数理解析研究所講究録
第1815巻 2012年 58-69

近似グレブナー基底の理論と算法

佐々木 建昭 (Tateaki Sasaki) *
筑波大学 名誉教授 (数学系)
PROFESSOR EMERITUS, UNIVERSITY OF TSUKUBA

Abstract

近似グレブナー基底は過去15年間、世界の多くの著名な研究者が追求したが、未だに概念すら確立されていない。筆者は過去4年間、浮動小数グレブナー基底を研究し、Buchberger算法の不安定性のメカニズムを解明して不安定性を除去する方法を考案するとともに、組織的で不正確な項キャンセルの大きさをかなり正確に知る方法も提案した。後者は近似グレブナー基底の算法開発に不可欠である。これらの成果の上に、本稿では近似多項式イデアルを定義することからスタートして、近似グレブナー基底の理論を構築し、算法を提案する。理論では、従来の項簡約に代わり"精度防御簡約"なる項簡約を導入する。得られる基底は従来のグレブナー基底に比べて冗長な要素を含むが、これにより精度を可能な限り失わないで簡約ができるようになる。

1 はじめに

本稿では、**浮動小数グレブナー基底**とは、『浮動小数を用いて既知の算法でグレブナー基底を精度よく計算すること』とし、**近似グレブナー基底**とは、『近似イデアルを定義し、それを基礎に近似グレブナー基底を定め、理論を構築して算法を考案すること』とする。このことは例えば、多項式GCDを浮動小数係数で計算することは近似GCDとは言わず、許容度に基づいて定義されたGCDを近似GCDと言うことと類似である。この意味から言うと、近似グレブナー基底は未だ定義さえされておらず、もちろん算法もない。一方、浮動小数グレブナー基底の方は近年非常に研究が進展し、計算時間さえ気にしなければ、ある程度大きな系も計算できるようになった。本稿は、浮動小数グレブナー基底の研究で得られた知見に基づいて、近似グレブナー基底の理論構築と算法開発に挑む。入力多項式の係数は浮動小数とは限定しないが、誤差を含み浮動小数で表されているとする。従来の研究については論文 [6, 13, 15, 14, 3, 16, 17, 18, 5, 1, 10, 11, 8, 7, 12, 9] を参照。

浮動小数グレブナー基底をBuchberger算法で計算すると、多くの場合、計算は非常に不安定である。理由は、巨大な桁落ちが頻繁に生じて係数の精度を落とすこと、有意な桁が失われた"完全誤差項"が多く生成され、それが主項になって以後の計算を完全に破壊すること、の2点である。そのため、多くの研究者が行列法（入力多項式の係数ベクトル

*sasaki@math.tsukuba.ac.jp

からなる行列を変換することでグレブナー基底の要素多項式を計算する方法）を研究しているが、膨大な計算時間を要するなどの問題がある。本稿では、愚直に Buchberger 算法(の拡張)を考察する。

浮動小数グレブナー基底研究で得られた知見は2章で説明するが、重要なのは次の2点である；論文 [11] 参照。第1点：非常に大きな桁落ちと完全誤差項は**正確な項キャンセル**による。このうち、組織的な項キャンセルによる巨大桁落ちは数値行列をピボッティングなしで消去する際の桁落ちのようなものだが、計算精度を最初に高精度化することで完全に無害化できる (**高精度法**)。第2点：入力多項式間に**近似線形従属関係**が存在する場合、組織的で**不正確な項キャンセル**が発生する。後者は高精度法でも無害化できず、本質的な項キャンセルであると言える。

グレブナー基底はイデアルの基底であるから、近似グレブナー基底には当然、基底要素が張る**近似イデアル**がなければならない。これが近似グレブナー基底が近似 GCD や近似因数分解と大きく異なる点である。近似グレブナー基底と銘打った論文はこれまで数多く発表されたが、近似イデアルまで考察した論文を筆者は知らない。近似イデアルの定義は第3章で行うが、そこでは理論的には入力多項式間の近似線形従属関係が本質的な役割を演じ、実際的には不正確な組織的項キャンセルの推定法が重要な役割を果たす。いずれも浮動小数グレブナー基底の研究で得られた成果が基礎になっている。

近似イデアルを定義できれば、近似グレブナー基底は近似なしの場合の定式化を倣って定式化できる。ただ一つの問題は、近似イデアルには精度が落ちた多項式も含まれるが、それらが近似グレブナー基底により与許容度で近似ゼロに簡約できることを如何に保証するかである。筆者は、従来の項簡約に替えて “精度防御簡約” と命名した新たな簡約を導入することでこれを保証する。精度防御簡約の導入により、当然、近似グレブナー基底は浮動小数グレブナー基底とは異なり一見、冗長な要素を含むものとなる。これらのことは全ての読者にとって目新しいことと思われるので、第6章で例を用いて説明する。

2 浮動小数グレブナー基底からの知見

本稿では precision を**計算精度**、accuracy を**精度**と区別する。固定計算精度の浮動小数の集合を $\mathbb{F}$ と表す。変数 $x_1, \ldots, x_\nu$ を $\boldsymbol{x}$ と表し、$\mathbb{F}[\boldsymbol{x}]$ の要素を P, Q などと表す。$\|P\|$ は多項式 P の**ノルム**を表す：本稿では無限大ノルム (数係数の絶対値の最大値) を採用する。$\succ$ は $\mathbb{F}[\boldsymbol{x}]$ の要素に対し設定された**順序**とする。$\succ$ に関する P の最高順位項を**主項**といい $\mathrm{lt}(P)$ と表す。主項の係数を**主係数**といい $\mathrm{lc}(P)$ と表す。変数のべきの積を**べき積**という。主項に現れるべき積を**主べき積**といって $\mathrm{lpp}(P)$ と表す：$\mathrm{lt}(P) = \mathrm{lc}(P)\mathrm{lpp}(P)$。$P$ と Q の **S 多項式**を $\mathrm{Spol}(P,Q)$ と表し、P の Q による**主項簡約**を $\mathrm{Lred}(P,Q)$ と表し $P \xrightarrow{Q}$ と図示することにする。$P \xrightarrow{Q}\!\!\!\!\rightarrow \tilde{P}$ は $\tilde{P}$ が G 既約になるまでの簡約を意味する。なお、本稿では S 多項式生成と主項簡約だけを用いてグレブナー基底を計算する。簡約基底が欲しい場合には計算の最後で非主項を簡約すればよい。

浮動小数 f が誤差 e を含むとき、$|e/f|$ を f の**精度**といい $\mathrm{acc}(f)$ と表す。e は未知だが、大雑把な大きさは分っているとする。浮動小数 f, g は同符号で、上位 k ビットが同じで

