数式処理 *J.JSSAC (2007)*
Vol. 14, No. 1, pp. 3 - 15

CGB 特集

包括的グレブナー基底（系）入門

佐藤 洋祐*

東京理科大学

1 はじめに

複素数体 (代数的閉体) 上の連立代数方程式はグレブナー基底を用いることで完璧に解くことができる. それでは，方程式がパラメーターを含むような場合はどうであろうか. 以下のような連立方程式を考えてみよう.

$$\begin{cases} x^3 + 2xyz - z^2 - s = 0 \\ 3x^2 + 2yz - 2x\lambda = 0 \\ xz - y\lambda = 0 \\ xy - z - z\lambda = 0 \\ x^2 + y^2 + z^2 - a = 0 \end{cases}$$

これは，a をパラメーターとする制約条件 $x^2 + y^2 + z^2 - a = 0$ のもとで $x^3 + 2xyz - z^2$ の極値 s とそれを与える x, y, z の値を求めるためのラグランジェの未定乗数法による方程式である. s がみたすべき方程式を計算するには，イデアル $I = \langle x^3 + 2xyz - z^2 - s, 3x^2 + 2yz - 2x\lambda, xz - y\lambda, xy - z - z\lambda, x^2 + y^2 + z^2 - a\rangle \subseteq \mathbb{Q}[x, y, z, \lambda, s]$ に含まれる s の最小多項式，すなわち I に含まれる s のみからなる多項式の最小次数の多項式 f，を求めればよい. f を計算するには，例えば辞書式順序 $\lambda > x > y > z > s$ のもとで I のグレブナー基底を求めればよい. 一般に，このグレブナー基底は a の取る値によって異なったものになり，従って f も異なったものになる. 例えば $a = 1$ の時は $f = 3456 * s^5 + 3395 * s^4 - 3652 * s^3 - 3395 * s^2 + 196 * s$ で，$a = \frac{36}{25}$ の時は $f = 105468750000000 * s^6 + 291390771484375 * s^5 - 132814589062500 * s^4 - 897144773400000 * s^3 - 543787731244800 * s^2 + 80790550806528 * s$ となる. ここで自然な疑問として， f を a に関して一様に求めることはできないであろうか，すなわち f を s と a からなる多項式として表すことができないであろうかという問題が考えつくであろう. I のグレブナー基底が a に関して一様に求めることができる，すなわち a にどんな値を代入したときでも $\{g_1(a, x, y, z, \lambda, s), \ldots, g_k(a, x, y, z, \lambda, s)\}$ が I のグレブナー基底になっているような多項式 $g_1(a, x, y, z, \lambda, s), \ldots, g_k(a, x, y, z, \lambda, s)$ が存在して計算できるならばこの問題

*ysato@rs.kagu.tus.ac.jp

は肯定的に解決する．ちょっと考えると以下の方法が思いつく．有理関数体 $\mathbb{Q}(a)$ を係数体とする多項式環 $\mathbb{Q}(a)[x,y,z,\lambda,s]$ において辞書式順序 $x>y>z>\lambda>s$ のもとでイデアル $\langle x^3+2xyz-z^2-s, 3x^2+2yz-2x\lambda, xz-y\lambda, xy-z-z\lambda, x^2+y^2+z^2-a\rangle \subseteq \mathbb{Q}(a)[x,y,z,\lambda,s]$ のグレブナー基底を求める方法である．実際にこの既約グレブナー基底を求めると，それに含まれる s と a のみからなる多項式は以下のものが１つだけあることが分かる．(既約グレブナー基底は最大単項の係数は１として定義されるが，本稿ではこの条件をみたしていないものも含めて既約グレブナー基底とよぶことにする.)

$$3456*s^7+(1728*a^2+7632*a+947)*s^6+(-544*a^3+5588*a^2+1442*a+108)*s^5+(-1856*a^5-6500*a^4+449*a^3+495*a^2+108*a)*s^4+(-3040*a^6-5640*a^5-1458*a^4-108*a^3)*s^3+(128*a^8-1132*a^7-1396*a^6-495*a^5-108*a^4)*s^2+(128*a^9+52*a^8+16*a^7)*s$$

a に 1 あるいは $\frac{36}{25}$ を代入しても，上の f は得られない．このことから，$\mathbb{Q}(a)[x,y,z,\lambda,s]$ において求めたグレブナー基底が a に関して一様なグレブナー基底にはなっていないことがわかる．一様なグレブナー基底の構成はそんなに簡単なことではないのである．このような一様なグレブナー基底について，１９９２年に Weispfenning によって初めてその存在と計算アルゴリズムが発見された [17]. 彼はこれを包括的グレブナー基底 (Comprehensive Gröbner basis, 以下 CGB と記す) とよび，今日では一般的な名称になっている．きちんとした定義や構成方法については次節で述べるが，上の例は少々複雑なので，以下のような簡単な例を考えてみよう．a をパラメーターとするイデアル $\langle a*x+y^2-1, y^3-1\rangle \subseteq \mathbb{Q}[x,y]$ の辞書式順序 $x>y$ のもとでの CGB は $\{-a*y*x+y-1, a*x+y^2-1, y^3-1\}$ で与えられる．実際 a に 0 を代入すると，$\{y-1, y^2-1, y^3-1\}$ が得られるがこれは $\langle y^2-1, y^3-1\rangle$ のグレブナー基底になっている．a が 0 でないときは $-a*y*x+y-1=-y*(a*x+y^2-1)+(y^3-1)\in\langle a*x+y^2-1, y^3-1\rangle$ であり $\{a*x+y^2-1, y^3-1\}$ はグレブナー基底であるので $\{-a*y*x+y-1, a*x+y^2-1, y^3-1\}$ は $\langle a*x+y^2-1, y^3-1\rangle$ のグレブナー基底になっている．ここで注意すべき点は，どちらも既約グレブナー基底にはなっていないことである．一般に CGB は既約グレブナー基底を与えることができない．一様な既約グレブナー基底を得るためには，パラメーター空間を分割する必要がある．厳密な定義は次節で与えるが，パラメーターの空間をパラメーターのみたす条件によって分割して，１つの分割においては一様なグレブナー基底が与えられているとき，これを包括的グレブナー基底系 (Comprehensive Gröbner System, 以下 CGS と記す) とよぶ．上の例では，$\{(\{a=0\}, \{y-1\}), (\{a\neq 0\}, \{a*x+y^2-1, y^3-1\})\}$ が $\langle a*x+y^2-1, y^3-1\rangle$ の CGS である．CGS では，それぞれの分割におけるグレブナー基底が既約グレブナー基底になるようにできる．CGS の存在とアルゴリズムも [17] ではじめて明らかになったのであるが，CGB と CGS は表裏一体の関係にあり，実際 CGB は CGS を用いて構成される．

