Backward error analysis of approximate Grbner basis

Kosaku Nagasaka

Kobe University [*]

E-mail: nagasaka@main.h.kobe-u.ac.jp

Abstract

Computing a Gröbner basis for the given polynomial system with inexact (erroneous) coefficients is one of the challenging problems in symbolic-numeric computations and there are several approaches to find an approximate Gröbner basis that are usually computed by floating-point numbers. However, in general the resulting basis is not a Gröbner basis and does not generate the given ideal. This is the problem of all kinds of approximate Gröbner bases even though there are some workarounds (with concepts of approximate basis, ideal and so on). In this paper, we introduce a proof of concept method and open questions to find an exact result from those approximate Gröbner bases, that is a Gröbner basis of the ideal generated by a nearby polynomial set in the exact sense.

Key words: Gröbner Bases, Symbolic-Numeric Computations, Inexact

1 Introduction

For the given ideal with a finite set of polynomials, computing its Gröbner basis can be done by the well known Buchberger's algorithm and this basis is very useful (see the graduate text book [1] for example). However, in practical situations, the given polynomials with empirical data on their coefficients may have a priori errors and the conventional algorithms can not work in general. This is the problem so called "approximate Gröbner basis" which has been studied in the past several decades and is one of the challenging problems in symbolic-numeric computations.

In Sasaki and Kako [17], this problem is classified into the first and second kinds of problem. The first kind is computing a Gröbner basis for the ideal generated by the given polynomials with exact coefficients by numerical arithmetic (e.g. floating-point arithmetic). The second kind is for the given polynomials with inexact coefficients having a priori errors. In this case, we have to operate with a priori errors whether we compute a basis by exact arithmetic or not.

For the first kind, Shirayanagi [19, 20] proposed algorithms using stabilization techniques [21] by which we can compute a numerical sequence of Gröbner bases with exact inputs and the sequence converges to the exact result. By this, we can recover the exact coefficients

[*] 3-11 Tsurukabuto, Nada-ku, Kobe 657-8501 JAPAN.

from the resulting numerical coefficients by just rationalizing them since the upper bounds of computational cost and space are computable and theoretically finite. Therefore, for the first kind, finding exact result from approximate Gröbner basis is not the problem (see the example in [13]).

For the second kind, there are several studies ([22, 24, 10, 18, 15, 3] and citations therein) from the numerical point of view. In this case, it is difficult to find exact result from approximate Gröbner basis since the correct input (set of polynomials) is unknown. In contrast, computing a comprehensive Gröbner system [25] for polynomials with parameters instead of inexact coefficients is an approach [26] using exact arithmetic hence we do not have to recover any exact result (it's already in the exact representation). However, in general, a comprehensive Gröbner system has a huge number of segments and its computation time is slow though there are several improvements (see [9] and citations therein). This is a reason that approximate Gröbner basis is still challenging and there are studies using floating-point arithmetic.

1.1 The Problem to be Solved

Computing approximate Gröbner basis can be thought as just one of symbolic-numeric computations for polynomials (see [16] for some note from the methological point of view): approximate polynomial GCD, approximate factorization and so on. However, there is a big difference between approximate Gröbner basis and others that the backward error analyses are naturally given or not (i.e. easy or not).

For example, we consider an approximate GCD of the following $\tilde{f}(x)$ and $\tilde{g}(x)$ by the algorithm proposed in [8].

$$\begin{aligned}\tilde{f}(x) &= 54x^6 - 36x^5 - 192x^4 + 42x^3 + 76x^2 - 62x + 15,\\ \tilde{g}(x) &= 73x^5 + 36x^4 - 103x^3 - 70x^2 - 48x + 35.\end{aligned}$$

In this case, we have an approximate common divisor $\tilde{h}(x) = 1.00x^2 + 0.99x - 0.55$ with tolerance $\varepsilon = 10^{-8}$:

$$\begin{aligned}\tilde{f}(x) &\approx \tilde{h}(x)(54.34x^4 - 90.20x^3 - 72.20x^2 + 63.65x - 27.07),\\ \tilde{g}(x) &\approx \tilde{h}(x)(72.84x^3 - 36.20x^2 - 26.83x - 63.33).\end{aligned}$$

The resulting approximate common divisor $\tilde{h}(x)$ can be characterized as the polynomial GCD of the following polynomials in the exact sense, by rationalizing the coefficients.

$$\begin{aligned}f(x) &= \tfrac{2717}{50}x^6 - \tfrac{182017}{5000}x^5 - \tfrac{38277}{200}x^4 + \tfrac{20891}{500}x^3 + \tfrac{151307}{2000}x^2 - \tfrac{154517}{2500}x + \tfrac{29777}{2000},\\ g(x) &= \tfrac{1821}{25}x^5 + \tfrac{2011985808912}{56026069819}x^4 - \tfrac{10273}{100}x^3 - \tfrac{657655415835}{9397534153}x^2 - \tfrac{239701}{5000}x + \tfrac{69663}{2000}.\end{aligned}$$

Moreover, we consider an approximate factorization of the following polynomial (from the example in [7]).

$$\begin{aligned}\tilde{f}(x,y,z) = {} & 81x^4 + 72x^2y^2 + \tfrac{3}{1292}x^2z^2 - 648x^2 + 16y^4\\ & + \tfrac{1}{969}y^2z^2 - 288y^2 - \tfrac{837227}{1292}z^4 - \tfrac{3}{323}z^2 + 1296.\end{aligned}$$

In this case, we have the following approximate factorization $\tilde{f}_1(x,y,z)\tilde{f}_2(x,y,z)$ with tolerance $\varepsilon = 4.54478 \times 10^{-6}$.

$$\begin{aligned}\tilde{f}_1(x,y,z) &= 9.000x^2 + 4.000y^2 - 25.46z^2 - 36.00,\\ \tilde{f}_2(x,y,z) &= 9.000x^2 + 4.000y^2 + 25.46z^2 - 36.00.\end{aligned}$$

