数理解析研究所講究録
第 1814 巻 2012 年 160-167

近似 Gröbner 基底に向けて
– RREF と STLS による安定化の試み –

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

本講演では，近似 Gröbner 基底の計算を RREF と STLS を活用して安定化させる取り組みについて報告します。まず，近似 Gröbner 基底については，これまでに多くの研究論文等 [10, 11, 14, 16, 3, 12, 9, 8, 7] が発表されています。ただし，これらはその問題設定（ないしはアプローチ）に大きな違いがあり，それを改めて確認しておきます。以下は，佐々木と加古 [9] による問題のクラス分けですが，本講演で扱うのは**第二種**になります。

第一種の近似 Gröbner 基底

厳密な係数を持つ多項式が与えられ，それを生成系とするイデアルの Gröbner 基底を数値的な演算 (例えば浮動小数点数) で求めるものを指します。例えば，Floating point gröbner bases. Shirayanagi, K. (1996) などが，これに分類されます。

入力データ		中間データ		出力データ
正確/厳密	⇒	浮動小数点数	⇒	浮動小数点数

浮動小数点数を 厳密な数 に使っている

第二種の近似 Gröbner 基底

先天的な誤差を持つ不正確な係数を持つ多項式が与えられ，それを生成系とするイデアルの Gröbner 基底を求めるものを指します。利用する演算の種類は問いませんが，先天的な誤差への対応が不可避となります。例えば，Computing floating-point gröbner bases stably. Sasaki & Kako (2007) などが，これに分類されます。

入力データ		中間データ		出力データ
浮動小数点数	⇒	浮動小数点数	⇒	浮動小数点数

近似計算を 避けられない

まず，第二種の近似 Gröbner 基底に対する誤解について説明しておきます。次のような先天的誤差がある入力 $\tilde{F}$ に対して，全次数辞書式順序の近似 Gröbner 基底を既知のアルゴリズムで求めると，以下のような G を基底として得ることができます。

$$\tilde{F} = \{2.000005x + 3.000001y,\ 0.999999xy - 2.000003\} \Rightarrow G = \left\{1.0x + 1.5y,\ 1.0y^2 + 1.33334\right\}$$

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

このような結果を見ると，次の厳密な例（F の全次数辞書式順序の Gröbner 基底が G）が真の値で，それに対して何らかの誤差が付け加えられたものが先の $\tilde{F}$ ではないかと考えたくなりますが，これは必ずしも成り立つとは限りません（というか，一般にこれは成り立ちません）。

$$F = \{2x + 3y,\ xy - 2\} \ \Rightarrow\ G = \left\{x + \frac{3y}{2},\ y^2 + \frac{4}{3}\right\}$$

例えば，先天的誤差の大きさが係数毎に最大 10^{-5} であることが分かっている，次のような入力があったとします。我々は，この入力を $\{2x + 3y,\ xy - 2\}$ に誤差が入ったと考えてしまいがちです。

$$\{0.000001x^2 + 2.000005x + 3.000001y,\ 0.999999xy - 2.000003\}$$

しかしながら，$\{2x + 3y,\ xy - 2\}$ も以下のようないくつかの**可能性**の一つに過ぎず，どれが本当かは判断不可能です。そもそも，これが判断可能ならば近似 Gröbner 基底の計算は，厳密な基底計算と同じ方法で良いことになってしまいます。

$$\begin{array}{lrll}
\{ & 0.000002x^2 + 2.000006x + 3.000002y, & 0.999998xy - 2.000004 & \}, \\
\{ & -0.000001x^2 + 2.000005x + 3.000001y, & 0.999999xy - 2.000003 & \}, \\
\{ & 0.000001x^2 + 2.000005x + 2.999999y, & 0.999999xy - 2.000003 & \}, \\
\{ & 2.000000x + 3.000000y, & 1.000000xy - 2.000000 & \}, \\
 & & & \text{などなど}
\end{array}$$