[17] が発表されて以来，Weispfenning らのグループによるソフトウエアの改良は行なわれてきたものの ([3])，CGS と CGB に関する研究の本質的な進展はほとんどなかったが，最近になって [9, 10, 15] において効率的な計算方法に関して飛躍的な発展がなされた．

一方，理論においても [14, 18] が発表された．[14] は [16] で導入され [12, 13] 等でさらに研究が進んだ可換フォンノイマン正規環を係数環とする多項式環におけるグレブナー基底の

理論を用いて CGS を自然に扱うことができることを明らかにし，従来の理論では不可能であった CGS の標準形が自然に定義されること等を明らかにした. [18] は [10] の内容を一般化したものであるが，ある種の条件のもとで CGB の標準形が定義できることを明らかにした. この結果はその後の [9] に使われている.

以下では，まず 2 節で [17] で与えられた CGS と CGB の計算アルゴリズムについて平易な解説を与える. 3 節では CGS や CGB と密接な関係をもつ [1, 4, 5, 6, 7] 等のグレブナー基底の安定性についての結果を紹介する. 4 節では [8, 9, 10, 11, 14, 15, 18] における最近の CGS と CGB の研究について紹介する. 最後に 5 節では，現在利用可能なソフトウエアについて，入手方法及び実行例も含めて紹介する. 2 節の内容はグレブナー基底に関する基礎知識 (S-多項式とブッフバーガーアルゴリズム) さえあれば読解可能である. 3 節と 4 節の内容は多岐にわたるので，証明はつけていないが概要は十分理解できるであろう. より詳しく知りたい読者には十分な引用文献を紹介しているのでそれらを参照されたい. 尚， [8, 15] については，本号の「Dynamic Evaluation を用いた Discrete Comprehensive Gröbner Bases の計算 (倉田陽介)」と「グレブナー基底を用いた包括的グレブナー基底計算 (鈴木 晃)」で詳しく解説されている.

2 CGB と CGS

この節以降では K は任意の体を表すものとし，$\overline{K}$ はその代数的閉体とする. 標数は任意のものを取り得るが，これらの概念に不慣れな読者は K は有理数体 $\mathbb{Q}$，$\overline{K}$ は複素数体 $\mathbb{C}$ であると思って読むと読み易いであろう.
以下では主変数を表す記号として $X_1, \ldots, X_n$，パラメーターを表す記号として $A_1, \ldots, A_m$ を用いることにする. $X_1, \ldots, X_n$ や $A_1, \ldots, A_m$ を一まとまりにしてそれぞれ $\bar{X}$ と $\bar{A}$ で表すことにする.

定義 **2.1** m を自然数とする. $\overline{K}^m$ の分割とは $\cup_{i=1}^{l} S_i = \overline{K}^m$ をみたす互いに交わらない，すなわち相異なる i, j にたいして $S_i \cap S_j = \emptyset$ となるような，$\overline{K}^m$ の部分集合の集まり $\{S_1, \ldots, S_l\}$ と定義する. ここで，さらに各 S_i は次のような形で表現されているものとする. $K[\bar{A}]$ のある多項式 $p_1, \ldots, p_s, q_1, \ldots, q_t$ にたいして，$S_i = \{\bar{a} \in \overline{K}^m | p_1(\bar{a}) = 0, \ldots, p_s(\bar{a}) = 0, q_1(\bar{a}) \neq 0, \ldots, q_t(\bar{a}) \neq 0\}$ と表される. 各 S_i を分割部とよぶ. また，S_i とそれを定める等式と非等式の集合 $\{p_1 = 0, \ldots, p_s = 0, q_1 \neq 0, \ldots, q_t \neq 0\}$ を同一視することにする.

定義 **2.2** $K[\bar{A}, \bar{X}]$ の有限集合 $F = \{f_1(\bar{A}, \bar{X}), \ldots, f_k(\bar{A}, \bar{X})\}$ にたいして，$\overline{K}^m$ の分割 $\{S_1, \ldots, S_l\}$ と $K[\bar{A}, \bar{X}]$ の有限集合の集合 $\{G_1, \ldots, G_l\}$ が各 i について，任意の $\bar{a} \in S_i$ にたいして $G_i(\bar{a}) = \{g(\bar{a}, \bar{X}) | g(\bar{A}, \bar{X}) \in G_i\}$ が $\overline{K}[\bar{X}]$ におけるイデアル $\langle f_1(\bar{a}, \bar{X}), \ldots, f_k(\bar{a}, \bar{X})\rangle$ のグレブナー基底になっているとき，$\mathcal{G} = \{(S_1, G_1), \ldots, (S_l, G_l)\}$ を F の *CGS* とよぶ. さらに任意の $\bar{a} \in S_i$ にたいして $G_i(\bar{a}) \setminus \{0\}$ が既約グレブナー基底になっているなら $\mathcal{G}$ は既約 *CGS(reduced CGS)* とよばれる. 各 i にたいして，G_i が $K[\bar{A}, \bar{X}]$ におけるイデアル $\langle f_1(\bar{A}, \bar{X}), \ldots, f_k(\bar{A}, \bar{X})\rangle$ に含まれるとき，$\mathcal{G}$ は F の忠実な *CGS(faithful CGS)* とよばれる.