The resulting approximate factorization $\tilde{f}_1(x,y,z)\tilde{f}_2(x,y,z)$ can be characterized as the factorization of the following polynomial in the exact sense, by rationalizing the coefficients.

$$f(x,y,z) = 81x^4 + 72x^2y^2 - 648x^2 + 16y^4 - 288y^2 - \tfrac{78400}{121}z^4 + 1296.$$

In contrast, approximate Gröbner basis does not have this behavior in general. To see this, we consider an approximate Gröbner basis of the ideal generated by the following set of polynomials $\tilde{F}_{app}$ by Mathematica [12], and $\tilde{G}_{app}$ is the resulting approximate Gröbner basis w.r.t. the graded lexicographic order $(x \succ y)$.

$$\begin{aligned}\tilde{F}_{app} &= \{0.01084x^3y + 0.891x^3,\ 0.503xy^3 + 0.1129x + 0.02201\},\\ \tilde{G}_{app} &= \{1.0xy^3 + 0.224453x + 0.0437575,\ 1.0x^3 - 7.87965 \times 10^{-8}x^2\}.\end{aligned}$$

However, we have the following result if we compute a Gröbner basis of the ideal generated by $\tilde{G}_{app}$ with the rationalized coefficients in the exact sense (we show only first 6 decimal places to save the paper space instead of full rational representations).

$$G_{app} \approx \{1.23120 \times 10^{30}y^3 + 6.83710 \times 10^{35},\ 3.12500 \times 10^{21}x - 2.46239 \times 10^{14}\}.$$

Moreover, the following G_{ex} is a Gröbner basis of the ideal generated by $\tilde{F}_{app}$ with the rationalized coefficients, which is different from G_{app} and $\tilde{G}_{app}$ (only first 6 decimal places instead of full rational representations). Note that we consider the second kind of problem hence in general the resulting G_{ex} is not the basis we want (for details, see [15] and [16]).

$$G_{ex} \approx \{5.55931 \times 10^{17}x - 4.38054 \times 10^{10},\ 271.000y + 22275.0\}.$$

Therefore, the resulting approximate Gröbner basis is not a Gröbner basis and does not generate the given ideal in the exact sense. This behavior is common for algorithms computing approximate Gröbner basis hence we have a very natural question: "What is that we computed?" This is the problem of all kinds of approximate Gröbner bases even though there are some workarounds (with concepts of approximate basis, ideal and so on). In this paper, we introduce a proof of concept method and open questions to find an exact result from those approximate Gröbner bases, that is a Gröbner basis of the ideal generated by a nearby polynomial set in the exact sense (so we can have a backward error analysis).

At first we introduce the notations we use and the basic structure of our method in the next section **2**. In the section **3**, we discuss about a nearby Gröbner basis for the resulting approximate Gröbner basis for which we consider a nearby polynomial system generating its sub-ideal in the section **4**. In general, the resulting nearby polynomial system does not generate the ideal generated by the nearby Gröbner basis. We give some results on this problem in the section **5**. Finally, in the section **6**, we give some remarks and open questions for further work.

2 Preliminary Discussion

We always assume that we have $\tilde{F}_{app} = \{\tilde{f}_1, \ldots, \tilde{f}_n\} \in \mathbb{F}[\vec{x}] = \mathbb{F}[x_1, \ldots, x_\ell]$ and $\tilde{G}_{app} = \{\tilde{g}_1, \ldots, \tilde{g}_m\} \in \mathbb{F}[\vec{x}]$ as the given polynomial system in variables $\vec{x} = x_1, \ldots, x_\ell$ with inexact coefficients and its (**minimal** and approximate/numerical) Gröbner basis w.r.t. the term order $\prec$, respectively, where $\mathbb{F}$ is the set of floating-point numbers hence $\mathbb{F} \subset \mathbb{R}$ (the real number field but note that $\mathbb{F}$ is not a field). Note that we discuss about polynomials over $\mathbb{R}$ however it is easy to extend them to polynomials over $\mathbb{C}$ (but we restrict to over $\mathbb{R}$ for simplicity's sake).

Though there are several definitions of approximate Gröbner basis, we just assume that it is minimal (in the ordinary sense) hence for all $\tilde{g} \in \tilde{G}_{app}$ we have $\forall \tilde{h} \in \tilde{G}_{app} \setminus \{\tilde{g}\}, \mathrm{ht}(\tilde{h}) \nmid \mathrm{ht}(\tilde{g})$ where $\mathrm{ht}(f)$ denotes the head term of polynomial $f(\vec{x})$ w.r.t. the term order. Note that "term" means a power product of variables and "monomial" means a term with a coefficient. In this paper, we always represent a numerical object of some mathematical object with the tilde symbol: $\tilde{\cdot}$ hence $\tilde{a}$ is over $\mathbb{F}$ and a is over $\mathbb{R}$. We consider the following problem.

Problem 1 [Nearby Gröbner basis and system] For the given $\tilde{F}_{app}, \tilde{G}_{app} \subset \mathbb{F}[\vec{x}]$, compute $F_{cl}, G_{cl} \subset \mathbb{R}[\vec{x}]$ such that G_{cl} is a Gröbner basis of ideal(F_{cl}) in the exact sense and F_{cl} and G_{cl} are close to $\tilde{F}_{app}$ and $\tilde{G}_{app}$, respectively. $\triangleleft$

For this problem we propose the following 3 steps:

1. Compute a close enough exact Gröbner basis G_{cl} of its self to the given approximate Gröbner basis $\tilde{G}_{app}$ (discussed in the section **3**).

2. Compute a close enough system F'_{cl} that is a subset of the ideal generated by the resulting exact Gröbner basis G_{cl} (discussed in the section **4**). This can be combined with the above.

3. Compute a close enough system F_{cl} whose Gröbner basis is the resulting exact Gröbner basis G_{cl} (discussed in the section **5**).