次のビットが異なるとする。このとき $f-g$ では上位の k ビットがキャンセルするので、$\mathrm{acc}(f-g) = 2^k \max\{\mathrm{acc}(f), \mathrm{acc}(g)\}$ となり誤差が 2^k 倍に拡大される。これを大きさ 2^k の**桁落ち**といい、そのときに生じる誤差を**桁落ち誤差**という。浮動小数演算では**丸め誤差**も無視できない。理論 (特に初期理論) では丸め誤差が無視されることが多々あるが、その場合には等式や不等式はガード数を考慮して、下記のように解釈する必要がある。

前提 1 浮動小数 f と g に対し、"$f = g$" および "$f > g$" はそれぞれ $|f-g| < e$ および $f - g > e$ と解釈する。ここで、e は演算中に発生する丸め誤差の上界である。 ◇

論文 [12] において著者らは、Buchberger 算法における項キャンセルを次のように分類した；この分類は他の多くの演算にも適用できる。

- **組織的**な項キャンセル
  - 正確なキャンセル ⇐ 無害化が可能
  - 不正確なキャンセル ⇐ 近似線形従属
- **偶発的**な項キャンセル
  - 正確なキャンセル ⇐ 項除去が可能
  - 不正確なキャンセル ⇐ 対処は難しい

ここで、**組織的で正確な項キャンセル**とは多項式の一部が異なる経路を経て自分自身とキャンセルするものであり、**偶発的な項キャンセル**とは異なる二つの項の係数が何回かの演算で偶然に値が非常に似てきてキャンセルするものである。組織的な項キャンセルは、その計算に関与する多項式の行列表現を使って解析できる；詳細は本報告書の筆者の別稿「グレブナー基底算法における項キャンセルの一般論」を参照されたい。それによると、組織的で正確な項キャンセルは頻繁に起きることが分かり、主項が相対的に小さい多項式がからむと大きな桁落ちが発生しうることも分る。行列表現はさらに、行列の幾つかの行の間に近似線形従属関係が成立する場合にも大きな項キャンセルが生じることを教える。これが**組織的で不正確な項キャンセル**であり、これら以外に組織的項キャンセルはない。一方、偶発的な項キャンセルは数値計算でも起き、入力多項式が実験データから作られた場合などでは、大きな桁落ちが発生する確率は極めて低い。しかし、本稿の例のように、小さい有理数を浮動小数に変換して入力多項式を生成する場合などにはかなりの頻度で発生する。偶発的項キャンセルについては、正確なものへは対処できるが、不正確なものへの対処は厄介である（異なる経路で計算する、などの不完全な手はある）。

組織的項キャンセルへの対処法を説明する。正確なものには高精度法 [11] で対処する。浮動小数の加減算 $(f+g)-f'$ で説明する。簡単のため、$\mathrm{acc}(f) = \mathrm{acc}(g) = \mathrm{acc}(f')$, $f = f'$, $|f| \gg |g|$ とする ($f-f'$ は誤差のみの数)。$f+g$ では g の下位桁に f の誤差が入り込む。次の減算 $(f+g)-f'$ では、g に加えられた f は g から除かれるが、g の下位桁には誤差 $f-f'$ が残り下位桁が汚れる。そこで、入力直後に f と g を高精度化しておけば、高精度化された部分には無意味な数が詰め込まれるが、コンピュータは詰め込まれた数も含めて計算精度の範囲で正しく計算するから、g の入力部分には誤差 $f-f'$ は入らない。これが高精度法であり、計算時間が増えるものの、完璧な方法である。

不正確な組織的項キャンセルについては、それを除去することは必要なく、その大きさを決定することが重要である。現在までに三つの推定法が提案された [11, 8, 12]。第一の

方法は、項キャンセルによる桁落ちが微小主項の多項式により引き起こされることに着目し、微小主項の多項式が関与するS多項式生成と項簡約をモニターする。そして、多項式全体に亘る組織的桁落ちが起きたら、その桁落ちが起きた加減算を正確にキャンセルする多項式を除外して再実行する。すると、後に残った桁落ちは不正確なものとみなせる [11]。この方法はかなり複雑であり、ほんの少しだけ小さい主項の多項式による桁落ちを見逃す欠点がある。第二の方法は、正確な組織的項キャンセルがピボッティングなしの行列消去に対応することに着目し、計算の節目毎に、節目間の計算を数値行列に変換し、その行列をピボッティングつきガウス法で消去して、多項式を再計算する。すると、再計算された多項式に残る桁落ちは不正確な項キャンセルによるとみなせる [12]。この方法は魅力的であるが、正確な組織的項キャンセルを十分に除去するには行列サイズをかなり大きくする(多くの計算ステップを行列に取り込む)必要があり、計算時間がかかるのが難点である。第三の方法は、入力多項式を高精度化したあとで、各係数の初期誤差に対応するビットに**マーク**をし(そのビットが表す大きさ程度の数を加える)、二通りにマークした初期系AとBを用意する。そして、系Aと系Bから出発して二つのグレブナー基底を計算する。正確な組織的項キャンセルは二つの系の各係数の下位ビットを損傷するだけだが、不正確な項キャンセルは上位ビットを喪失させる。したがって、系Aと系Bの対応する係数で上位から一致するビット数を見れば、不正確な組織的桁落ち量を係数毎に推定できる [8]。この方法は桁落ちの推定値が統計的にふらつく欠点はあるが、現在最も扱い易い。

なお、桁落ち誤差が累積して係数の精度が全て失われると計算は破綻するので、各係数に起きた桁落ちの総量を知るため、係数は有効浮動小数で表す [4]。

3 近似グレブナー基底の定義

偶発的項キャンセルのうち正確なものは、初期精度$\varepsilon_{\text{init}}$を完全に失わせるので、マーキング法で対応する項を検出し除去できる。しかし、不正確なものへの対処法は探求中で、未だ確たる方法を示せない。したがって、本稿では次の前提の下で議論を進める。

前提2 以下では『**不正確な偶発的項キャンセルが起きないか回避できるなら**』との条件下で議論を進めるが、この条件は補題や定理には記述しない。 ◇

定義1 (**多項式の精度**) 多項式Pの各係数の精度を組織的で不正確な項キャンセルだけで定める。このとき、係数の精度の最大値をPの精度といい$\mathrm{acc}(P)$と表す。 ◇

