

Parametric Polynomial Spectral Factorization

篠原直行*

立教大学 理学部 化学科

概 要

For polynomial spectral factorization, an important mathematical tool in signal processing and control, an algebraic approach has been proposed by Kanno et al. that enables us to carry out parametric polynomial spectral factorization (PPSF). In this paper, we investigate the efficiency of PPSF by means of comprehensive Gröbner systems (CGS). More specifically, three methods are devised and their computabilities are compared. A discussion on which approach is most suited to PPSF is provided, and furthermore remarks are given on how much improvement of CGS computation can be made for solving general practical problems.

1 はじめに

科学や工学の分野では、パラメータを変数として残すことによって、解析や設計の性能をパラメータを直接操作することによって容易に観察できることが望まれており、そのような要求を満たす手法として代数的手法が提案されている。

多項式スペクトル分解は信号処理や線形動的システムの解析設計において有効な基本的数学ツールであり、多項式スペクトル分解を解くための多くのアプローチが提案されている。ただ、それらのほとんどは高速な浮動小数点演算に適用したものであり、パラメータを陽に扱うことに適していない。しかし、Kanno, Yokoyama, Anai, Hara [1] による、多項式スペクトル分解とグレブナー基底の興味深い関係を利用した代数的手法はパラメータつきの場合の多項式スペクトル分解を解くことを可能にした。

Kanno, Yokoyama, Anai, Hara のアプローチでは、パラメータつきの場合の多項式スペクトル分解を解くために、後述の補題 2 の連立代数方程式からえられるイデアルの shape basis を計算する必要がある。本論文では、スペクトル因子の次数を n としたときに、実際にどの程度大きな n に対してまで shape basis を計算することができるかということに注目し、そのために三つの解法を提案しその有効性を検証した。具体的には、まずパラメータつきの場合のグレブナー基底の計算に Nabeshima のアプローチ [3] を適用し、それを基本的なツールとして、saturation テクニックを用いた二つの手法と、パラメータつきの場合の一変数多項式の最大公約因子を利用する手法である。

*shnhr@tvs.rikkyo.ne.jp

これらの検証は、後述の (4) に現れる $a_0, a_2, \dots, a_{2n-2}$, つまり n 個のパラメータをもつ多項式スペクトル分解に対して行った。ただ実際問題は、これらの a_i はそれぞれ独立したパラメータではなく、 $a_i \in K[p_1, p_2]$ のように 1 個または 2 個のパラメータを変数とする多項式で与えられる。従って実際的な問題に現れるパラメータの個数は 1 個か 2 個であるため、本論文での検証にかかる時間的・空間的計算コストは実際の問題より大きいと考えられ、その検証で得られた結果は実際的な問題を解く際のキーポイントを与える。

本論文の構成は以下のとおり。第 2 節では [1] のアプローチについて簡単に述べる。すなわち、最初にパラメータなしの場合の多項式スペクトル分解とその解法について説明し、その後でパラメータつきの場合について説明する。第 3 節では、パラメータつきの場合のグレブナー基底の計算方法、つまり包括的グレブナー基底系の計算手法である Nabeshima のアプローチ [3] について述べる。第 4 節ではパラメータつきの多項式スペクトル分解を解くための三種類の手法について提案する。第 5 節ではそれら三種類の手法による結果を述べ、第 6 節で考察と今後の研究課題についてまとめる。

ここで本論文で使われる記号について説明する。

K : 体. 本論文では K を \mathbb{Q} または \mathbb{R} と想定する。

L : K の代数的閉包. 本論文では \mathbb{C} と想定する。

$pp(X)$: 変数集合 X の項の集合。

\prec_X : $pp(X)$ における項順序。

$\prec_{X_1, \dots, X_\ell}$: $\prec_{X_1}, \dots, \prec_{X_\ell}$ による $X_1 \prec \dots \prec X_\ell$ なる $pp(X_1 \cup \dots \cup X_\ell)$ 上の消去順序。

$lpp_X(f)$: \prec_X に対する頭項。

$lc_X(f)$: \prec_X に対する頭係数。

$\mathbb{V}(\{f_1, \dots, f_k\}) := \{\alpha \in L^m \mid f_1(\alpha) = \dots = f_k(\alpha) = 0, f_i \in K[X], \#X = m\}$.

$\sigma: K[P][X] \rightarrow L[X]$ なる特化準同型写像. 但し、 P をパラメータの集合、 X を変数の集合とする。

$GB(F, \prec)$: 有限多項式集合 F の項順序 \prec によるグレブナー基底。

$RGB(F, \prec)$: 有限多項式集合 F の項順序 \prec による簡約グレブナー基底。

$CGS(F, \prec)$: パラメトリックな有限多項式集合 F の項順序 \prec による包括的グレブナー基底系。

$A = \{a_0, \dots, a_{2(n-1)}\}$.

$B = \{b_0, \dots, b_{n-1}\}$.

2 パラメータつきの多項式スペクトル分解

この節では多項式スペクトル分解の定義などを紹介し、Kanno, Yokoyama, Anai, Hara のアプローチ [1] によるパラメータつきの多項式スペクトル分解の解法を簡単に説明する。

2.1 パラメータなしの場合の多項式スペクトル分解

議論を簡略化するために、まずパラメータなしの場合の多項式スペクトル分解から説明する。従ってこの節では a_i はパラメータではなく実数とする。

下の実係数多項式 $f(x)$ は虚軸上に根を持たないとする:

$$f(x) = x^{2n} + a_{2n-2}x^{2n-2} + \cdots + a_2x^2 + a_0 \in \mathbb{R}[x]. \quad (1)$$

この仮定は, 制御系問題においては自然に与えられる. $f(x)$ が偶関数であることから, α が $f(x)$ の根であるならば $-\alpha$ も $f(x)$ の根である. また, $f(x)$ は虚軸上に根を持たないことから, $f(x)$ の n 個の根は複素左半平面に属し, 他の根は複素右半平面に属することがわかる. 多項式スペクトル分解とは, 下の定義で与えられる $f(x)$ の因子 $g(x)$ を求めることである:

定義 1 (多項式スペクトル分解)

(1) の多項式 $f(x)$ の多項式スペクトル分解とは次の因数分解をいう:

$$f(x) = (-1)^n g(x)g(-x). \quad (2)$$

ただし,

$$g(x) = x^n + b_{n-1}x^{n-1} + \cdots + b_1x + b_0 \in \mathbb{R}[x] \quad (3)$$

の根はすべて複素左半平面に含まれるものとする. また, このような $g(x)$ を $f(x)$ のスペクトル因子と呼ぶ.