Moreover, we use the following notations. By $\mathrm{hc}(f)$ we denote a coefficient of head monomial of $f(\vec{x})$. For a set of polynomials F, we denotes the set of head terms of elements in F by $\mathrm{ht}(F)$. For a polynomial $f(\vec{x})$ we denote the set of terms of monomials with non-zero coefficients in $f(\vec{x})$ by $\mathrm{supp}(f)$. We denote the S-polynomial of $f(\vec{x})$ and $g(\vec{x})$ by $\mathrm{Spoly}(f, g)$ and the normal form of $f(\vec{x})$ w.r.t. G by $\overline{f(\vec{x})}^G$. For a polynomial $f(\vec{x})$, we denote the dense coefficient vector of $f(\vec{x})$ w.r.t. the term order by $\overrightarrow{f(\vec{x})}$ or just $\overrightarrow{f}$.

3 Nearby Gröbner Basis

In this section, we consider the following sub-problem.

Problem 2 [Nearby Gröbner basis of its self]
For the given $\tilde{G}_{app} \subset \mathbb{F}[\vec{x}]$, compute $G_{cl} \subset \mathbb{R}[\vec{x}]$ such that G_{cl} is a Gröbner basis of ideal(G_{cl}) in the exact sense and G_{cl} is close to $\tilde{G}_{app}$. $\triangleleft$

We formalize the problem as follows. For each $\tilde{g}_i \in \tilde{G}_{app}$, let $g_i(\vec{x})$ be a parametric polynomial over $\mathbb{R}[\vec{p}]$ such that

$$g_i(\vec{x}) = \sum_{t_j \in \mathrm{supp}(\tilde{g})} p_{ij} t_j \in \mathbb{R}[\vec{p}][\vec{x}] \tag{3.1}$$

and put $G_{par} = \{g_1, \ldots, g_m\}$, the set of polynomials $g_i(\vec{x})$. The problem is solved if we can find a specialization $\mathfrak{S} : \mathbb{R}[\vec{p}] \to \mathbb{R}$ such that $\mathfrak{S}(G_{par})$ is a Gröbner basis of its self and close to $\tilde{G}_{app}$. For this computation, we can use the algorithms for Comprehensive Gröbner System (see [9] and citations therein). However, computing a comprehensive Gröbner system of G_{par} is too much for our purpose since we only need a one branch (the resulting basis is the given set of polynomials). The following trivial lemma makes the problem easier to solve under our assumptions.

Lemma 1 (Existence of a nearby basis)
For any $\tilde{G}_{app} \subset \mathbb{F}[\vec{x}]$, there always exists $G_{cl} \subset \mathbb{R}[\vec{x}]$. ◁

Proof By the assumption, $\tilde{G}_{app}$ is minimal (though it may not be a Gröbner basis) hence $\mathrm{ht}(\tilde{G}_{app})$ is a Gröbner basis of its self. □

By this trivial lemma, we focus to compute a close Gröbner basis with the same head terms as those of $\tilde{G}_{app}$. This is reasonable since the head terms play the essential role in the Gröbner basis theory hence any approximate Gröbner basis may (or should) take care of them. With this restriction the problem is similar to the inverse Gröbner basis problem [23], however, our input is not a monomial ideal and we have to minimize the difference from the given basis. Therefore, the problem is equivalent to an optimization problem:

$$\begin{array}{l} \underset{\vec{p}}{\text{minimize}} \ \sum_{i=1}^{m} \|g_i(\vec{x}) - \tilde{g}_i(\vec{x})\| \\ \text{subject to } \mathrm{ht}(g_i) = \mathrm{ht}(\tilde{g}_i) \text{ and } \overline{\mathrm{Spoly}(g_i, g_j)}^{G_{par}} = 0 \end{array} \tag{3.2}$$

where $\|\cdot\|$ is the Euclidean norm (but not limited to this norm). Moreover, we can convert $\overline{\mathrm{Spoly}(g_i, g_j)}^{G_{par}} = 0$ into polynomial constraints in $\vec{p}$ since we can put $\mathrm{hc}(g_i) = 1$ by the assumption $\mathrm{ht}(g_i) = \mathrm{ht}(\tilde{g}_i)$ hence any monomial reductions can be done over $\mathbb{R}[\vec{p}]$. In this case, we have to minimize $\sum_{i=1}^{m} \|p_{i1} \times g_i(\vec{x}) - \tilde{g}_i(\vec{x})\|$ instead.

This optimization problem can be solved by the well known method of Lagrange multipliers or cylindrical algebraic decomposition since the problem 2 does not require the optimum G_{cl} but only close to $\tilde{G}_{app}$.

Example 1 [Nearby Gröbner basis of its self]
We compute G_{cl} for the approximate Gröbner basis $\tilde{G}_{app} = \{1.0xy^3 + 0.224453x + 0.0437575,\ 1.0x^3 - 7.87965 \times 10^{-8} x^2\}$ given in the section 1.1. At first, we construct a set of parametric polynomials:

$$G_{par} = \{g_1(\vec{x}) = xy^3 + p_{12}x + p_{13},\ g_2(\vec{x}) = x^3 + p_{22}x^2\}.$$

The corresponding polynomial constraints are computed as

$$\mathrm{Spoly}(g_1, g_2) = -p_{22}x^2y^3 + p_{12}x^3 + p_{13}x^2 \Longrightarrow \overline{\mathrm{Spoly}(g_1, g_2)}^{G_{par}} = p_{13}x^2 + p_{13}p_{22}x.$$

Therefore, we solve the optimization problem:

$$\begin{gathered}\underset{(p_{11},p_{12},p_{13},p_{21},p_{22})}{\text{minimize}} \quad \|p_{11}g_1-\tilde{g}_1\|+\|p_{21}g_2-\tilde{g}_2\| \\ \text{subject to } p_{13}=0 \text{ and } p_{13}p_{22}=0.\end{gathered}$$

The target function has an optimum at

$$p_{11}=1,\ p_{12}=\frac{1122266401590457}{5000000000000000},\ p_{13}=0,\ p_{21}=1,\ p_{22}=-\frac{246239085291621}{3125000000000000000000}.$$

Finally, we have the following solution for the problem 2.

$$\begin{aligned}G_{cl}=&\{xy^3+\tfrac{1122266401590457}{5000000000000000}x,\ x^3-\tfrac{246239085291621}{3125000000000000000000}x^2\}\\ \approx&\{xy^3+0.224453x,\ x^3-7.87965\times 10^{-8}x^2\}.\end{aligned}$$

◁

In this subsection, we parameterized the given polynomials as in the equation (3.1) hence implicitly we restricted the support of parametric polynomial to that of the given polynomial. In general, we have to assume that the parameterized polynomial $g_i(\vec{x})$ has all the possible terms but the number of such terms may be infinite (e.g. the lexicographic order). The aim of this paper is answering to the natural question: "What is that we computed?" so our implicit assumption in the equation (3.1) is acceptable (or required).

