

MI Preprint Series

**Kyushu University
The Global COE Program
Math-for-Industry Education & Research Hub**

Derivation of specific conditions with Comprehensive Groebner Systems

**Katsusuke Nabeshima
& Hiroshi Yoshida**

MI 2010-11

(Received March 5, 2010)

Faculty of Mathematics
Kyushu University
Fukuoka, JAPAN

Derivation of specific conditions with Comprehensive Gröbner Systems

Katsusuke Nabeshima · Hiroshi Yoshida

Abstract Here we present an efficient calculation of comprehensive Gröbner systems to derive specific conditions for neural circuits as well as electric circuits. Comprehensive Gröbner systems (CGS) have been applied to problems with a small number of parameters such as the automatic geometric theorem proving and the inverse kinematics problem of a robot arm. In CGS, however, a larger number of parameters make its calculation less tractable. Therefore, we take ‘not-equal’ conditions into account during CGS calculation, resulting in a reduced format of CGS of parametric systems even though many parameters exist. Using our implemented CGS, we derive specific conditions such as resonant conditions that play an important role in physical, mechanical, and biological phenomena. The obtained conditions lead to analysis of realistic neural circuits having many parameters, and provide us with a possibility of positive CGS.

Keywords Resonant Condition · Electric and Neural Circuits · Symbolic Computation · Comprehensive Gröbner Systems

Mathematics Subject Classification (2000) 35B34 · 68W30 · 13P10

1 Introduction

Gröbner bases have been applied to various systems such as electric and digital systems [2, 21, 30]. Considering parameters as well as variables during Gröbner basis calculation is called comprehensive Gröbner systems (CGS). CGS can classify the roots of parametric polynomial equations [27]. On the other side, in mechanical, electric, or neural systems, when parameters satisfy some specific condition, resonance, synchronization, or entrainment occurs. This coincidence has inspired us to exploit CGS to derive specific conditions of a system.

Analysis of specific conditions is important in many fields because a system under resonance shows remarkable characters. Indeed, once resonance vibration occurs in a mechanical system, the amplitude of the system grows larger and larger [12, §22–§27]. In electric circuits, a resonant condition is essential for tuning [1]. Further, in chemical reactions, synchronization or entrainment as a kind of resonance plays an important role in chemical oscillations [10, §5]. In biomedical systems, metabolic kinetics of chemicals inside a human body is usually described as a compartmental model [3]. Resonant conditions in a compartmental model was also explored [26].

Thus, derivation of specific conditions is important for understanding a system, but a system with many parameters is difficult for obtaining specific conditions by hand. Therefore, we propose an application of Comprehensive Gröbner systems (CGS) to obtaining specific conditions, exemplifying electric and neural systems.

K. Nabeshima
Graduate School of Information Science and Technology, Osaka University, Machikaneyama 1-1, Toyonaka 560-0043, Osaka, JAPAN.
Japan Science and Technology Agency, CREST, Sanbancho, Chiyoda-ku, Tokyo, 102-0075, JAPAN.
E-mail: nabeshima@math.sci.osaka-u.ac.jp

H. Yoshida
Faculty of Mathematics, Kyushu University, Itou, Motouoka 744, Nishi-ku, FUKUOKA 819-0395 Japan.
E-mail: phiroshi@math.kyushu-u.ac.jp

Solving parametric algebraic systems is one of the big topics in computer algebra [6, 9, 13, 27]. Comprehensive Gröbner systems (bases) are key ingredients and are fundamental tools for applications of parametric polynomial systems in several fields both inside and outside mathematics.

Comprehensive Gröbner systems for parametric ideals were introduced, constructed, and studied by Weispfenning [27] in 1992. After Weispfenning's paper was published, Dolzhan and Sturm have implemented and published the software [5]. There was, however, no big development about comprehensive Gröbner systems (or bases) for ten years. Last ten years, the big developments were made by Kurata, Montes, Nabeshima, Sato, Suzuki, and Weispfenning.

[Efficient computation]

- Montes published the new algorithm for computing CGS and its software in 2002 and 2006 [14, 15].
- Suzuki and Sato published the new algorithm for computing CGS in 2006 [25].
- Nabeshima and Kurata improved Suzuki-Sato's algorithm ([25]) in 2007 and 2010 [11, 17].

[Theory]

- Suzuki and Sato presented an alternative definition of comprehensive Gröbner bases (systems) in terms of Gröbner bases in polynomial rings over commutative von Neumann regular rings [24]. This Gröbner basis is called 'alternative comprehensive Gröbner basis (ACGB).' Alternative comprehensive Gröbner bases have the following nice properties which do not hold in standard comprehensive Gröbner bases:
 1. There is a canonical form of an alternative comprehensive Gröbner basis in a natural way.
 2. We can use reductions of an alternative comprehensive Gröbner basis.
- Weispfenning presented a concept of canonical comprehensive Gröbner bases under very general assumptions on the parameter ring [28]. After this paper was published, this result was applied for improving Montes' algorithm by Montes [14].

There are mainly three kinds of algorithms for computing CGS: Weispfenning-Montes' algorithm, Suzuki-Sato's algorithm [25], and ACGB algorithm [24]. In general, Suzuki-Sato's algorithm is the fastest among them in several cases. Therefore, in this paper, we improve and apply Suzuki-Sato's algorithm for solving our problems of electric and neural circuits.

Roughly speaking, a CGS is a parametric Gröbner basis for a parametric polynomial ideal. A CGS is used as a systematic tool to classify the roots of parametric polynomial equations. Therefore, the number of applications of CGS is large. Some interesting applications of CGS have already been introduced:

- Modules of syzygies [18, 19].
- The study of parametric varieties, their size, and their dimension functions [18].
- The study of singular points of a conic [15].
- The inverse kinematics problem for a simple robot [15].
- The automatic geometric theorem proving [16, 29].
- CGS in rings of differential operators [18].
- Kanno, *et al.* [8] have explored a potential of algebraic approach for *parametric polynomial spectral factorization* (PPSF). For computing the parametric spectral factor of a parametric polynomial, one has to classify the roots of parametric polynomial. For this purpose, they devised an algorithm for using CGS computation.

In [22], Shinohara presented three CGS based algorithms for computing PPSF and showed that the fastest one took 72 hours in computation of PPSF of a quartic parametric polynomials. If there are many parameters in parametric problems, it is difficult to obtain results of the applications above. In fact, they showed simple examples in the application above. Therefore, from a practical point of view, an efficient CGS implementation is necessary in order to solve large problems.

On a computer algebra system equipped with Gröbner bases computations, it is easy to implement Suzuki-Sato's algorithm. Nevertheless, Suzuki-Sato's algorithm often produces many segments whose parameter spaces overlap each other, so that the computational cost will swell without careful treatments of superfluous segments. Thus, we need an algorithm whose number of steps becomes small, and an optimal means to obtain irredundant segments which Suzuki and Sato referred to, but did not describe. In this paper, we define '**supports**' as parameter spaces in order to check overlap and unnecessary segments. We describe useful basic manipulations using supports to implement CGS algorithm bases on Suzuki-Sato's.

Moreover, another advantage of our algorithm is the following. Our problems, electric and neural circuits, need to consider 'not-equal ($\neq 0$)' conditions of parameters. To overcome difficulty in calculating CGS against many parameters, we take 'not-equal' conditions into account, leading to an efficient calculation and much reduced format of outputs of

CGS. Suzuki-Sato's algorithm treats only 'equal (= 0)' conditions of parameters. Our algorithm for computing CGS treats both of 'equal' and 'not-equal' conditions of parameters. In this point, our algorithm is more useful than Suzuki-Sato's algorithm for analyzing properties of our problems. Further, our algorithm outputs CGS on a restricted parameter space because we do not need to compute CGS on the whole parameter space. Thus, our algorithm is superior in several cases. In fact, we were able to obtain specific conditions for electric and neural circuits with many parameters by using our implemented CGS.

2 Problem

A problem with respect to an electric or neural circuit is described as a system of equations of the following form:

$$f(V_\ell(t), \mathbf{p}, \sin(\omega t), \cos(\omega t), \int V_j(t)dt, dV_k(t)/dt), \quad (1)$$

where $V_\ell(t)$ ($\ell \in \{i, j, k\}$) denotes a function in t and a vector \mathbf{p} denotes parameters for CGS. This formula (1) includes a linear relation of $\sin(\omega t), \cos(\omega t), \int dt, d/dt$, denoted by f .

The aim is to obtain specific conditions by CGS and to analyze them.

3 Comprehensive Gröbner Systems

In this section, we give the detail of our algorithm for computing CGS. Our algorithm is based on Suzuki-Sato's algorithm. We improve Suzuki-Sato's algorithm [25] for solving our problems.

Let $\mathbf{A} := \{A_1, \dots, A_m\}$ and $\mathbf{X} := \{X_1, \dots, X_n\}$ be finite sets of variables such that $\mathbf{A} \cap \mathbf{X} = \emptyset$. We define K and L as fields such that L is an algebraic closure of K . Let $\text{pp}(\mathbf{X})$, $\text{pp}(\mathbf{A})$ and $\text{pp}(\mathbf{A}, \mathbf{X})$ be the sets of power products of \mathbf{X} , \mathbf{A} and $\mathbf{A} \cup \mathbf{X}$, respectively. We define \mathbb{N} , \mathbb{Q} and \mathbb{C} as the set of natural numbers with 0, the field of rational numbers, and the field of complex numbers, respectively. In this paper, we define $K[\mathbf{A}, \mathbf{X}]$ as a polynomial ring over a field K , and define $K[\mathbf{A}][\mathbf{X}] := (K[\mathbf{A}])[\mathbf{X}]$ as a polynomial ring over a polynomial ring $K[\mathbf{A}]$ (coefficients in a polynomial ring). For a polynomial $f \neq 0$ in a polynomial ring equipped with a term order $<$, let $\text{lpp}_<(f)$, $\text{lc}_<(f)$, and $\text{lm}_<(f)$ be the leading power product of f with respect to $<$, the coefficient of $\text{lm}_<(f)$, and $\text{lc}_<(f) \cdot \text{lpp}_<(f)$, respectively. The degree of f in X_i , denoted by $\deg_{X_i}(f)$, is defined as the degree of f when viewed as a univariate polynomial in X_i . $<_{\mathbf{A}, \mathbf{X}}$ is defined as a term order on $\text{pp}(\mathbf{A}, \mathbf{X})$ such that \mathbf{X} is always greater than \mathbf{A} . That is, any term in $\text{pp}(\mathbf{X})$ is greater than any term in $\text{pp}(\mathbf{A})$. Let $<_{\mathbf{A}}$ and $<_{\mathbf{X}}$ be its restriction on $\text{pp}(\mathbf{A})$ and $\text{pp}(\mathbf{X})$, respectively. For any subset F of $K[\mathbf{A}]$, $\mathbb{V}(F)$ denotes the algebraic set defined by F , that is $\mathbb{V}(F) := \{a \in L^m \mid \forall f \in F, f(a) = 0\}$. Specially, we define $\mathbb{V}(1) := \emptyset$ and $\mathbb{V}(0) := L^m$. For any $\alpha \in L^m$, we define the canonical specialization homomorphism $\sigma_\alpha : K[\mathbf{A}] \rightarrow L$ induced by α , and we naturally extend it to $\sigma_\alpha : K[\mathbf{A}][\mathbf{X}] \rightarrow L[\mathbf{X}]$.