これらを踏まえて，第二種の問題についてまとめると次のようになります。ある有限の多項式集合 $F = \{f_1, \ldots, f_k\} \subset \mathbf{C}[\vec{x}]$ を生成系とするイデアル $I \subseteq \mathbf{C}[\vec{x}]$ の Gröbner 基底かそれに類するものを計算することが目的になります。ただし，集合 F は与えられずに，係数が先天的な誤差を持つ不正確な有限な多項式集合 $\tilde{F} = \{\tilde{f}_1, \ldots, \tilde{f}_{\tilde{k}}\} \subset \mathbf{C}[\vec{x}]$ しか与えられません。また，可能性としては集合に含まれる多項式の数自体が異なる可能性もあります（つまり，$k \neq \tilde{k}$）。結果として，本来の多項式集合 F を知る方法がない場合に，入力された不正確な情報である多項式集合 $\tilde{F}$ に対して，我々は何を求めるべきかというのが重要な問題となります。

この第二種の問題に対する数学的に厳密なアプローチとして，Weispfenning[15, 16] による Comprehensive Gröbner system （包括的 Gröbner システム，CGS）があります。CGS によるアプローチでは，入力に含まれる先天的な誤差をパラメータで表現し，近似的な側面を排することで厳密計算のみで結果を得られます。つまり，下図のように $\tilde{F} \in \mathbf{C}[\vec{x}]$ の不正確な入力を $\tilde{F} \in \mathbf{C}[\vec{\alpha}][\vec{x}]$ という集合と考えることで，問題を扱うアプローチと言えます。

不正確		**誤差の上限が既知ならば**		**パラメータ**
3.1415927	$\Rightarrow$	[3.14159265, 3.14159274]	$\Rightarrow$	$\frac{31415927}{10000000} + a$
誤差を含む		**区間数などで表現**		**厳密な表現**

このアプローチにより第二種の問題は解決したかのように思えますが，先天的な誤差の大きさがシャープに抑えられることは少なく，仮に抑えられたとしても，CGS の結果はパラメータの条件毎に大量の Gröbner 基底が含まれているため，実際の問題に適用することは非常に困難です。また，近年の研究 [6] などによって効率化されつつありますが，CGS の計算自体も難しい問題であることにも留意して下さい。

2 Gröbner 基底と線形空間

Gröbner 基底計算を行列計算の活用で高速化するなどの研究が古くから行われており，実に多様な結果が得られています。例えば，古くは D. Lazard (1983)[4] から最近の A. Suzuki (2009, CASC)[13] など多くの

研究 [2, 3, 1] がおこなわれています。私も何度か発表を，線形空間の基底としてグレブナ基底を計算する方法のまとめ，RREF によるグレブナー基底計算について，A Study on Gröbner Basis with Inexact Input (CASC2009)[7] などのタイトルで行っています。例えば，(前述の) 次の厳密な Gröbner 基底計算を取り上げ，行列を用いて求めることを考えてみましょう。

$$F = \{2x+3y,\ xy-2\} \Rightarrow G = \left\{x + \frac{3y}{2},\ y^2 + \frac{4}{3}\right\}$$

入力された多項式集合の各要素の係数ベクトルを取り出し，次のような行列 $\mathcal{M}_T(F)$ を構築します。この行列の行ベクトルは，項集合を制限したイデアルの部分集合（加群）に対応する線形空間を張っています。

$$\mathcal{M}_T(F) = \begin{array}{c} \begin{pmatrix} 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 3 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -2 \end{pmatrix} \\ \begin{array}{cccccccccc} x^3 & x^2y & xy^2 & y^3 & x^2 & xy & y^2 & x & y & 1 \end{array} \end{array}$$

この行列に対して，ガウスの消去法などで標準形（Reduced Row Echelon Form）を求めることで次の行列 $\overline{\mathcal{M}_T(F)}$ が得られます。この操作は，Gröbner 基底を求める一般的な方法である Buchberger アルゴリズムにおける S 多項式の計算と簡約化を同時に行っていることに相当します。

$$\overline{\mathcal{M}_T(F)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{9}{2} & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \frac{4}{3} & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & \frac{4}{3} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \frac{3}{2} & 0 \end{pmatrix}$$