We note that in our preliminary implementation on Mathematica we use the built-in function: FindInstance with NMinimize for simplicity's sake since the method of Lagrange multipliers is not practical. We just find a numerical local optimum and find exact values satisfying the constraints near the numerical optimum.

4 Nearby Polynomial System

In this section, we consider the following sub-problem.

Problem 3 [Nearby polynomial system]
For the given $\tilde{F}_{app}\subset\mathbb{F}[\vec{x}]$ and $G_{cl}\subset\mathbb{R}[\vec{x}]$, compute $F'_{cl}\subset\mathbb{R}[\vec{x}]$ generated by ideal(G_{cl}) in the exact sense and F'_{cl} is close to $\tilde{F}_{app}$. ◁

This is equivalent to find the polynomials $h_{ij}\in\mathbb{R}[\vec{x}]$:

$$f_i(\vec{x})=\sum_{j=1}^{m}h_{ij}(\vec{x})g_j(\vec{x}) \quad (i=1,\ldots,n) \tag{4.1}$$

satisfying $\sum_{i=1}^{n}\|f_i(\vec{x})-\tilde{f}_i(\vec{x})\|$ is small. As in the implicit assumption of the equation (3.1), we need a discussion for the supports of polynomials $f_i(\vec{x})$. The supports of $\tilde{f}_i(\vec{x})$ are not reliable since $\tilde{F}_{app}$ is the given polynomial system with inexact coefficients. The reliable information we have is only the given G_{cl} since this is over $\mathbb{R}$ and is a Gröbner basis of its self in the exact sense.

Hence we assume that the support of $f_i(\vec{x})$ is the union of $\mathrm{supp}(\tilde{h}_{ij}(\vec{x})g_j(\vec{x}))$ and $\mathrm{supp}(r_i(\vec{x}))$ over $\mathbb{R}[a,b]$ satisfying

$$\tilde{f}_i'(\vec{x}) = \sum_{j=1}^{m} \tilde{h}_{ij}(\vec{x})g_j(\vec{x}) + \tilde{r}_i(\vec{x}),\ \tilde{r}_i(\vec{x}) \prec g_1(\vec{x}), \ldots, g_n(\vec{x})$$

where $\tilde{f}_i'(\vec{x}) = a\tilde{f}_i(\vec{x}) + b\sum t_{jk} \times g_j(\vec{x}) \in \mathbb{R}[a,b][\vec{x}]$ and the condition of the sum is $\mathrm{supp}(t_{jk} \times g_j(\vec{x})) \cap \mathrm{supp}(\tilde{f}_i(\vec{x})) \neq \phi$ for any term t_{jk} in $\mathbb{R}[\vec{x}]$.

Therefore, the problem 2 under this assumption is equivalent to the following least square problem over $\mathbb{R}$.

$$\underset{\overrightarrow{f_i} \in \mathbb{R}^{\#\mathrm{supp}(f_i)}}{\text{minimize}}\ \|\overrightarrow{f_i} - \overrightarrow{\tilde{f}_i}\|,\quad \mathcal{M}_i\overrightarrow{h_i} = \overrightarrow{f_i} \tag{4.2}$$

where $\mathcal{M}_i$ is similar to the Macaulay matrix (see [11, 2, 10, 15] for details). We note that we can construct $\mathcal{M}_i$ of full column rank since G_{cl} is a (minimal) Gröbner basis hence the residual of monomial reduction by G_{cl} is unique and does not depend on the order of reducers used in G_{cl}. This can be solved by the exact arithmetic (the generalized inverse by LSP decomposition, see [5, 6] for example).

Example 2 [Continued from Example 1]
We compute F_{cl}' for $\tilde{F}_{app}$ and G_{cl} from the example 1:

$$\begin{aligned}\tilde{F}_{app} &= \{\tilde{f}_1(\vec{x}) = 0.01084x^3y + 0.891x^3,\ \tilde{f}_2(\vec{x}) = 0.503xy^3 + 0.1129x + 0.02201\},\\ G_{cl} &= \{g_1(\vec{x}) = xy^3 + \tfrac{1122266401590457}{5000000000000000}x,\ g_2(\vec{x}) = x^3 - \tfrac{246239085291621}{3125000000000000000000}x^2\}.\end{aligned}$$

According to the discussion above, we have $\mathrm{supp}(f_1(\vec{x})) = \{x^3y,\ x^3,\ x^2y,\ x^2\}$ and $\mathrm{supp}(f_2(\vec{x})) = \{xy^3,\ x,\ 1\}$. Hence the problem is equivalent to the following set of least squares.