In this paper, angle brackets $\langle \cdot \rangle$ are defined as follows: let $f_1, \dots, f_l \in R$, where R is a commutative ring with identity. Then $\langle f_1, \dots, f_l \rangle := \{\sum_{i=1}^l h_i f_i \mid h_1, \dots, h_l \in R\}$.

Definition 1 (Comprehensive Gröbner Systems) Let F be a subset of $K[\mathbf{A}][\mathbf{X}]$, $\mathcal{A}_1, \dots, \mathcal{A}_l$ algebraically constructible subsets of L^m , and let G_1, \dots, G_l be subsets of $K[\mathbf{A}][\mathbf{X}]$. Let \mathcal{S} be a subset of L^m such that $\mathcal{S} \subseteq \mathcal{A}_1 \cup \dots \cup \mathcal{A}_l$. A finite set $\mathcal{G} = \{(\mathcal{A}_1, G_1), \dots, (\mathcal{A}_l, G_l)\}$ of pairs is called a **comprehensive Gröbner system (CGS) on \mathcal{S}** for $\langle F \rangle$ if $\sigma_\alpha(G_i)$ is a Gröbner basis of the ideal $\langle \sigma_\alpha(F) \rangle$ in $L[\mathbf{X}]$ for each $i = 1, \dots, l$ and $\alpha \in \mathcal{A}_i$. Each (\mathcal{A}_i, G_i) is called a **segment** of \mathcal{G} . We simply say \mathcal{G} is a comprehensive Gröbner system for $\langle F \rangle$ if $\mathcal{S} = L^m$.

For a segment (\mathcal{A}, G) , the algebraically constructible set \mathcal{A} is also called its parameter space or simply called its case. In this paper, we use an algebraically constructible set that has a form:

$$\mathbb{V}(f_1, \dots, f_k) \setminus \mathbb{V}(g_1, \dots, g_l) \subseteq L^m,$$

where $f_1, \dots, f_k, g_1, \dots, g_l \in K[\mathbf{A}]$. In this paper, the meaning of parameters is that parameters can take arbitrary values. We give an example of a CGS:

Example 1 Let $F = \{ax^2y + y, bx^2y^2 + ax + y\} \subset \mathbb{C}[a, b][x, y]$, a, b be parameters, x, y be variables, and $<_{\{x, y\}}$ be the lexicographic order such that $x <_{\{x, y\}} y$. Then a comprehensive Gröbner system for $\langle F \rangle$ with respect to $<_{\{x, y\}}$ is

$$\mathcal{G} = \left\{ \begin{array}{l} (\mathbb{C}^2 \setminus \mathbb{V}(ab), \{-b^2y^5 + 2aby^4 - a^2y^3 - a^3y, a^2x - by^2 + ay\}), \\ (\mathbb{V}(b) \setminus \mathbb{V}(a, b), \{ax + y, y^3 + ay\}), \\ (\mathbb{V}(a), \{y\}) \end{array} \right\}.$$

From Definition 1, one can understand \mathcal{G} as follows:

1. If parameters belong to $\mathbb{C}^2 \setminus \mathbb{V}(ab)$ (i.e., $ab \neq 0$), then $\{-b^2y^5 + 2aby^4 - a^2y^3 - a^3y, a^2x - by^2 + ay\}$ is a Gröbner basis of $\langle F \rangle$ with respect to $<_{\{x, y\}}$ in $\mathbb{C}[x, y]$.
2. If parameters belong to $\mathbb{V}(b) \setminus \mathbb{V}(a, b)$ (i.e., $b = 0, a \neq 0$), then $\{(by^3 - ay^2)x + ay, ayx^2 + y, -b^2y^5 + 2bay^4 - a^2y^3 - a^3y, a^2x - by^2 + a\}$ is a Gröbner basis.
3. If parameters belong to $\mathbb{V}(a)$ (i.e., $a = 0$), then $\{y\}$ is a Gröbner basis.

In general, the properties of parametric ideals are dependent on values of parameters. Therefore, CGS are a useful tool to analyze parametric polynomial systems.

3.1 Criteria

Recently, some criteria for computing CGS, have been introduced by Nabeshima and Kurata [11, 17]. In this subsection, we describe the criteria in order to make an efficient CGS implementation.

First, we introduce the definition of a set of **minimal head term**, based on [11]. Let F be a set of polynomials in $K[\mathbf{A}][\mathbf{X}]$. Then a set of minimal head term of F with respect to $<_{\mathbf{X}}$ is defined as follows:

$\text{MHT}_{<_{\mathbf{X}}}(F) :=$

$$\{\text{lpp}_{<_{\mathbf{X}}}(g) \in \text{pp}(\mathbf{X}) \mid \text{lpp}_{<_{\mathbf{X}}}(f_1) \nmid \text{lpp}_{<_{\mathbf{X}}}(f_2), \text{lpp}_{<_{\mathbf{X}}}(f_1) \neq \text{lpp}_{<_{\mathbf{X}}}(f_2), f_1, f_2 \in F\}.$$

Theorem 1 ([17]) *Let F be a subset of $K[\mathbf{A}][\mathbf{X}]$, S be a subset of $K[\mathbf{A}]$, and H be a Gröbner basis for $\langle S \cup F \rangle$ with respect to $<_{\mathbf{X}}$. Moreover, let $H_1 := H \cap \langle S \rangle$ and $G_1 := H \setminus H_1$. Select g from $\text{MHT}_{<_{\mathbf{X}}}(G_1)$, and set $r := 1 / \text{lc}_{<_{\mathbf{X}}}(g)$ (r is a new variable) and $g' := \text{lpp}_{<_{\mathbf{X}}}(g) + r \cdot (g - \text{lm}_{<_{\mathbf{X}}}(g))$. Suppose that $G_2 := (G_1 \setminus \{g\}) \cup \{g'\} \subseteq K[r, \mathbf{A}][\mathbf{X}]$, and G_3 is a Gröbner basis of $\langle G_2 \cup S \rangle$ with respect to $<_{\mathbf{X}}$ in $K[r, \mathbf{A}][\mathbf{X}]$, $G_4 := \{f \in K[\mathbf{A}][\mathbf{X}] : f \neq 0, f = \text{lc}_{<_{\mathbf{X}}}(g)^k \cdot \sigma_{r=1}(q), \deg_r(q) = k \in \mathbb{N}, q \in G_3\}$, and $\{h_1, \dots, h_e\} := \{\text{lc}_{<_{\mathbf{X}}}(f) \in K[\mathbf{A}] : f \in G_3 \setminus (G_3 \cap \langle S \rangle)\}$.*

Then for any $\alpha \in \mathbb{V}(S) \setminus (\mathbb{V}(\text{lc}_{<_{\mathbf{X}}}(g)) \cup \mathbb{V}(\text{LCM}(h_1, \dots, h_e)))$, $\sigma_{\alpha}(G_4)$ is a Gröbner basis for $\langle \sigma_{\alpha}(F) \rangle$ w.r.t. $<_{\mathbf{X}}$ in $L[\mathbf{X}]$ ($\sigma_{r=1}(q)$ means substituting 1 for the variable r of q . $\text{LCM}(h)$ is the least common multiple of h).

If we apply Theorem 1 directly, then we need an additional variable ‘ r .’ As the author noticed in [17], this approach may give rise to time-consuming Gröbner basis computation. In Theorem 1, however, if we select a monomial as g from $\text{MHT}_{<_{\mathbf{X}}}(G_1)$, then we do not need an additional variable r . We apply this selection for our algorithm for computing CGS.

Theorem 2 ([11]) *Let I be an ideal in $K[\mathbf{A}][\mathbf{X}]$, and let $G = \{g_1, \dots, g_s\}$ be a Gröbner basis of I with respect to $<_{\mathbf{X}}$. We assume that the g_i s are ordered in such a way that $g_1, \dots, g_r \notin K[\mathbf{A}]$ (for $1 \leq r \leq s$) and $g_{r+1}, \dots, g_s \in K[\mathbf{A}]$, and let $G' = \{g_1, \dots, g_r\}$. If a specialization σ satisfies that $\sigma(g_{r+1}) = \dots = \sigma(g_s) = 0$ and $\text{MHT}_{<_{\mathbf{X}}}(G) = \text{MHT}_{<_{\mathbf{X}}}(\sigma(G))$, then $\sigma(G')$ is a Gröbner basis for $\langle \sigma(I) \rangle$ with respect to $<_{\mathbf{X}}$ in $L[\mathbf{X}]$.*