注釈 前提2の下では、$\mathrm{acc}(P)$はPに生じる不正確な組織的項キャンセルだけで定まる。このキャンセルは入力多項式間の近似線形従属性に基づくから、簡約手順を行列表現してみれば分るように、計算のある段階の加減算で答となる多項式のノルムが落ちる。一方、不正確な偶発的項キャンセルは一部の係数だけを桁落ちさせる。 ◇

入力多項式系を$\{F_1, \cdots, F_t\} \subset \mathbb{F}[\boldsymbol{x}]$とし、**入力精度**を$\varepsilon_{\text{init}} = \max\{\mathrm{acc}(F_1), \cdots, \mathrm{acc}(F_r)\}$とする。最初にやらねばならぬことは**近似イデアル**の定義である。近似イデアルの精度としては$\varepsilon_{\text{init}}$をとるのが自然である。

従来のイデアルの定義 $\{P = a_1F_1 + \cdots + a_rF_r \mid a_i \in \mathbb{C}[\boldsymbol{x}]\}$ から出発する。もとの定義とはズレるが、$(a_1, \cdots, a_r)$ を P に対するシジジー (syzygy) と呼ぶことにする。許容度を入れるため、まず各入力多項式を規格化する：$\|F_1\| = \cdots = \|F_r\| = 1$。近似イデアルの定義では、$\|P\|$ は意味を持たず、P の各係数がどれほどの精度を持っているかが本質的でなければならない。実際、イデアルの要素か否かが $\|P\|$ で決まるならば、$\|P\| < \varepsilon_{\text{init}}$ であっても、巨大な数 b をかければ $\|bP\| > \varepsilon_{\text{init}}$ となるので、おかしなことになる。一方、P の各係数が十分な精度を持っているなら、$\|P\|$ に拘らずイデアルに含めるべきである。そこで、$a_1F_1 + \cdots + a_rF_r$ における桁落ちを考える。これは浮動小数グレブナー基底で扱ったテーマに他ならない。グレブナー基底計算における組織的で正確な項キャンセルはシジジーが一意的でない（$b_1F_1 + \cdots + b_rF_r = 0$ を満たす $(b_1, \cdots, b_r)$ は無数に存在する）ことに対応する。そこで、正確な組織的項キャンセルを除いてシジジーを定義する。

定義 2 (極小シジジー) 多項式 $F_1, \ldots, F_r$ は $\|F_1\| = \cdots = \|F_r\| = 1$ と規格化されているとする。$P = a_1F_1 + \cdots + a_rF_r$ を満たすシジジーの中で $\max\{\|a_1\|, \cdots, \|a_r\|\}$ が最小となるものを P の極小シジジーといい、$(A_1, \cdots, A_r)$ と表す。 ♢

注釈 極小シジジーは、理論上導入するだけで、実際に計算する訳ではない。実際上は、P に生じる不正確な組織的項キャンセルの大きさが分かればよく、2章に解説した方法で推定できる。なお、極小シジジーに近いシジジーは、P を $F_1, \ldots, F_r$ の係数ベクトルで行列表現し、ピボッティングつきガウス法で消去すれば、副産物として得られる。 ♢

定義 3 (近似ゼロ) 多項式 $F_1, \ldots, F_r \in \mathbb{F}[\boldsymbol{x}]$ は $\|F_1\| = \cdots = \|F_r\| = 1$ と規格化されており、$P = A_1F_1 + \cdots + A_rF_r$ において $(A_1, \cdots, A_r)$ は極小シジジーとする。P が次式を満たすとき、P は許容度 ε_{app} で近似ゼロといい、$P = 0$ $(\text{tol } \varepsilon_{\text{app}})$ と表す。

$$\|P\| < \varepsilon_{\text{app}} \max\{\|A_1\|, \cdots, \|A_r\|\}, \quad \text{ただし} \ \ 0 < \varepsilon_{\text{app}} \ll 1. \qquad \diamondsuit \tag{3.1}$$

定義 4 (近似イデアル) 多項式 $F_1, \ldots, F_r \in \mathbb{F}[\boldsymbol{x}]$ は $\|F_1\| = \cdots = \|F_r\| = 1$ と規格化されており、それらの初期精度を $\varepsilon_{\text{init}}$ とする。下記集合を $F_1, \ldots, F_r$ で生成される精度 $\varepsilon_{\text{init}}$ の近似イデアルといい、従来のイデアル $\langle F_1, \cdots, F_r\rangle$ に倣い $\langle F_1, \cdots, F_r; \varepsilon_{\text{init}}\rangle$ と表す。

$$\{P = A_1F_1 + \cdots + A_rF_r \mid P \neq 0 \ \ (\text{tol } \varepsilon_{\text{init}}),\ (A_1, \cdots, A_r) \text{は極小シジジー}\}. \quad \diamondsuit \tag{3.2}$$

上記の近似イデアルは、与多項式 $F_1, \ldots, F_r$ で精度ギリギリまで表現できる多項式を全て含み、精度が失われて無意味な多項式は一切含まない。線形結合 $A_1F_1 + \cdots + A_rF_r$ は従来の定義を踏襲するのみならず、2章でその重要性を強調した近似線形従属性も取り込んでいる。したがって、上記の定義は合理的であるのみならず、近似グレブナー基底の定式化に有用であろう。なお、従来のイデアルでは $F_1, \ldots, F_r$ の中の最高順位より P の順位が低くなり得るが、これに対応して、近似イデアルでは P の精度が $\varepsilon_{\text{init}}$ より大きくなり得る (精度劣化を起こす)。すなわち次式が成立する。

$$\langle F_1, \cdots, F_r; \varepsilon_1\rangle \supseteq \langle F_1, \cdots, F_r; \varepsilon_2\rangle \quad \text{if} \ \ \varepsilon_1 < \varepsilon_2. \tag{3.3}$$

筆者は近似グレブナー基底をBuchberger算法を修正して計算する。Buchberger算法で最も重要な概念は項簡約である。多項式PとQがあり、$\mathrm{lpp}(Q)$は$\mathrm{lpp}(P)$を割り切るが、$\mathrm{acc}(P) \ll \mathrm{acc}(Q)$であるとする。従来の算法では$P \xrightarrow{Q}\!\!\!\!\rightarrow \tilde{P}$を計算したあと、$P$を$\tilde{P}$で置き換えるのが普通である。しかし、$\mathrm{acc}(\tilde{P}) \gg \mathrm{acc}(P)$なので、$P$を$\tilde{P}$で置き換えれば系全体が精度劣化する。そこで、従来の項簡約に替えて限定的な項簡約を導入する。