定義 **2.3** $K[\bar{A}, \bar{X}]$ の有限集合 $F = \{f_1(\bar{A}, \bar{X}), \ldots, f_k(\bar{A}, \bar{X})\}$ にたいし $K[\bar{A}, \bar{X}]$ の有限集合 $G =$

$\{g_1(\bar{A},\bar{X}),\ldots,g_s(\bar{A},\bar{X})\}$ が，任意の $\bar{a}\in\overline{K}^m$ にたいして $G(\bar{a})=\{g_1(\bar{a},\bar{X}),\ldots,g_s(\bar{a},\bar{X})\}$ が，$\overline{K}[\bar{X}]$ におけるイデアル $\langle f_1(\bar{a},\bar{X}),\ldots,f_k(\bar{a},\bar{X})\rangle$ のグレブナー基底になっているとき，G は F の *CGB* とよばれる.

次の定理は明らかであるが，CGB の構成にとって非常に重要である.

定理 **2.1** $\mathcal{G}=\{G_1,\ldots,G_l\}$ を F の忠実な *CGS* であるとすると，$\cup_{i=1}^{l}G_i$ は F の *CGB* になる.

例 **2.1** $F=A*X+Y^2-1,Y^3-1\subseteq\mathbb{Q}[A,X,Y]$ にたいし *(X, Y* が主変数，*A* がパラメーター*)*，辞書式順序 $X>Y$ のもとで，$\mathcal{G}_1=\{(\{A=0\},\{Y-1\}),(\{A\neq 0\},A*X+Y^2-1,Y^3-1)\}$ は F の既約 *CGS* であるが忠実な *CGS* にはなっていない. $\mathcal{G}_2=\{(\{A=0\},\{-A*Y*X+Y-1\}),(\{A\neq 0\},A*X+Y^2-1,Y^3-1)\}$ は F の忠実な *CGS* である. したがって，$\{-A*Y*X+Y-1,A*X+Y^2-1,Y^3-1\}$ は F の *CGB* になる.

CGS の計算アルゴリズムは [17] によって導入されたのであるが，そのアイデアは非常に自然であり簡単である. まず，例として $F=\{A*X+Y^2-1,B*Y^3-1\}\subseteq\mathbb{Q}[A,B,X,Y]$ の CGS がどのようにして計算されるかみてみよう. ここで X,Y が主変数で A,B がパラメーターである. 単項順序は $X>Y$ なる辞書式順序とする. $f_1=A*X+Y^2-1,f_2=B*Y^3-1$ とおく. まずは f_1 と f_2 の S-多項式 $S(f_1,f_2)$ を計算しなければならないがパラメーターの値によって最大単項式が変わってしまうので，この点を考慮しなければならない. f_1 の最大単項式は $A\neq 0$ の場合は $A*X$ で $A=0$ の場合は Y^2，f_2 の最大単項式は $B\neq 0$ の場合は $B*Y^3$ で $B=0$ の場合は -1 である. 従って，以下の４つの場合によって $S(f_1,f_2)$ は異なったものになる.
ケース１. $A\neq 0$ かつ $B\neq 0$
この場合 f_1 の最大単項式は $A*X$ で f_2 の最大単項式は $B*Y^3$ なので，ブッフバーガーの第一クライテリオンによって，F はグレブナー基底であることが言えるので，$G_1=\{A*X+Y^2-1,B*Y^3-1\}$ となる.
ケース２. $A\neq 0$ かつ $B=0$
ケース３. $A=0$ かつ $B=0$
いずれの場合も $f_2=-1$ となるので $G_2=\{1\}$(ケース２) $G_3=\{1\}$(ケース３) となる.
ケース４. $A=0$ かつ $B\neq 0$
この場合 f_1 の最大単項式は Y^2 で f_2 の最大単項式は $B*Y^3$ なので，$S(f_1,f_2)=B*Y*f_1-f_2=-B*Y+1$，これを f_3 とおく. f_3 の最大単項式は $-B*Y$ なので，$S(f_2,f_3)=-B*f_2-B*Y^2*f_3=B*Y^2-B$. これを $f_1=Y^2-1$ で割った余りは０. $S(f_1,f_3)=-B*f_1-Y*f_3=-Y+B$. $-Y+B$ を f_3 で割った余りを求めると B^2-1，これを f_4 とおく.
サブケース１. $B^2-1=0$
この場合 $G_4=\{f_1,f_2,f_3\}=\{Y^2-1,B*Y^3-1,-B*Y+1\}$ となる.
サブケース２. $B^2-1\neq 0$
この場合 ０でない定数を含むことになるので $G_5=\{1\}$ となる.

以上より，求める CGS は $\{(\{A\neq 0,B\neq 0\},\{A*X+Y^2-1,B*Y^3-1\}),(\{A\neq 0,B=0\},\{1\}),(\{A=0,B=0\},\{1\}),(\{A=0,B^2-1=0,B\neq 0\},\{Y^2-1,B*Y^3-1,-B*Y+1\}),(\{A=0,B^2-1\neq 0,B\neq$

$0\},\{1\})\}$ となる.