In Theorem 2, for given Gröbner bases G and G' , we next consider what condition of σ satisfies. First, we give a notation for explaining a solution of the question. We write

$$g_i = a_{i,1} \bar{X}^{\alpha_{i,1}} + a_{i,2} \bar{X}^{\alpha_{i,2}} + \dots + a_{i,l_i} \bar{X}^{\alpha_{i,l_i}}, \quad (a_{i,j} \in K[\mathbf{A}], \bar{X}^{\alpha_{i,j}} \in \text{pp}(\mathbf{X}))$$

for $g_i \in G' = \{g_1, \dots, g_r\}$, and terms in g_i are ordered in descending with respect to $<_{\mathbf{X}}$, that is, $\bar{X}^{\alpha_{i,l_i}} <_{\mathbf{X}} \dots <_{\mathbf{X}} \bar{X}^{\alpha_{i,2}} <_{\mathbf{X}} \bar{X}^{\alpha_{i,1}}$. Note that $\text{lpp}_{<_{\mathbf{X}}}(g_i) = \bar{X}^{\alpha_{i,1}}$ and $\text{lc}_{<_{\mathbf{X}}}(g_i) = a_{i,1}$. Next, we assume that the polynomials in G' are ordered in such a way that $\text{MHT}_{<_{\mathbf{X}}}(G') = \{\text{lpp}_{<_{\mathbf{X}}}(g_1), \dots, \text{lpp}_{<_{\mathbf{X}}}(g_k)\}$ for $1 \leq k \leq r$. Thus, $\text{lpp}_{<_{\mathbf{X}}}(g_{k+1}), \dots, \text{lpp}_{<_{\mathbf{X}}}(g_r)$ are divisible by some power products in $\text{MHT}_{<_{\mathbf{X}}}(G')$. For $k+1 \leq i \leq r$ and power product in g_i , let $\bar{X}^{\alpha_{i,l_i+1}}$ be the highest term which is not divisible by some power products in $\text{MHT}_{<_{\mathbf{X}}}(G')$ if it exists. Thus, $\bar{X}^{\alpha_{i,1}}, \dots, \bar{X}^{\alpha_{i,l_i}}$ are divisible by some power products in $\text{MHT}_{<_{\mathbf{X}}}(G')$. Moreover, we also assume that the polynomials in $\{g_{k+1}, \dots, g_r\}$ are ordered in such a way that g_i has a power product which is not divisible by every polynomial in $\text{MHT}_{<_{\mathbf{X}}}(G')$ for $k+1 \leq i \leq r'$, and every power products in g_j is divisible by some power products in $\text{MHT}_{<_{\mathbf{X}}}(G')$ for $r' < j \leq r$.

Under the notation and ordering rule given above, we give a condition of σ .

Proposition 1 ([11]) *Let I be an ideal in $K[\mathbf{A}][\mathbf{X}]$, and $G = \{g_1, \dots, g_s\}$ a Gröbner basis of I with respect to $\prec_{\mathbf{x}}$. We assume that g_i s are ordered and written in the given rule $g_1, \dots, g_r \notin K[\mathbf{A}]$ and $g_{r+1}, \dots, g_s \in K[\mathbf{A}]$, and let $G' = \{g_1, \dots, g_r\}$. Then for all*

$$a \in \mathbb{V}(g_{r+1}, \dots, g_s) \setminus (\mathbb{V}(a_{1,1} \cdots a_{k,1}) \cup \mathbb{V}(a_{k+1,1}, \dots, a_{k+1, u_{k+1}}) \cup \cdots \cup \mathbb{V}(a_{r',1}, \dots, a_{r', u_{r'}})),$$

$\sigma_a(G')$ is a Gröbner basis of $\langle \sigma_a(I) \rangle$ with respect to $\prec_{\mathbf{x}}$ in $L[\mathbf{X}]$.

Remark: If we apply original Suzuki-Sato's criterion [25] to the set G' , then we obtain $\mathbb{V}(g_{r+1}, \dots, g_s) \setminus \mathbb{V}(a_{1,1} \cdots a_{k,1} \cdot a_{k+1,1} \cdots a_{r',1})$. Obviously, $\mathbb{V}(a_{1,1} \cdots a_{k,1} \cdot a_{k+1,1} \cdots a_{r',1}) = \mathbb{V}(a_{1,1} \cdots a_{k,1}) \cup \mathbb{V}(a_{k+1,1}) \cup \cdots \cup \mathbb{V}(a_{r', u_1})$. We know that $\mathbb{V}(a_{k+1,1}) \supset \mathbb{V}(a_{k+1,1}, \dots, a_{k+1, u_{k+1}})$, \dots , $\mathbb{V}(a_{r', u_{r'}}) \supset \mathbb{V}(a_{r',1}, \dots, a_{r', u_1})$. That is, the condition of Proposition 1 is stronger than Suzuki-Sato's one.

Proposition 1 leads to the following algorithm which outputs the condition of parameters $(\mathbb{V}(a_{1,1} \cdots a_{k,1}) \cup \mathbb{V}(a_{k+1,1}, \dots, a_{k+1, u_{k+1}}) \cup \cdots \cup \mathbb{V}(a_{r',1}, \dots, a_{r', u_{r'}}))$ and a set of polynomial G' .

Algorithm Kurata

Specification: Kurata(G, \prec)

Input: G : a Gröbner basis, \prec : a term order,

Output: L : a set of sets of polynomials in $K[\mathbf{A}]$, H : a set of polynomials in $K[\mathbf{A}]$,

G' : a set of polynomials in $K[\mathbf{A}][\mathbf{X}]$,

begin

$L \leftarrow \emptyset$; $MH \leftarrow \text{MHT}_{\prec}(G)$; $B \leftarrow G \setminus MH$; $G' \leftarrow MH$

while $B \neq \emptyset$ **do**

 Select g from B ; $B \leftarrow B \setminus \{g\}$; $D \leftarrow \emptyset$; $g' \leftarrow g$

while $g \neq 0$ **do**

if $\exists f \in MH$ s.t. $\text{lpp}_{\prec}(f) \parallel \text{lpp}_{\prec}(g)$ **then**

$D \leftarrow D \cup \text{lc}_{\prec}(g)$; $g \leftarrow g - \text{lm}_{\prec}(g)$

else

$A \leftarrow 1$; $G' \leftarrow G' \cup \{g'\}$

break-while

end-if

$A \leftarrow 0$

end-while

if $A = 1$ **then**

$L \leftarrow L \cup D$

end-if

end-while

$H \leftarrow \{\text{lc}_{\prec}(h) \mid \forall h \in MH\}$

return $[L, H, G']$

end

We apply Theorem 1 with the selecting monomial strategy and Proposition 1 to our algorithm for computing CGS. In the next subsection, we give some basic manipulations for an efficient CGS implementation.

3.2 Supports

In this subsection, we introduce a key tool of our algorithm for computing CGS, and some computational algorithms. An algorithm based on CGS often yields many superfluous segments, so that it gives rise to computational difficulties if we do not apply optimizations. In order to avoid these difficulties, we have to reduce useless segments, which appear during a computation, by treating their parameter spaces carefully.

In [17, 25], a parameter space was merely defined as $\mathbb{V}(T) \setminus \mathbb{V}(S)$ where T, S are sets of polynomials. Even if $\mathbb{V}(T) \setminus \mathbb{V}(S) = \emptyset$, their algorithms [17, 25] do not care this emptiness. Here we introduce '**supports**' for parameter spaces, because we can reduce useless segments and unnecessary computation by checking supports.

Definition 2 (Support) Let T and S be finite sets of polynomials in $K[\mathbf{A}]$ such that $\mathbb{V}(T) \supseteq \mathbb{V}(S)$. Then $\mathbb{V}(T) \setminus \mathbb{V}(S)$ is called a **support**. If $\mathbb{V}(S) = \mathbb{V}(1) = \emptyset$, then $\mathbb{V}(t)$ is also called a support.

During a CGS computation, we need to check whether a support $\mathbb{V}(T) \setminus \mathbb{V}(S)$ satisfies $\mathbb{V}(T) \setminus \mathbb{V}(S) = \emptyset$ or not. The next algorithm tells us how to check whether a support is an empty set or not.

Algorithm SupportIsZERO

Specification: SupportIsZERO(P)

Check whether a support P satisfies $P = \emptyset$ or not.

Input: P : a support,

Output: return 1 if $P = \emptyset$, return 0 otherwise.

begin

$\mathbb{V}(S) \setminus \mathbb{V}(T) \leftarrow P$

if $\mathbb{V}(S) = \mathbb{V}(T)$ **then**

return 1

else

return 0

end-if

In order to trim redundant branches during a CGS computation, we need three algorithms for supports. One algorithm is Intersection for computing the intersection of two supports as follows:

Let $p_1 = \mathbb{V}(t_1) \setminus \mathbb{V}(s_1)$ and $p_2 = \mathbb{V}(t_2) \setminus \mathbb{V}(s_2)$ be supports where t_1, t_2, s_1, s_2 are finite sets of $K[\mathbf{A}]$, and $\mathbb{V}(s_1), \mathbb{V}(s_2) \neq \mathbb{V}(1)$. Then we perform the following computation:

$$\begin{aligned} p_1 \cap p_2 &= (\mathbb{V}(t_1) \setminus \mathbb{V}(s_1)) \cap (\mathbb{V}(t_2) \setminus \mathbb{V}(s_2)) \\ &= (\mathbb{V}(t_1) \cap \mathbb{V}(t_2)) \setminus (\mathbb{V}(s_1) \cup \mathbb{V}(s_2)) \\ &= \mathbb{V}(t_1 \cup t_2) \setminus ((\mathbb{V}(s_1) \cap \mathbb{V}(t_2)) \cup (\mathbb{V}(s_2) \cap \mathbb{V}(t_1))) \\ &= \mathbb{V}(t_1 \cup t_2) \setminus \mathbb{V}(\mathbf{Prod}(s_1 \cup t_2, s_2 \cup t_1)), \end{aligned}$$

where $\mathbf{Prod}(s, t) := \{f \cdot g \mid f \in s, g \in t\}$. The last term $\mathbb{V}(t_1 \cup t_2) \setminus \mathbb{V}(\mathbf{Prod}(s_1 \cup t_2, s_2 \cup t_1))$ is also a support¹. In this paper, the next algorithm Intersection outputs the intersection of two supports, based on the computations above and SupportIsZERO.

Algorithm Intersection

Specification: Intersection(P, Q)

Compute $P \cap Q$.

Input: P, Q : supports,

Output: return $P \cap Q$ if it is not the empty, return 0 otherwise.

begin

$\mathbb{V}(t_1) \setminus \mathbb{V}(s_1) \leftarrow P$

$\mathbb{V}(t_2) \setminus \mathbb{V}(s_2) \leftarrow Q$

$A \leftarrow \text{SupportIsZERO}(\mathbb{V}(t_1 \cup t_2) \setminus \mathbb{V}(\mathbf{Prod}(s_1 \cup t_2, s_2 \cup t_1)))$

if $A = 1$ **then**

return 0

else

return $\mathbb{V}(t_1 \cup t_2) \setminus \mathbb{V}(\mathbf{Prod}(s_1 \cup t_2, s_2 \cup t_1))$

end-if

end

The second algorithm is the following:

During a CGS computation, we need to check whether a given parameter space is in a set of parameter spaces already computed. If so, we do not need to compute a CGS on the parameter space. We implement this procedure as follows:

¹ If $\mathbb{V}(s_1)$ or $\mathbb{V}(s_2)$ is $\mathbb{V}(1)$, then we need another procedure of the computation. It is not, however, difficult to implement it.