定義5 (精度防御簡約) $\Gamma = \{G_1, G_2, \cdots\} \subset \mathbb{F}[\boldsymbol{x}]$とし、$F \in \mathbb{F}[\boldsymbol{x}]$の$\Gamma$による簡約$F = F_1 \xrightarrow{G_1} F_2 \xrightarrow{G_2} \cdots \longrightarrow \tilde{F}$を考える。各簡約$F_i \xrightarrow{G_i} F_{i+1}$では$G_i$として$\mathrm{acc}(G_i) \leq \mathrm{acc}(F_i)$を満たすもののみを用いる。$\tilde{F}$が$\Gamma$既約なとき、この簡約を$F \overset{\Gamma}{\cdots\!\!\rightarrow\!\!\!\!\rightarrow} \tilde{F}$と表す。 ◇

従来のグレブナー基底の定義の一つは、$\Gamma = \{G_1, \cdots, G_s\}$に対して次の2条件を課すものである。(1) $\langle F_1, \cdots, F_r\rangle = \langle G_1, \cdots, G_s\rangle$、(2) $\mathrm{Spol}(G_i, G_j) \overset{\Gamma}{\longrightarrow\!\!\!\!\rightarrow} 0 \ (\forall i \neq j)$。これに倣い、筆者は近似グレブナー基底を次のように定義する。定義の妥当性は、次章で算法を与えたのち、第5章で述べる。

定義6 (近似グレブナー基底) $\Phi = \{F_1, \cdots, F_r\} \subset \mathbb{F}[\boldsymbol{x}]$は初期精度$\varepsilon_{\mathrm{init}}$の初期基底で、$\varepsilon_{\mathrm{init}} \leq \varepsilon_{\mathrm{app}} \ll 1$とする。$\Gamma = \{G_1, \cdots, G_s\} \subset \mathbb{F}[\boldsymbol{x}]$は、下記2条件を満足するとき、近似イデアル$\langle F_1, \ldots, F_r; \varepsilon_{\mathrm{init}}\rangle$の許容度$\varepsilon_{\mathrm{app}}$の近似グレブナー基底であるという。

(1) $\langle F_1, \cdots, F_r; \varepsilon_{\mathrm{init}}\rangle = \langle G_1, \cdots, G_s; \varepsilon_{\mathrm{init}}\rangle$、

(2) $\mathrm{Spol}(G_i, G_j) \overset{\Gamma}{\cdots\!\!\rightarrow\!\!\!\!\rightarrow} 0 \ (\mathrm{tol}\ \varepsilon_{\mathrm{app}}) \quad (\forall i \neq j)$。 ◇

4 近似グレブナー基底を計算するアルゴリズム

次頁算法は、不正確な偶発的項キャンセルが起きないか、回避できる場合に対する最も簡単なものである；前提2参照。なお、等号や不等号については前提1を参照。

精度防御簡約は従来の簡約に比べて弱いが、次の二つの性質を持っている。性質i) $\cdots\!\!\rightarrow\!\!\!\!\rightarrow$は停止する。性質ii) appGBで多項式$P$が多項式$Q$よりも先に生成されていれば、通常$\mathrm{acc}(P) \leq \mathrm{acc}(Q)$となり、$\mathrm{lpp}(P) \mid \mathrm{lpp}(Q)$であれば$Q$は$P$で精度防御簡約できる。通常でない場合とは、多項式$P_1, P_2$が、$P_i = \hat{P}_i + \breve{P}_i$, $\mathrm{acc}(\hat{P}_i) < \mathrm{acc}(\breve{P}_i)$と表され、$\mathrm{Spol}(P_1, P_2)$または$\mathrm{Lred}(P_1, P_2)$において$\breve{P}_1$と$\breve{P}_2$が正確にキャンセルする場合である。

定理1 プロシジャappGBは有限回の手順で停止する。

証明 Buchberger算法と比べてappGBは、1) ゼロ判定に替えて近似ゼロ判定を採用する、2) $\longrightarrow\!\!\!\!\rightarrow$の代わりに$\cdots\!\!\rightarrow\!\!\!\!\rightarrow$を用いる、3) 精度が落ちた多項式を一時的に$\Psi$に保存する、の3点のみが異なっている。1)はBuchberger算法よりも強く停止性に寄与する。

2)と3)について、まずappGBで$\varepsilon = \varepsilon_{\mathrm{init}}$の場合を考える。上述の性質ii)により、精度$\varepsilon_{\mathrm{init}}$の多項式全体の集合に対しては、精度防御簡約は計算結果の精度が落ちない限り従来の簡約と同じであることが分る。よって、Dicksonの補題([2]参照)よりステップStp2で生成されるS多項式の個数が有限であることが分り、Ψに保存される多項式の個数も有限

Procedure appGB(Φ, ε_{app}) ==
local: $\Psi := \{\}$; $\varepsilon := \varepsilon_{\text{init}}$;
Stp1: While $\forall F \in \Phi$ is reducible by $\Phi' = \Phi \setminus \{F\}$,
do $F \cdots\xrightarrow{\Phi'}\!\!\!\!\rightarrow \tilde{F}$; $\Phi := \Phi \setminus \{F\}$;
If $\tilde{F} = 0$ or $\text{acc}(\tilde{F}) > \varepsilon_{\text{app}}$ then nil
else if $\text{acc}(\tilde{F}) \leq \varepsilon$ then $\Phi := \Phi \cup \{\tilde{F}\}$
else $\Psi := \Psi \cup \{\tilde{F}\}$;
od ;
If $\#(\Phi) = 1$ then goto Stp3;
Stp2: For a new pair (F_i, F_j), $i < j$, taken from Φ,
do $\text{Spol}(F_i, F_j) \cdots\xrightarrow{\Phi}\!\!\!\!\rightarrow \tilde{F}$;
If $\tilde{F} = 0$ or $\text{acc}(\tilde{F}) > \varepsilon_{\text{app}}$ then goto chk;
If $\text{acc}(\tilde{F}) \leq \varepsilon$ then
$\Phi := \Phi \cup \{\tilde{F}\}$ and goto Stp1
else $\Psi := \Psi \cup \{\tilde{F}\}$ and goto chk;
od ;
chk: If there is any unchecked pair then goto Stp2;
Stp3: If $\Psi = \{\}$ then return Φ;
Let F be the smallest accuracy element of Ψ;
Move F from Ψ to Φ ;
$\varepsilon := \text{acc}(F)$ and goto Stp2 ;
end.