このとき，各行ベクトルは求めたかった Gröbner 基底の基底多項式の係数ベクトルになっています。すなわち，次の集合 G_T は入力 F の Gröbner 基底になります。冗長な基底多項式を消去などすれば，先の簡約 Gröbner 基底が得られます。

$$G_T = \left\{x^3 - \frac{9y}{2},\ yx^2 + 3y,\ xy^2 - 2y,\ y^3 + \frac{4y}{3},\ x^2 + 3,\ xy - 2,\ y^2 + \frac{4}{3},\ x + \frac{3y}{2}\right\}$$

2.1 近似 Gröbner 基底の線形空間的考え

参考文献 [7] では，行列の数値的な階数を用いて近似 Gröbner 基底について定義を与えています。これについて少し説明しておきます。まず，線形代数の基礎的な事実として次が成り立っています。

$$\vec{u} \in \sum_{i=1}^{n} \mathbf{C}\vec{v}_i \iff \operatorname{rank}(\vec{v}_1 \cdots \vec{v}_n)^t = \operatorname{rank}(\vec{u}\,\vec{v}_1 \cdots \vec{v}_n)^t$$

いま問題にしているのは近似計算ですから，厳密な階数に代えて行列の Numerical Rank を使うことが考えられます。なお，行列 M の Numerical Rank は次式で定義されます。

$$\mathrm{rank}_\varepsilon(M) = \min_{\|M-M'\|_2 \le \varepsilon} \mathrm{rank}(M')$$

この定義を使うと，誤差を許容したベクトルのメンバーシップ問題を次のように考えることができます。

$$\vec{u} \in_\varepsilon \sum_{i=1}^{n} \mathbf{C}\vec{v}_i \iff \mathrm{rank}_\varepsilon(\vec{v}_1 \cdots \vec{v}_n)^t = \mathrm{rank}_\varepsilon(\vec{u} \;\; \vec{v}_1 \cdots \vec{v}_n)^t$$

これを単純にイデアルのメンバーシップ問題に適用すると，次のように考えることができます。

$$u(\vec{x}) \in_\varepsilon \sum_{i=1}^{n} \mathbf{C}v_i(\vec{x}) \iff \mathrm{rank}_\varepsilon(\vec{v}_1 \cdots \vec{v}_n)^t = \mathrm{rank}_\varepsilon(\vec{u} \;\; \vec{v}_1 \cdots \vec{v}_n)^t$$

近似的なイデアルのメンバーシップ問題が定義されたので，厳密の場合と同じくさらに近似 Gröbner 基底が定義されます。$G = \{g_1, \ldots, g_r\}$ が $\mathrm{ideal}(\tilde{F})$ の項順序 $\succ$ ，許容度 $\varepsilon \in \mathbf{R}_{\ge 0}$ の近似 Gröbner 基底であることを，参考文献 [7] では，次の関係式が満たされることと定義しています。

1. $\forall i, j \in \{1, \ldots, r\}$, $\mathrm{lcm}(\mathrm{ht}(g_i), \mathrm{ht}(g_j)) \in \mathcal{T}$
2. $\forall i, j \in \{1, \ldots, r\}$, $\overline{\mathrm{Spoly}(g_i, g_j)}^G =_{\mathcal{T},\varepsilon} 0$

記法などを省略して説明しているため，詳細については参考文献をご覧ください。しかしながら，この定義に基づく近似 Gröbner 基底の計算方法は確立されていません。本稿では，同じ Numerical Rank を用いるものの，より現実的な対応を模索しています。

3 Nearest Proper Gröbner Basis

近似 Gröbner 基底の計算では，先天的な誤差と新たに発生する計算誤差が問題となります。その両方を同時に解決することは難しいため，先天的な誤差を含む入力が発生する状況について考察すると，そのような入力に対して基底を求めたい理由は，次のようなものと考えられます。