Algorithm Check_space**Specification:** Check_space(Q, SP)Check whether Q is in SP or not.**Input:** Q : a support, SP : a set of supports,**Output:** return 1 if Q is in SP , return 0 otherwise.**begin****while** $SP \neq \emptyset$ **then**Select P from SP ; $SP \leftarrow SP \setminus \{P\}$ $A \leftarrow \text{Intersection}(P, Q)$ **if** $A = Q$ **then**

return 1

end-if**end-while**

return 0

end

The third algorithm is the following:

If we compute a CGS with ‘equal ($= 0$)’ or ‘not-equal ($\neq 0$)’ conditions, we first transform this condition into a support for our CGS computation. We next compute a CGS on this support as follows:

Algorithm Make_supp**Specification:** Make_supp(T, S)Transform $\mathbb{V}(T) \setminus \mathbb{V}(S)$ into a support.**Input:** T, S : a set of polynomials,**Output:** return 0 if $\mathbb{V}(T) \setminus \mathbb{V}(S)$ is an empty, return $\mathbb{V}(T) \setminus \mathbb{V}(T \cup S)$, otherwise.**begin****if** SupportIsZERO($\mathbb{V}(T) \setminus \mathbb{V}(T \cup S)$)=1 **then**

return 0

elsereturn $\mathbb{V}(T) \setminus \mathbb{V}(T \cup S)$ **end**

We give a simple example of Make_supp. If ‘equal ($= 0$)’ and ‘not-equal ($\neq 0$)’ conditions are $\{a = 0, b(b - 2) \neq 0\}$, then the algorithm Make_supp outputs $\mathbb{V}(a) \setminus \mathbb{V}(a, b(b - 2))$ as a support.

All the algorithms in this subsection are essential for our CGS algorithm in the next subsection.

3.3 Main Algorithm

Here we give a new algorithm for computing CGS which is efficient for our problems, electric and neural circuits. We assume that the algorithm factorize(h) outputs a set of all irreducible factors of h in $K[\mathbf{A}]$, where $h \in K[\mathbf{A}]$.

Algorithm newCGS**Specification:** newCGS($F, T, S, <$)Compute a CGS on $\mathbb{V}(T) \setminus \mathbb{V}(S)$.**Input:** F : a set of polynomials in $K[\mathbf{A}][\mathbf{X}]$, $<$: a term order, T : a set of polynomials in $K[\mathbf{A}]$ (‘equal ($= 0$)’ conditions), S : a set of polynomials in $K[\mathbf{A}]$ (‘not-equal ($\neq 0$)’ conditions),**Output:** a CGS on $\mathbb{V}(T) \setminus \mathbb{V}(S)$ w.r.t $<$.**begin**

Consider that ‘SPACE’ is a global variable.

SPACE $\leftarrow \emptyset$ $\mathbb{V}(T) \setminus \mathbb{V}(S) \leftarrow \text{Make_supp}(T, S)$ return CGS_main($F, T, S, <$)**end**

Algorithm CGS_main**Specification:** CGS_main($F, T, S, <$)Compute a CGS on $\mathbb{V}(T) \setminus \mathbb{V}(S)$.**Input:** F : a set of polynomials in $K[\mathbf{A}][\mathbf{X}]$, $\mathbb{V}(T) \setminus \mathbb{V}(S)$: a support, $<$: a term order,**Output:** a CGS on $\mathbb{V}(T) \setminus \mathbb{V}(S)$.**begin**

```

1:  $G_1 \leftarrow$  Compute the reduced Gröbner basis for  $\langle F \cup T \rangle$  w.r.t.  $<_{\mathbf{A}, \mathbf{X}}$  in  $K[\mathbf{A}, \mathbf{X}]$ 
2:  $G_2 \leftarrow G_1 \setminus (G_1 \cap \langle T \rangle)$ ;
3:  $MH \leftarrow MHT_{<}(G_2)$ 
4:  $M_0 \leftarrow \{f \in MH \mid \exists g \text{ s.t. } \text{lpp}_{<}(f) \mid \text{lpp}_{<}(g), f \text{ is a monomial.}\}$ 
5: if  $M_0 \neq \emptyset$  then
6:   while  $M_0 \neq \emptyset$  do
7:     Select  $p$  from  $M_0$ 
8:      $\mathbb{V}(T_1) \setminus \mathbb{V}(S_1) \leftarrow \text{Intersection}(\mathbb{V}(T) \setminus \mathbb{V}(S), \mathbb{V}(0) \setminus \mathbb{V}(\text{lc}_{<}(p)))$ 
9:     if  $\text{SupportIsZERO}(\mathbb{V}(T_1) \setminus \mathbb{V}(S_1)) \neq 1$  then
10:       $G_2 \leftarrow (G_2 \setminus \{p\}) \cup \{\text{lpp}_{<}(p)\}$ ;  $U \leftarrow \text{factorize}(\text{lc}_{<}(p))$ 
11:       $ALL \leftarrow \text{CGS\_main}(G_2, T_1, S_1, <)$ 
12:      break-while
13:    end-if
14:     $M_0 \leftarrow M_0 \setminus \{p\}$ 
15:  end-while
16:  if  $M_0 \neq \emptyset$  then
17:    while  $U \neq \emptyset$  do
18:      Select  $u$  from  $U$ ;  $U \leftarrow U \setminus \{u\}$ 
19:       $\mathbb{V}(T_2) \setminus \mathbb{V}(S_2) \leftarrow \text{Intersection}(\mathbb{V}(T) \setminus \mathbb{V}(S), \mathbb{V}(u) \setminus \mathbb{V}(1))$ 
20:      if  $\text{SupportIsZERO}(\mathbb{V}(T_2) \setminus \mathbb{V}(S_2)) \neq 1$  then
21:         $ALL \leftarrow ALL \cup \text{CGS\_main}(F, T_2, S_2, <)$ 
22:      end-if
23:    end-while
24:  end-if
25: return  $ALL$ 
26: else
27:  $[L, H, G'] \leftarrow \text{Kurata}(G_2, <)$ 
28:  $[A_1, A_2] \leftarrow \text{make\_segment}(L, H, G')$ 
29: if  $A_1 \neq 1$  then
30:    $SPACE \leftarrow SPACE \cup \{A_1\}$ 
31:    $ALL \leftarrow \{(A_1, A_2)\}$ 
32: else
33:    $ALL \leftarrow \emptyset$ 
34: end-if
35:  $CU \leftarrow H \cup L$ 
36: while  $CU \neq \emptyset$  do
37:   Select  $C_1$  from  $CU$ ;  $CU \leftarrow CU \setminus \{C_1\}$ 
38:    $\mathbb{V}(T_2) \setminus \mathbb{V}(S_2) \leftarrow \text{Intersection}(\mathbb{V}(T) \setminus \mathbb{V}(S), \mathbb{V}(C_1) \setminus \mathbb{V}(1))$ 
39:   if  $\text{SupportIsZERO}(\mathbb{V}(T_2) \setminus \mathbb{V}(S_2)) = 0$  then
40:     if  $\text{Check\_space}(\mathbb{V}(T_2) \setminus \mathbb{V}(S_2), SPACE) = 0$  then
41:        $ALL \leftarrow ALL \cup \text{CGS\_main}(F, T_2, S_2, <)$ 
42:     end-if
43:   end-if
44: end-while
45: return  $ALL$ 
46: end-if
end

```

Subalgorithm Make_segment**Specification:** Make_segement($L, H, G', \mathbb{V}(T) \setminus \mathbb{V}(S)$)

Create a segment .

Input: L : a set of sets of polynomials in $K[\mathbf{A}]$, H : a set of polynomials,
 G' : a set of polynomials in $K[\mathbf{A}][\mathbf{X}]$,**Output:** a segment (TS : a support, G' : a set of polynomials).**begin** $\{h_1, \dots, h_l\} \leftarrow \{lc_{<}(h) \mid \forall h \in H\}$ TS \leftarrow Intersection($\mathbb{V}(T) \setminus \mathbb{V}(S), \mathbb{V}(0) \setminus \mathbb{V}(h_1 \cdots h_l)$)**if** SupportIsZERO(TS) = 0 **then** $E \leftarrow 0$ **while** $L \neq \emptyset$ **do**Select L_1 from L ; $L \leftarrow L \setminus \{L_1\}$ TS \leftarrow Intersection(TS, $\mathbb{V}(0) \setminus \mathbb{V}(L_1)$)**if** SupportIsZERO(TS) = 1 **then** $E \leftarrow 1$ **break-while****end-if****end-while****if** $E = 0$ **then**return [TS, G']**else**return [1, \emptyset]**end-if****else**return [1, \emptyset]**end-if****end**

The algorithm newCGS always terminates, and outputs a CGS of $\langle F \rangle$ on $\mathbb{V}(T) \setminus \mathbb{V}(S)$. Since these proofs are essentially the same as the proofs of [17, 25], we do not give the proofs (See [17, 25]).

A big advantage of newCGS is that we can input ‘not-equal ($\neq 0$)’ and ‘equal ($= 0$)’ conditions. As these conditions trim redundant branches in lines 8, 19, 40 during CGS computation, the algorithm runs efficiently.

If there exist many parameters, then this approach is useful to obtain CGS. In fact, this approach is suitable for our problems, electric and neural circuits. If we input $\{0\}$ as T , and $\{1\}$ as S (i.e., $\mathbb{V}(0) \setminus \mathbb{V}(1) = L^m$), then we are able to obtain a normal CGS on the whole parameter space, too.