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \\ c_1 & 0 \\ 0 & c_1 \end{pmatrix}\overrightarrow{h_1} = \begin{pmatrix} \frac{271}{25000} \\ \frac{891}{1000} \\ 0 \\ 0 \end{pmatrix},\ \begin{pmatrix} 1 \\ c_2 \\ 0 \end{pmatrix}\overrightarrow{h_2} = \begin{pmatrix} \frac{503}{1000} \\ \frac{1129}{10000} \\ \frac{2201}{100000} \end{pmatrix}$$

where $c_1 = -\frac{246239085291621}{3125000000000000000000}$ and $c_2 = \frac{1122266401590457}{5000000000000000}$. By computing the generalized inverses (Moore-Penrose inverses) and multiply them, we have the following $F_{cl}' \in \mathbb{R}[\vec{x}]$.

$$\begin{aligned}F_{cl}' = \{&\tfrac{105859375000000000000000000000000000000000}{9765625000000060633687125254201498612807641}x^3y + \tfrac{8701171875000000000000000000000000000000000}{9765625000000060633687125254201498612807641}x^3\\ &-\tfrac{8341349014253661375000000000000000}{9765625000000060633687125254201498612807641}x^2y - \tfrac{685621953108857221875000000000000000}{9765625000000060633687125254201498612807641}x^2,\\ &\tfrac{13208519383697812976500000000000}{26259481876138792906019153468849}xy^3 + \tfrac{29646955038160690740895624266330521}{262594818761387929060191534688490000}x\}\\ \approx \{&0.01084x^3y + 0.891x^3 - 8.54154 \times 10^{-10}x^2y - 7.02077 \times 10^{-8}x^2,\ 0.503xy^3 + 0.1129x\}.\end{aligned}$$

In this case, we can easily confirm that a Gröbner basis of the resulting polynomial system F_{cl}' is the given basis G_{cl} hence F_{cl}' is also a solution of the problem 5 in the next section. ◁

The above problems 2 and 3 can be combined into the following problem so that resulting F'_{cl} is much better than that is computed from G_{cl} separately.

Problem 4 [Nearby polynomial system directly] For the given $\tilde{F}_{app}, \tilde{G}_{app} \subset \mathbb{F}[\vec{x}]$, compute $F'_{cl}, G_{cl} \subset \mathbb{R}[\vec{x}]$ that are close to $\tilde{F}_{app}, \tilde{G}_{app}$, respectively, such that F'_{cl} is generated by ideal(G_{cl}) and G_{cl} is a Gröbner basis of its self in the exact sense. ◁

We combine the minimization problems (3.2) and (4.2). Since the generalized inverse can be computed over any field so we extend the Macaulay like matrix $\mathcal{M}$ over $\mathbb{R}$ to that over the rational function field $\mathbb{R}(\vec{p})$, we can solve the minimization problem (4.2) over $\mathbb{R}(\vec{p})$. Then we solve the minimization problem (3.2). The resulting F'_{cl} is not different from the problem 2 since G_{cl} is a (minimal) Gröbner basis hence we construct $\mathcal{M}$ of full column rank as noted before.

However, if we solve the following modified version of the minimization problem (3.2) the resulting F'_{cl} is different.

$$\begin{array}{l} \underset{\vec{p}}{\text{minimize}}\ \sum_{i=1}^{m} \|f_i(\vec{x}) - \tilde{f}_i(\vec{x})\| \\ \text{subject to } \mathrm{ht}(g_i) = \mathrm{ht}(\tilde{g}_i) \text{ and } \overline{\mathrm{Spoly}(g_i, g_j)}^{G_{par}} = 0. \end{array}$$

For example, we get the following result for the example 2.

$$\begin{array}{l} G_{cl} = \{xy^3 + \frac{1129}{5030}x,\ x^3\}, \\ F'_{cl} = \{\frac{271}{25000}x^3y + \frac{891}{1000}x^3,\ \frac{503}{1000}xy^3 + \frac{1129}{10000}x\}. \end{array}$$

5 Nearby System of the Ideal

In general, the resulting F'_{cl} of problems 3 and 4 does not generate the ideal generated by the given G_{cl} and only satisfies ideal$(F'_{cl}) \subseteq$ ideal(G_{cl}). This condition is not enough for understanding the relation between the given erroneous polynomial system and its approximate Gröbner basis hence in this section, we consider the following sub-problem.

Problem 5 [Nearby polynomial system of ideal] For the given $\tilde{F}_{app} \subset \mathbb{F}[\vec{x}]$ and $G_{cl} \subset \mathbb{R}[\vec{x}]$, compute $F_{cl} \subset \mathbb{R}[\vec{x}]$ satisfying ideal$(F_{cl}) =$ ideal(G_{cl}) in the exact sense and F_{cl} is close to $\tilde{F}_{app}$.◁

In this paper, we only give a sufficient condition for a special case, that there is no solution to this problem. The minimal generators of the ideal (see [14, 4] for related problems) is our main tool.

Definition 1 (Minimal Generators)
Let $G \subset \mathbb{R}[\vec{x}]$ be a generating set of the ideal $I \subseteq \mathbb{R}[\vec{x}]$. If $I \neq$ ideal$(G \setminus \{g\})$ for any $g \in G$, then G is called a **minimal generating set** *of I and the polynomials in G are called* **minimal generators** *of I.* ◁

Lemma 2 *Let $I \subseteq \mathbb{R}[\vec{x}]$ be the ideal generated by a minimal Gröbner basis $G = \{g_1, \dots, g_m\}$. If all the elements in G is a monomial, the cardinality of any generating set of I is larger than or equal to m.* ◁

Proof We assume that the lemma is not valid hence there exists a generating set $F = \{f_1, \ldots, f_n\}$ with $n < m$. G and F are generating sets of I hence $I = \mathrm{ideal}(G) = \mathrm{ideal}(F)$. This means that there exist $h_{ij}, s_{jk} \in \mathbb{R}[\vec{x}]$ such that

$$g_i(\vec{x}) = \sum_{j=1}^{n} h_{ij}(\vec{x}) f_j(\vec{x}), \;\; f_j(\vec{x}) = \sum_{k=1}^{m} s_{jk}(\vec{x}) g_k(\vec{x}).$$

Composing those two expressions, we obtain

$$g_i(\vec{x}) = \sum_{k=1}^{m} \left(\sum_{j=1}^{n} h_{ij}(\vec{x}) s_{jk}(\vec{x}) \right) g_k(\vec{x}).$$

Though we do not have $\sum_{j=1}^{n} h_{ij}(\vec{x}) s_{jk}(\vec{x}) = \delta_{ik}$ in general, $\sum_{j=1}^{n} h_{ij}(\vec{x}) s_{jk}(\vec{x})$ must have a constant term for $k = i$ since $g_1(\vec{x}), \ldots, g_m(\vec{x})$ are monomials and G is minimal. In contrast, by the same reason, $\sum_{j=1}^{n} h_{ij}(\vec{x}) s_{jk}(\vec{x})$ must not have any constant term for $k \neq i$.