次に, 多項式スペクトル分解を未定係数法で解くことを考える. つまり b_0, \dots, b_{n-1} を変数とし, 主変数 x に関する (2) の右辺と左辺の係数比較から得られる連立代数方程式を解く. この連立代数方程式から得られる多項式集合は次の補題に示される性質を持つ:

補題 2 ([1], Lemma 5.)

(1) と (3) で与えられた $f(x)$ と $g(x)$ に対して, b_0, \dots, b_{n-1} を変数とする. このとき (2) の右辺と左辺の x による係数比較から下記の連立代数方程式を得る:

$$\begin{aligned} & (-1)b_{n-1}^2 + 2b_{n-2} - a_{2(n-1)} = 0, \\ & \quad \quad \quad \vdots \\ & (-1)^k b_{n-k}^2 + \sum_{\substack{1 \leq i \leq 2k-1, \\ i \neq k}} (-1)^i b_{n-i} b_{n-2k+i} + 2b_{n-2k} - a_{2(n-k)} = 0 \text{ for } 2k \leq n, \\ & \quad \quad \quad \vdots \\ & (-1)^k b_{n-k}^2 + \sum_{\substack{2k-n \leq i \leq n, \\ i \neq k}} (-1)^i b_{n-i} b_{n-2k+i} - a_{2(n-k)} = 0 \text{ for } 2k > n, \\ & \quad \quad \quad \vdots \\ & (-1)^n b_0^2 - a_0 = 0. \end{aligned} \quad (4)$$

このとき, この連立代数方程式から得られる多項式集合を \mathcal{G} とすると, \mathcal{G} 自身は $b_{n-1} > \dots > b_0$ なる全次数逆辞書式順序による $\langle \mathcal{G} \rangle$ に対する, 符号を除いて簡約グレブナー基底になっている.

定義 3

補題 2 の $\langle \mathcal{G} \rangle$ をスペクトル分解のイデアルと呼ぶ。

ここで、グレブナー基底の計算による多項式スペクトル分解の解法を簡単に説明する。

定義 4

スペクトル分解のイデアル \mathcal{G} を法とした b_{n-1} 倍写像に対して、それを行列で表現したものを $M_{b_{n-1}}$ とし、さらに $M_{b_{n-1}}$ の特性多項式を S_f とする。また、

$$S_f = \prod_{\alpha_i \neq \alpha_j (i \neq j)} (x - \alpha_i)^{e_i} \in \mathbb{C}[x],$$

に対して

$$T_f = \prod_{e_i=1} (x - \alpha_i)^{e_i}$$

を S_f の simple part と呼ぶ。

[1] の Lemma 4 により、 T_f の最大実根と次の定理の shape basis [4] を求めることで $f(x)$ のスペクトル因子 $g(x)$ を得ることができる:

定理 5 ([1], Theorem 11.)

任意の $\{b_0, \dots, b_{n-2}\} > b_{n-1}$ なる消去順序に対して、イデアル $\langle \mathcal{G} \cup \{T_f(b_{n-1})\} \rangle$ は shape basis S を持つ:

$$S = \{T_f(b_{n-1}), b_{n-2} - h_{n-2}(b_{n-1}), \dots, b_0 - h_0(b_{n-1})\}. \quad (5)$$

但し、 h_i は T_f より次数の低い一変数多項式である。さらにイデアル $\langle \mathcal{G} \cup \{T_f(b_{n-1})\} \rangle$ は f のスペクトル因子をもたらす零点をもつ。

つまり、 α を $T_f(b_{n-1})$ の最大実根としたときに、

$$g(x) = x^n + \alpha x^{n-1} + h_{n-2}(\alpha) x^{n-2} + \dots + h_1(\alpha) x + h_0(\alpha)$$

が $f(x)$ のスペクトル因子である。

本手法ではこの shape basis を計算することを目標とし、 T_f の最大実根の計算についてはふれない。パラメータなしの場合の shape basis の計算手順を下に整理する。

アルゴリズム 6 (パラメータなしの場合の shape basis の計算)

(NonPara-Step 0) : \mathcal{G} を法としたときの b_{n-1} 倍写像を表す行列 $M_{b_{n-1}}$ を求める。

(NonPara-Step 1) : $M_{b_{n-1}}$ の特性方程式 S_f を計算する。

(NonPara-Step 2) : S_f の simple part T_f を計算する。

(NonPara-Step 3) : $RGB(\{T_f\} \cup \mathcal{G}, \langle_B)$ を計算する。但し、 \langle_B は $B \setminus \{b_{n-1}\} >> b_{n-1}$ なる消去順序とする。

2.2 パラメータつきの多項式スペクトル分解 (shape basis の計算)

この論文のメインテーマはパラメータつき, つまり $f(x)$ の係数 a_i を実数ではなくパラメータとした場合の shape basis S の計算方法についてである. さらに $g(x)$ の次数 n に対してどの程度の大きさ n に対してまで問題を解けるかということと, パラメータつき多項式スペクトル分解を解くための後述の三つの手法の有効性について検証する.

パラメータつきの場合も, パラメータなしの場合とほぼ同様の手順で計算できることが [1] の 2.5 節で示されている. 実際には $n = 3, 4$ のときに, パラメータつきの場合の shape basis を, 後述の三つの手法それぞれで計算を試みた. 第 4 節で説明するが, $n = 3, 4$ のときは (NonPara-Step 0) に対応する部分の計算が不要であるため, パラメータつきの場合の shape basis は以下のステップで計算される.

アルゴリズム 7 (パラメータつきの場合の shape basis の計算)

(Para-Step 1): Suzuki-Sato のアプローチにより, 包括的グレブナー基底系の最初の断片を計算することで特性方程式 S_f を得る.

(Para-Step 2): 各断片における S_f の simple part T_f を計算する.

(Para-Step 3): 各断片における $\{T_f\} \cup \mathcal{G}$ に対して包括的グレブナー基底系を計算する.

包括的グレブナー基底系と断片はパラメータつきの場合のグレブナー基底に関係するもので詳しくは第 3 節で説明する.

3 包括的グレブナー基底系 (CGS) とその計算

3.1 包括的グレブナー基底系 (CGS)

(Para-Step 1) から (Para-Step 3) において, パラメータつきのグレブナー基底の計算が必要となる. そのために, この節では包括的グレブナー基底系 (CGS) とその計算方法について, つまり [5] と [3] で議論されていることを例を使って簡単に説明する. 詳しくは [6] の第 2 章を参照.