The algorithm newCGS has been implemented on the computer algebra system, Risa/Asir [20]. We give a simple example how our implementation outputs. Let $\{ax^2y + bx + 3, (b - 2)xy + bx + 5\}$ be a set of polynomials in $\mathbb{C}[x, y]$, x, y be variable, and a, b be parameters. The term order is the lexicographic such that $y <_{\{x, y\}} x$. If we assume ‘not-equal’ conditions: $a \neq 0 \wedge b \neq 0$ (i.e., $ab \neq 0$), then our implementation outputs the following \mathcal{G}_1 as a CGS on $\mathbb{C}^2 \setminus \mathbb{V}(ab)$.

$$\mathcal{G}_1 = \left\{ \begin{array}{l} (\mathbb{V}(b - 2) \setminus \mathbb{V}(a, b - 2), \{25ay - 8, 2x + 5\}), \\ (\mathbb{C}^2 \setminus \mathbb{V}((b^2 - 2b)a), \{-5ayx + (3b - 6)y - 2b, (3b^2 - 12b + 12)y^2 + (25a + \\ b^2 - 2b)y - 2b^2, 5bax + (3b^2 - 12b + 12)y + 25a - 2b^2 + 4b\}) \end{array} \right\}.$$

\mathcal{G}_1 has two segments. Our implementation also outputs a CGS on the whole parameter space (i.e., \mathbb{C}^3) as follows:

$$\left\{ \begin{array}{l} (\mathbb{V}(b) \setminus \mathbb{V}(a, b), \{12y + 25a, 5ax + 6\}), \\ (\mathbb{V}(b - 2) \setminus \mathbb{V}(a, b - 2), \{25ay - 8, 2x + 5\}), \\ (\mathbb{V}(a, b), \{1\}), \\ (\mathbb{V}(a, b - 2), \{1\}), \\ (\mathbb{V}(a) \setminus \mathbb{V}(b^2 - 2b, a), \{(3b - 6)y - 2b, bx + 3\}), \\ (\mathbb{C}^2 \setminus \mathbb{V}((b^2 - 2b)a), \{-5ayx + (3b - 6)y - 2b, (3b^2 - 12b + 12)y^2 + (25a + \\ b^2 - 2b)y - 2b^2, 5bax + (3b^2 - 12b + 12)y + 25a - 2b^2 + 4b\}) \end{array} \right\}.$$

Problem	Algorithm	CA system	No. of segments	CPU time (sec.)
S_1	newCGS	Risa/Asir	6	25.17
	Suzuki-Sato	Risa/Asir	36	343.2
	Montes	Maple 9.5	–	> 30 min
	NabeCGS	Risa/Asir	–	> 30 min
	NabeCGS[1]	Risa/Asir	36	746.4
	NabeCGS[2]	Risa/Asir	36	785.7
S_2	newCGS	Risa/Asir	7	0.203
	Suzuki-Sato	Risa/Asir	16	0.281
	Montes	Maple 9.5	14	12.125
	NabeCGS	Risa/Asir	16	1.687
	NabeCGS[1]	Risa/Asir	16	0.328
	NabeCGS[2]	Risa/Asir	16	0.36
S_3	newCGS	Risa/Asir	12	2.297
	Suzuki-Sato	Risa/Asir	25	7.047
	Montes	Maple 9.5	–	> 30 min
	NabeCGS	Risa/Asir	–	> 30 min
	NabeCGS[1]	Risa/Asir	27	7.968
	NabeCGS[2]	Risa/Asir	42	12.61
S_4	newCGS	Risa/Asir	27	33.2
	Suzuki-Sato	Risa/Asir	55	52.39
	Montes	Maple 9.5	–	> 30 min
	NabeCGS	Risa/Asir	–	> 30 min
	NabeCGS[1]	Risa/Asir	–	> 30 min
	NabeCGS[2]	Risa/Asir	–	> 30 min
S_5	newCGS	Risa/Asir	16	27.44
	Suzuki-Sato	Risa/Asir	–	> 30 min
	Montes	Maple 9.5	–	> 30 min
	NabeCGS	Risa/Asir	–	> 30 min
	NabeCGS[1]	Risa/Asir	38	37.88
	NabeCGS[2]	Risa/Asir	–	> 30 min

Table 1 Comparisons of our implementation with other ones. ‘newCGS’ is our implementation in this paper.

This output has six segments.

3.4 Comparisons

In this subsection, we compare our implementations newCGS with other implementations: the Maple-implementation² by Montes [14], the Risa/Asir-implementation based on pure Suzuki-Sato algorithm [25] by Nabeshima, and the Risa/Asir-implementation³ based on [17] by Nabeshima. Five problems S_1, S_2, S_3, S_4 , and S_5 are the followings. All measures are taken on a PC [CPU: Pentium 1.73 GHZ, Memory 2GB RAM, OS: Windows XP]. Note that NabeCGS has a normal strategy presented in [17] and that NabeCGS[s] ($s \in \mathbb{N}$) means the selection strategy presented in [17]. Let $X, Y, Z, S, X_1, X_2, Y_1, Y_2$ be variables, and a, b, c be parameters. The term order is fixed at the lexicographic such that $S < Z < Y < X$ or $S < Y_2 < Y_1 < X_2 < X_1$.

$$S_1: \{X^5 - a, Y^6 - b, X + Y - Z\}$$

$$S_2: \{P, Q, (X_1 - X_2)^2 + (Y_1 - Y_2)^2 - S, \frac{\partial P}{\partial X_1} \frac{\partial Q}{\partial Y_2} - \frac{\partial P}{\partial Y_1} \frac{\partial Q}{\partial X_2}, \frac{\partial P}{\partial X_1} (Y_1 - Y_2) - \frac{\partial P}{\partial Y_1} (X_1 - X_2)\}$$
 with $P = aX_1^2 + bY_1$ and $Q = cY_2^2 + dX_2$.

$$S_3: \text{The same polynomial set as } S_2 \text{ with } P = X_1^2 + Y_1^2 + a \text{ and } Q = Y_2^2 - bX_2^2 + c.$$

$$S_4: \{P - Z, X^2 + Y^2 + Z^2 - S, X + \frac{\partial P}{\partial X} Z, Y + \frac{\partial P}{\partial Y} Z\}$$
 with $P = (X - a)^2 + bY^2 + b$.

$$S_5: \text{The same polynomial set as } S_4 \text{ with } P = (X - a)^2 + bY^2 + a^2 - b.$$

Table 1 shows that our implementation is more efficient than other ones for computing CGS. Specially, our implementation outputted small numbers of segments, because of trimming redundant branches.

² <http://www-ma2.upc.edu/~montes/>

³ <http://www.math.sci.osaka-u.ac.jp/~nabeshima/PGB/>

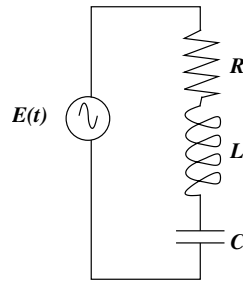


Fig. 1 An example of electric circuits. R , L , and C denote a resistor, inductor, and capacitor, respectively. $E(t)$ is an alternating voltage source.

4 Solving problems

We have to convert Formula (1) in §2 into a set of polynomials to which we can apply CGS. For this purpose, first, we substitute $V_i(t) = v_{i1} \sin(\omega t) + v_{i2} \cos(\omega t)$ into Formula (1). Next, we calculate a set of coefficients w.r.t. $\sin(\omega t)$ and $\cos(\omega t)$. This is a set of polynomials over $\mathbb{Q}[\mathbf{p}, v_{i1}, v_{i2}]$ ($i \geq 1$). Last, in the next sub-section, we show CGS where ‘equal’ and ‘not-equal’ conditions are considered as mentioned in §1.

In this section, we used a workstation with Intel[®] Xeon[®] W5590 CPU 3.33 GHz processor.

4.1 Example 1: Basic electric circuit

An electric circuit is illustrated in Fig. 1. The dynamics of this circuit can be described as the following equation:

$$RI(t) + L \frac{dI(t)}{dt} + \frac{1}{C} \int I(t) dt = e \sin(\omega t), \quad (2)$$

where $I(t)$ denotes a current, and $E(t) = e \sin(\omega t)$ denotes a sine-wave alternating voltage source. As above-mentioned, to convert Formula (2) into a set of polynomials, we substituted $I(t)$ with $x \sin(\omega t) + y \cos(\omega t)$. Then we obtained a set of coefficients w.r.t. $\sin(\omega t)$ and $\cos(\omega t)$:

$$\{-x + RC\omega y + L\omega^2 Cx, -L\omega^2 Cy + RC\omega x + y - eC\omega\} \quad (3)$$

In addition to the above set, we considered also $x^2 + y^2 - a_m^2$, where a_m corresponds to the amplitude of $I(t)$ based on [21]. Therefore, for this electric circuit, we had to compute a CGS of the following set of polynomials,

$$\{-x + RC\omega y + L\omega^2 Cx, -L\omega^2 Cy + RC\omega x + y - eC\omega, x^2 + y^2 - a_m^2\}. \quad (4)$$

We applied our implemented CGS to the set (4) with parameters: $\{e, C, R, L, \omega\}$ and variables: $\{x, y, a_m\}$ on the computer algebra system, Risa/Asir (Ver. 20090215), which yielded the result as shown in Table 2 (a). With ‘not-equal’ conditions of e, C, R, L , and ω , this calculation took around 0.6 sec.

To compare with our implemented CGS, we also applied the same set to a CGS which was implemented⁴ by Suzuki and Sato [25]. Over MAPLE 11.02 and with parameters: $\{e, C, R, L, \omega\}$ and variables: $\{x, y, a_m\}$, it yielded eight segments as shown in Table 2 (b). This calculation took around 0.06 sec. Considering the ‘not-equal’ conditions, only case 4) $-1 + L\omega^2 C = 0, (Re) \neq 0$ in Table 2 (b) is reasonable, which is consistent with case 3) $\nabla(LC\omega^2 - 1) \setminus \nabla(LC\omega^2 - 1, eR)$ in Table 2 (a).

4.2 Example 2: Neural Circuits

Another promising application of our implemented CGS is analysis of neural circuits. In fact, neural circuits can be described as an electric circuits [4]. Here we analyze a branching and ladder-like neural circuits [7] with an input impulse as illustrated in Fig. 2.

