数理解析研究所講究録
第1976巻 2015年 27-30

実数領域における包括的グレブナー基底系と限量子消去
Comprehensive Gröbner System over real number field and quantifier elimination

深作 亮也*
東京理科大学
Ryoya Fukasaku
Tokyo University of Science

1 はじめに

包括的グレブナー基底系 (Comprehensive Gröbner System: CGS) を利用した限量子消去 (Quantifier Elimination: QE) アルゴリズムは [11] で示され, [1] で改良された. 本稿においてこのアルゴリズムを CGS-QE と呼ぶこととする. CGS-QE アルゴリズムはパラメトリックなイデアルを操作するため, 入力に含まれた等式制約をすべて利用することができる. 従って, 入力に等式制約が多い場合は非常に効率的となる.

CGS-QE アルゴリズムは主に CGS 計算によって構成される (つまり, CGS-QE の計算効率は CGS の計算効率に大きく依存する). 近年, [2, 3, 4, 5, 6, 7, 8, 9, 10] といった効率的な CGS 計算アルゴリズムが発表されている. これらの CGS 計算アルゴリズムを利用することによって, CGS-QE の効率化を図ることができる. さらに, CGS-QE において実数領域パラメータ空間上の CGS 計算をすることもできる (つまり, CGS-QE において実数の性質を利用して無駄な計算を省くことができるかもしれない). 本稿では実数領域パラメータ空間上の CGS の計算効率と複素数領域パラメータ空間上の CGS の計算効率の比較を行う.

本稿は次のように構成される. 2 節では CGS の概略を示す. 3 節では CGS 計算アルゴリズムを示す. 4 節では実数領域パラメータ空間上の CGS の計算効率と複素数領域パラメータ空間上の CGS の計算効率の比較に関する実験結果を示す.

2 包括的グレブナー基底系

本稿では次の記号を利用する. $\bar{y} = y_1, \ldots, y_{n_y}$, $\bar{x} = x_1 \ldots x_{n_x}$ とする. $T(\bar{x})$ は $\bar{x}$ からなる項全体とする. ここで h を $\mathbb{Q}[\bar{y}, \bar{x}]$ の多項式とする. このとき, $\mathbb{Q}[\bar{y}, \bar{x}]$ は係数 $\mathbb{Q}[\bar{y}]$ の多項式環 $(\mathbb{Q}[\bar{y}])[\bar{x}]$ とみなす. $T(\bar{x})$ の項順序 $\succ$ を固定したとき, $LM(h)$, $LT(h)$, $LC(h)$ をそれぞれ $h \in \mathbb{Q}[\bar{y}, \bar{x}]$ の $\succ$ に関する先頭単項式, 先頭項, 先頭係数とする ($LM(h) = LC(h)LT(h)$ に注意する). $\mathbb{Q}$ 上の多項式環のイデアル I に対して, $\mathbb{C}, \mathbb{R}$ 上の多様体をそれぞれ $\mathbb{V}_{\mathbb{C}}(I), \mathbb{V}_{\mathbb{R}}(I)$ と記述する. $\mathbb{R}[\bar{x}]$ 上の有限集合 F で生成されるイデアルは $\langle F \rangle$ と記述する.

K を $\mathbb{R}$ もしくは $\mathbb{C}$ とする. まずは分割と分割部についての定義を示す.

*1414704@ed.tus.ac.jp

定義 1
K^{n_v} 上の部分集合による $\{\mathcal{S}_1,\ldots,\mathcal{S}_s\}$ は以下を満たすとき K^{n_v} の分割とよばれる：

1. $\cup_{i=1}^{s}\mathcal{S}_i = K^{n_v}$.
2. 相異なる i,j について $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset$.

各 $\mathcal{S}_i$ は分割部とよばれる.

次に CGS の定義を与える.

定義 2
$\succ$ を $T(\bar{x})$ の項順序とする. $\mathbb{Q}[\bar{y},\bar{x}]$ 上の有限集合 F に対し, 以下を満たすとき有限集合 $\mathcal{G} = \{(\mathcal{S}_1,G_1),\ldots,(\mathcal{S}_s,G_s)\}$ をパラメータ $\bar{y}$ と主変数 $\bar{x}$ の $\succ$ に関する K 上の CGS とよぶ：

1. 各 G_i が $\mathbb{Q}[\bar{y},\bar{x}]$ の有限部分集合である.
2. $\{\mathcal{S}_1,\ldots,\mathcal{S}_s\}$ が K^{n_v} の分割である.
3. 各 $i\in\{1,\ldots,s\}$ について任意の $\bar{c}\in\mathcal{S}_i$ を考えたとき, $G_i(\bar{c},\bar{x}) = \{g(\bar{c},\bar{x}) : g\in G_i\}$ が $\langle F(\bar{c},\bar{x})\rangle$ の $\succ$ に関するグレブナー基底である.