定義 8 (包括的グレブナー基底系 (CGS))

X を変数の集合, P をパラメータの集合とし, $m = \#P$ とする. さらに, F を $K[X][P]$ の有限部分集合とし, $S \subset L^m$ を代数構成的集合, $>$ を項順序とする. 組の有限集合

$$\mathcal{F} = \{(Eq_1, Not_1, G_1), \dots, (Eq_\ell, Not_\ell, G_\ell)\} \quad (6)$$

が $\langle F \rangle$ の $>$ に関する S 上の包括的グレブナー系 (CGS) であるとは以下を満たすときにいう.

1. $Eq_1, \dots, Eq_\ell, Not_1, \dots, Not_\ell$ は $K[P]$ の有限部分集合, また G_1, \dots, G_ℓ は $K[P][X]$ の有限部分集合.

2. $\mathbb{V}(Eq_1) \setminus \mathbb{V}(Not_1), \dots, \mathbb{V}(Eq_\ell) \setminus \mathbb{V}(Not_\ell)$ は L^m の構成的部分集合.

3. $S \subset \cup_{i=1}^{\ell} \mathbb{V}(Eq_i) \setminus \mathbb{V}(Not_i)$.

4. 各 $i = 1, \dots, \ell$ と各 $a \in \mathbb{V}(Eq_i) \setminus \mathbb{V}(Not_i)$ に対して $\sigma_a(G_i) \setminus \{0\}$ が $\langle \sigma_a(F) \rangle$ の $L[X]$ 上での $>$ に関するグレブナー基底をなす.

特に L^m 上 $\langle F \rangle$ の包括的グレブナー基底系を単に $\langle F \rangle$ の \succ に関する包括的グレブナー基底系と呼ぶ。また、各代数構成的集合 $\mathbb{V}(Eq_i) \setminus \mathbb{V}(Not_i)$ をパラメータ空間 L^m の細胞と呼び、各組 (Eq_i, Not_i, G_i) を断片と呼ぶ。

簡単にいえば、包括的グレブナー基底系とはパラメータ空間を各細胞に分割して得られるグレブナー基底の family である。集合 (6) の各組の左二つの集合 Eq_i, Not_i がパラメータの制約を表すもので、 Eq_i は等式制約を意味し、 Not_i は非等式制約を意味する集合である。

例 9

$A = \{a_0, a_2\}$ をパラメータの集合、 $B = \{b_0\}$ を変数の集合とする。多項式集合 $F = \{a_0 b_0, a_2 b_0^2\} \subset \mathbb{Q}[A, B]$ の包括的グレブナー基底系 CGS は次のような集合であらわされる。

$$CGS = \{(\emptyset, \{a_0 a_2\}, \{b_0\}), (\{a_0\}, \{a_2\}, \{b_0^2\}), (\{a_2\}, \{a_0\}, \{b_0\}), (\{a_0, a_2\}, \emptyset, \{0\})\}.$$

例えば、二番目の断片 $(\{a_0\}, \{a_2\}, \{b_0^2\})$ は、 $a_0 \neq 0, a_2 = 0$ のとき F のグレブナー基底は $\{b_0^2\}$ を意味する。

3.2 Suzuki-Sato のアプローチ

本手法では CGS の計算に Nabeshima のアプローチ [3] を用いた。この手法は包括的グレブナー基底の計算方法である Suzuki-Sato のアプローチ [5] を断片の個数が少なくなるように改良したものである。断片の個数が少なくなることによって計算時間の短縮が期待でき、また実際に計算結果を扱う際にも断片の個数が少ないほうが望ましいことは明らかである。そこで、まず Suzuki-Sato のアプローチから説明を始める。

A をパラメータの集合、 B を変数の集合とし、 $F \subset K[A, B]$ を有限集合とする。このとき Suzuki-Sato のアプローチでは項順序 \prec_B による多項式の頭係数に注目した以下のステップを繰り返していく：

アルゴリズム 10 (Suzuki-Sato のアプローチ)

Input 有限集合 $F \subset K[A, B]$,

Output F に対する CGS.

begin

(S-S-Step 1), A も変数の集合とみなし $\prec_{A,B}$ に関する F のグレブナー基底 G' を計算する。

(S-S-Step 2), $Eq := G' \cap K[A]$ を求める。

(S-S-Step 3), $Not := \{lc_B(g) \in K[A] : g \in G' \setminus Eq, lc_B(g) \notin K\}$ を求める。

(S-S-Step 4), $G = G' \setminus Eq$ とし、断片 (Eq, Not, G) を生成する。

(S-S-Step 5), 各 $not \in Not$ に対して $F = G' \cup \{not\}$ とし (S-S-Step 1) へ。

end

つまり、全ての $g \in G$ に対して、 g の \prec_B に関する頭係数 $lc_B(g)$ が 0 にならないようなパラメータ空間では、 G は F の \prec_B に関するグレブナー基底になっている。

実際に例 9 を Suzuki-Sato のアプローチで計算すると, (S-S-Step 1) で $G'_1 = \{a_0b_0, a_2b_0^2\}(= F)$ となり, (S-S-Step 2) で $Eq_1 = \emptyset$, (S-S-Step 3) で $Not_1 = \{a_0, a_2\}$ をえる. 従って (S-S-Step 4) で最初の断片 $(\emptyset, \{a_0, a_2\}, \{a_0b_0, a_2b_0^2\}) = (Eq_1, Not_1, G_1)$ を得る. 次に (S-S-Step 5) で $a_0 = 0$ と $a_2 = 0$ のそれぞれの場合に対応するグレブナー基底の計算に進んでいく. $a_0 = 0$ の場合については, $F = \{a_0\} \cup G'_1$ として (S-S-Step 1) から (S-S-Step 4) までの計算を行い, 断片 $(\{a_0\}, \{a_2\}, \{a_2b_0^2\}) = (Eq_1, Not_1, G_1)$ を得るわけである. そして, 最終的には以下の四つの断片からなる包括的グレブナー基底系 CGS を得る:

$$CGS(F, \prec_{A,B}) = \{(\emptyset, \{a_0, a_2\}, \{a_0b_0, a_2b_0^2\}), (\{a_0\}, \{a_2\}, \{a_2b_0^2\}), (\{a_2\}, \{a_0\}, \{a_0b_0\}), (\{a_0, a_2\}, \emptyset, \{0\})\}.$$

3.3 Nabeshima のアプローチとその改良

3.3.1 Nabeshima のアプローチの概要

ここで Nabeshima のアプローチの概要を説明する. 詳細については [3] を参照.