- 誤差に埋もれている代数関係があるに違いない。なんとか誤差を考慮にいれた上で求めて欲しい。

そこで，次のような概念を定義します。

定理 1 (Nearest Proper Gröbner Basis)
有限な多項式集合 $F = \{f_1, \ldots, f_k\} \subset \mathbf{C}[\vec{x}]$ に対して，$\tilde{F} = \{\tilde{f}_1, \ldots, \tilde{f}_k\} \subset \mathbf{C}[\vec{x}]$ とおく。ここで，$\tilde{f}_i = f_i + \delta_i$, $\delta_i \in \mathbf{C}[\vec{x}]$。また，$\tilde{F}$ を生成系とするイデアル I は $I \neq \mathbf{C}[\vec{x}]$ を満たすとする。このとき，摂動部分 δ_i が最小となるイデアル I の *Gröbner* 基底を，F の **Nearest Proper Gröbner Basis** と定義する。

この定義の意図を次の多項式集合で説明します。入力となる多項式集合 $\tilde{F}$ の要素数も少なく，各要素多項式の次数も低いため，現在知られている近似 Gröbner 基底アルゴリズム（近似的なゼロ判定の利用など）で，比較的安定した計算が可能です。

$$\tilde{F} = \{2.000005x + 3.000001y,\ 0.999999xy - 2.000003\} \Rightarrow G = \left\{1.0x + 1.5y,\ 1.0y^2 + 1.33334\right\}$$

しかしながら，各要素多項式の 1.2 倍と 0.5 倍の差を数値的に丸めた多項式を入力集合に追加した次の例では結果が大きく変わります。もちろん，アルゴリズムや打ち切り精度などの調整によっては，先の結果と同じものを得ることも可能かもしれませんが，それが正しい結果であるかは別問題です。

$$\tilde{F} = \{2.000005x + 3.000001y,\ 0.999999xy - 2.000003,\ -0.49995xy + 2.4001x + 3.59999y + 1.00001\} \\ \Rightarrow\ G = \{1.0\}$$

どの計算結果が正しいか判断することは困難です。しかし，この計算結果のように「$G = \{1\}$」と不用意になることは，何らかの構造が期待される場合には不適切な可能性が高いです。先の「Nearest Proper Gröbner Basis」の定義の意図は，このような結果にならないよう先天的な誤差をなるべく排除して一定の構造を導き出そうというものです。この例であれば，若干要素多項式の係数を次のように摂動させることで，一定の構造を見つけることが可能となります。

$$\tilde{F} = \{2.00005x + 2.99997y,\ 0.999981xy - 2.00000,\ -0.499986xy + 2.40006x + 3.60001y + 1.00001\} \\ \Rightarrow\ G = \left\{1.0x + 1.49996y,\ 1.0y^2 + 1.3334\right\}$$

このような摂動を行う先天的な誤差を排除する前処理の方法を次章から提案します。

4 RREF と STLS による前処理

摂動を行うことで先天的な誤差を排除し，入力された多項式集合で生成されるイデアルを真のイデアル ($I \neq \mathbf{C}[\vec{x}]$) に近づけるにはどうしたら良いでしょうか。本発表では，行列の階数とイデアルの自明さに成り立つ関係を用いることにします。即ち，十分大きな行列 $\mathcal{M}_T$ を構成すれば，その列数と同じ階数を持つこととイデアルが自明 ($I = \mathbf{C}[\vec{x}]$) であることは同値になる性質を使います。先天的な誤差の排除には，十分大きな行列 $\mathcal{M}_T$ を構成し，その階数が列数より低くなるように構造を保ったまま行列を摂動させれば良いことになります。構造を持った行列に対して，階数落ちの行列を求めることは，近似 GCD や近似因数分解で提案されている状況に似ており，そこで使われている Structured Total Least Squares の解法で解決できると考えられます。