各 $G_i(\bar{c},\bar{x})$ が簡約 (極小) であれば $\mathcal{G}$ も簡約 (極小) とよばれる. (モニックであることは必要ないとする.)

次に実数の性質を利用して無駄な計算を省くことができるような CGS の例を示す.

例 3
a,b をパラメータとする. このとき, イデアル $I = \langle (a^2+1)x+by^2-z^2, y+z\rangle$ を考え, 項順序 $\succ$ を $x\succ_{lex} y\succ_{lex} z$ とする. ここで I の $\succ$ に関する $\mathbb{C}$ 上の CGS を $\mathcal{G}_{\mathbb{C}}$, I の $\succ$ に関する $\mathbb{R}$ 上の CGS を $\mathcal{G}_{\mathbb{R}}$ とする. このとき, $\mathcal{G}_{\mathbb{C}}$, $\mathcal{G}_{\mathbb{R}}$ は以下のような形となる:

i $\mathcal{G}_{\mathbb{C}} = \{(\mathbb{C}^2\setminus \mathbb{V}_{\mathbb{C}}(a^2+1), \{(a^2+1)x+by^2-z^2, y+z\}),$
$(\mathbb{V}_{\mathbb{C}}(a^2+1,b-1), \{y-z\}), (\mathbb{V}_{\mathbb{C}}(a^2+1)\setminus \mathbb{V}_{\mathbb{C}}(b-1), \{1\})\}.$

ii $\mathcal{G}_{\mathbb{R}} = \{(\mathbb{R}^2, \{(a^2+1)x+by^2-z^2, y+z\})\}.$

3 節で CGS 計算アルゴリズムを示した後に再度この例について考察を与える.

3 CGS 計算アルゴリズム

以下は [9] で示された CGS 計算アルゴリズムの概略である.

Algorithm 1 CGS

Input: a finite $F \subset \mathbb{Q}[\bar{y}, \bar{x}]$, a term order $\succ$ on $T(\bar{y}, \bar{x})$ s.t. $\bar{x} \gg \bar{y}$, its restriction $\succ_{\bar{x}}$ on $T(\bar{x})$, a field K;
Output: a CGS of $\langle F \rangle$ w.r.t. $\succ_{\bar{x}}$ over K;

```
1: G←reducedGB(⟨F⟩,≻);
2: if 1 ∈ G then
3:    return {(K^{n_y}, {1})};
4: else
5:    G ← CGSMain(F, ≻, K);
6:    S' ← ∪{P : (P, G) ∈ G};
7:    S ← K^{n_y} \ S';
8:    return {S, {1})} ∪ G;
9: end if
```

Algorithm 2 CGSMain

Input: F, $\succ$, K;

```
1: if V_K(F ∩ Q[ȳ]) = ∅ then
2:    return {(V(F) ∩ Q[ȳ], {1})}
3: end if
4: G ←reducedGB(⟨F⟩,≻);
5: if 1 ∈ G then
6:    return ∅
7: else
8:    {LT(g_1), ..., LT(g_l)} ← the minimal basis of LT(G \ Q[ȳ]);
9:    for 1 ≤ i ≤ l do
10:      H_i ← {LC(g) ∈ Q[Ȳ] : g ∈ G s.t. LT(g) = LT(g_i)};
11:   end for
12:   S_1 ← V_K(G ∩ Q[ȳ]) \ ∪_i V_K(H_i);
13:   G_1 ← G \ {g ∈ G : ∀i ∈ {1, ..., l}(LT(g) ≠ LT(g_i))};
14:   return {(S_1, G_1)} ∪ ∪_i CGSMain(F ∪ H_i, ≻, K);
15: end if
```

再度, 前節の例を考察する.