上の計算における S-多項式の計算と割算には注意が必要である. ケース 4 において $S(f_1,f_2) = B*Y*f_1 - f_2$ としたが，体上の多項式環における S-多項式の通常の定義では $S(f_1,f_2) = Y*f_1 - \frac{1}{B}*f_2$ であり分数式が入る. これをさけるために最大単項式の係数で割るかわりに他方の多項式に乗ずることで分数式を避けることができる. また，$-Y+B$ を $-B*Y+1$ で割った余りは体上の多項式環における通常の割算の定義では $B-\frac{1}{B}$ であるが，この場合も分数式を避けるために分母の B を乗じた B^2-1 を余りとする.

さて，上でもとめた CGB は既約にはなっていない. 既約な CGB を求めるには通常のグレブナー基底の計算と同じように，各分割部において既約になるように変形すればよい. 上の場合は $(\{A=0, B^2-1=0, B\neq 0\},\{Y^2-1, B*Y^3-1, -B*Y+1\})$ が既約ではないので，これを既約化して $(\{A=0, B^2-1=0, B\neq 0\},\{-B*Y+1\})$ が得られる.
忠実な CGB の構成はちょっとした工夫をほどこすことで可能になる. 上の計算では，例えばケース 4 において $A=0$ なので $f_1 = Y^2-1$ として $S(f_1,f_2) = B*Y*(Y^2-1)-(B*Y^3-1)$ としたが $A*X$ もそのまま残して $S(f_1,f_2) = B*Y*(A*X+Y^2-1)-(B*Y^3-1) = A*B*X*Y-B*Y+1$ を f_3 とおくと，この分割において f_3 は $-B*Y+1$ と等しく，しかもイデアル $\langle A*X+Y^2-1, B*Y^3-1\rangle$ の中に留まっている. このように分割条件の等号の式 (この場合は A) を係数にもつ単項式を捨て去らずに，見かけ上残しておくことによって計算の途中に現れるすべての多項式がもとのイデアルの中に留まるようにできる. S-多項式や割算の計算において，0 になる単項式，すなわち分割条件の等号の式がかかった単項式 (この場合は $B*Y*(A*X)$) を無視することで，各分割部の中では全く同じ計算となり，忠実な CGB が計算できる.

一般の場合のアルゴリズムも同様である. 単項式の係数部 (パラメーターのみの多項式) が 0 に等しいか等しくないかによってパラメーター空間を分割しながら，それぞれの分割部においてブッフバーガーアルゴリズムを適用していくのである. アルゴリズムの停止性は通常のブッフバーガーアルゴリズムの停止性と同様にディクソンの補題により成り立つ. 数学的にきちんと述べると以下のようになる. 分割の木構造を考えると，これは有限に枝分かれしているので，もし計算が停止しないとするとケーニッヒの補題により無限のパス，すなわち分割部がより細分化されていくような無限の計算が含まれることになる. これには無限個の極小な単項が含まれることになりディクソンの補題に矛盾することになる.

上の例において分割部 $\{A=0, B^2-1=0, B\neq 0\}$ の表現は冗長である. $B^2-1=0$ なら $B\neq 0$ がなりたつので $B\neq 0$ は必要ない. 分割部の表現を必要最小限にするのは可能であるがそれにはパラメーターを変数とする多項式環におけるグレブナー基底の計算が必要になる. 上の例には現れていないが，矛盾する分割部すなわち $S_i = \{\bar{a}\in \overline{K}^m | p_1(\bar{a})=0,\ldots,p_s(\bar{a})=0, q_1(\bar{a})\neq 0,\ldots,q_t(\bar{a})\neq 0\} = \emptyset$ であるような分割部 $\{p_1=0,\ldots,p_s=0,q_1\neq 0,\ldots,q_t\neq 0\}$ が現れることもある. このような分割部の検証もやはりパラメーターを変数とする多項式環におけるグレブナー基底の計算により可能である.

3 グレブナー基底の安定性

1 節で見たように，$F = \{f_1(\bar{A}, \bar{X}), \ldots, f_k(\bar{A}, \bar{X})\} \subseteq K[\bar{A}, \bar{X}]$ にたいし，有理関数体 $K(\bar{A})$ を係数体とする多項式環 $K(\bar{A})[\bar{X}]$ における F のグレブナー基底を計算しても，それは一般には F の CGB にはならない. しかしながら，このようにして求めたグレブナー基底にたいしてもパラメーターに代入する $\bar{a}$ の値によっては $\langle f_1(\bar{a}, \bar{X}), \ldots, f_k(\bar{a}, \bar{X})\rangle$ のグレブナー基底になる. 前節の例 $F = \{A * X + Y^2 - 1, B * Y^3 - 1\} \subseteq \mathbb{Q}[A, B, X, Y]$ にたいして，$Q(A, B)[X, Y]$ におけるグレブナー基底は F 自身である. これに含まれる多項式の最大単項式の係数がすべて 0 に等しくないような代入にたいして，この場合は $A \neq 0$ かつ $B \neq 0$ となるような代入にたいしてはグレブナー基底になる. 一般に以下の定理が成り立つ.

定理 **3.1** $F = \{f_1(\bar{A}, \bar{X}), \ldots, f_k(\bar{A}, \bar{X})\} \subseteq K[\bar{A}, \bar{X}]$ にたいし，有理関数体 $K(\bar{A})$ を係数体とする多項式環 $K(\bar{A})[\bar{X}]$ における F のグレブナー基底を $G = \{g_1(\bar{A}, \bar{X}), \ldots, g_s(\bar{A}, \bar{X})\}$ とする. ただし係数は分数式は含まず，すべて $K[\bar{A}]$ の要素であるとする. 各 i にたいして $g_1(\bar{A}, \bar{X})$ の最大単項式の係数を $h_i(\bar{A})$ とすると，すべての i について $h_i(\bar{a}) \neq 0$ なる $\bar{a} \in \overline{K}^m$ にたいして $G(\bar{a}) = \{g_1(\bar{a}, \bar{X}), \ldots, g_s(\bar{a}, \bar{X})\}$ は $\{f_1(\bar{a}, \bar{X}), \ldots, f_k(\bar{a}, \bar{X})\}$ のグレブナー基底になる.