定義 2 (Structured Total Least Squares)
与えられた $A \in \mathbb{R}^{m \times n}$ と $b \in \mathbb{R}^{n \times 1}$ に対して，$\min_{\Delta A, \Delta b, x} \|[\Delta A\ \Delta b]\|_F^2$ なる ΔA と Δb を求めよ。ただし，$(A + \Delta A)x = b + \Delta b$ が自明でない解 x を持ち，かつ $[A+\Delta A\ b+\Delta b]$ は $[A\ b]$ と同じアフィン構造を持つこと。

なお，次のように式変形することで，フルランクの行列の階数落ち行列を求める問題と STLS は同じ問題となります（なので，近似代数で最近良く使われています）。

$$(A + \Delta A)x = b + \Delta b \iff [A+\Delta A\ b+\Delta b] \begin{bmatrix} x \\ -1 \end{bmatrix} = 0$$

STLS の主な解法としては，P. Lemmerling[5] によれば次のようなものがあります。[1]

RiSVD (Riemanniam SVD) 非線形一般化 SVD として，STLS を解く。

STLN (Structured Total Least Norm) 制約付き非線形最適化 (二次計画) 問題として，STLS を解く。

STLN by ECLS (Equality Constrained LS) 等号制約付き LS として，STLS を解く。

[1] 以下の実験では，最適化計算は Mathematica の NMinimize 関数を使用しました。

5 実験結果

実際に，前述の方法で前処理を行うことで近似 Gröbner 基底の計算が改善されるかの実験を行ったので，それを報告します。まず，次のような入力に対して，全次数辞書式の近似 Gröbner 基底を計算することを考えます。

$$\tilde{F} = \{2.000005x + 3.000001y,\ 0.999999xy - 2.000003,\ -0.49995xy + 2.4001x + 3.59999y + 1.00001\}$$

この入力に対して次のような行列 $\mathcal{M}_T(F)$ を構築します。

$$\begin{pmatrix}
2.00001 & 3.00000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 2.00001 & 3.00000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 2.00001 & 3.00000 & 0 & 0 & 0 & 0 \\
0 & 0 & 2.00001 & 3.00000 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 2.00001 & 3.00000 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.00001 & 3.00000 & 0 \\
0 & 0.999999 & 0 & 0 & 0 & 0 & 0 & -2.00000 & 0 & 0 \\
0 & 0 & 0.999999 & 0 & 0 & 0 & 0 & 0 & -2.00000 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.999999 & 0 & 0 & 0 & -2.00000 \\
0 & -0.49995 & 0 & 0 & 2.4001 & 3.59999 & 0 & 1.00001 & 0 & 0 \\
0 & 0 & -0.49995 & 0 & 0 & 2.4001 & 3.59999 & 0 & 1.00001 & 0 \\
0 & 0 & 0 & 0 & 0 & -0.49995 & 0 & 2.4001 & 3.59999 & 1.00001
\end{pmatrix}$$

この行列に対して，ガウスの消去法などで標準形（Reduced Row Echelon Form）を求めると次を得ます。

$$\begin{pmatrix}
2.00001 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 2.29128 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2.71896 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 2.03232 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 2.98863 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 3.24957 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1.5746 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 3.13997 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.000154372 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.000197261 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

列に関してフルランクとなっていることがわかります。結果からもわかりますが，行列 $\mathcal{M}_T$ の特異値は次のようになっており，僅かの摂動により階数落ちの行列に変化させられる可能性を見て取れます。

$$\{6.97057, 6.06323, 4.69849, 4.2975, 3.6574, 2.29914, 2.12079, 1.96341, 0.0000605786, 0.0000204548\}$$

そこで，行列 $\mathcal{M}_T$ に対して STLN(ECLS) を適用することで，構造を保ったまま階数落ち行列を求めてみます。何度か再帰計算を行うことになりますが，1 回目の計算で，特異値は次のように若干小さくなります。

$$\{6.97057, 6.06323, 4.6985, 4.29751, 3.6574, 2.29912, 2.12079, 1.96341, 0.0000244546, 0.0000110997\}$$