となって、ステップStp1,Stp2は停止する。次に $\varepsilon > \varepsilon_{\text{init}}$ の場合を考える。上述の性質ii)によれば、稀にεより小さい精度の多項式が生成されることがある。精度εで生成されるS多項式の簡約が、その多項式により促進されることはあっても遅延されることはない。よって、この場合にもステップStp1,Stp2は停止することが言える。 ♢

補題1 $\Phi = \{F_1, \cdots, F_r\} \subset \mathbb{F}[\boldsymbol{x}]$ とし、$\text{acc}(F_i) = \varepsilon_{\text{init}}$ $(i = 1, \ldots, r)$ とする。
Claim-1. $S = \text{Spol}(F_i, F_j)$ $(i \neq j)$ に対して $\langle F_1, \cdots, F_r, S; \varepsilon_{\text{init}}\rangle = \langle F_1, \cdots, F_r; \varepsilon_{\text{init}}\rangle$。
Claim-2. $\Phi' = \Phi \setminus \{F_i\}$ とし、$F_i \cdots\xrightarrow{\Phi'}\!\!\!\!\rightarrow \tilde{F}_i$ とすれば、
$$\langle F_1, \cdots, F_i, \cdots, F_r; \varepsilon_{\text{init}}\rangle = \langle F_1, \cdots, \tilde{F}_i, \cdots, F_r; \varepsilon_{\text{init}}\rangle。$$

証明 $S = AF_i + BF_j$, $A, B \in \mathbb{C}[\boldsymbol{x}]$, と表せるので、Claim-1は明白である。
次に、$F_i \xrightarrow{F_j} F_i'$ $(i \neq j)$, ただし $\text{acc}(F_j) \leq \text{acc}(F_i)$, を考える。$F_i$は単項式$T$により $F_i = TF_j + F_i'$ と表せ、$\text{acc}(TF_j) \leq \text{acc}(F_i)$ かつ $TF_j \succ F_i'$ である。したがって、任意の $A_i, A_j \in \mathbb{C}[\boldsymbol{x}]$ に対して $A_iF_i + A_jF_j$ を $A_iF_i' + (A_iT + A_j)F_j$ で置き換えても、精度は劣化

しない。これを繰り返せばClaim-2が得られる。 ◇

定理2 (主定理) プロシジャappGBの出力は近似イデアル $\langle F_1, \cdots, F_r; \varepsilon_{\text{init}} \rangle$ の許容度 ε_{app} の近似グレブナー基底である。ただし、$\varepsilon_{\text{app}} \geq \varepsilon_{\text{init}}$ とする。

証明 定理1はappGBの出力が有限集合であることを保証する。その出力を Γ とする。定義6の条件(1)は補題1により保証される。Γ の任意の要素対 (G_i, G_j) に対してappGBは $\text{Spol}(G_i, G_j) \cdots\!\xrightarrow{\Gamma}\!\!\!\!\rightarrow 0 \ \ (\text{tol}\ \varepsilon_{\text{app}})$ をチェックするので、条件(2)も満足される。 ◇

5 近似グレブナー基底の基本的性質

数学では“多項式環の任意の元に対して…”とする定理が多くある。しかし、浮動小数係数多項式の場合、任意回数の演算を繰り返すうちに丸め誤差が際限なく増える可能性がある。そこで、下記定理3, 4では、“任意の多項式”を文中のように制限する。

定理3 $\Gamma = \{G_1, \cdots, G_s\}$ は許容度 ε_{app} の近似グレブナー基底とし、単項式 M は、$P \prec M$ なる任意の多項式 $P \in \mathbb{F}[\boldsymbol{x}]$ を Γ で簡約する際の丸め誤差は無視できる、そんなものであるとする。このとき、$P \prec M$ なる任意の多項式 $P \in \langle G_1, \cdots, G_s; \varepsilon_{\text{init}} \rangle$ に対して $P \cdots\!\xrightarrow{\Gamma}\!\!\!\!\rightarrow \tilde{P}$ とすれば、$\tilde{P} = 0 \ \ (\text{tol}\ \varepsilon_{\text{app}})$ である。

証明 $P = A_1G_1 + \cdots + A_rG_r$ を満たす $A_1, \ldots, A_r \in \mathbb{C}[\boldsymbol{x}]$ が存在する。このとき、$\{\text{lt}(A_1G_1), \cdots, \text{lt}(A_rG_r)\}$ の中で最高の順位を P の**表現順位**ということにする。P を Γ で簡約するか P の表現を変えるなら、P が $A'_1G_1 + \cdots + A'_rG_r$ と変換され、表現順位が下がることを示せばよい。一般性を失うことなく $\text{lpp}(A_1G_1) = \cdots = \text{lpp}(A_lG_l) \succ \text{lpp}(A_{l+j}G_{l+j})$, $(j=1, \ldots, s-l)$, とする。$l = 1$ ならば A_1G_1 を G_1 で簡約すればよい。$l > 1$ でも $\text{lpp}(P) = \text{lpp}(A_1G_1)$ ならば、$\text{lt}(P)$ と同順位の主項がある限り $A_1G_1, \ldots, A_lG_l$ を $G_1, \ldots, G_l$ で簡約すればよい。最後に残るのは $l > 1$ かつ $\text{lpp}(P) \prec \text{lpp}(A_1G_1)$ の場合である。この場合、$(\star)$: $\text{lt}(A_1G_1) + \cdots + \text{lt}(A_lG_l) = 0 \ \ (\text{tol}\ \varepsilon_{\text{app}})$ が成立する。$\text{lt}(A_i) = a_iU_i$ および $\text{lc}(G_i) = c_i$, $(i=1, \ldots, l)$、ただし U_i はべき積で $a_i, c_i \in \mathbb{C}$、とすれば、次の恒等式が成立する。

$$\begin{aligned} &\text{lt}(A_1)G_1 + \cdots + \text{lc}(A_l)G_l = a_1c_1U_1(G_1/c_1) + \cdots + a_lc_lU_l(G_l/c_l) \qquad (5.1)\\ =\ &(a_1c_1)[U_1G_1/c_1 - U_2G_2/c_2] + (a_1c_1 + a_2c_2)[U_2G_2/c_2 - U_3G_3/c_3] + \cdots \\ &+ (a_1c_1 + \cdots + a_{l-1}c_{l-1})[U_{l-1}G_{l-1}/c_{l-1} - U_lG_l/c_l] + (a_1c_1 + \cdots + a_lc_l)U_lG_l/c_l. \end{aligned}$$