Therefore, we have the following equations in $\mathbb{R}[\vec{x}]/\mathrm{ideal}(\vec{x})$.

$$\sum_{j=1}^{n} h_{ij}(\vec{x}) s_{jk}(\vec{x}) = \delta_{ik} \;\; (i = 1, \ldots, n) \; (k = 1, \ldots, m).$$

This is equivalent to the following matrix representation.

$$HS = E_m, \; H = (h_{ij}) \in \mathbb{R}^{m \times n}, \; S = (s_{jk}) \in \mathbb{R}^{n \times m}$$

where E_m is the identity matrix of dimension m. The assumption $n < m$ means that this system does not have any solution hence the lemma is valid. □

We note that the lemma is valid only for monomial ideals. In general, there are several generating sets of I with fewer elements than that of minimal generating sets of Gröbner bases of I. For example, the ideal generated by $F = \{-4x^3 - 4x^2 + xy - 5y - 4, \; 7x^3 - 9x^2y + 6xy^2\}$ has the reduced Gröbner basis with 4 elements and its minimal generating set has at least 3 elements.

Example 3 [Unfaithful Gröbner basis]
We show an example of the lemma with the input $\tilde{F}_{app}$ below.

$$\begin{aligned} \tilde{F}_{app} = \{ & 0.6533x^3 + 0.1359xy^2 + 0.08952xyz + 0.5586y^2z + 0.1688yz^2 + 0.9009z^3 + 0.7794x^2 \\ & \quad + 0.4555xy + 0.1825xz + 0.4381y^2 + 0.3805yz, \\ & 0.5142z^3 + 0.981xy + 0.1469yz + 0.06382z^2, \\ & 0.3685x^3 + 0.2276xyz + 0.4191xz^2 + 0.27289y^3 + 0.427y^2z + 0.2659yz^2 + 0.9568yz\}. \end{aligned}$$

For this input, the following $\tilde{G}_{app}$ is one of approximate Gröbner bases w.r.t. the graded lexicographic order ($x \succ y \succ z$), computed by Mathematica.

$$\tilde{G}_{app} = \{1.0z^2, \; 1.0yz, \; 1.0y^2, \; 1.0xz, \; 1.0xy, \; 1.0x^2\}.$$

Trivially, we have $G_{cl} = \{z^2, yz, y^2, xz, xy, x^2\}$ and this is a minimal generating set of $\mathrm{ideal}(G_{cl})$. Therefore, by the lemma 2, there is no polynomial system close to $\tilde{F}_{app}$ whose Gröbner basis is G_{cl} since it has only 3 elements which is fewer than that of G_{cl}. ◁

Note that the resulting approximate Gröbner basis depends on the algorithm and assumptions (precision, accuracy and so on) we used. Hence the above example does not mean that Mathematica's approximate Gröbner basis is odd, and by the suggestion from the lemma 2 we can have a chance to change assumptions or algorithms. This is a merit of doing a backward error analysis for approximate Gröbner basis.

6 Conclusion

Computing an approximate Gröbner basis basically comes from numerical computations and do not come from theoretical computations though we can use comprehensive Gröbner system to solve the problem partially. However, as shown in this paper, for some backward error analysis of the resulting approximate Gröbner basis, we have the following interesting problems.

- For the given set of polynomials F, compute a close enough G which is a Gröbner basis of ideal(G).
- For the given set of polynomials F and Gröbner basis G of ideal(G), compute a close enough set of polynomials to F, which generates a sub-ideal of ideal(G).
- For the given set of polynomials F and Gröbner basis G of ideal(G), compute a close enough set of polynomials to F, which generates ideal(G).

We give a proof of concept method for solving some of the above problems. However, basically they are not solved and still open since the optimizations for the former problems are not well analyzed and the necessary and sufficient condition for the last problem is not given. The author thinks that solving these problems is very important for approximate Gröbner basis and if solved we can discuss approximate Gröbner basis is well defined or not in the exact context. Some further examples can be found in the appendix.

Acknowledgements

The author wishes to thank Prof. Kaltofen and Lichtblau for their comments at Hybrid2011 which led me to the problem in this paper though it is not solved yet. This work was supported in part by Japanese Ministry of Education, Culture, Sports, Science and Technology under Grant-in-Aid for Young Scientists, MEXT KAKENHI (22700011).

References

[1] T. Becker, V. Weispfenning, and H. Kredel. *Gröbner bases: a computational approach to commutative algebra*. Graduate texts in mathematics. Springer-Verlag, 1993.

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

[3] J.-C. Faugère and Y. Liang. Pivoting in Extended Rings for Computing Approximate Gröbner Bases. *Mathematics in Computer Science*, 5:179–194, 2011.

[4] Hans and Schoutens. Computing the minimal number of equations defining an affine curve ideal-theoretically. *Journal of Pure and Applied Algebra*, 177(1):95 – 101, 2003.

[5] O. H. Ibarra, S. Moran, and R. Hui. A generalization of the fast lup matrix decomposition algorithm and applications. *Journal of Algorithms*, 3(1):45 – 56, 1982.

[6] C.-P. Jeannerod. Lsp matrix decomposition revisited. Technical Report Research Report 2006-28, École normale supérieure de Lyon, LIP, 2006.

[7] E. Kaltofen, J. P. May, Z. Yang, and L. Zhi. Approximate factorization of multivariate polynomials using singular value decomposition. *Journal of Symbolic Computation*, 43(5):359 – 376, 2008.

