Article Submitted to Journal of Symbolic Computation

A new algorithm for discussing Gröbner bases with parameters

Antonio MONTES *[1]

[1] *Dep. Matemàtica Aplicada 2,*
Universitat Politècnica de Catalunya, Spain,
e-mail: montes@ma2.upc.es

Abstract

Let F be a set of polynomials in the variables $\overline{x} = x_1, \ldots, x_n$ with coefficients in $R[\overline{a}]$, where R is a UFD of characteristic zero and $\overline{a} = a_1, \ldots, a_m$ a set of parameters. In this paper we present a new algorithm for discussing Gröbner bases with parameters. The algorithm, called DISPGB, obtains all the cases over the parameters leading to different reduced Gröbner basis, when the parameters in F are substituted in an extension field K of R. This new algorithm improves Weispfenning CGB algorithm, obtaining a reduced set of compatible and disjoint cases. A final algorithm, called GENCASE, uses the output of DISPGB, and determines the minimal singular variety outside of which the Gröbner basis specializes properly. It outputs the corresponding general case and the rest of singular cases inside the minimal singular variety. As a consequence of these constructive methods, the discussion becomes satisfactory and a rich geometrical interpretation is provided in the applications.

DISPGB is based on a generalization of Gaussian elimination (GGE), combined with Gröbner bases techniques, quasi-canonical representation of specializations and a dichotomic tree discussion. It can be easily parallelized, and in that case its time-complexity is of the same order as that of the general case.

1. Introduction.

In many practical applications it is required to determine the reduced Gröbner basis of an ideal of polynomials whose coefficients depend on some parameters. The main problem in this context is to obtain the distinct reduced Gröbner basis for all cases of the parameters.

Examples of this situation can be found, for instance, in constructive algebraic

*Partially supported by GEN. CAT. 1999SGR00356 and DGES-MEC PB98-0933.

geometry when determining conditions for a family of curves to have singular points (see subsection 10.2); in robotics, in order to determine conditions on the magnitudes of a robot configuration with certain degrees of freedom and to solve the inverse kinematics problem (see subsection 10.3); in the load-flow problem for a given electrical network (Montes, 1995, 1998), (see subsection 10.4); in automatic theorem proving, and so on.

A direct approach to this problem can be derived applying Weispfenning (1992) CGB Comprehensive Gröbner Basis algorithm (see also Becker & Weispfenning (1993)). Other contributions to this problem can be found in Kalkbrener (1997). Nevertheless, the output achieved with CGB algorithm is, in general, not simple enough for applications, as it generates more cases than necessary and leads to a complex discussion. In this paper, we present an improved algorithm, called DISPGB, that simplifies the general discussion, as well as a second algorithm, called GENCASE, that takes DISPGB output and provides the minimal singular variety, outside of which Gröbner basis specialize properly. Moreover, it outputs the corresponding general case and the rest of singular cases inside the minimal singular variety.

The main idea of our approach is to focus on the essentially different reduced Gröbner basis, more than computing a CGB. An adapted generalized Gaussian elimination is used for rearranging the input basis of the ideal in such a way that the singular variety (i.e. the algebraic hypersurface in the parameter space containing the exceptional cases) is simplified. Then a dichotomic tree discussion is carried out using quasi-canonical representations of specialization. Finally, using GENCASE algorithm, we are able to obtain the minimal singular variety and the rest of singular cases. This is an important point that is not obtained by classical methods. It provides a satisfactory discussion leading to a lot of geometrical interpretation in applications (see section 10).

In the case of linear systems, Sit (1992) gives an algorithm for the discussion using determinants. For this case, determinants provide the theoretical elements for defining the minimal singular variety. Our algorithm does not use determinants at all, but provides a polynomial in the parameters directly related to the value of the system determinant. Moreover, this discriminant polynomial is also obtained for general non-linear systems, where no analogue of the system determinant is known.

In order to state precisely the problem, let $F = \{f_1, \ldots, f_s\} \subseteq R[\overline{x}, \overline{a}]$ be a set of polynomials in the variables $\overline{x} = x_1, \ldots, x_n$ and the parameters $\overline{a} = a_1, \ldots, a_m$, where R is a unique factorization domain (UFD) of characteristic zero. We can consider F as a polynomial set in the variables $\overline{x}$ with coefficients in $R[\overline{a}]$. The extension to rings of characteristic p, would need further work, and is not considered in this paper.

The objective of DISPGB is to obtain the distinct reduced Gröbner basis for all cases of the parameters. Under this perspective, a case is a set of polynomial equalities and inequalities over the parameters $\overline{a}$. It would be desirable to distinguish cases only when their corresponding reduced Gröbner bases have different leading power products sets in the variables $\overline{x}$. This is a very ambitious requirement. Nevertheless, this objective is completely reached for the general case using GENCASE algorithm. For the special cases this objective is not far from being reached as it will become clear in the examples.

The problem of specialization of Gröbner bases has been actually studied by many authors. The general basic problem can be formulated in the following way. Let R, R' be Noetherian commutative rings with identity, and $\sigma : R \to R'$ be a ring homomorphism. When does a Gröbner basis G of an ideal $I \subseteq R[\overline{x}]$ map to a Gröbner basis of $\langle\sigma(I)\rangle \subset R'[\overline{x}]$ under the natural extension $\sigma : R[\overline{x}] \to R'[\overline{x}]$?