上式右辺の最後の項は $(\star)$ より0であることが分る。また、$[U_iG_i/c_i - U_{i+1}G_{i+1}/c_{i+1}]$ は $\text{Spol}(G_i, G_{i+1})$ の単項式倍である。ここで、$\text{Spol}(G_i, G_{i+1}) \cdots\!\xrightarrow{\Gamma}\!\!\!\!\rightarrow 0 \ \ (\text{tol}\ \varepsilon_{\text{app}})$ であり、$A_iG_i = \text{lt}(A_i)G_i + \text{rt}(A_i)G_i$, $\text{lt}(A_i) \succ \text{rt}(A_i)$, なので、$P$ は $\text{lpp}(A'_iG_i) \prec \text{lpp}(A_1G_1)$ なる $A'_1, \ldots, A'_s$ により $P = A'_1G_1 + \cdots + A'_sG_s + P'$ と表され、かつ $\text{acc}(P') > \varepsilon_{\text{app}}$ であることがわかる。 ◇

補題2 $\Gamma = \{G_1, \cdots, G_s\}$ は許容度 ε_{app} の近似グレブナー基底とし、任意の $1 \le i < j \le s$ について、$\mathrm{Spol}(G_i, G_j) = u_{ij}G_i - v_{ij}G_j$ とおく、ただし u_{ij}, v_{ij} は単項式である。$\mathrm{Spol}(G_i, G_j)$ を Γ で精度防御簡約した結果を次のように表す；ここで $w_{ijk} \in \mathbb{C}[\boldsymbol{x}]$。

$$\mathrm{Spol}(G_i, G_j) = -\textstyle\sum_{k=1}^{s} w_{ijk}G_k + \delta G_{ij}, \quad u_{ij}G_i \succ w_{ijk}G_k, \quad \mathrm{acc}(\delta G_{ij}) > \varepsilon_{\text{app}}. \tag{5.2}$$

このとき、下記 $R_{i,j}$ は $G_1, \ldots, G_s$ の許容度 ε_{app} での近似線形従属関係である。

$$R_{i,j} : w_{ij1}G_1 + \cdots + (w_{iji}+u_{ij})G_i + \cdots + (w_{ijj}-v_{ij})G_j + \cdots + w_{ijs}G_s. \tag{5.3}$$

証明 $\mathrm{Spol}(G_i, G_j) \cdots\!\overset{\Gamma}{\longrightarrow}\!\!\!\!\rightarrow 0$ $(\mathrm{tol}\ \varepsilon_{\text{app}})$ から明白。 ◇

定理4 単項式 M は定理3で定義され、数 $\varepsilon_1(=\varepsilon_{\text{init}}), \varepsilon_2, \ldots, \varepsilon_k$ は appGB でパラメータ ε が順にとる値であるとする $(\varepsilon_1 < \varepsilon_2 < \cdots < \varepsilon_k \le \varepsilon_{\text{app}})$。appGB の出力を下記とする。

$$\begin{gathered}\Gamma = \{G_{s_1+1}, \ldots, G_{s_1+t_1}, \ \cdots, \ G_{s_k+1}, \ldots, G_{s_k+t_k}\}, \\ \mathrm{acc}(G_{s_i+j}) = \varepsilon_i \quad (i = 1, \ldots, k;\ j = 1, \ldots, t_i).\end{gathered} \tag{5.4}$$

R_{j_1,j_2} は補題2の $R_{i,j}$ のように、$\mathrm{Spol}(G_{j_1}, G_{j_2})$ から作られる近似線形従属関係であるとすれば、$\{R_{j_1,j_2} \mid 1 \le j_1 < j_2 \le s_i+t_i\}$ は、$F_1, \ldots, F_r$ の近似線形従属関係のうち、順位が M 以下で許容度が ε_i 以下のものの基底となる。

証明 許容度 ε_i の任意の近似線形従属関係を $R = A_1G_1 + \cdots + A_sG_s$ とする。一般性を失わず $\mathrm{lpp}(A_1G_1) = \cdots = \mathrm{lpp}(A_lG_l) \succ \mathrm{lpp}(A_{l+j}G_{l+j})$, $(j=1, \ldots, s-l)$, とする。仮定より、$(\star)$: $\mathrm{lt}(A_1G_1) + \cdots + \mathrm{lt}(A_lG_l) = 0$ $(\mathrm{tol}\ \varepsilon_{\text{app}})$ が成立するので、$l \ge 2$ である。定理3の証明と同様、$\mathrm{lt}(A_i) = a_iU_i$ および $\mathrm{lc}(G_i) = c_i$, $(i=1, \ldots, l)$, とおき、(5.1)を考える。右辺の最後の項は $(\star)$ より0である。また、$[U_iG_i/c_i - U_{i+1}G_{i+1}/c_{i+1}] = V\,\mathrm{Spol}(G_i, G_{i+1})$, V は単項式, と表せる。補題2より、$\mathrm{Spol}(G_i, G_{i+1}) = R_{i,i+1} + \sum_{k=1}^{s} w_{i,i+1,k}G_k + \delta G_{i,i+1}$ なので、R から $VR_{i,i+1}$ を引き去れば R の表現順位が下がる。この操作を繰り返せば、R は許容度 ε_i で近似ゼロになる。 ◇

6 簡単な例

実際の計算では、不正確な組織的項キャンセルの推定値がズレたり、不正確な偶発的項キャンセルが起きたりするので、実際計算用の精度を定義する。

定義6 (最大・最小精度) 多項式 P はプロシジャ appGB を高精度マーキング法で駆動して計算されたものとする。P の係数の実際の精度の中で、最大および最小のものをそれぞれ最大精度, 最小精度といい、$\mathrm{acmax}(P)$, $\mathrm{acmin}(P)$ と表す。

以下の例は、いずれも入力多項式の各係数を最初に倍精度浮動小数に変換し(その結果、各係数には 2×10^{-16} 程度の相対誤差が入る)、40桁精度の有効浮動小数に変換したあと、高精度マーキング法で近似グレブナー基底を計算した。

例 1 (イデアルの次元が変化する系) f_1, f_2, f_3 を下記で与える。これら多項式間には近似線形従属関係 $\|56/57\, y f_1 - 57/56\, x f_2 - 2xy f_3\| \simeq 0.000041$ が成立する。

$$\begin{cases} f_1 = 57/56\, x^2 y + 68/67\, xz^2 - 79/78\, xy + 89/88\, x, \\ f_2 = 56/57\, xy^2 - 67/68\, yz^2 + 78/79\, y^2 - 88/89\, y, \\ f_3 = \qquad\qquad z^2 - \quad y \quad + \quad 1. \end{cases} \tag{6.1}$$