注意 4
例 3 を仮定し, アルゴリズム **CGS** の適用を考える. まず, $K = \mathbb{C}$ の場合を考えると, $\mathcal{G}_{\mathbb{C}}$ の第 2 要素と第 3 要素は $\mathcal{G}_{\mathbb{C}}$ の第 1 要素の下の再帰計算による結果であることがわかる. 次に $K = \mathbb{R}$ の場合を考えると, $\mathcal{G}_{\mathbb{R}}$ はその第 1 要素の計算終了後, 無駄な再帰計算をしないことがわかる. CGS-QE において $\mathcal{G}_{\mathbb{C}}$ を計算した場合は無駄な再帰計算が発生することになる.

4 実験結果

筆者はアルゴリズム **CGS** を Maple 上に実装した. 本節ではアルゴリズム **CGS** で $K = \mathbb{R}$ の場合の計算時間と $K = \mathbb{C}$ の場合の計算時間を以下の例で比較する.

例 5
イデアル $I = \langle x_0^2 - 2x_0x_{10} + x_{10}^2 + x_2^2 - 2x_2x_9 + x_9^2 - 1, x_0^2 - 2x_0x_{10} + 2x_{10}x_5 + x_2^2 - 2x_2x_9 - x_3^2 + 2x_3x_9 - x_5^2, x_1^2 - 2x_1x_{10} + 2x_{10}x_5 - x_3^2 + 2x_3x_9 + x_4^2 - 2x_4x_9 - x_5^2 \rangle$ を考え, $x_0, x_1, x_2, x_3, x_4, x_5$ をパラメータとする. このとき, 項順序 $x_{10} \succ_{drl} x_9$ に関する CGS をアルゴリズム **CGS** で計算すると, $K = \mathbb{R}$ の場合の計算時間は 1.108 秒であったが $K = \mathbb{C}$ の場合の計算時間は 29.569 秒であった.

5 まとめ

例 5 で見たように CGS の全体の計算時間が $K = \mathbb{R}$ の方がよい場合があることがわかった. しかしながら, 一般に $K = \mathbb{R}$ におけるアルゴリズム **CGSMain** のステップ 1 の判定の計算量は大きい. 従って, CGS-QE で CGS の計算をする場合, $K = \mathbb{R}$ に関するステップ 1 の判定とステップ 12 の再帰計算を並列に行うことが CGS-QE の効率化につながると考えている.

参 考 文 献

[1] Fukasaku, R., Iwane, H. and Sato, Y: Real Quantifier Elimination by Computation of Comprehensive Gröbner Systems, Proceedings of International Symposium on Symbolic and Algebraic Computation, 2015, pp.173-180

[2] Kapur, D., Sun, Y. and Wang, D.: A New Algorithm for Computing Comprehensive Gröbner Systems, Proceedings of International Symposium on Symbolic and Algebraic Computation, 2010, pp.29-36

[3] Kurata, Y.: Improving Suzuki-Sato's CGS Algorithm by Using Stability of Gröbner Bases and Basic Manipulations for Efficient Implementation, Communications of JSSAC Vol.1, 2011, pp.39-66

[4] Montes, A.: A new algorithm for discussing Gröbner bases with parameters, Journal of Symbolic Computation Vol.33-2, 2002, pp.183-208

[5] Manubens, M. and Montes, A.: Improving DISPGB algorithm using the discriminant ideal, Journal of Symbolic Computation Vol.41, 2006, pp.1245-1263

[6] Manubens, M. and Montes, A.: Minimal Canonical Comprehensive Gröbner System, Journal of Symbolic Computation Vol.44, 2009, pp.463-478

[7] Montes, A. and Wibmer, M.: Gröbner Bases for Polynomial Systems with parameters, Journal of Symbolic Computation Vol.45, 2010, pp.1391-1425

[8] Nabeshima, K.: A Speed-Up of the Algorithm for Computing Comprehensive Gröbner Systems, Proceedings of International Symposium on Symbolic and Algebraic Computation, 2007, pp.299-306

[9] Nabeshima, K.: Stability Conditions of Monomial Bases and Comprehensive Gröbner systems, Lecture Notes in Computer Science Vol.7442, 2012, pp.248-259

[10] Suzuki, A. and Sato, Y.: A Simple Algorithm to Compute Comprehensive Groöbner Bases Using Gröbner Bases, Proceedings of International Symposium on Symbolic and Algebraic Computation, 2006, pp.326-331

[11] Weispfenning, V.: A New Approach to Quantifier Elimination for Real Algebra, Quantifier Elimination and Cylindrical Algebraic Decomposition, 1998, pp.376-392