Nabeshima のアプローチでは断片の個数を減らすために, (S-S-Step 1) で得られた G' において, ある $f \in G'$ に対して

$$lpp_B(g) \mid lpp_B(f) \in G', f \neq g, g \notin K[A] \quad (7)$$

となるような多項式 $g \in G'$ に注目する. このような多項式 g を消去多項式と呼ぶことにする. 要点は $lc_B(g) \neq 0$ と仮定すれば, (7) を満たす多項式 $f \in G'$ の頭係数 $lc_B(f)$ が 0 であるかどうかの場合分けをしなくてもよいことである.

例えば例 9 において a_0b_0 は消去多項式であり, この消去多項式に対して (7) を満たす多項式 $a_2b_0^2$ の \prec_B による頭係数は a_2 である. $a_0 \neq 0$ のときの $F = \{a_0b_0, a_2b_0^2\}$ のグレブナー基底を考える場合, それは $F' = \{b_0, a_2b_0^2\}$ のグレブナー基底を求めると同じである. 従って $a_0 \neq 0$ のときは a_2 が 0 であるかどうかの場合分けを考える必要がない.

つまり Nabeshima のアプローチは, 消去多項式 g が存在するならば, まずパラメータ空間を $lc_B(g) \in K[A]$ が 0 か 0 でないかの二つに場合分けし, そのあとで Suzuki-Sato のアプローチを実行する.

3.3.2 Rabinovich trick の導入

本手法では Nabeshima のアプローチに Rabinovich trick [2] を導入することで実装の簡略化を行った. Rabinovich trick とは, $f \in K[X]$ が 0 でないときの有限集合 $F \subset K[X]$ のグレブナー基底を求める場合, $r \notin X$ なる新たな変数 r を用意し, $\{fr - 1\} \cup F$ に対して $\prec_{X \cup \{r\}}$ によるグレブナー基底を計算する方法である. Nabeshima のアプローチは $K[A]$ に属するある多項式が 0 でない場合のグレブナー基底を計算するため, Rabinovich trick を取り入れることができる. 今後, r を Nabeshima のアプローチで用いる Rabinovich trick のために用意した変数とし,

$$Ar := \{r, a_0, a_2, \dots, a_{2(n-1)}\}$$

とする.

アルゴリズム 11 (Rabinovich trick を用いた Nabeshima のアプローチ)

Input 有限集合 $F \subset K[A, B]$,

Output F に対する CGS.

begin

1. $Gr : F$ に対する $\prec_{A,r,B}$ の簡約グレブナー基底.
 $G : r$ の消去法 [3] により Gr から r を消去.
if $G \subset K[A]$ **then stop**.
2. **if** G に消去多項式 $g \in K[A, B] \setminus K[A]$ が存在する **then** ステップ 3 **else** ステップ 4 **へ**.
3. $G \cup \{lc_B(g)r - 1\}$ と $G \cup \{lc_B(g)\}$ に対して, ステップ 1 に行き, それぞれを F として実行.
 ただし $G \cup \{lc_B(g)r - 1\}$ が以前に現れたものと一致する場合はステップ 4 **へ**.
4. $Eq := G \cap K[A]$, $Not := \{lc_B(f) : f \in G \setminus Eq, lc_B(f) \notin K\}$.
 断片 $(Eq, Not, G \setminus Eq)$ を生成
 各 $not \in Not$ に対して $F := G \cup \{not\}$ としてステップ 1 **へ**.

end

ここで, Rabinovich trick を用いた Nabeshima のアプローチによる計算方法を例を使って説明する.

例 12

$P := \{a, b, c\}$ をパラメータの集合, $\{x\}$ を変数の集合とし, $f = ax^2 + bx + c \in K[P, x]$ とする. さらに Rabinovich trick 用の変数を r とする. このとき $\{f, \frac{d}{dx}f\} = \{ax^2 + bx + c, 2ax + b\} \subset K[P, x]$ の CGS を Nabeshima のアプローチで計算し, $a \neq 0$ かつ $b^2 - 4ac = 0$ のときに f が重根を持つことを確認する.

Nabeshima のアプローチでは項順序 $\prec_{P \cup \{r\}, x}$ が $P \cup \{r\} \ll \{x\}$ なる消去順序でなければならぬ. ここでは $c < b < a < r < x$ なる辞書式順序とする.

$$H_1 = RGB(F, \prec_{P \cup \{r\}, x}) = \{-4ca + b^2, bx + 2c, 2ax + b\}$$

消去多項式の候補は $bx + 2c, 2ax + b$ の二つであり, ここでは $2ax + b$ を選ぶ.

$a \neq 0$ なる場合を考えるので, Rabinovich trick により

$$RGB(\{ar - 1\} \cup H_1, \prec_{P \cup \{r\}, x}) = \{-4ca + b^2, -rb^2 + 4c, ar - 1, 2x + rb\}$$

を得, 断片は $(Eq_1, Not_1, G_1) = (\{-4ca + b^2, -rb^2 + 4c, ar - 1\}, \{a\}, \{2x + rb\})$ となる. Eq_1 は $r = 1/a$ に注意して $Eq_1 = \{-4ca + b^2\}$ に書きかえられる. よって, f が重根を持つ条件 $a \neq 0$ かつ $-4ca + b^2 = 0$ を得る.

3.3.3 r の消去の省略

アルゴリズム 11 では, 消去多項式の選択後に非等式制約 $f_i \neq 0$ が与えられた場合, Gr に含まれる r に $1/f_i$ を代入することによって r を消す操作をするが, 本手法では断片に r を残すことを提案する. その理由は, r の消去にかかる計算量がその断片の非等式制約集合に含まれ

る多項式によっては小さくないことと, shape basis の計算において, r の消去を省略することで計算量の低減を図れるからである.

アルゴリズム 13 (Rabinovich trick を用い, r の消去をしない Nabeshima のアプローチ)

Input 有限集合 $F \subset K[A, B]$,

Output F に対する CGS.

begin

1. $Gr : F$ に対する $\langle_{A, B}$ の簡約グレブナー基底.

if $Gr \subset K[Ar]$ then stop.

2. if Gr に消去多項式 $g \in K[Ar, B] \setminus K[Ar]$ が存在する then ステップ 3 へ else ステップ 4 へ.

3. $\ell : lc_B(g)$ で r を消去したもの.

