GCD）

次の多項式ペア（精度 10 桁で正規化。許容度 10^{-5} 。GCD は 5 次固定）を対象に実験しました。以下に現れる $f_1(x)$ と $g_1(x)$ は，記載していませんが，係数をランダムに $[-99, 99] \subset \mathbb{Z}$ から選んだ多項式です。

- ランダム生成の多項式（各 $i = 1, \dots, 20$ に 100 組ずつ）

$$f(x) = f_1(x)d(x), \quad g(x) = g_1(x)d(x), \quad d(x) = \sum_{j=0}^5 d_j x^j, \\ \deg(f) = 10, \quad \deg(g) = 10 + k_i, \quad k_i = \begin{cases} 20(i-1) & (i \leq 10) \\ 40(i-1) & (i > 10) \end{cases}$$

- 上記に誤差を追加（各 $i = 1, \dots, 20$ に 100 組ずつ）

$$f(x) + = 10^{-8} \Delta_f(x) / \|\Delta_f\|_2, \quad g(x) + = 10^{-8} \Delta_g(x) / \|\Delta_g\|_2$$

- 単位円の内部と外部に複数根（各 $i = 1, \dots, 20$ に 100 組ずつ）

$$\begin{aligned} d(x) &= \prod_{j=1}^2 (x - \omega_{d,j}) \prod_{j=1}^3 (x - \hat{\omega}_{d,j}) \quad (O(10^{-2}), O(10^2)), \\ f(x) &= d(x)f_1(x), \quad g_2(x) = d(x)g_1(x) \quad (\text{円内外に 5 つずつ根}), \\ g(x) &= g_2(x) \sum_{j=0}^{k_i} g_j x^j, \quad g_j \in [-99, 99] \subset \mathbb{Z} \end{aligned}$$

これらの多項式ペアの特徴を，枢軸選択の視点からまとめると次のとおりです。

- 入力次数は $f(x)$ と $g(x)$ で大きく異なる
 - $\deg(f) = 10, \deg(g) = 10, 30, \dots, 170, 190, 410, 450, \dots, 770$
 - QRGCD/ExQRGCD では，約 10 列を除き，多くが列選択の対象となる
 - UVGCD では，初期が $\deg(g) - 8$ 列であり，その多くが列選択の対象となる
- $f(x)$ と $g(x)$ の ∞ -norm は $i = 1$ を除き，基本 $\|f\|_\infty \geq \|g\|_\infty$ であり，行選択相当
- 行選択前処理済みの順を A 配列，逆順（ $\|f\|_\infty \leq \|g\|_\infty$ ）を B 配列と略記する
 - QRGCD/ExQRGCD では，A 配列ではほぼ全て行選択の前処理済みになる
 - QRGCD/ExQRGCD では，B 配列ではほぼ全て行選択の前処理済みなしになる
 - UVGCD では，多項式の順序による前処理は行選択相当とならないためケアしない

なお，以降の実験結果では，行選択の効果を確認するため， $i = 1$ の結果は除外しています。

4.1.1 実験結果の概要

数値実験の結果から，検出された近似 GCD の次数について，違いが多少あったもののみを表にしたのが，表 1 から表 4 です。割合は，その実験においてもっとも優れた結果（下線部）を 1 としたときの比です。表やその他の実験結果から読み取れることをまとめておきます。

QRGCD

- ランダム生成と誤差項のセットでは検出次数に変化なし
- 単位円内外根のセットでは僅かに変化あり
 - A 配列または B 配列の枢軸選択ありが僅かに良い（1% 未満）

ExQRGCD

- ランダム生成と誤差項のセットでは検出次数に変化なし
- 単位円内外根のセットでは僅かに変化あり
 - 違いはあるが，0.1% 未満の極々僅かな違い

UVGCD どのセットでも検出次数に変化なし

	A 配列	B 配列	A 配列 (枢軸選択あり)	B 配列 (枢軸選択あり)
平均	7.60947	7.60947	<u>7.62474</u>	7.62421
割合	0.997998	0.997998	1	0.999931

表 1: QRGCD (次数差大, 低次 GCD, double)

	A 配列	B 配列	A 配列 (枢軸選択あり)	B 配列 (枢軸選択あり)
平均	7.60947	7.60947	7.62263	<u>7.62316</u>
割合	0.998205	0.998205	0.999931	1

表 2: QRGCD (次数差大, 低次 GCD, dcomplex)