参考のため、イデアル $\langle f_1, f_2, f_3 \rangle$ の $\mathbb{Q}$ 上のグレブナー基底を与えておく。

$$\begin{cases} g_2 = z^6 - 0.6986\cdots z^4 - 1.0012\cdots z^2 + 0.6974\cdots, \\ g_3 = y - z^2 - 1, \\ g_1 = x - 0.1490\cdots y^2 z^2 + y^2\text{-}, yz^4\text{-}, yz^2\text{-}, \text{and } y\text{-terms}. \end{cases}$$

許容度 $\varepsilon_{\text{app}} = 0.0001$ に対する近似グレブナー基底は次のものとなる。

$$\begin{cases} G_3 = \#\text{EF}(1.000000000000000000\cdots\text{e}{-}0,\ 5.0\text{e}{-}39)\, y \\ \quad - \#\text{EF}(1.000000000000000000\cdots\text{e}{-}0,\ 5.0\text{e}{-}39)\, z^2 \\ \quad - \#\text{EF}(1.000000000000000000\cdots\text{e}{-}0,\ 5.0\text{e}{-}39), \\ G_2 = \#\text{EF}(1.000000000000000000\cdots\text{e}{-}0,\ 5.0\text{e}{-}39)\, xz^4 \\ \quad + xz^2\text{-term} + x\text{-term} + y^2\text{-term} - \cdots - y\text{-term}, \\ G_1 = \#\text{EF}(1.000000000000000000\cdots\text{e}{-}0,\ 5.0\text{e}{-}39)\, x^2 z^2 \\ \quad + x^2\text{-term} - xy\text{-term} + xz^2\text{-term} + x\text{-term}. \end{cases}$$

ここで、$\text{acmin}(G_i), \text{acmax}(G_i) = O(\varepsilon_{\text{init}})$, $(i = 1, 2, 3)$, である。計算は非常に簡単である：$F_3 = G_3$, $F_2 \cdots\xrightarrow{F_3}\!\!\!\!\rightarrow G_2$, $F_1 \cdots\xrightarrow{F_3}\!\!\!\!\rightarrow G_1$ となり、任意の対 (i, j) に対して $\text{Spol}(G_i.G_j) \cdots\!\!\rightarrow 0$ となる。その過程で、下記多項式が許容度約 1/1580 で近似ゼロと判定された。

$$xz^4 - 0.00408\cdots xz^2 - 1.004\cdots x.$$

$\mathbb{Q}$ 上のイデアルは 0 次元であるが、上記近似イデアルは 2 次元であることに注意。

許容度 $\varepsilon_{\text{app}} = 10^{-13}$ では、近似グレブナー基底として次が得られた。

$$\begin{cases} G_1,\ G_2,\ G_3 \text{ are the same as above}, \\ G_4 = \#\text{EF}(0.999999999999999999\cdots\text{e}{-}0,\ 1.5\text{e}{-}31)\, z^6 \\ \quad - \#\text{EF}(0.686678690478816648\cdots\text{e}{-}0,\ 3.0\text{e}{-}31)\, z^4 \\ \quad - \#\text{EF}(1.001246563216279752\cdots\text{e}{-}0,\ 3.0\text{e}{-}31)\, z^2 \\ \quad + \#\text{EF}(0.697421305831601912\cdots\text{e}{-}0,\ 3.0\text{e}{-}31), \\ G_5 = \#\text{EF}(0.999999999999999999\cdots\text{e}{-}0,\ 5.0\text{e}{-}29)\, x \\ \quad - \#\text{EF}(0.669949437931807210\cdots\text{e}{-}0,\ 2.5\text{e}{-}29)\, y^2 z^2 \\ \quad + y^2\text{-term} + yz^4\text{-term} - yz^2\text{-term} - y\text{-term}, \\ G_6 = \#\text{EF}(0.999999999999999999\cdots\text{e}{-}0,\ 1.5\text{e}{-}34)\, xz^2 \\ \quad + \#\text{EF}(0.999999999999999999\cdots\text{e}{-}0,\ 5.0\text{e}{-}39)\, x \\ \quad - y^2 z^2\text{-term} + y^2\text{-term} + yz^4\text{-term} + \cdots. \end{cases}$$

ここで、最小精度と最大精度は、G_4 では $15800\,\varepsilon_{\rm init}$ と $106000\,\varepsilon_{\rm init}$、$G_5$ では $60400\,\varepsilon_{\rm init}$ と $122000\,\varepsilon_{\rm init}$、$G_6$ では $60400\,\varepsilon_{\rm init}$ と $105000\,\varepsilon_{\rm init}$ である。最大精度と最小精度の比は 2～7 なので、計算はほぼ安定に行われたと言える。$\mathrm{acmax}(G_5) \gg \mathrm{acmax}(G_1), \mathrm{acmax}(G_2)$ なので、精度防御簡約の観点から G_5 は G_1, G_2 を簡約できないことに注意されたい。なお、$G_1, G_2, G_3, G_4, G_5, G_6$ で張られるイデアルは 0 次元である。 ◇

例 2 (イデアルの次元が変化しない系) 多項式 f_1, f_2, f_3 を下記で与える。これらの間には近似線形従属関係 $f_1 - zf_2 + 2yf_3/3 = (y^2z + yz^2)/30000$ が成立する。

$$\begin{cases} f_1 := yz((xy-3y) - (2xz-z))/3, \\ f_2 := y\,(xy - 30001/10000y)/3, \\ f_3 := z\,(2xz - 9999/10000z)/2. \end{cases}$$

イデアル $\langle f_1, f_2, f_3\rangle$ の $\mathbb{Q}$ 上のグレブナー基底 (全次数順序) は下記となる。

$$\{g_2 = f_2, \quad g_3 = f_3, \quad g_1 = y^2z + yz^2, \quad g_4 = yz^2\}.$$

40 桁精度の有効浮動小数に変換された係数をマーキングする。マークは乱数で、系 A で $+\varepsilon$ とマークした場合は系 B では $-\varepsilon$ とマークする。たとえば入力多項式 f_1 は

$$\begin{aligned} F_{1\mathrm{A}} &= \#\mathrm{EF}(1.000000000000000594\cdots\mathrm{e}{-0},\ 1.0\mathrm{e}{-38}\,xy^2z \\ &\quad - \#\mathrm{EF}(1.999999999999998954\cdots\mathrm{e}{-0},\ 2.0\mathrm{e}{-38}\,xyz^2 + \cdots, \\ F_{1\mathrm{B}} &= \#\mathrm{EF}(0.999999999999999405\cdots\mathrm{e}{-0},\ 1.0\mathrm{e}{-38}\,xy^2z \\ &\quad - \#\mathrm{EF}(2.000000000000001045\cdots\mathrm{e}{-0},\ 2.0\mathrm{e}{-38}\,xyz^2 + \cdots, \end{aligned}$$