Gr の r に $r \cdot \ell$ を代入したものを Gr' と $Gr \cup \{lc_B(g)\}$ に対して, ステップ 1 に行き, それぞれを F として実行. ただし Gr' が以前に現れたものと一致する場合はステップ 4 へ.

4. $Cond := Gr \cap K[Ar]$.

$Not : \{lc_B(f) : f \in Gr \setminus Eq, lc_B(f) \notin K\}$ から r を消去したもの.

断片 $(Cond, Not, Gr \setminus Eq)$ を生成.

各 $not \in Not$ に対して $F := Gr \cup \{not\}$ としてステップ 1 へ.

end

例えば (Para-Step 3) では, simple part $T_{f,i} \in K[A, b_{n-1}]$ に対応する断片 $(Eq_i, Not_i, \{T_{f,i}\})$ ($Eq_i, Not_i \subset K[A]$) を計算した後に,

$$\mathcal{G} \cup \{T_{f,i}\} \cup Eq_i \cup \left\{ r \prod_{q \in Not_i} q - 1 \right\} \quad (8)$$

に対して CGS を計算する. つまりパラメータ空間をいくつかの細胞に分けた後に, その各細胞をさらに小さい細胞へ分割する計算をする. このような 2 度目以降の細胞分割を行う場合に r の消去計算を省略できることを (Para-Step 3) での計算を例にして以下に説明する.

パラメータ空間から細胞への分割をアルゴリズム 13 でおこなうと, $s_i \in K[A, r], T'_f \in K[Ar, b_{n-1}]$ として,

$$Gr = \{s_1, \dots, s_\ell, T'_f\}$$

なるグレブナー基底 Gr が得られる. 等式制約集合 Eq を得るには, $Cond = Gr \cap K[Ar]$ の r に $1 / \prod_{q \in Not} q$ を代入し分母をはらえばよい. また同様にして $\{T'_f\}$ から r を消去して simple part を意味する集合 $\{T_f\} \subset K[A, b_{n-1}]$ を得る. ここで, この Eq, Not と $\{T_f\}$ から (8) を計算するが,

$$\langle Eq \cup \{r \prod_{q \in Not} q - 1\} \rangle = \langle Cond \rangle$$

に注意すれば r の消去を省略して $Cond$ を保持しておけばよいことがわかる.

本手法で残る問題は, 2 度目以降の細胞分割の新たな非等式制約集合 Not' を $Cond$ などの r を含む集合にどのように反映させるかである. しかしそれは [3] で Not' を反映させるには r を $1/\prod_{q \in Not'} q$ と扱えばよいという議論により, r を含む全ての多項式の r に $r \cdot \prod_{q \in Not' \setminus Not} q$ を代入することで可能である.

例えば $ba + c = 0$ かつ $a \neq 0$ なる条件 $Cond$ は

$$\langle Cond \rangle = \langle \{ab + c\} \cup \{ar - 1\} \rangle$$

であり, これに $b \neq 0$ の条件を加えたものは

$$\langle Cond \rangle = \langle \{ab + c\} \cup \{b(ar) - 1\} \rangle$$

である.

r を除去しない Nabeshima のアプローチによる CGS の計算結果を

$$\{(Cond_1, Not_1, G_1), \dots, (Cond_k, Not_k, G_k)\}$$

と書くことにする.

4 パラメータつきの場合の shape basis の計算

この節ではパラメータつきの場合の多項式スペクトル分解を解くために必要な shape basis S の計算方法 (Para-Step1) から (Para-Step3) について議論する.

まず, スペクトル分解のイデアルを与える集合 \mathcal{G} を計算するが, これは (2) の係数比較により得られる. 次にパラメータつきの場合における, \mathcal{G} を法としたときの b_{n-1} 倍写像行列 $M_{b_{n-1}}$ の特性方程式 S_f の計算について説明する.

4.1 Para-Step 1

ここでパラメータつきの場合でかつ $n = 3, 4$ の場合, S_f は Suzuki-Sato のアプローチによる $CGS(\mathcal{G}, \langle_{A,B})$ の最初の断片のグレブナー基底 G_1 に含まれていることを示す. そのために次の定理を用意する.

定理 14

\langle_B を $b_{n-1} \ll B \setminus \{b_{n-1}\}$ なるブロック順序とする. さらに $G := RGB(\mathcal{G}, \langle_{A,B})$, $Eq_1 = G \cap K[A]$, $G_1 := G \setminus Eq_1$, $Not_1 := \{lc_B(g) : g \in G_1, lc_B(g) \notin K\}$, $\alpha_1 \in \mathbb{V}(Eq_1) \setminus \mathbb{V}(Not_1)$ とする.

このとき $h \in G_1$ で $h \in K[A, b_{n-1}]$ なるものがただ一つ存在し, さらに $\langle_{A,B}$ において h がモニックであるならば, $\sigma_{\alpha_1}(h)$ は $\sigma_{\alpha_1}(G_1)$ を法としたときの b_{n-1} の最小多項式である.

証明 [5] のアルゴリズム CGSMain と [5] の Theorem 2.3 により, (Eq_1, Not_1, G_1) は Suzuki-Sato のアプローチによる CGS の最初の断片であり, さらに $\sigma_{\alpha_1}(G_1) \subset L[B]$ は \langle_B による $\sigma_{\alpha_1}(\mathcal{G})$ に対するグレブナー基底である.

仮定より, $h \in G_1 \cap K[A, b_{n-1}]$ なるものがただ一つ存在し, \langle_B は $b_{n-1} \ll B \setminus \{b_{n-1}\}$ なるブロック順序であることから, $\sigma_{\alpha_1}(h)$ は法 $\langle \sigma_{\alpha_1}(G_1) \rangle$ による b_{n-1} の最小多項式である. ■

注意 15

実際に $n = 3, 4$ の場合, Suzuki-Sato のアプローチによる CGS の最初の断片の計算により定理 14 の h の条件を満たすものが存在することが計算結果から分かっており, それらをそれぞれ $n = 3$ に対しては m_{b_2} , $n = 4$ に対しては m_{b_3} と書く. さらにその計算により

$$\deg_{\mathbb{S}_{b_2}}(m_{b_2}) = 2^3, \deg_{\mathbb{S}_{b_3}}(m_{b_3}) = 2^4 \quad (9)$$

であることが得られる.