⁴ <http://kurt.scitec.kobe-u.ac.jp/~sakira/CGBusingGB/>

(a) Segment	Gröbner basis
1) $\mathbb{V}(0) \setminus \mathbb{V}(L^4 e C^3 R \omega^7 + L^2 e C^3 R^3 \omega^5 - 3L^3 e C^2 R \omega^5 \dots)$	$\{(-L^2 C^2 \omega^4 + \dots, \dots)\}$
2) $\mathbb{V}(L^2 C^2 \omega^4 + (C^2 R^2 - 2LC)\omega^2 + 1) \setminus \mathbb{V}(LC\omega^2 - 1, R, \dots)$	$\{1\}$
3) $\mathbb{V}(LC\omega^2 - 1) \setminus \mathbb{V}(LC\omega^2 - 1, eR)$	$\{y, -Rx + e, ex - a_m^2 R\}$

(b) Segment	Gröbner basis
1) $-1 + L\omega^2 C = 0, e = 0, R \neq 0$	$\{x^2 + y^2 - a_m^2, a_m^2 R, xR, yR\}$
2) $e = 0, R = 0, (-1 + L\omega^2 C) \neq 0$	$\{x^2 + y^2 - a_m^2, -x + L\omega^2 Cx, L\omega^2 Cy - y, a_m^2 CL\omega^2 - a_m^2\}$
3) $R = 0, (e(-1 + L\omega^2 C)) \neq 0$	$\{a_m^2 CL\omega^2 + yeC\omega - a_m^2, L\omega^2 Cy + eC\omega - y, x^2 + y^2 - a_m^2, xe, -x + L\omega^2 Cx, y^2 e - a_m^2 e\}$
4) $-1 + L\omega^2 C = 0, (Re) \neq 0$	$\{x^2 + y^2 - a_m^2, xR - e, yR, a_m^2 R - xe, ye\}$
5) $C = 0$	$\{a_m^2, x, y\}$
6) $-1 + L\omega^2 C = 0, e = 0, R = 0$	$\{x^2 + y^2 - a_m^2\}$
7) $\omega = 0$	$\{a_m^2, x, y\}$
8) $(RC\omega(-1 + L\omega^2 C)) \neq 0$	$\{-x + RC\omega y + L\omega^2 Cx, a_m^2 CL\omega^2 + yeC\omega - a_m^2, x^2 + y^2 - a_m^2, a_m^2 R - xe, L\omega^2 Cy - RC\omega x + eC\omega - y\}$

Table 2 Output of CGS. (a) By our implemented method. (b) By Suzuki-Sato algorithm [25].

4.2.1 Branching Neural Circuit

The system of differential equations that describes the branching neural circuit is the following:

$$\begin{cases} c_m dV_1(t)/dt = -g_l V_1(t) + e \sin(\omega t) + g_{12}(V_2(t) - V_1(t)) + g_{13}(V_3(t) - V_1(t)), \\ c_m dV_2(t)/dt = -g_l V_2(t) + g_{21}(V_1(t) - V_2(t)), \\ c_m dV_3(t)/dt = -g_l V_3(t) + g_{31}(V_1(t) - V_3(t)), \end{cases} \quad (5)$$

where c_m and $V_i(t)$ denote membrane capacitance and potential of neuron i (More exactly, one has to say ‘compartment i ’ because in this example, we use multi-compartment model [4, §6.4].) The parameter g_{ij} designates the resistive coupling from i to j . $g_l V_i(t)$ is a simple form of the membrane current of neuron i , and $e \sin(\omega t)$ is an input impulse into neuron 1. To convert Formula (5) into a set of polynomials, we substituted $V_i(t)$ with $x_i \sin(\omega t) + y_i \cos(\omega t)$, and obtained a set of coefficients w.r.t. $\sin(\omega t)$ and $\cos(\omega t)$:

$$\begin{aligned} & \{ c_m x_3 \omega - g_{31} y_1 + g_{12} y_3 + g_{31} y_3, -g_l x_3 + g_{31} x_1 + c y_3 \omega - g_{31} x_3, c_m x_2 \omega - g_{21} y_1 + g_l y_2 \\ & + g_{21} y_2, g_l x_2 - g_{21} x_1 - c_m y_2 \omega + g_{21} x_2, c_m x_1 \omega - g_{12} y_2 + g_l y_1 + g_{12} y_1 - g_{13} y_3 + g_{13} y_1, \\ & -c_m y_1 \omega - g_{12} x_2 + g_{12} x_1 - g_{13} x_3 + g_l x_1 + g_{13} x_1 - e \}. \end{aligned} \quad (6)$$

As mentioned in the electric circuit, in addition to the above set, we consider also $(x_2 + x_3)^2 + (y_2 + y_3)^2 - 4a_m^2$, where a_m represents the averaged amplitude of sum of neurons 2 and 3.

We applied our CGS to the set (6) $\cup \{(x_2 + x_3)^2 + (y_2 + y_3)^2 - 4a_m^2\}$ with eight parameters: $\{w, e, c, g_{12}, g_{13}, g_l, g_{21}, g_{31}\}$, seven variables: $\{a_m, y_3, y_2, y_1, x_3, x_2, x_1\}$, and ‘not-equal’ conditions: $\omega \neq 0 \wedge e \neq 0 \wedge c_m \neq 0 \wedge g_{12} \neq 0 \wedge g_{13} \neq 0 \wedge g_l \neq 0 \wedge g_{21} \neq 0 \wedge g_{31} \neq 0$. This calculation took around 1000 sec. The output of the CGS had 14 segments, one of which we found biologically-meaningful. This parameter space was $\mathbb{V}(g_{31} - g_{21}) \setminus \mathbb{V}(\dots)$, which means that some characteristic phenomenon occurs when the resistive coupling of neuron 2 to 1 is equal to that of 3 to 1. The CGS by Suzuki-Sato consumed 8GB memory and halted without an output in three days.

4.2.2 Ladder-like Neural Circuit

In the same manner as the branching neural circuits, we obtained a set of polynomials with respect to the ladder-like neural circuit:

$$\begin{aligned} & \{ c_m x_1 \omega + g_l y_1 - g_{12} y_2 + g_{12} y_1 - g_{13} y_3 + g_{13} y_1, e + c_m y_1 \omega - g_l x_1 - g_{12} x_1 \\ & + g_{12} x_2 - g_{13} x_1 + g_{13} x_3, -g_{42} x_2 - c_m y_4 \omega + g_l x_4 - g_{43} x_3 + g_{42} x_4 + g_{43} x_4, g_l y_4 \\ & - g_{42} y_2 + g_{42} y_4 + c_m x_4 \omega - g_{43} y_3 + g_{43} y_4, g_{21} x_1 + c_m y_2 \omega - g_l x_2 + g_{24} x_4 - g_{21} x_2 \\ & - g_{24} x_2, -g_{31} x_3 + c_m y_3 \omega - g_l x_3 - g_{34} x_3 + g_{31} x_1 + g_{34} x_4, -g_l y_3 - g_{31} y_3 + g_{31} y_1 \\ & - c_m x_3 \omega - g_{34} y_3 + g_{34} y_4, g_l y_2 - g_{21} y_1 + g_{21} y_2 + c_m x_2 \omega - g_{24} y_4 + g_{24} y_2, x_4^2 + y_4^2 - a_m^2 \}, \end{aligned} \quad (7)$$

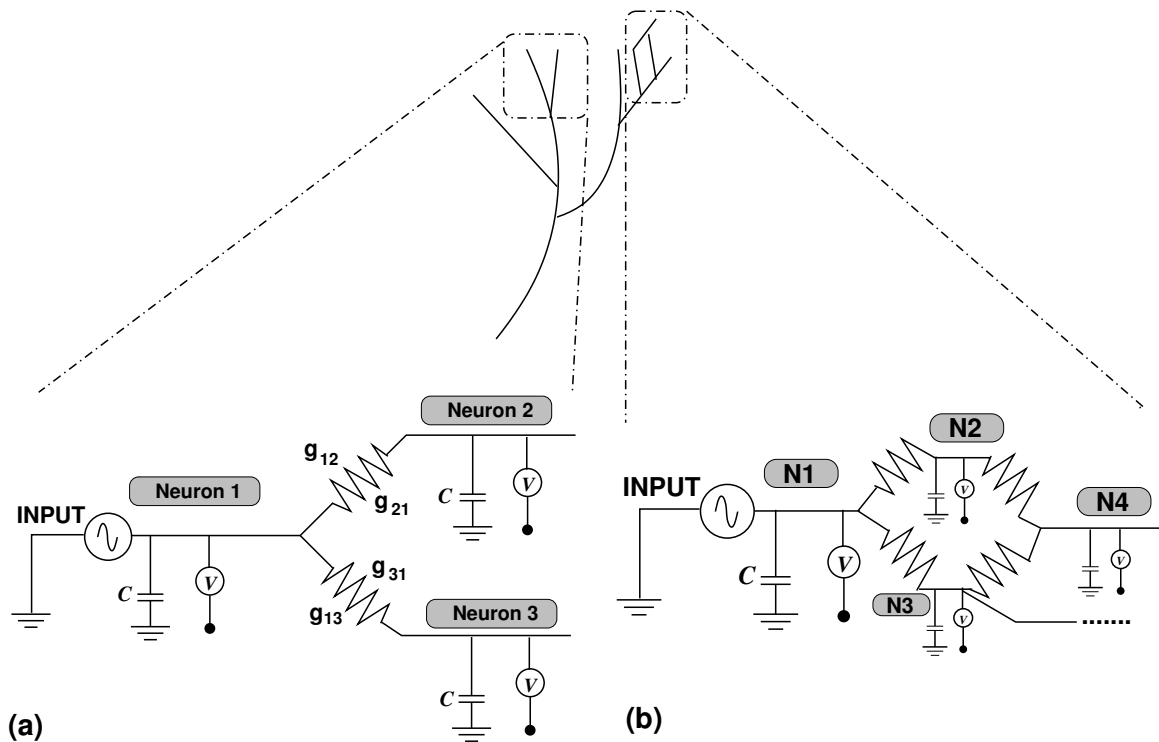


Fig. 2 Two example of neural circuits. (a) A branching example which was originally introduced in [4, Fig. 6.16]. C corresponds to a capacitor. g_{ij} corresponds to resistance from neuron i to j , where g_{ij} is not necessarily equal to g_{ji} due to compartmental interpretation of neurons. (b) A ladder-like example, which is based on ‘ladder-like patterns’ of distribution of sympathetic preganglionic neurons [7].

where, in the last element, $x_4^2 + y_4^2 - a_m^2$, a_m represents the amplitude of neuron 4.