この定理がなりたつことは容易にわかる. 最大単項式の係数がいずれも 0 でないような代入にたいしては G の多項式による $K(\bar{A})[\bar{X}]$ における S-多項式と割算の計算が保存される，すなわち $g_i, g_j \in G$ の $K(\bar{A})[\bar{X}]$ における S-多項式を f とすると $f(\bar{a}, \bar{X})$ は $g_i(\bar{a}, \bar{X}), g_j(\bar{a}, \bar{X})$ の $K[\bar{X}]$ における S-多項式になり，任意の多項式 $p(\bar{A}, \bar{X}) \in K(\bar{A})[\bar{X}]$ にたいし $p(\bar{A}, \bar{X})$ を G で割った余りを $q(\bar{A}, \bar{X})$ とすると $q(\bar{a}, \bar{X})$ は $p(\bar{a}, \bar{X})$ を $G(\bar{a})$ で割った余りになる. したがって $G(\bar{a})$ はグレブナー基底になる.

$K(\bar{A})[\bar{X}]$ におけるグレブナー基底は $K[\bar{A}, \bar{X}]$ におけるグレブナー基底を用いて計算することができる.

定理 **3.2** $\bar{A}, \bar{X}$ の単項順序を各変数 X_i が $\bar{A}$ のみから構成されるどんな単項よりも大きいとする. *(*これを $\bar{X} >> \bar{A}$ と略記する.*)* $F = \{f_1(\bar{A}, \bar{X}), \ldots, f_k(\bar{A}, \bar{X})\} \subseteq K[\bar{A}, \bar{X}]$ にたいしこのような単項順序のもとでのグレブナー基底を G とすると，G は $K(\bar{A})[\bar{X}]$ においても F のグレブナー基底になる. ここで $\bar{X}$ の単項順序は $\bar{A}, \bar{X}$ の単項順序を $\bar{X}$ に制限したものとする.

テキスト [2] の Lemma 8.93 として同じ内容が証明されているので，証明はそちらを参照されたい. この定理により，次の定理は前の定理の一般化になっている.

定理 **3.3** $F = \{f_1(\bar{A}, \bar{X}), \ldots, f_k(\bar{A}, \bar{X})\} \subseteq K[\bar{A}, \bar{X}]$ にたいし $\bar{X} >> \bar{A}$ なる単項順序のもとでのグレブナー基底を G とする. 一般に G には $\bar{A}$ のみをむくむ多項式 $g'_1(\bar{A}), \ldots, g'_t(\bar{A})$ が含まれるので，$G = \{g_1(\bar{A}, \bar{X}), \ldots, g_s(\bar{A}, \bar{X}), g'_1(\bar{A}), \ldots, g'_t(\bar{A})\}$ とおくと，すべての g'_i にたいして $g'_i(\bar{a}) = 0$ でありかつ各 g_j について g_j を $\bar{X}$ の多項式とみなしたときの最大単項式の係数 $h_j(\bar{A})$ にたいして $h_j(\bar{a}) \neq 0$ ならば，$G(\bar{a}) = \{g_1(\bar{a}, \bar{X}), \ldots, g_s(\bar{a}, \bar{X})\}$ は $\{f_1(\bar{a}, \bar{X}), \ldots, f_k(\bar{a}, \bar{X})\}$ のグレブナー基底になる.

この定理は [7] の Theorem3.1 から容易に導かれる. 証明を知りたい読者はこの文献と後述の [15] の Lemma2.2 を参照されたい.

一般にパラメーター $\bar{A}$ を含んだ式のグレブナー基底 G にたいし，$G(\bar{a})$ が再びグレブナー基底になるとき，G は $\bar{a}$ で安定であるという. グレブナー基底の安定性に関する結果は CGB が発表される以前から知られている. 主な結果を以下に述べておく.

定理 **3.4** 主変数が一つしかないような $F = \{f_1(\bar{A}, X), \ldots, f_k(\bar{A}, X)\}$ にたいし，イデアル $I = \langle F \rangle \subseteq K[\bar{A}, X]$ の X に関する消去イデアル $I \cap K[\bar{A}]$ が 0 次元であるとき，単項順序 $X >> \bar{A}$ のもとでの F のグレブナー基底を G とすると，G は全ての $\bar{a} \in \overline{K}^m$ で安定である.

この結果は [5, 6] によって独立に発表された. 最近になって [4] で消去イデアルに関する条件がなくても成り立つことが示されている. これらの結果は連立代数方程式の解法には非常に有効であるが，主変数が一つしかないという非常に特殊な条件のもとでの結果なので，主変数が複数個あるときの一般の CGB や CGS の計算には使えそうにない.
[5, 6] の結果はその後 [1, 7] によって以下の形へ拡張された.

定理 **3.5** $F = \{f_1(\bar{A}, \bar{X}), \ldots, f_k(\bar{A}, \bar{X})\}$ にたいし，イデアル $I = \langle F \rangle \subseteq K[\bar{A}, \bar{X}]$ の $\bar{X}$ に関する消去イデアル $I \cap K[\bar{A}]$ が 0 次元の根基イデアルあるとき，単項順序 $\bar{X} >> \bar{A}$ のもとでの F のグレブナー基底を G とすると，G は全ての $\bar{a} \in \overline{K}^m$ で安定である.

[1] では，単項順序 $\bar{X} >> \bar{A}$ が辞書式順序の場合に限って証明されているが，[7] において $\bar{X} >> \bar{A}$ なるすべての単項順序にたいしても成り立つことが示されている. この結果は後述する [15] のアルゴリズムの高速化に使用されている.