4.2 数値実験の内容 (次数差大, 高次 GCD)

次の多項式ペア (精度 10 桁で正規化。許容度 10^{-5} 。GCD は 50 次固定) を対象に実験しました。以下に現れる $f_1(x)$ と $g_1(x)$ は, 前の実験と同じく, 係数をランダムに $[-99, 99] \subset \mathbb{Z}$ から選んだ多項式です。

- ランダム生成の多項式 (各 $i = 1, \dots, 10$ に 100 組ずつ)

$$f(x) = f_1(x)d(x), g(x) = g_1(x)d(x), d(x) = \sum_{j=0}^{50} d_j x^j, \deg(f) = 100, \deg(g) = 100i$$

- 上記に誤差を追加 (各 $i = 1, \dots, 10$ に 100 組ずつ)

$$f(x) + = 10^{-8} \Delta_f(x) / \|\Delta_f\|_2, g(x) + = 10^{-8} \Delta_g(x) / \|\Delta_g\|_2$$

- 単位円の内部と外部に複数根 (各 $i = 1, \dots, 10$ に 100 組ずつ)

$$\begin{aligned} d(x) &= \prod_{j=1}^{25} (x - \omega_{d,j}) \prod_{j=1}^{25} (x - \hat{\omega}_{d,j}) \quad (O(10^{-2}), O(10^2)), \\ f(x) &= d(x)f_1(x), g_2(x) = d(x)g_1(x) \quad (\text{円内外に 50 個ずつ根}), \\ g(x) &= g_2(x) \sum_{j=0}^{100(i-1)} g_j x^j, g_j \in [-99, 99] \subset \mathbb{Z} \end{aligned}$$

これらの多項式ペアの特徴を, 枢軸選択の視点からまとめると次のとおりです。

- 入力次数は $f(x)$ と $g(x)$ で大きく異なる
 - $\deg(f) = 100, \deg(g) = 100, 200, \dots, 1000$
 - QRGCD/ExQRGCD では, 約 100 列を除き, 多くが列選択の対象となる
- $f(x)$ と $g(x)$ の ∞ -norm は $i = 1$ を除き, 基本 $\|f\|_\infty \geq \|g\|_\infty$ であり, 行選択相当
- A 配列, B 配列については, 前の実験と同じ

なお, 以降の実験結果では, 行選択の効果を確認するため, $i = 1$ の結果は除外しています。また, UVGCD に関しては, 変化が期待されないと考えたため, この実験はしていません。

4.2.1 実験結果の概要

数値実験の結果から, 検出された近似 GCD の次数について, 違いが多少あったもののみを表にしたのが, 表 5 から表 10 です。割合は, その実験においてもっとも優れた結果 (下線部) を 1 としたときの比です。表やその他の実験結果から読み取れることをまとめておきます。

	A 配列	B 配列	A 配列 (枢軸選択あり)	B 配列 (枢軸選択あり)
平均	<u>7.07895</u>	7.07579	7.07474	7.07632
割合	1	0.999554	0.999405	0.999628

表 3: ExQRGCD (次数差大, 低次 GCD, double)

	A 配列	B 配列	A 配列 (枢軸選択あり)	B 配列 (枢軸選択あり)
平均	7.07579	7.07632	<u>7.07737</u>	<u>7.07737</u>
割合	0.999777	0.999851	1	1

表 4: ExQRGCD (次数差大, 低次 GCD, dcomplex)

QRGCD

- ランダム生成のセットでは僅かに変化あり
 - A または B 配列の枢軸選択 なし が, 極僅かに良い (1% 未満)
- 誤差項を加えたセットでは検出次数に (ほぼ) 変化なし
- 単位円内外根のセットでは僅かに変化あり
 - B 配列の枢軸選択ありが, 僅かに良い (1% 前後)

ExQRGCD

- ランダム生成のセットでは僅かに変化あり
 - A または B 配列の枢軸選択 なし が, 極僅かに良い (1% 未満)
- 誤差項を加えたセットでは僅かに変化あり
 - A 配列の枢軸選択ありが, 極僅かに良い (1% 未満)
- 単位円内外根のセットでは僅かに変化あり
 - A または B 配列の枢軸選択 なし が, 極僅かに良い (1% 未満)