[1] の Lemma 6 の前で, $LB(\sigma_\alpha(G_1)) = 2^n$ であることが示されている. 従って, b_{n-1} 倍写像 $M_{b_{n-1}}$ は 2^n 次の正方行列であるため, その特性方程式 S_f に対して次が成り立つ:

$$\deg(S_f) = 2^n. \quad (10)$$

[4] の補題 4.37 より, $M_{b_{n-1}}$ の最小多項式は $m_{b_{n-1}}$ であるため,

$$m_{b_{n-1}} \mid S_f \quad (11)$$

である.

(9), (10), (11) より, $n = 3, 4$ の場合は

$$S_f \in G_1$$

であることがわかる.

4.2 Para-Step 2 と Para-Step 3

パラメータつきの場合も S_f の simple part T_f を計算する必要がある. その計算方法として, 本論文では saturation technique ([4], p178) を用いた手法とパラメータつきの場合の GCD 演算 を用いた手法について議論する. また, saturation technique を用いた手法については, 直接 shape basis を求める方法と先に T_f を求めてから shape basis を計算する二つの方法を提案する.

4.2.1 Saturation technique による計算 1 (Sat 1)

ここで Rabinovich trick の一種である saturation technique について紹介する.

アルゴリズム 16 (Saturation technique)

Input 多項式 $f \in K[X]$, 有限集合 $F \subset K[X]$,

Output $G : f \neq 0$ のときの F の \prec_X に関するグレブナー基底.

begin

z を $z \notin X$ なる新たな変数とし, $\prec_{X,\{z\}}$ を \prec_X による $X \ll \{z\}$ なる消去順序とする.

$F \cup \{fz - 1\}$ に対する $\prec_{X,\{z\}}$ によるグレブナー基底 G_1 を計算する.

$G = G_1 \cap K[X]$ を返す.

end

まず, saturation technique を使い, (Para-Step 2) と (Para-Step 3) を同時に計算する方法から説明する. β を T_f の任意の根とすると,

$$S_f(\beta) = 0, S_f'(\beta) \neq 0$$

でなければならない. 従って, $S_f \in \langle \mathcal{G} \rangle$ であることから, $S_f' \neq 0$ のときの \mathcal{G} のグレブナー基底を計算すればよいことになる. 従って計算手順は以下のとおりである.

アルゴリズム 17 (Sat 1)

begin

(Para-Step 1): Suzuki-Sato のアプローチにより, 包括的グレブナー基底系の最初の断片を計算することで特性方程式 S_f を得る.

(Sat1-Step 1): $\langle_{A,B,\{y\}}$ は $A \ll b_{n-1} \ll B \setminus \{b_{n-1}\} \ll y$ なる消去順序とする. Nabeshima のアプローチにより $CGS(\mathcal{G} \cup \{S_f'(b_{n-1})y - 1\}, \langle_{A,B,\{y\}})$ を計算し, 各断片で y を含む多項式を除去する.

end

4.2.2 Saturation technique による計算 2 (Sat 2)

今度は saturation technique を使った, もう一つの計算手順 [Sat 2] について説明する. [Sat 1] との違いは, T_f とそれに対応する断片を先に計算し, その各断片に対して shape basis を計算するところである. 従って計算手順は以下のとおりである.

アルゴリズム 18 (Sat 2)

begin

(Para-Step 1): Suzuki-Sato のアプローチにより, 包括的グレブナー基底系の最初の断片を計算することで特性方程式 S_f を得る.

(Sat2-Step 1): Nabeshima のアプローチにより $CGS(\{S_f(b_{n-1}), S_f'(b_{n-1})y - 1\}, \langle_{A,\{b_{n-1}\},\{y\}})$ を計算し, 各断片で y を含む多項式を除去することで $\{(Cond_1, Not_1, T_{f,1}), \dots, (Cond_k, Not_k, T_{f,k})\}$ を得る.

(Sat2-Step 2): $\langle_{A,B}$ は $A \ll b_{n-1} \ll B \setminus \{b_{n-1}\}$ なる消去順序とする. 各 $(Cond_i, Not_i, T_{f,i})$ に対して, Nabeshima のアプローチにより $CGS(\mathcal{G} \cup Cond_i \cup \{T_{f,i}\}, \langle_{A,B})$ を計算する.

end

4.2.3 パラメータつきの場合の GCD 演算 による計算 (PGCD)

P をパラメータの集合, x を変数とし, $f_1, f_2 \in K[P, x]$ とする. $\{f_1, f_2\} \subset K[P, x]$ の $L[x]$ 上の GCD の family を CGS であらわすことができる. なぜならば, $CGS(\{f_1, f_2\}) = \{(Cond, Not, G)\}$ と書いて, $\alpha \in \mathbb{V}(Eq) \setminus \mathbb{V}(Not)$ に対して $\sigma_\alpha(G)$ は $\sigma_\alpha(\{f_1, f_2\})$ の $L[x]$ 上のグレブナー基底である. また, $L[x]$ は単項イデアル整域であるため $L[x]$ 上の簡約グレブナー基底を $\{g\}$ の形でかける.

実際に Nabeshima のアプローチで, 消去多項式を最小次数のもので選べば

$$CGS(\{f_1, f_2\}, \langle_{P \cup \{r\}, \{x\}}) = \{(Cond_i, Not_i, \{g_i\})\}$$

の形に書ける.

ただ, ここで注意しなければならないことは, Nabeshima のアプローチで $CGS(f_1, f_2)$ を計算すると, f_1 と f_2 が互いに素となる断片, つまり $(Eq, Not, \{1\})$ となる断片が含まれない場合があることである.

ここで実際にその例を紹介する.

例 19

$A = \{a_0, a_2\}$ をパラメータの集合, $B = \{b_3\}$ を変数の集合とし, $U = a_2b_3^2 + a_0$ とする. このとき $CGS_N(\{U, U'\}, \prec_{A,r,B})$ の計算過程を見ていく.

1: $G_1 := RGB(\{U, U'\}, \prec_{A,r,B}) = \{a_0, a_2b_3\}$ を得る.

($a_2 \neq 0$ の断片の計算へ)

2: 消去多項式が存在しないので $a_2 \neq 0$ の条件を付け加える. つまり $F_1 := G_1 \cup \{a_2r - 1\}$.

3: $G'_1 := RGB(F_1, \prec_{A,r,B}) = \{a_0, a_2r - 1, b_3\}$ を得る.

4: 断片 $C_1 := (Cond_1, Not_1, \{g_1\}) = (\{a_0, a_2r - 1\}, \{a_2\}, \{b_3\})$ をえる.