4 最近の研究動向

CGB あるいは CGS の研究は [17] が発表されてから１０年近くの間ほとんど進展がなかった. 主な理由として考えられることは，まず [17] で発表された CGB と CGS の構成方法が非常に自然で，数学的にも特に難しい道具を必要としない初歩的なものであったため，野心的な研究者の関心を引かなかったことがあげられる. 次に考えられることは CGB や CGS の重要性を認識している研究者があまり多くはいなかった点であろう. 個々の問題を解くときに CGB や CGS が本質的に必要になる場合でも，$K(\bar{A})[\bar{X}]$ におけるグレブナー基底や単項順序 $\bar{X} >> \bar{A}$ のもとでの $K[\bar{A}, \bar{X}]$ におけるグレブナー基底の計算だけで何とか解決できると錯覚している人は今でも結構見かける. 例えば，代数的閉体上の多項式による等式と非等式から構成される一階論理式の限量子消去は CGS を使って初めて可能になるのであるが，消去イデアルの計算だけで得られる不十分な解で満足している人は結構多い. しかしながら，最近になって計算アルゴリズムの効率化と CGB と CGS の数学理論の両方において飛躍的な発展があった. 以下では双方について簡単に紹介する.

4.1 効率的計算

２節で述べた CGB と CGS の構成方法はグレブナー基底の安定性に関する結果を一切利用していない. 一般に $K(\bar{A})[\bar{X}]$ におけるグレブナー基底や単項順序 $\bar{X} >> \bar{A}$ のもとでの $K[\bar{A}, \bar{X}]$

におけるグレブナー基底の計算は 2 節で述べた CGB と CGS の構成方法による計算に比べ圧倒的に高速である. したがって，いったんこれらのグレブナー基底を求めておけば，安定性に関する結果を使って，扱わなければいけないパラメーター空間を小さくすることができる. [10] では，定理 3.1 にもとづくアルゴリズムを提唱している. この結果はその後 [18] で一般化され [9] において，より効率的なアルゴリズムになっている. このアルゴリズムは数式処理システム Maple 上で実装され公開されているが，残念ながら Maple のグレブナー基底の計算が極めて遅いせいもあって，Weispfennig らのグループが数式処理システム Reduce 上で開発した [17] のアルゴリズムにもとづいたプログラム ([3]) よりも一般には低速である.
本号の「グレブナー基底を用いた包括的グレブナー基底計算 (鈴木 晃)」で詳しい解説がされている [15] のアルゴリズムは基本的に定理 3.3 にもとづいている. このアルゴリズム (Suzuki-Sato Algorithm) は体 K 上の多項式環におけるグレブナー基底の計算だけを用いて構成されている. このため，通常のグレブナー基底の計算を備えた数式処理システムであれば，それを用いて容易に実装することができる. 実際，このアルゴリズムは Risa/Asir，Singular，Maple 上で実装されている. K が有理数体 $\mathbb{Q}$ のとき，パラメーター $\bar{A}$ の個数がそれほど多くない場合は，一般に $\mathbb{Q}[\bar{A},\bar{X}]$ におけるグレブナー基底の計算の方が $\mathbb{Q}(\bar{A})[\bar{X}]$ におけるグレブナー基底の計算よりも圧倒的に高速である. 非常に低速な Maple9.5 のグレブナー基底を用いたプログラムでも Weispfenning や Montes のプログラムよりも高速になっている. Risa/Asir 版のものは，いろいろな工夫も採り入れてあるのでさらに高速になっている. 実際，本稿の冒頭で紹介した例の CGS や CGB は Weispfenning や Montes のプログラムでは数時間計算させても計算できなかったのが数十秒程度で計算される.
Suzuki-Sato Algorithm にはパラメーターの個数が多い場合には，著しく性能が劣るという欠点がある. この点を改良したアルゴリズムが [11] で発表されている. [8] では特別な場合の CGB 計算の効率的計算法が論じられているが，この手法は一般の CGS 計算の高速化にも応用できると思われる. 詳細は本号の「Dynamic Evaluation を用いた Discrete Comprehensive Gröbner Bases の計算 (倉田陽介)」を参照されたい.

4.2 標準形

体を係数とする多項式環における既約グレブナー基底は唯一つ存在する (最大単項の係数を 1 としていない本稿の定義では定数倍を除いてということになるが) ので，既約グレブナー基底がイデアルの標準形を与える. それではパラメーターを含むイデアルの標準形はどのようにして与えられるのであろうか. CGB あるいは CGS が何らかの形で標準形を与えてくれそうに思われる. しかしながら一般に CGB や CGS はアルゴリズムによって変わったものになり，アルゴリズムと独立にこれらの標準形を定義することは長い間できなかった. [18] においてこの問題に進展が生まれたものの，本質的な回答をもたらしたのは [14] である.
多項式 $f(\bar{A}) \in K[\bar{A}]$ は $\overline{K}^m$ から $\overline{K}$ への関数とみなすことができる. $\overline{K}^m$ から $\overline{K}$ への関数全体 ($K^{(\overline{K}^m)}$ と記す) は可換環になるので $K[\bar{A}]$ は $K^{(\overline{K}^m)}$ の部分環になる. $K^{(\overline{K}^m)}$ は整域ではないが可換フォンノイマン正規環とよばれる構造の環になっている. 可換フォンノイマン正規環を係数環とする多項式環においてグレブナー基底およびその標準形が構成できることが [16] で発表されている. [14] では以下のことが示された. $F = \{f_1(\bar{A},\bar{X}),\ldots,f_k(\bar{A},\bar{X})\} \subseteq K[\bar{A},\bar{X}]$ にた

いし，F の $K^{(\overline{K}^m)}[\bar{X}]$ におけるグレブナー基底が本質的に F の CGS になる. またこのグレブナー基底の標準形がパラメーターを含むイデアルの標準形になる. 実際 [17] や [9, 10] のアルゴリズムは可換フォンノイマン正規環を係数環とする多項式環におけるグレブナー基底の計算アルゴリズムの一つになっているのである.
以上の結果については, 本号の「Comprehensive Gröbner bases and von Neumann regular rings(鍋島克輔)」で詳しい解説が与えられているので参照されたい.