5 まとめと考察

枢軸選択ありなしで違いの出た実験結果のみをまとめると次のようになりました。

次数差大, 低次 GCD

- 円内外根: QRGCD: A または B 配列枢軸選択あり (1% 未満)
- 円内外根: ExQRGCD: 0.1% 未満の極々僅かな違い

次数差大, 高次 GCD

- 摂動あり: ExQRGCD: A 配列枢軸選択あり (1% 未満)
- 円内外根: QRGCD: B 配列枢軸選択あり (1% 前後)
- 円内外根: ExQRGCD: A または B 配列枢軸選択 なし (1% 未満)

	A 配列	B 配列	A 配列 (枢軸選択あり)	B 配列 (枢軸選択あり)
平均	61.9133	62.1456	62.1156	<u>62.7967</u>
割合	0.985933	0.989631	0.989154	1
標準偏差	13.5263	13.4202	13.5919	13.6236

表 5: QRGCD (高次 GCD, 円内外根, double)

	A 配列	B 配列	A 配列 (枢軸選択あり)	B 配列 (枢軸選択あり)
平均	62.0111	62.0000	62.1133	<u>62.3911</u>
割合	0.993909	0.993731	0.995548	1
標準偏差	13.5108	13.5167	13.6564	13.5325

表 6: QRGCD (高次 GCD, 円内外根, dcomplex)

これらの結果から, QRGCD は B 配列の枢軸選択ありで僅かに改善する可能性はあるものの, 影響は非常に小さいと考えられます。その他, ExQRGCD はどちらも言えない結果となり, UVGCD は近似 GCD 候補を Gauss-Newton 法で改善することあつてか枢軸選択の影響を受けませんでした。枢軸選択なしの QR 分解の方が, 多くの環境において高速に計算可能なため, これらの近似 GCD アルゴリズムで枢軸選択を行うメリットはないと考えられます。

参 考 文 献

- [1] Bini, D.A., Boito, P.: A fast algorithm for approximate polynomial GCD based on structured matrix computations. In: Numerical methods for structured matrices and applications. Volume 199 of Oper. Theory Adv. Appl. Birkhäuser Verlag, Basel (2010) 155–173
- [2] Boito, P.: Structured Matrix Based Methods for Approximate GCD. Ph.D. Thesis. Department of Mathematics, University of Pisa, Italia (2007)
- [3] Corless, R.M., Watt, S.M., Zhi, L.: QR factoring to compute the GCD of univariate approximate polynomials. IEEE Trans. Signal Process. **52**(12) (2004) 3394–3402
- [4] 増井貴明, 長坂耕作: SNAP パッケージと QRGCD アルゴリズムの改善. 京都大学数理解析研究所講究録. No. 1843 (2013) 101–113
- [5] Zeng, Z.: The numerical greatest common divisor of univariate polynomials. In: Randomization, relaxation, and complexity in polynomial equation solving. Volume 556 of Contemp. Math. Amer. Math. Soc., Providence, RI (2011) 187–217

	A 配列	B 配列	A 配列 (枢軸選択あり)	B 配列 (枢軸選択あり)
平均	<u>93.6778</u>	93.5467	93.6678	93.5333
割合	1	0.998600	0.999893	0.998458
標準偏差	5.06868	5.18821	5.08973	5.14755

表 7: ExQRGCD (高次 GCD, 円内外根, double)

	A 配列	B 配列	A 配列 (枢軸選択あり)	B 配列 (枢軸選択あり)
平均	93.3467	<u>93.7522</u>	93.5244	93.5800
割合	0.995674	1	0.997570	0.998163
標準偏差	5.81948	4.66271	5.60561	5.19365

表 8: ExQRGCD (高次 GCD, 円内外根, dcomplex)

	A 配列	B 配列	A 配列 (枢軸選択あり)	B 配列 (枢軸選択あり)
平均	34.8056	34.8067	<u>34.8611</u>	34.7422
割合	0.998406	0.998438	1	0.996590
標準偏差	18.4662	18.4911	18.4638	18.4433

表 9: ExQRGCD (高次 GCD, 摂動あり, double)

	A 配列	B 配列	A 配列 (枢軸選択あり)	B 配列 (枢軸選択あり)
平均	34.8044	34.7744	<u>34.8056</u>	34.7222
割合	0.999968	0.999106	1	0.997606
標準偏差	18.4648	18.4702	18.4701	18.5183

表 10: ExQRGCD (高次 GCD, 摂動あり, dcomplex)