We applied our CGS to the set (7) with 12 parameters: $\{\omega, e, c_m, g_l, g_{12}, g_{13}, g_{42}, g_{43}, g_{21}, g_{24}, g_{31}, g_{34}\}$ and 9 variables: $\{a_m, y_4, y_3, y_2, y_1, x_4, x_3, x_2, x_1\}$. In this circuit, for simplicity, we set ‘equal’ conditions: $g_{31} = g_{13} = g_{21} = g_{12} \wedge g_{34} = g_{43} = g_{24} = g_{42}$ together with ‘not-equal’ conditions: $\omega \neq 0 \wedge c_m \neq 0 \wedge g_l \neq 0 \wedge g_{12} \neq 0 \wedge g_{24} \neq 0$ during CGS calculation. It took around 845 min and output 8 segments whose parameter spaces were of the form: $\mathbb{V}(f_i) \setminus \mathbb{V}(\dots)$ ($i = 1, 2, \dots, 8$). f_1, f_2, \dots, f_8 are (1) 0, (2) $c_m^4 \omega^4 + (2c_m^2 g_l^2 + (6g_{31} + 6g_{34})c_m^2 g_l + (9g_{31}^2 + 2g_{34}g_{31} + 9g_{34}^2)c_m^2) \omega^2 + g_l^4 + (6g_{31} + 6g_{34})g_l^3 + (9g_{31}^2 + 34g_{34}g_{31} + 9g_{34}^2)g_l^2 + (48g_{34}g_{31}^2 + 48g_{34}^2 g_{31})g_l + 64g_{34}^2 g_{31}^2$, (3) $c_m^2 \omega^2 + g_l^2 + (2g_{31} + 2g_{34})g_l + g_{31}^2 + 2g_{34}g_{31} + g_{34}^2$, (4) $c_m^2 \omega^2 + g_l^2$, (5) $2g_l^2 + (3g_{31} + 3g_{34})g_l + 4g_{34}g_{31}$, (6) $2g_l + 3g_{31} + 3g_{34}$, (7) $g_l + g_{31} + g_{34}$, and (8) $g_l - g_{31} + 3g_{34}$.

The derivation of the sets (4), (6), (7) is given in <http://sites.google.com/site/neuropoke/>.

5 Discussion

In the basic electric circuit, through our CGS, we have found a resonant condition, $\omega = 1/\sqrt{LC}$ indicated by segment 3) $\mathbb{V}(LC\omega^2 - 1) \setminus \mathbb{V}(LC\omega^2 - 1, eR)$ in Table 2 (a). Further, in this segment, a Gröbner basis in terms of lexicographical term order, $a_m < y < x$ is calculated as $\{a_m^2 R^2 - e^2, y, -Rx + e\}$, showing that the amplitude of $I(t)$, a_m becomes e/R under the resonant condition.

In the branching neural circuit, we obtained one acceptable segment, $g_{31} - g_{21}$ under a biological assumption, $\omega > 0 \wedge e > 0 \wedge c_m > 0 \wedge g_{12} > 0 \wedge g_{13} > 0 \wedge g_l > 0 \wedge g_{21} > 0 \wedge g_{31} > 0$. It should be noted that $g_{31} - g_{21}$ is, of course, a simple term, but it is difficult to confirm by hand that the only term exists as a biologically-meaningful segment against 8 parameters. In this segment, we calculated a Gröbner basis in terms of lexicographical term order, $a_m < x_1 < x_2 < x_3 < y_1 < y_2 < y_3$, revealing that each of the variables: $y_3, y_2, y_1, x_3, x_2, x_1, a_m$ can be described as a function of the parameters: $\{\omega, c_m, e, g_{31}, g_{12}, g_{13}, g_l\}$. For example, a can explicitly be described as follows: $a_m = eg_{31}/\sqrt{(g_l^2 + c_m^2 \omega^2)F}$

with $F = g_{12}^2 + g_{13}^2 + g_{31}^2 + 2g_{31}g_l + g_l^2 + 2g_{13}(g_{31} + g_l) + 2g_{12}(g_{13} + g_{31} + g_l) + c_m^2\omega^2$. Thus, through our CGS, we have succeeded in deriving a relation of parameters, where the system has some characteristic. It was easy to find out these relations because only 14 segments exist as an output of CGS. The number of segments depends on the number of parameters under ‘not-equal’ conditions. Indeed, there exist 14, 28, 41, and 45 segments when 8, 4, 1, and 0 parameters are under ‘not-equal’ conditions, respectively, showing that many unnecessary segments will appear without a biological assumption, *i.e.*, a positive-value condition.

Last, in the ladder-like neural circuit, we used the ‘equal’ conditions as well as the ‘not-equal’ conditions. We obtained 8 segments, only the last segment ((8): $g_l - g_{31} + 3g_{34}$) of which is acceptable under a biological assumption $\omega > 0 \wedge e > 0 \wedge c_m > 0 \wedge g_l > 0 \wedge g_{31} > 0 \wedge g_{34} > 0$. These facts tell us a possibility that a ‘positive-value’ assumption during CGS calculation will reduce both of its calculation time and the number of segments. ‘Positive comprehensive Gröbner system’ is, therefore, a promising future work⁵.

6 Summary

Here we have implemented a CGS to derive specific conditions with the ‘equal’ and ‘not-equal’ conditions considered. These conditions decrease the number of segments as an output of CGS, which enables us to find out a particular relation of parameters at ease. A close view of the obtained parameter spaces provided us with a future work, ‘positive CGS (pCGS).’

Acknowledgements Author H.Y. was supported by a Grant-in-Aid for Young Scientists (B) (grant 21740077) from the Ministry of Education, Culture, Sports, Science and Technology of Japan (MEXT). This study was also supported in part by the Program for Improvement of Research Environment for Young Researchers from Special Coordination Funds for Promoting Science and Technology (SCF) commissioned by the Japan Science and Technology Agency (JST).

References

1. Boylestad, R., Nashelsky, L.: Electronic devices and circuit theory. 10th edn. Prentice Hall, Upper Saddle River (2008)
2. Brickensteina, M., Dreyer, A., Greuel, G.M., Wedler, M., Wienand, O.: New developments in the theory of Gröbner bases and applications to formal verification. *J. Pure Appl. Algebra* **213**(8), 1612–1635 (2009). DOI 10.1016/j.jpaa.2008.11.043
3. Cobelli, C., Foster, D., Toffolo, G.: Tracer Kinetics in Biomedical research: From data to model. Kluwer Academic/Plenum Publishers, New York (2000)
4. Dayan, P., Abbott, L.F.: Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems. The MIT Press, Cambridge, Massachusetts (2001)
5. Dolzmann, A., Sturm, T.: Redlog: Computer algebra meets computer logic. *ACM SIGSAM Bulletin* **31**(2), 2–9 (1997)
6. Gao, X., Chou, S.: Solving Parametric Algebraic Systems. In: International Symposium on Symbolic and Algebraic Computation, pp. 335–341. ACM Press (1992)
7. Hosoya, Y., Okado, N., Sugiura, Y., Kohno, K.: Coincidence of “ladder-like patterns” in distributions of monoaminergic terminals and sympathetic preganglionic neurons in the rat spinal cord. *Exp. Brain Res.* **86**, 224–228 (1991)
8. Kanno, M., Anai, H., Yokoyama, K., Hara, S.: Parametric Optimization in Control Using the Sum of Roots for Parametric Polynomial Spectral Factorization. In: International Symposium on Symbolic and Algebraic Computation, pp. 211–218. ACM Press (2007)
9. Kapur, D.: An Approach for Solving Systems of Parametric Polynomial Equations. In: Saraswat, Hentenryck (eds.) Principles and Practice of Constraint Programming, pp. 217–244. MIT press (1993)
10. Kuramoto, K.: Chemical Oscillations, Waves, and Turbulence. Springer-Verlag, Heidelberg (1984)
11. Kurata, Y.: Improving Suzuki-Sato’s CGS Algorithm by Using Stability of Gröbner Bases. In: presentation at the workshop CA-DAIA (2009). Personal communication
12. Landau, L.D., Lifshitz, E.: Mechanics, *Course of Theoretical Physics*, vol. 1. 3rd edn. Pergamon Press, Oxford (1976)

⁵ ‘Positive quantifier elimination (QE)’ has recently been studied [23].

13. Lazard, D., Rouillier, F.: Solving parametric polynomial systems. *J. Symbolic Comput.* **43**(3), 636–667 (2007)
14. Manubens, M., Montes, A.: Improving DISPGB algorithm using the discriminant ideal. *J. Symbolic Comput.* **41**, 1245–1263 (2006)
15. Montes, A.: A new algorithm for discussing Gröbner basis with parameters. *J. Symbolic Comput.* **33**(1-2), 183–208 (2002)
16. Montes, A., Recio, T.: Automatic discovery of geometry theorems using minimal canonical comprehensive Gröbner system. In: ADG 2006, *Lecture Notes in Artificial Intelligence* **4869**, pp. 113–138. Springer (2007)
17. Nabeshima, K.: A Speed-Up of the Algorithm for Computing Comprehensive Gröbner Systems. In: *International Symposium on Symbolic and Algebraic Computation*, pp. 299–306. AMC-Press (2007)
18. Nabeshima, K.: PGB: A package for computing parametric polynomial systems. In: the Joint Conference of ASCM2009 and MACIS2009, *MI (Math-for-Industry) Lecture Note* **Vol. 22**, pp. 111–122. Kyushu University (2009)
19. Nabeshima, K.: On the Computation of Parametric Gröbner Bases for Modules and Syzygies. *Japan Journal of Industrial and Applied Mathematics* (2010). In press
20. Noro, M., Takeshima, T.: Risa/Asir- A Computer Algebra System. In: Wang, P. (ed.) *International Symposium on Symbolic and Algebraic Computation*, pp. 387–396. AMC-Press (1992)
21. Okumura, K.: Application of Gröbner bases to nonlinear circuit analysis [in Japanese]. *Systems, control and information* **48**(2), 45–50 (2004)
22. Shinohara, N.: Parametric Polynomial Spectral factorization [in Japanese]. *Bulletin of the Japan Society for Symbolic and Algebraic Computation* **16**(1), 3–19 (2009)
23. Sturm, T., Weber, A.: Investigating genetic methods to solve Hopf bifurcation problems in Algebraic Biology. In: Horimoto, K., Regensburger, G., Rosenkranz, M., Yoshida, H. (eds.) *Algebraic Biology, Lecture Notes in Computer Science*, vol. 5147, pp. 200–215. Springer-Verlag, Heidelberg (2008)
24. Suzuki, A., Sato, Y.: An alternative approach to Comprehensive Gröbner bases. *J. Symbolic Comput.* **36**(3-4), 649–667 (2003)
25. Suzuki, A., Sato, Y.: A Simple Algorithm to compute Comprehensive Gröbner Bases using Gröbner bases. In: *International Symposium on Symbolic and Algebraic Computation*, pp. 326–331. ACM Press (2006)
26. Wang, C.: Resonance effect in compartmental analysis and its detection. *Math. Biosci.* **36**, 109–117 (1977)
27. Weispfenning, V.: Comprehensive Gröbner bases. *J. Symbolic Comput.* **14**(1), 1–29 (1992)
28. Weispfenning, V.: Canonical Comprehensive Gröbner bases. *J. Symbolic Comput.* **36**(3-4), 669–683 (2003)
29. X.F. Chen, P.L., Wang, D.: Proving Geometric Theorem by Partitioned-Parametric Gröbner Bases. In: ADG 2004, *Lecture Notes in Artificial Intelligence* **3763**, pp. 34–43. Springer (2004)
30. Yagi, M., Hisakado, T.: Algebraic expression of solutions for harmonic balance method using error bound. IEICE technical report. *Nonlinear problems* **106**(413), 33–38 (2006)