5 利用可能なソフトウエア

商用の Maple や Mathematica あるいはフリーに利用できる Maxima や Mupad 等の汎用数式処理システムのほとんどがグレブナー基底計算プログラムを有している. しかしながら, CGB や CGS を計算することができるものはほとんど皆無である. 以下では, 現在入手可能なソフトウエアを紹介する.

5.1 Reduce の Redlog パッケージ

古くからある汎用数式処理システム Reduce の Redlog パッケージに組み込まれている gsys は忠実な CGS を計算する. cgb は gsys を利用して CGB を計算する. 3.7 版以降の Reduce には標準でこのパッケージが組み込まれているので特別なインストールの必要はない. Reduce は 3.8 版が現時点での最新版であるが 3.7 版と比べ多倍長整数演算がかなり高速になっているので gsys や cgb を使用するときは 3.8 版を用いることを推奨する. Reduce は本来商用の数式処理システムであるが Maple や Mathematica に比べるとかなり安価に購入できる. 3.8 版のシングルライセンスは 1 万円程である. パッケージの使用法は次のサイトから入手できる.
http://www.fmi.uni-passau.de/~redlog/
http://www.fmi.uni-passau.de/~reduce/cgb/
以下では大まかな使用例を紹介する.

```
REDUCE 3.8, 15-Apr-2004, patched to 30-May-2005 ...
1: load cgb;
2: on cgbgs;
3: cgb{x};
{x}
4: torder({x,y},lex);
{{},lex}
5: cgb{a*x+y^2-1,b*y^3-1};
     2     2                    2         2
{(a*b )*x*y  + (a*b)*x*y + (a*b )*x - (b  - 1),
 (a*b)*x*y - b*y + 1,
        2
 a*x + y  - 1,
    3
```

```
 b*y  - 1}
6: gsys{a*x+y^2-1,b*y^3-1};
{{a <> 0 and b <> 0,
          2        3
  {a*x + y  - 1,b*y  - 1}},
                       3
 {a <> 0 and b = 0,{b*y  - 1}},
 {b + 1 <> 0 and b - 1 <> 0 and b <> 0 and a = 0,
       2     2                   2        2
  {(a*b )*x*y  + (a*b)*x*y + (a*b )*x - (b  - 1)}},
                         2
 {b <> 0 and a = 0 and b  - 1 = 0,
  {(a*b)*x*y - b*y + 1,
          2
   a*x + y  - 1,
      3
   b*y  - 1}},
                      3
 {a = 0 and b = 0,{b*y  - 1}}}

7:
```

torder で単項順序を指定する. ここで指定されない変数は自動的にパラメーターとみなされる. 辞書式順序 lex の他に全次数逆辞書式順序 revgradlex 等が利用できる. 3.8 版にはちょっとしたバグがあるようで，最初に単項順序の指定をするとエラーがでるので，3: cgb{x}; のような計算を最初にやらせておかなければいけない. on cgbgs; はパラメーターの矛盾する分割部を検出するためのオプションである. ただし非等式は考慮していないので不完全なものになっている. 分割部として {a*b<>0 and a=0} のようなものが出てくる場合もある. このオプションはなくてもよいが，大抵の場合はオプション付きの方が高速である.

5.2 Montes のプログラム

[9, 10] で発表されたアルゴリズムを筆者の Montes が Maple 上に実装したプログラムが公開されている. 以下のサイトから入手可能である.

http://www-ma2.upc.edu/~montes/

グラフィカルなインターフェースも含め種々のツールが利用できるようになっているが，上記の論文の内容を理解していないと使いこなせないかもしれない.

5.3 鈴木晃のプログラム

[15] で発表されたアルゴリズムを Risa/Asir，Singular，Maple 上で実装したプログラムが以下のサイトから入手できる.

http://kurt.scitec.kobe-u.ac.jp/~sakira/CGBusingGB/
アルゴリズムの詳細は上記の論文を参照されたいが，詳しい理論がわからなくても，本稿で紹介した CGB と CGS の概念さえ理解していれば使用可能である. Risa/Asir 版が一番充実しているので，これについて以下では大まかな使用例を紹介する.

```
[1247] load("./acgs.rr")$
_0
[1316] GB = acgs.rcgs([a*x+y^2-1,b*y^3-1],[x,y],2)$
[1317] acgs.bases2str(GB);
((b)(a)!=0)[b*y^3-1,a*x+y^2-1]
((a)=0,(b-1)(b+1)=0)[y-b]
[1318] GB = acgs.cgb([a*x+y^2-1,b*y^3-1],[x,y],2)$
[1319] acgs.bases2str(GB);
((a)=0)[(-b^2*a*y^2-b*a*y-b^2*a)*x+b^2-1,(b*a*y^2+b*a)*x+y-b]
((b)=0)[b*y^3-1]
((b)(a)!=0)[b*y^3-1,a*x+y^2-1]
[1320]
```