($a_2 = 0$ の断片の計算へ)

5: $F_2 := \{a_2\} \cup G_1$ とする.

6: $G'_2 := RGB(F_2, \prec_{A,r,B}) = \{a_0, a_2\}$ を得る.

7: 断片 $C_2 := (Cond_2, Not_2, \{g_2\}) = (\{a_0, a_2\}, \{\}, \{0\})$ をえる.

よって $CGS_N(\{U, U'\}, \prec_{A,r,B})$ は $\{C_1, C_2\}$, となり, $a_0 \neq 0$ に対応する断片は含まれていないことがわかる. この例は非常にシンプルであるため U と U' が互いに素であるための条件は $a_0 \neq 0$ であることがわかるが, 通常はこの条件の計算はかなり複雑なものとなる.

この場合に互いに素となる断片が生成されない理由は, そもそも Suzuki-Satou のアプローチがグレブナー基底が $\{1\}$ とならない全ての断片を計算する手法であることにある.

そこで, 互いに素となるとき条件の計算方法について説明する. 必要なのは f_1, f_2 が共通根を持たない条件, つまり終結式 $res(f_1, f_2)$ が 0 でない条件である. 従って, $res(f_1, f_2) \in K[A, r]$ を計算すればよいことがわかる.

以上より Nabeshima のアプローチと終結式の計算を組み合わせることで全ての場合の断片を計算できることがわかる.

次にパラメータつきの場合の GCD 演算を使った T_f の計算方法について説明する. パラメータなしの場合で GCD 演算を使って T_f を計算するには

$$U(x) = \gcd\left(S_f(x), \frac{d}{dx}S_f(x)\right), \quad V(x) = \gcd\left(U(x), \frac{d}{dx}U(x)\right)$$

とする. さらに $S_f = \prod_{e_i=1}(x - \alpha_i)^{e_i} \prod_{e_j=2}(x - \alpha_j)^{e_j} \prod_{e_k \geq 3}(x - \alpha_k)^{e_k}$ と書ける. そのとき

$$U = \prod_{e_j=2}(x - \alpha_j)^{e_j-1} \prod_{e_k \geq 3}(x - \alpha_k)^{e_k-1}, \quad V = \prod_{e_k \geq 3}(x - \alpha_k)^{e_k-2}$$

が成り立つ. 従って

$$\frac{S_f \cdot V}{U^2} = \frac{\prod_{e_i=1}(x - \alpha_i)^{e_i} \prod_{e_j=2}(x - \alpha_j)^{e_j} \prod_{e_k \geq 3}(x - \alpha_k)^{e_k} \prod_{e_k \geq 3}(x - \alpha_k)^{e_k-2}}{\prod_{e_j=2}(x - \alpha_j)^{2(e_j-1)} \prod_{e_k \geq 3}(x - \alpha_k)^{2(e_k-1)}} = \prod (x - \alpha_i) = T_f$$

が成り立つ. 従って T_f の計算は GCD 演算のみで計算できることがわかり, パラメータつきの場合は CGS を使って計算できることがわかる.

注意すべきは $V = 1$ の条件のみなので計算過程は以下ようになる.

[PGCD]

(Para-Step 1) : Suzuki-Sato のアプローチにより, $CGS(\mathcal{G}, \langle_{A,B})$ を計算し S_f を得る.

(PGCD-Step 1) : Nabeshima のアプローチにより $CGS(\{S_f, S'_f\}, \langle_{Ar, \{b_{n-1}\}}) = \{(Cond_U, Not_U, \{U_i\})\}_i$ を計算.

(PGCD-Step 2) : $CGS(\{U, U'\} \cup Cond_U, \langle_{Ar, \{b_{n-1}\}}) = \{(Cond_V, Not_V, \{V\})\}$ を計算.

$\gamma = res(U, U') \in K[A, r]$ を計算し, Nabeshima のアプローチの r の消去法に基づき γ を $\gamma' \in K[A]$ に変換する.

$Cond_U \in K[A, r]$ の r の部分に $r \cdot \gamma'$ を代入したものを $Cond'_U$ とすることで, $Cond_V = RGB(Cond'_U, \langle_{A,B})$ を得る.

そして, $Cond_U$ における $V = 1$ のときの断片 $(Cond_V, Not_U \cup \{\gamma'\}, \{1\})$ を生成する.

(PGCD-Step 3) : 各 $Cond_V$ 上での $T_f = S_f \cdot V/U^2$ を計算.

(PGCD-Step 4) : $CGS(T_f \cup Cond_V \cup \mathcal{G}, \langle_{Ar,B}) = \{S, Cond_S\}$ を計算.

4.3 三つの手法の違い

これらの方法の違いは, 図 1 で示されるように, 実装の複雑さと計算コストの大きさである. [Sat 1] は実装は最もシンプルであるが計算コストは最も大きく, [PGCD] はその逆であり, [Sat 2] は実装も計算もその中間に位置する. 実装については上述のステップの数を比べれば明らかで, 計算コストの大きさの違いは項順序によるものであり, その原因は消去順序のブロックの数と変数の個数である. 経験的に, 変数の個数よりもブロックの個数のほうがその影響が大きいことがわかっている.

注意 20

Shape basis を求めるには, \langle_B は $b_{n-1} \prec B \setminus \{b_{n-1}\}$ なる消去順序でなければならない. 従って \langle_B は $b_{n-1} \prec b_{n-2} \prec \dots \prec b_0$ なる辞書式順序でもよい. Shape 基底の計算において, この二種類の項順序 \langle_B で実験をしてみたところ大差はないことがわかった. したがって \langle_B のブロック数は 1 とみなしてもよいといえる.