[8] E. Kaltofen, Z. Yang, and L. Zhi. Approximate greatest common divisors of several polynomials with l inearly constrained coefficients and singular polynomials. In *ISSAC 2006*, pages 169–176. ACM, 2006.

[9] D. Kapur, Y. Sun, and D. Wang. A new algorithm for computing comprehensive Gröbner systems. In *Proceedings of the 2010 International Symposium on Symbolic and Algebraic Computation*, ISSAC '10, pages 29–36, New York, NY, USA, 2010. ACM.

[10] 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.

[11] 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.

[12] D. Lichtblau. Gröbner bases in mathematica 3.0. *The Mathematica Journal*, 6(4):81 – 88, 1996.

[13] D. Lichtblau. Exact computation using approximate gröbner bases. Work presented at ACA 2008, Applications of Computer Algebra. Session: Gröbner Bases and their Applications, 2008. http://library. wolfram.com/infocenter/Conferences/7537/.

[14] Y. Luo and Z. Lu. An estimation of the number of elements of minimal generators for a polynomial ideal. MM Research Preprints Vol.25, pages 179–189, KLMM, AMSS, Academia Sinica, 2006.

[15] K. Nagasaka. Computing a structured gröbner basis approximately. In *Proceedings of the 36th international symposium on Symbolic and algebraic computation*, ISSAC '11, pages 273–280, New York, NY, USA, 2011. ACM.

[16] K. Nagasaka. A symbolic-numeric approach to gröbner basis with inexact input. Work presented at Hybrid 2011, Fields Institute Workshop on Hybrid Methodologies for Symbolic-Numeric Computation, 2011. http://www.cs.uwaterloo.ca/conferences/ hybrid2011/slides/KosakuNagasaka.pdf.

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

[18] T. Sasaki and F. Kako. Term cancellations in computing floating-point Gröbner bases. In *Proceedings of CASC 2010*, volume 6244 of *Lecture Notes in Comput. Sci.*, pages 220–231, Berlin, 2010. Springer.

[19] 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.

[20] 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.

[21] K. Shirayanagi and M. Sweedler. A theory of stabilizing algebraic algorithms. *Technical Report 95-28*, pages 1–92, 1995. http://www.ss.u-tokai.ac.jp/s̃hirayan/msitr95-28.pdf.

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

[23] A. Taylor. The inverse gröbner basis problem in codimension two. *Journal of Symbolic Computation*, 33(2):221 – 238, 2002.

[24] 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.

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

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

Appendix

We show some further examples for approximate Gröbner basis [3, 15, 17] other than Mathematica's. Note that all the examples in this paper can be found in the following URL with our preliminary implementation on Mathematica.
http://wwwmain.h.kobe-u.ac.jp/˜nagasaka/research/snap/issac12.nb

Example 4
The following $\tilde{F}_{app}$ and $\tilde{G}_{app}$ are cited from the example 6.1 in [3] (w.r.t. the graded reverse lexicographic order with $x \succ y \succ z$).

$$\begin{aligned}\tilde{F}_{app} =\{&0.10519760 \times 10^{-5}x - 0.70383719y - 0.74858720z + 1,\\ &x - 0.10365584 \times 10^{-5}y + 0.99083786z - 0.74199025,\\ &0.12288986x + y - 0.11161051 \times 10^{-5}z - 0.32687369,\\ &0.49279128 \times 10^{-1}x + y + 0.63703207z - 0.98207322\},\\ \tilde{G}_{app} =\{&x + 0.24863582,\ y - 0.35742965,\ z - 0.99978662\}.\end{aligned}$$

Trivially, we have the following solution for the problem 2.

$$G_{cl} = \left\{x + \frac{12431791}{50000000}, y - \frac{7148593}{20000000}, z - \frac{49989331}{50000000}\right\}.$$

We have the following solution for the problem 3 (we show only them in floating-point representation with a limited number of decimal places since the full rational representations are too large here).

$$\begin{aligned}F'_{cl} \approx \{&1.05101 \times 10^{-6}x - 0.703837y - 0.748587z + 1.0,\\ &1.0x - 1.03566 \times 10^{-6}y + 0.990838z - 0.74199,\\ &0.12289x + 1.0y - 1.12662 \times 10^{-6}z - 0.326874,\\ &0.0492791x + 1.0y + 0.637032z - 0.982073\}.\end{aligned}$$

In this case, the resulting F'_{cl} is close to $\tilde{F}_{app}$ and its difference is about 1.93532×10^{-8} in the Euclidean norm. This result indicates that the given $\tilde{G}_{app}$ is enough close to Gröbner bases of nearby systems of $\tilde{F}_{app}$.

Moreover, if we minimize the difference between F'_{cl} and $\tilde{F}_{app}$ as in the end of the section 4. we get the following result (we show only them in floating-point representation).

$$\begin{aligned}G_{cl} &\approx \{x + 0.248636,\ y - 0.35743,\ z - 0.999787\},\\ F'_{cl} &\approx \{1.05172 \times 10^{-6}x - 0.703837y - 0.748587z + 1.0,\\ &\quad 1.0x - 1.03676 \times 10^{-6}y + 0.990838z - 0.74199,\\ &\quad 0.12289x + 1.0y - 1.11655 \times 10^{-6}z - 0.326874,\\ &\quad 0.0492791x + 1.0y + 0.637032z - 0.982073\}.\end{aligned}$$

The resulting F'_{cl} and G_{cl} are close to the given system and Gröbner basis, respectively. The differences in the Euclidean norm are 5.19825×10^{-9} and 2.49691×10^{-8}, respectively. $\triangleleft$

Example 5
The following $\tilde{F}_{app}$ and $\tilde{G}_{app}$ are cited from the example 3 in [15] (w.r.t. the graded lexicographic order with $x \succ y$).