上の例は KNOPPIX/MATH の Risa/Asir を実行している. acgs.rr では *gr* と *primdec*(一部のオプションのみ) を使用しているのでそれらが自動的に load されない環境ではまずこれらのプログラムを load しておく必要がある. 単項順序の指定は gr と同様である. Redlog と同様ここで指定されない変数は自動的にパラメーターとみなされる. acgs.rcgs は既約 CGS を計算する. グレブナー基底が {1} になるときは出力されないようになっている. acgs.cgb は忠実な CGS を計算する. これは Redlog の gsys と基本的に同じものであるが，分割のオーバーラップを許している. 分割部 $a = 0$ と $b = 0$ には共通部分が含まれる. 以下参考までに，本稿の冒頭であげた例について acgs.rsgs で計算した分割部とその最小多項式をあげておく.

```
((a)(a-1)(4*a+1)(25*a-36)(27*a+28)(32*a-27)(32*a^2+13*a+4)
(256*a^2+32*a+9)(1536*a^3-512*a^2-376*a-207)!=0)
[3456*s^7+(1728*a^2+7632*a+947)*s^6+
(-544*a^3+5588*a^2+1442*a+108)*s^5+
(-1856*a^5-6500*a^4+449*a^3+495*a^2+108*a)*s^4+
(-3040*a^6-5640*a^5-1458*a^4-108*a^3)*s^3+
(128*a^8-1132*a^7-1396*a^6-495*a^5-108*a^4)*s^2+
(128*a^9+52*a^8+16*a^7)*s]

((a)=0)
[-3456*s^3-947*s^2-108*s,161*s*r+144*s^2+117*s]

((a-1)=0)
[3456*s^5+3395*s^4-3652*s^3-3395*s^2+196*s]
```

```
((4*a+1)=0)
[884736*s^5-218368*s^4+13120*s^3-3412*s^2-11*s]

((25*a-36)=0)
[-10546875000000*s^6-291390771484375*s^5+132814589062500*s^4+
897144773400000*s^3+543787731244800*s^2-80790550806528*s]

((27*a+28)=0)
[-36150980669568*s^6+40750049294487*s^5-40386053130444*s^4+
42011688712896*s^3-75493893888*s^2-3831991697408*s]

((32*a-27)=0)
[-3710851743744*s^6-6121066594304*s^5-178391089152*s^4+
3774177773568*s^3+1446080721696*s^2-58500493839*s]

((32*a^2+13*a+4)=0)
[-115964116992*s^6+(-232532213760*a-24528289792)*s^5+
(28518121472*a+20740833280)*s^4+(8863488000*a-49160192)*s^3+
(-1418717024*a-634712960)*s^2+(10231593*a+5925940)*s]

((256*a^2+32*a+9)=0)
[-115964116992*s^6+(-263335182336*a-21583888384)*s^5+
(-5371854848*a+5723062272)*s^4+(1404076032*a+273715200)*s^3+
(123064064*a-20777328)*s^2+(-212432*a-8919)*s]

((1536*a^3-512*a^2-376*a-207)=0)
[2293235712*s^6+(-382205952*a^2+4299816960*a+413392896)*s^5+
(1092206592*a^2-287815680*a-373434624)*s^4+
(-1593879552*a^2-999447552*a-233575488)*s^3+
(-697857024*a^2-288580092*a-92661273)*s^2+
(25585844*a^2+11829649*a+3816873)*s]
```

参 考 文 献

[1] Becker,T. (1994). On Gröbner Bases under Specialization. Applicable Algebra in Engineering, Communication and Computing. 5, 1–8.

[2] Becker,T. and Weispfenning,V. (1993). Gröbner Bases. Graduate Texts in Mathematics 141, Springer-Verlag.

[3] Dolzmann, A. and Sturm, T. (1997). Redlog: Computer algebra meets computer logic, ACM SIGSAM Bulletin, 31, 2, 2–9.

[4] Fortuna, E. Gianni, P. Trager, B. (2001). Degree reduction under specialization. Journal of Pure and Applied Algebra 164, 153–163.

[5] Gianni,P. (1989). Properties of Gröbner bases under specializations. EUROCAL '87, J. H. Davenport Ed., Springer LNCS 378, 293–297.

[6] Kalkbrener,M. (1989). Solving systems of algebraic equations by using Gröbner bases. EUROCAL '87, J. H. Davenport Ed., Springer LNCS 378, 282–292.

[7] Kalkbrener, K. (1997). On the stability of gröbner bases under specialization, J. Symb. Comp. 24, 1, 51–58.

[8] Kurata, Y. and Noro, M.(2007). To appear in International Symposium on Symbolic and Algebraic Computation (ISSAC 2007), Proceedings.

[9] Manubens, M. and Montes, A. (2006). Improving DISPGB Algorithm Using the Discriminant Ideal, J. Symb. Comp. 41, 11, 1245–1263.

[10] Montes, A. (2002). A new algorithm for discussing Gröbner basis with parameters, J. Symb. Comp. 33, 1-2, 183–208.

[11] Nabeshima, K.(2007). To appear in International Symposium on Symbolic and Algebraic Computation (ISSAC 2007), Proceedings.

[12] Sato, Y. (1998). A new type of canonical Gröbner bases in polynomial rings over Von Neumann regular rings. International Symposium on Symbolic and Algebraic Computation (ISSAC 98), Proceedings, 317–321.

[13] Sato, Y. and Suzuki, A. (2001). Discrete Comprehensive Gröbner Bases. International Symposium on Symbolic and Algebraic Computation (ISSAC 2001), Proceedings, 292–296.

[14] Suzuki, A. and Sato, Y. (2003). An Alternative approach to Comprehensive Gröbner Bases. J. Symb. Comp. 36/3-4, 649–667.

[15] Suzuki, A. and Sato, Y. (2006). A Simple Algorithm to compute Comprehensive Gröbner Bases using Gröbner Bases. International Symposium on Symbolic and Algebraic Computation (ISSAC 2006), Proceedings, pp 326–331.

[16] Weispfenning, V. (1989). Gröbner bases in polynomial ideals over commutative regular rings, EUROCAL '87, J. H. Davenport Ed., Springer LNCS **378**, 336–347.

[17] Weispfenning, V. (1992). Comprehensive Gröbner bases, J. Symb. Comp. 14/1, 1–29.

[18] Weispfenning, V. (2003). Canonical Comprehensive Gröbner bases, J. Symb. Comp. 36, 669-683.