List of MI Preprint Series, Kyushu University

The Global COE Program
Math-for-Industry Education & Research Hub

MI

- MI2008-1 Takahiro ITO, Shuichi INOKUCHI & Yoshihiro MIZOGUCHI
Abstract collision systems simulated by cellular automata
- MI2008-2 Eiji ONODERA
The initial value problem for a third-order dispersive flow into compact almost Hermitian manifolds
- MI2008-3 Hiroaki KIDO
On isosceles sets in the 4-dimensional Euclidean space
- MI2008-4 Hirofumi NOTSU
Numerical computations of cavity flow problems by a pressure stabilized characteristic-curve finite element scheme
- MI2008-5 Yoshiyasu OZEKI
Torsion points of abelian varieties with values in infinite extensions over a p -adic field
- MI2008-6 Yoshiyuki TOMIYAMA
Lifting Galois representations over arbitrary number fields
- MI2008-7 Takehiro HIROTSU & Setsuo TANIGUCHI
The random walk model revisited
- MI2008-8 Silvia GANDY, Masaaki KANNO, Hirokazu ANAI & Kazuhiro YOKOYAMA
Optimizing a particular real root of a polynomial by a special cylindrical algebraic decomposition
- MI2008-9 Kazufumi KIMOTO, Sho MATSUMOTO & Masato WAKAYAMA
Alpha-determinant cyclic modules and Jacobi polynomials

- MI2008-10 Sangyeol LEE & Hiroki MASUDA
Jarque-Bera Normality Test for the Driving Lévy Process of a Discretely Observed Univariate SDE
- MI2008-11 Hiroyuki CHIHARA & Eiji ONODERA
A third order dispersive flow for closed curves into almost Hermitian manifolds
- MI2008-12 Takehiko KINOSHITA, Kouji HASHIMOTO and Mitsuhiro T. NAKAO
On the L^2 a priori error estimates to the finite element solution of elliptic problems with singular adjoint operator
- MI2008-13 Jacques FARAUT and Masato WAKAYAMA
Hermitian symmetric spaces of tube type and multivariate Meixner-Pollaczek polynomials
- MI2008-14 Takashi NAKAMURA
Riemann zeta-values, Euler polynomials and the best constant of Sobolev inequality
- MI2008-15 Takashi NAKAMURA
Some topics related to Hurwitz-Lerch zeta functions
- MI2009-1 Yasuhide FUKUMOTO
Global time evolution of viscous vortex rings
- MI2009-2 Hidetoshi MATSUI & Sadanori KONISHI
Regularized functional regression modeling for functional response and predictors
- MI2009-3 Hidetoshi MATSUI & Sadanori KONISHI
Variable selection for functional regression model via the L_1 regularization
- MI2009-4 Shuichi KAWANO & Sadanori KONISHI
Nonlinear logistic discrimination via regularized Gaussian basis expansions
- MI2009-5 Toshiro HIRANOUCI & Yuichiro TAGUCHI
Flat modules and Groebner bases over truncated discrete valuation rings

- MI2009-6 Kenji KAJIWARA & Yasuhiro OHTA
Bilinearization and Casorati determinant solutions to non-autonomous 1+1 dimensional discrete soliton equations
- MI2009-7 Yoshiyuki KAGEI
Asymptotic behavior of solutions of the compressible Navier-Stokes equation around the plane Couette flow
- MI2009-8 Shohei TATEISHI, Hidetoshi MATSUI & Sadanori KONISHI
Nonlinear regression modeling via the lasso-type regularization
- MI2009-9 Takeshi TAKAISHI & Masato KIMURA
Phase field model for mode III crack growth in two dimensional elasticity
- MI2009-10 Shingo SAITO
Generalisation of Mack's formula for claims reserving with arbitrary exponents for the variance assumption
- MI2009-11 Kenji KAJIWARA, Masanobu KANEKO, Atsushi NOBE & Teruhisa TSUDA
Ultradiscretization of a solvable two-dimensional chaotic map associated with the Hesse cubic curve
- MI2009-12 Tetsu MASUDA
Hypergeometric q -functions of the q -Painlevé system of type $E_8^{(1)}$
- MI2009-13 Hidenao IWANE, Hitoshi YANAMI, Hirokazu ANAI & Kazuhiro YOKOYAMA
A Practical Implementation of a Symbolic-Numeric Cylindrical Algebraic Decomposition for Quantifier Elimination
- MI2009-14 Yasunori MAEKAWA
On Gaussian decay estimates of solutions to some linear elliptic equations and its applications
- MI2009-15 Yuya ISHIHARA & Yoshiyuki KAGEI
Large time behavior of the semigroup on L^p spaces associated with the linearized compressible Navier-Stokes equation in a cylindrical domain

- MI2009-16 Chikashi ARITA, Atsuo KUNIBA, Kazumitsu SAKAI & Tsuyoshi SAWABE
Spectrum in multi-species asymmetric simple exclusion process on a ring
- MI2009-17 Masato WAKAYAMA & Keitaro YAMAMOTO
Non-linear algebraic differential equations satisfied by certain family of elliptic functions
- MI2009-18 Me Me NAING & Yasuhide FUKUMOTO
Local Instability of an Elliptical Flow Subjected to a Coriolis Force
- MI2009-19 Mitsunori KAYANO & Sadanori KONISHI
Sparse functional principal component analysis via regularized basis expansions and its application
- MI2009-20 Shuichi KAWANO & Sadanori KONISHI
Semi-supervised logistic discrimination via regularized Gaussian basis expansions
- MI2009-21 Hiroshi YOSHIDA, Yoshihiro MIWA & Masanobu KANEKO
Elliptic curves and Fibonacci numbers arising from Lindenmayer system with symbolic computations
- MI2009-22 Eiji ONODERA
A remark on the global existence of a third order dispersive flow into locally Hermitian symmetric spaces
- MI2009-23 Stjepan LUGOMER & Yasuhide FUKUMOTO
Generation of ribbons, helicoids and complex scherk surface in laser-matter Interactions
- MI2009-24 Yu KAWAKAMI
Recent progress in value distribution of the hyperbolic Gauss map
- MI2009-25 Takehiko KINOSHITA & Mitsuhiro T. NAKAO
On very accurate enclosure of the optimal constant in the a priori error estimates for H_0^2 -projection

- MI2009-26 Manabu YOSHIDA
Ramification of local fields and Fontaine's property (Pm)
- MI2009-27 Yu KAWAKAMI
Value distribution of the hyperbolic Gauss maps for flat fronts in hyperbolic three-space
- MI2009-28 Masahisa TABATA
Numerical simulation of fluid movement in an hourglass by an energy-stable finite element scheme
- MI2009-29 Yoshiyuki KAGEI & Yasunori MAEKAWA
Asymptotic behaviors of solutions to evolution equations in the presence of translation and scaling invariance
- MI2009-30 Yoshiyuki KAGEI & Yasunori MAEKAWA
On asymptotic behaviors of solutions to parabolic systems modelling chemotaxis
- MI2009-31 Masato WAKAYAMA & Yoshinori YAMASAKI
Hecke's zeros and higher depth determinants
- MI2009-32 Olivier PIRONNEAU & Masahisa TABATA
Stability and convergence of a Galerkin-characteristics finite element scheme of lumped mass type
- MI2009-33 Chikashi ARITA
Queueing process with excluded-volume effect
- MI2009-34 Kenji KAJIWARA, Nobutaka NAKAZONO & Teruhisa TSUDA
Projective reduction of the discrete Painlevé system of type $(A_2 + A_1)^{(1)}$
- MI2009-35 Yosuke MIZUYAMA, Takamasa SHINDE, Masahisa TABATA & Daisuke TAGAMI
Finite element computation for scattering problems of micro-hologram using DtN map

- MI2009-36 Reiichiro KAWAI & Hiroki MASUDA
Exact simulation of finite variation tempered stable Ornstein-Uhlenbeck processes
- MI2009-37 Hiroki MASUDA
On statistical aspects in calibrating a geometric skewed stable asset price model
- MI2010-1 Hiroki MASUDA
Approximate self-weighted LAD estimation of discretely observed ergodic Ornstein-Uhlenbeck processes
- MI2010-2 Reiichiro KAWAI & Hiroki MASUDA
Infinite variation tempered stable Ornstein-Uhlenbeck processes with discrete observations
- MI2010-3 Kei HIROSE, Shuichi KAWANO, Daisuke MIIKE & Sadanori KONISHI
Hyper-parameter selection in Bayesian structural equation models
- MI2010-4 Nobuyuki IKEDA & Setsuo TANIGUCHI
The Itô-Nisio theorem, quadratic Wiener functionals, and 1-solitons
- MI2010-5 Shohei TATEISHI & Sadanori KONISHI
Nonlinear regression modeling and detecting change point via the relevance vector machine
- MI2010-6 Shuichi KAWANO, Toshihiro MISUMI & Sadanori KONISHI
Semi-supervised logistic discrimination via graph-based regularization
- MI2010-7 Teruhisa TSUDA
UC hierarchy and monodromy preserving deformation
- MI2010-8 Takahiro ITO
Abstract collision systems on groups
- MI2010-9 Hiroshi YOSHIDA, Kinji KIMURA, Naoki YOSHIDA, Junko TANAKA & Yoshihiro MIWA
An algebraic approach to underdetermined experiments

MI2010-10 Kei HIROSE & Sadanori KONISHI
Variable selection via the grouped weighted lasso for factor analysis models

MI2010-11 Katsusuke NABESHIMA & Hiroshi YOSHIDA
Derivation of specific conditions with Comprehensive Groebner Systems