よって, [PGCD] で使われている項順序は $\langle_{A, \{b_{n-1}\}}$ と $\langle_{A,B}$ とともにブロックの個数は 2 である. [Sat 2] の場合は $\langle_{Ar, \{b_{n-1}\}, \{y\}}$ と $\langle_{A,B}$ でブロックの数は 3 と 2 である. [Sat 1] が最も計算コストが大きいと考えられるのは, 項順序が $\langle_{A,B, \{y\}}$ でブロックの数が 3 であるのと [Sat 2] に比べて変数が多いからである.

$$S_f, T_f, U, V \in \mathbb{Q}[Ar, b_{n-1}]$$

[Sat1]

$$CGS(\mathcal{G} \cup \{S'_f y - 1\}, \prec_{Ar, B, \{y\}}) \cap \mathbb{Q}[Ar, B]$$

[Sat2]

$$CGS(\{S_f, S'_f y - 1\}, \prec_{Ar, \{b_{n-1}\}, \{y\}}) \cap \mathbb{Q}[Ar, b_{n-1}]$$

$$CGS(\mathcal{G} \cup Cond_i \cup \{T_{f,i}\}, \prec_{Ar, B})$$

[PGCD]

$$CGS(\{S_f, S'_f\}, \prec_{Ar, \{b_{n-1}\}})$$

$$CGS(\{U, U'\} \cup Cond_U, \prec_{Ar, \{b_{n-1}\}})$$

$$CGS(\{V, V'\} \cup Cond_V, \prec_{Ar, \{b_{n-1}\}})$$

$$T_f = S_f V / U^2 \text{ on } Cond_V.$$

$$CGS(\mathcal{G} \cup \{T_f\} \cup Cond_V, \prec_{Ar, B})$$

1

図 1: 三つの手法の違い

5 数値実験の結果

多項式スペクトル分解 (2) の解 $g(x)$ の次数を n とし, $n = 3$ と 4 の場合で数値実験を行った. 実験環境は Intel Quad Core Xeon X5355 2.66 GHz, メモリ 32 Gb である. また終結式の計算については Maple 11, それ以外の計算は Risa/Asir を用いた.

5.1 $n = 3$ のとき

最も計算コストの大きい [Sat 1] と最も小さい [ParaGCD] で数値実験を行った.

	[Sat 1]	[ParaGCD]
Shape basis に対応する断片の個数 (重複を含めて)	5	5

双方ともに 全ての場合分けに対応する shape 基底を数秒で計算することができた.

5.2 $n = 4$ のとき

[Sat 1] で計算をしたところ, 計算できたのは (Sat1-Step 1) における最初の断片のみで, それ以外の断片はメモリ不足で計算できなかった.

[Sat 2] の計算結果は, (Sat2-Step 1) においてたった 5 個の断片のみ計算することができた. しかし他の断片についてはメモリ不足のため計算できなかった.

[ParaGCD] ではすべての場合分けに対応する shape 基底を約 72 時間で計算することができた. 次の結果は $a_0 \neq 0$ の条件下で計算したものである.

Simple part T_f に対応するパラメータ空間の細胞分割は V に対する分割と等しいため, T_f

	U	V	Shape basis
断片の個数 (重複を込めて)	53	118	121

に対する断片は 118 個存在する. さらに, この結果より T_f に対する細胞分割と shape basis の細胞分割はほぼ一致することがわかる.

6 まとめ

$n = 3$ のときの計算コストは非常に小さく, どの三つの手法でも計算ができることがわかったが, $n = 4$ になると計算は飛躍的に難しくなることがわかり, その原因は変数とパラメータがともに一つずつ増えることにある. 従って $n = 5$ 以上の場合については, 非常に多くのメモリと時間が必要であるとされると予想される.

$n = 4$ のときの比較実験によって, 三つの手法のうち最も効果的なのは [PGCD] であることがわかった. また, パラメータ付きの shape 基底の計算で重要なのは包括的グレブナー基底系を計算するときの消去順序のブロックの個数であることがわかった.

パラメータの部分が斉次化されるようにパラメータの項順序に重みを付けることで係数膨張を抑えられ, 計算時間の短縮につながることがわかった. 例えば $\{a_0, a_2, a_4\}$ のときは $\{3, 2, 1\}$, $\{a_0, a_2, a_4, a_6\}$ の場合は $\{4, 3, 2, 1\}$ とするのが有効であり, 重みを付けることによって計算時間は約 $1/4$ になった.

7 今後の課題

Maple と Risa/Asir の得意な計算を活かすことで初めて計算が可能になった. 実際に Maple で終結式や擬剰余の演算を実装し, 消去順序によるグレブナー基底の計算については Risa/Asir を用いた. 今後の課題は同一の計算ツールを用いて正確な計算時間を計りたい.

今回は

$$f(x) = x^{2n} + a_{2(n-1)}x^{2(n-1)} + \dots + a_2x^2 + a_0 \in \mathbb{Z}[A, x],$$

つまり各係数をパラメータとしたパラメータが n 個の場合を考えた.

しかし実際の問題はもっとパラメータの個数が少ない, 例えばパラメータを 1, 2 個しか含まない多項式

$$f(x) = x^{2n} + a_{2(n-1)}(p_1, p_2)x^{2(n-1)} + \dots + a_2(p_1, p_2)x^2 + a_0(p_1, p_2) \in \mathbb{Z}[p_1, p_2, x], \quad a_i \in K[p_1, p_2]$$

などが考察の対象となる.

本論文の結果は実際的な問題を解くための有効な手法を与えるものであり, 例のようにパラメータが 1, 2 個しか含まない場合に, より大きな n に対して shape 基底系を計算できるかが今後の課題となる.

8 謝辞

パラメトリック多項式スペクトル分解の解法という興味深い問題を紹介してくださった新潟大学の菅野政明先生, セミナーでの的確なコメントなどありがとうございました. また論文

の構成や ACA での発表でご指導いただいた九州大学の穴井宏和先生に心から感謝いたします。筑波大学の小副川健さんに本論文の編集に協力していただきました。そのためのセミナーでの議論は有意義なものとなり大変感謝しております。最後にこの研究全般においてこの論文を書くにあたって立教大学の横山和弘先生にご指導いただきました。本当にありがとうございました。

参 考 文 献

- [1] M. Kanno, K. Yokoyama, H. Anai and S. Hara, "Parametric optimization in control using the sum of roots for parametric polynomial spectral factorization", Proceedings of the 2007 International Symposium on Symbolic and Algebraic Computation, ACM-Press, pp211-218, 2007.
- [2] D. Kapur, "Using Gröbner bases to reason about geometry problems", J. Symbolic Computation 2 (4), pp.399-408, 1986. ACM-Press, pp.299-306, 2007.
- [3] K. Nabeshima, "A speed-up algorithm for computing Comprehensive Gröbner systems", Proceedings of the International Symposium on Symbolic and Algebraic Computation, ACM-Press, pp.299-306, 2007.
- [4] 野呂正行, 横山和弘, グレブナー基底の計算 (基礎編), 東京大学出版, 2003.
- [5] A. Suzuki and Y. Sato, "A simple algorithm to compute Comprehensive Gröbner Bases using Gröbner bases", International Symposium on Symbolic and Algebraic Computation, ACM-Press, pp326-331, 2006.
- [6] 日比孝之, グレブナー基底の現在, 数学書房, 2006.