同じように2回目の計算をすると，更に最小特異値は小さくなっていきます。

$$\{6.97057, 6.06324, 4.6985, 4.29751, 3.6574, 2.29911, 2.12079, 1.96341, 0.00000609427, 0.00000563703\}$$

3回目の計算で，最小特異値は 4.45109×10^{-6} まで小さくなり，この時点で，元の多項式集合は次のように構造を保ったまま摂動したことになります。

$$\tilde{F} = \{2.00005x + 2.99997y,\ 0.999981xy - 2.00000,\ -0.499986xy + 2.40006x + 3.60001y + 1.00001\}$$

今回の実験では，P. Lemmerling[5] が提案している二次計画法への帰着を用いず，直接 Mathematica の最適化を行う組込み関数 NMinimize を用いているため，収束が悪く，STLN(ECLS) と RiSVD と STLN を組み合わせても実験をしてみました。先の問題と同じく全次数辞書式順序の近似 Gröbner 基底を次の入力に対して計算しようとしているします。

$$\tilde{F} = \{0.9999y^2 - 1.0001,\ 1.0000x^2 - 1.0003y,\ 0.000110088y - 0.000110099\}$$

適当な項集合を取り行列 $\mathcal{M}_T$ を構築し，特異値を計算すると次のような状況になっています。

$$\{2.04226,\ 2.00015,\ 1.73225,\ 1.7322,\ 1.52038,\ 1.41443,\ 1.41421,\ 1.00005,$$
$$1.0000,\ 0.99995,\ 0.720276,\ 0.000164316,\ 0.000163294,\ 0.0000163214,\ 0.0000163197\}$$

Mathematica による簡易的な STLN(ECLS) により，特異値は次のように小さくなります。

$$\{2.04226,\ 2.00015,\ 1.73225,\ 1.7322,\ 1.52038,\ 1.41443,\ 1.41421,\ 1.00005,$$
$$1.0000,\ 0.99995,\ 0.720276,\ 0.000180813,\ 0.000179701,\ 8.5009 \times 10^{-8},\ 8.5006 \times 10^{-8}\}$$

更に，Mathematica による簡易的な RiSVD により，特異値は次のように小さくなります。

$$\{2.04226,\ 2.00015,\ 1.73225,\ 1.7322,\ 1.52038,\ 1.41443,\ 1.41421,\ 1.00005,$$
$$1.0000,\ 0.99995,\ 0.720276,\ 0.000180894,\ 0.000179781,\ 4.41895 \times 10^{-9},\ 4.41877 \times 10^{-9}\}$$

最後に，Mathematica による簡易的な STLN を行うことで，次の特異値を持つ行列を得られます。

$$\{2.04226,\ 2.00015,\ 1.73225,\ 1.7322,\ 1.52038,\ 1.41443,\ 1.41421,\ 1.00005,$$
$$1.0000,\ 0.99995,\ 0.720276,\ 0.00018089,\ 0.000179777,\ 5.67011 \times 10^{-11},\ 5.66988 \times 10^{-11}\}$$

結果として，構造を保ったまま元の多項式集合は，次の多項式集合に摂動されたことになります。

$$\tilde{F} = \{0.9999y^2 - 1.0001,\ 1.0000x^2 - 1.0003y,\ 0.00009y - 0.00011\}$$

6 まとめ

本発表では，RREF（行列の標準形）と STLS（階数落ち行列の計算）による近似 Gröbner 基底計算に向けた前処理としての安定化の試みについて扱いました。要点としては，1) 近似 Gröbner 基底の前処理に数学的な意味として，最近 (Nearest) 真イデアル (Proper) の近似 Gröbner 基底という定義を与えたこと，2) その前処理には，RREF を利用する枠組みで STLS を利用可能であるという提案，3) その前処理により数値的に一定の安定化が図れる可能性が確認できたこと，になります。