$$\begin{aligned}\tilde{F}_{app} &= \{1.01x^2 - 2.09y^2 + 0.002,\ 4.03x^2y + 3.06xy,\ 2.04x^2y + 0.504x^2 + 1.504xy - 1.02y^2\},\\ \tilde{G}_{app} &= \{2.03847y^3 + 0.0485655x^2 + 0.745414xy - 0.100253y^2,\ 1.10491x^2 - 2.28084y^2\}.\end{aligned}$$

At first, we construct a set of parametric polynomials:

$$G_{par} = \{g_1(\vec{x}) = y^3 + p_{12}x^2 + p_{13}xy + p_{14}y^2,\ g_2(\vec{x}) = x^2 + p_{22}y^2\}.$$

In this case, the head terms of these two polynomials are co-prime hence there is no constraint on parameters. Therefore, we have $G_{cl} = \tilde{G}_{app}$ with just rationalized coefficients and we have the following solution for the problem 3 (we show only them in floating-point representation).

$$\begin{aligned} F'_{cl} \approx \{&1.01199x^2 - 2.08903y^2, \\ &4.03796x^2y + 0.00385432y^3 + 2.25236 \times 10^{-16}x^2 + 3.04946xy + 1.09111 \times 10^{-16}y^2, \\ &2.0241x^2y - 0.00770481y^3 + 0.495998x^2 + 1.52507xy - 1.02388y^2\}. \end{aligned}$$

The resulting F'_{cl} is far from the given system $\tilde{F}_{app}$ since $\tilde{F}_{app}$ is inconsistent. Moreover, if we minimize the difference between F'_{cl} and $\tilde{F}_{app}$, the optimization is a little bit hard so we could not find the optimum. The result is the followings (we show only them in floating-point representation).

$$\begin{aligned} G_{cl} &\approx \{y^3 + 0.367287xy + 0.0029292y^2,\ x^2 - 2.05519y^2\}, \\ F'_{cl} &\approx \{1.00722x^2 - 2.07003y^2, \\ &\quad 4.00659x^2y + 0.00383635y^3 + 0.00949385x^2 + 3.02575xy + 0.00461946y^2, \\ &\quad 2.0084x^2y - 0.00766937y^3 + 0.49866x^2 + 1.51321xy - 1.01277y^2\}. \end{aligned}$$

The resulting F'_{cl} is not closer than the above due to that we could not reach the optimum but this is one of other consistent systems near the given. ◁

Example 6
The following $\tilde{F}_{app}$ and $\tilde{G}_{app}$ are cited from the example 5 in [17] (w.r.t. the lexicographic order with $x \succ y$).

$$\begin{aligned} \tilde{F}_{app} &= \{x^3 + x^2y^2,\ x^2y^2 - y^3,\ -x^2y + 1.0001x^2 + xy^2 + 0.9999y^2\}, \\ \tilde{G}_{app} &= \{y^6 - 1.0001y^3,\ xy^2 + 0.9999y^4,\ x^2 + 0.9998xy^3 + 0.9999xy^2 + 0.9998y^2\}. \end{aligned}$$

In this case, we solve the optimization problem with 8 parameters and have the following solution for the problem 2 (we show only them in floating-point representation). We note that this is not the optimum solution since we could not reach the optimum in a reasonable period hence we computed a solution within the difference 10^{-6}. Moreover, the resulting G_{cl} is very close to $\tilde{G}_{app}$ but not the same.

$$\begin{aligned} G_{cl} \approx \{&1.00000y^6 - 1.00010y^3,\ 1.00000xy^2 + 0.999900y^4, \\ &1.00000x^2 + 0.999800xy^3 + 0.999900xy^2 + 0.999800y^2\}. \end{aligned}$$

With this result, we have the following solution for the problem 3 (we show only them in floating-point representation).

$$\begin{aligned} F'_{cl} \approx \{&1.00002x^3 + 0.999980x^2y^2 + 0.0000199594xy^4 - 0.0000199614y^6 - 0.0000199594y^3, \\ &1.00003x^2y^2 - 0.0000250508xy^4 + 0.0000250533y^6 - 0.999975y^3, \\ &-1.00000x^2y + 1.00010x^2 + 2.69562 \times 10^{-8}xy^3 \\ &\qquad +1.00000xy^2 - 2.69589 \times 10^{-8}y^5 - 9.54069 \times 10^{-8}y^4 + 0.999900y^2\}. \end{aligned}$$

The result indicates that $\tilde{F}_{app}$ lacks higher order terms in y to have its Gröbner basis with the shape of $\tilde{G}_{app}$. In fact, $F_\delta = \{x^3 + x^2y^2,\ x^2y^2 - y^3,\ -x^2y + (1+\delta)x^2 + xy^2 + (1-\delta)y^2\}$ has the following comprehensive Gröbner system and does not have any (minimal) Gröbner basis with y^6 if $\delta \neq 0$.

$$\begin{aligned}
&\delta = 0 \ \Rightarrow\ \{x^2 - y^5 - y^4 + y^2,\ -xy^2 - y^4, y^6 - y^3\},\\
&\delta = 1 \ \Rightarrow\ \{2x^2 + xy^2 - y^3,\ xy^3 + y^3,\ -y^4 + y^3\},\\
&\delta = -1 \ \Rightarrow\ \{x^3 + y^3,\ x^2y + y^3 - 2y^2,\ -xy^2 - y^3,\ y^4 - y^3\},\\
&\delta^3 - \delta \neq 0 \ \Rightarrow\ \{-\delta y^4 + \delta y^3,\ (-\delta^2 + \delta)xy^2 + (-\delta^2 + \delta)y^3,\\
&\qquad\qquad (-\delta^4 - \delta^3 + \delta^2 + \delta)x^2 + (2\delta^3 - 2\delta)y^3 + (\delta^4 - \delta^3 - \delta^2 + \delta)y^2\}.
\end{aligned}$$

However, we note that the aim of the method proposed in [17] is numerical stability hence the resulting $\tilde{F}_{app}$ is intended to have a similar shape of a Gröbner basis of F_δ with $\delta = 0$. $\triangleleft$