と二つの多項式に変換される。上述のマーキングにより $(P_{\mathrm{A}} + P_{\mathrm{B}})/2 = P$ がどの多項式にも成立する。係数 c に起きた不正確な桁落ちは $|c_{\mathrm{A}} - c_{\mathrm{B}}|/(|c_{\mathrm{A}} + c_{\mathrm{B}}|\,\varepsilon_{\rm init})$ と推定できる。ここで c_{A} と c_{B} はそれぞれ A 系と B 系の c に対応する係数である。

許容度 $\varepsilon_{\rm app} = 0.001$ に対する近似グレブナー基底は次のものとなる。

$$\begin{cases} G_3 = \#\mathrm{EF}(1.000000000000000000\cdots\mathrm{e}{-0},\ 5.0\mathrm{e}{-39})\,xz^2 \\ \quad - \#\mathrm{EF}(0.499950000000000005\cdots\mathrm{e}{-0},\ 2.5\mathrm{e}{-39})\,z^2, \\ G_2 = \#\mathrm{EF}(0.999999999999999999\cdots\mathrm{e}{-0},\ 5.0\mathrm{e}{-39})\,xy^2 \\ \quad - \#\mathrm{EF}(3.000100000000000155\cdots\mathrm{e}{-0},\ 1.5\mathrm{e}{-38})\,y^2, \\ G_4 = \#\mathrm{EF}(1.000000000000000000\cdots\mathrm{e}{-0},\ 6.0\mathrm{e}{-39})\,y^2z^2. \end{cases}$$

G_2 と G_3 は入力多項式そのもので、G_4 においても桁落ちは全くない。計算の過程で下記多項式が許容度約 1/10200 で近似ゼロと判定された。

$$0.9999\cdots\,y^2z + 0.9999\cdots\,yz^2.$$

許容度 $\varepsilon_{\rm app} = 10^{-13}$ に対する近似グレブナー基底は次のものとなる。

$$\begin{cases} G_2,\ G_3,\ G_4 \text{ are the same as above}, \\ G_6 = \#\mathrm{EF}(0.999999999999999999\cdots\mathrm{e}{-0},\ 6.0\mathrm{e}{-35})\,yz^2, \\ G_1 = \#\mathrm{EF}(0.999999999999999999\cdots\mathrm{e}{-0},\ 1.5\mathrm{e}{-34})\,y^2z \\ \quad + \#\mathrm{EF}(0.999999999999999444\cdots\mathrm{e}{-0},\ 5.0\mathrm{e}{-35})\,yz^2. \end{cases}$$

最小精度と最大精度は、G_6ではどちらも $10200\varepsilon_{\text{init}}$、$G_1$では $10200\varepsilon_{\text{init}}$ と $25700\varepsilon_{\text{init}}$ である。したがって、計算は非常に安定に行われたことがわかる。 ◇

参 考 文 献

[1] M. Bodrato and A. Zanoni. Intervals, syzygies, numerical Gröbner bases: a mixed study. *Proceedings of CASC2006 (Computer Algebra in Scientific Computing): Springer-Verlag LNCS* **4194**, 64-76, 2006.

[2] D. Cox, J. Little and D. O'Shea. *Ideals, Varieties, and Algorithms.* Springer-Verlag New York, 1997.

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

[4] F. Kako and T. Sasaki. Proposal of "effective" floating-point number. Preprint of Univ. Tsukuba, May 1997 (unpublished).

[5] A. Kondratyev, H.J. Stetter and S. Winkler. Numerical computation of Gröbner bases. *Proceedings of CASC2004 (Computer Algebra in Scientific Computing)*, 295-306, St. Petersburg, Russia, 2004.

[6] D. Lazard. Gröbner bases, Gaussian elimination and resolution of systems of algebraic equations. *Proceedings of EUROCAL 1983; Springer-Verlag LNCS* **162**, 146-156, 1983.

[7] K. Nagasaka. A Study on Gröbner basis with inexact input. *Proceedings of CASC2009 (Computer Algebra in Scientific Computing): Springer LNCS* **5743**, 248-258, 2009.

[8] T. Sasaki: A practical method for floating-point Gröbner basis computation. *Proceedings of The Joint Conf. of ASCM 2009 and MACIS 2009*; COE Lecture Note Vol. 22 (Kyushu Univ.), 167-176, 2009.

[9] T. Sasaki: A subresultant-like theory for Buchberger's procedure. Preprint of Univ. Tsukuba, Jan. 2010, 17 pages.

[10] T. Sasaki and F. Kako. Computing floating-point Gröbner base stably. *Proceedings of SNC2007 (Symbolic Numeric Computation)*, 180-189, London, Canada, 2007.

[11] T. Sasaki and F. Kako. Floating-point Gröbner basis computation with ill-conditionedness estimation. *Proceedings of ASCM2007 (Asian Symposium on Computer Mathematics): Springer LNAI* **5081**, 278-292, Deepak Kapur (Ed.), 2008.

[12] T. Sasaki and F. Kako. Term cancellations in computing floating-point Gröbner bases. *Proceedings of CASC2010 (Computer Algebra in Scientific Computing): Springer LNCS* **6244**, to appear.

[13] K. Shirayanagi. An algorithm to compute floating-point Gröbner bases. *Mathematical Computation with Maple V. Ideas and Applications*, Birkhäuser, 95-106, 1993.

[14] K. Shirayanagi and M. Sweedler,. Remarks on automatic algorithm stabilization. *J. Symb. Comput.*, **26**, 761-765, 1998.

[15] H.J. Stetter. Stabilization of polynomial systems solving with Gröbner bases. *Proceedings of ISSAC'97 (Intn'l Symposium on Symbolic and Algebraic Computation)*, 117-124, ACM Press, 1997.

[16] C. Traverso. Syzygies, and the stabilization of numerical Buchberger algorithm. *Proceedings of LMCS2002 (Logic, Mathematics and Computer Science)*, 244-255, RISC-Linz, Austria, 2002.

[17] C. Traverso and A. Zanoni. Numerical stability and stabilization of Gröbner basis computation. *Proceedings of ISSAC2002 (Intn'l Symposium on Symbolic and Algebraic Computation)*, 262-269, ACM Press, 2002.

[18] V. Weispfenning. Gröbner bases for inexact input data. *Proceedings of CASC2003 (Computer Algebra in Scientific Computing)*, 403-411, Passau, Germany, 2003.