今後の課題としては，1) 今回のような Toy Example でない場合の効果の検証，2) STLS は最小値を保証していないことへの対応，3) 複素数や既に真イデアルの場合に前処理をどう行うか，があります。最後の課題に付いては，複素数の取り扱いは STLS の直接的な拡張が既に知られており，それによって解消できるものの，既に真イデアルだった場合は（既に階数落ち行列となっている場合は），小行列で直接的に拡張するなどしなければなりませんが，余り現実的ではないように思えます。[2)]

[2)]発表後に，改善可能なことが判明しています。

参 考 文 献

[1] M. Byröd, K. Josephson, and K. Åström. Fast optimal three view triangulation. In Y. Yagi, I. S. Kweon, S. B. Kang, and H. Zha, editors, *Asian Conference on Computer Vision*, 2007.

[2] J.-C. Faugére. A new efficient algorithm for computing Gröbner bases (F_4). *J. Pure Appl. Algebra*, 139(1-3):61–88, 1999.

[3] A. Kondratyev, H. J. Stetter, and S. Winkler. Numerical computation of gröbner bases. In *Proceedings of CASC2004 (Computer Algebra in Scientific Computing)*, pages 295–306, 2004.

[4] D. Lazard. Gröbner bases, Gaussian elimination and resolution of systems of algebraic equations. In *Computer algebra (London, 1983)*, volume 162 of *Lecture Notes in Comput. Sci.*, pages 146–156. Springer, Berlin, 1983.

[5] P. Lemmerling. *Structured total least squares: analysis, algorithms and applications.* Ph.D. Thesis. Faculty of Applied Sciences, K.U. Leuven, Belgium, 1999.

[6] K. Nabeshima. A speed-up of the algorithm for computing comprehensive gröbner systems. In *ISSAC 2007: Proceedings of the 2007 international symposium on Symbolic and algebraic computation*, pages 299–306, New York, NY, USA, 2007. ACM.

[7] K. Nagasaka. A study on gröbner basis with inexact input. In *Proceegins of CASC 2009*, volume 5743 of *Lecture Notes in Comput. Sci.*, pages 247–258. Springer, Berlin, 2009.

[8] T. Sasaki and F. Kako. Computing floating-point gröbner bases stably. In *Proceedings of SNC 2007*, pages 180–189. ACM, New York, 2007.

[9] T. Sasaki and F. Kako. Floating-point gröbner basis computation with ill-conditionedness estimation. In *Proceedings of ASCM 2007*, volume 5081 of *Lecture Notes in Comput. Sci.*, pages 278–292. Springer, Berlin, 2008.

[10] K. Shirayanagi. An algorithm to compute floating point gröbner bases. In *Proceedings of the Maple summer workshop and symposium on Mathematical computation with Maple V : ideas and applications*, pages 95–106, Cambridge, MA, USA, 1993. Birkhauser Boston Inc.

[11] K. Shirayanagi. Floating point gröbner bases. In *Selected papers presented at the international IMACS symposium on Symbolic computation, new trends and developments*, pages 509–528, Amsterdam, The Netherlands, The Netherlands, 1996. Elsevier Science Publishers B. V.

[12] H. J. Stetter. Approximate gröbner bases – an impossible concept? In *Proceedings of SNC 2005 (Symbolic-Numeric Computation)*, pages 235–236, 2005.

[13] A. Suzuki. Computing gröbner bases within linear algebra. In *Proceegins of CASC 2009*, volume 5743 of *Lecture Notes in Comput. Sci.*, pages 310–321. Springer, Berlin, 2009.

[14] C. Traverso and A. Zanoni. Numerical stability and stabilization of groebner basis computation. In *ISSAC 2002: Proceedings of the 2002 international symposium on Symbolic and algebraic computation*, pages 262–269, New York, NY, USA, 2002. ACM.

[15] V. Weispfenning. Comprehensive Gröbner bases. *J. Symbolic Comput.*, 14(1):1–29, 1992.

[16] V. Weispfenning. Gröbner bases for inexact input data. In *Proceedings of CASC 2003 (Computer Algebra in Scientific Computing)*, pages 403–411, 2002.