In Kalkbrener (1997) conditions obtained by different authors for this question are reviewed (see Gianni (1989); Pauer (1992); Gräbe (1993); Assi (1994)). Kalkbrener (1997) also proves a necessary and sufficient condition when R' is a field.

We restrict ourselves to the practical case where the original ring R is a polynomial ring $R[\overline{a}]$ over a unique factorization domain R of zero characteristic and R' some extension field K of the quotient field $\mathrm{Quot}(R)$. The variables $\overline{a}$ are considered as parameters. In this case $R[\overline{a}]$ can be embedded in the quotient field $R(\overline{a})$ and we do not need the general algorithm for computing Gröbner basis over rings given in Möller (1988), but only the classical Buchberger's algorithm over fields.

The structure of this paper is as follows:

Sections 2 and 3 are devoted to the study of how specialization acts on the division and Gröbner basis algorithms. It is proved that they specialize to the corresponding algorithms in $K[\overline{x}]$ except for a $m-1$ dimensional variety (hypersurface). We call *singular variety* the hypersurface outside which Buchberger's algorithm specializes. It is not difficult to complete the algorithm in order to obtain a sufficient singular variety.

Nevertheless, the singular variety computed by Buchberger's algorithm is in general larger then is strictly necessary, and is algorithm depending. In order to minimize it we introduce the new Generalized Gaussian Elimination algorithm (GGE). This algorithm produces a new basis specializing properly for any specialization and is a good input for Buchberger's algorithm. It is also directly useful in applications, because it is able to simplify problems. Section 4 is devoted to the study of GGE algorithm (Montes, 1999).

In section 5 we describe how specializations can be specified in a quasi-canonical form in order to obtain a dichotomic discussion, that is a central point of DISPGB.

We are now ready to deal with the general DISPGB algorithm. Section 6 provides a general description of DISPGB that tries to make it understandable. Section 7 is devoted to two essential subalgorithms that take part in DISPGB. Section 8 is devoted to the control algorithms that manage the whole analysis; the final theorem describing DISPGB is also given there.

In section 9 a final theorem and the corresponding algorithm GENCASE is obtained allowing to define and determine, from the output of DISPGB, the unique *general case* and the corresponding *minimal singular variety.*

Finally some classical applications are given in the last section, showing both the elegance and power of DISPGB.

In what follows we assume that $\succ_{\overline{x}}$ is a given monomial order for the variables $\overline{x}$. We use the following notations. A power product in the variables $\overline{x}$ is denoted $\overline{x}^\alpha$. A monomial is a product of a coefficient in the coefficient ring or field and a power product of the variables. The leading monomial, coefficient and power product are denoted:

$$\mathrm{lm}(f, \succ_{\overline{x}}) = c\,\overline{x}^\alpha, \quad \mathrm{lc}(f, \succ_{\overline{x}}) = c, \quad \mathrm{lpp}(f, \succ_{\overline{x}}) = \overline{x}^\alpha.$$

We will also need a monomial order $\succ_{\overline{a}}$ involving the parameters $\overline{a}$ and the compatible elimination product order $\succ_{\overline{x}\,\overline{a}}$ (see Bayer & Stillman (1987)). It is defined by:

For all $\alpha, \gamma \in \mathbb{Z}^n_{\geq 0}$ and $\beta, \delta \in \mathbb{Z}^m_{\geq 0}$

$$\overline{x}^\alpha \overline{a}^\beta \succ_{\overline{x}\,\overline{a}} \overline{x}^\gamma \overline{a}^\delta \Longleftrightarrow \left\{ \overline{x}^\alpha \succ_{\overline{x}} \overline{x}^\gamma \quad \text{or} \quad \left(\overline{x}^\alpha = \overline{x}^\gamma \ \text{ and } \ \overline{a}^\beta \succ_{\overline{a}} \overline{a}^\delta \right) \right\}. \tag{1}$$

In order to avoid denominators and unnecessary factors in S-polynomials we define them as follows:

$$S(f,g) = \frac{\Gamma\,\overline{x}^\gamma}{\mathrm{lm}(f)}\,f - \frac{\Gamma\,\overline{x}^\gamma}{\mathrm{lm}(g)}\,g \tag{2}$$

where

$$\begin{array}{rcl} \Gamma & = & \mathrm{lcm}(\mathrm{lc}(f), \mathrm{lc}(g)) \\ \overline{x}^\gamma & = & \mathrm{lcm}(\mathrm{lpp}(f), \mathrm{lpp}(g)) \end{array}$$

with a slightly different normalization than usual.

For specializations coming from replacement of parameters, the corresponding ring homomorphism is:

$$\sigma : R[\overline{a}] \longrightarrow K \tag{3}$$

where K is any field extension of R (a UFD with characteristic 0) and $\sigma|R = Id$. The discussion is complete when we consider the algebraic closure $\overline{K}$ of K. Providing a specialization is equivalent to choosing $\overline{a}_0 \in K^m$, and setting $\sigma(\overline{a}) = \overline{a}_0$. We will denote also by σ the natural extension $\sigma' : R[\overline{a}][\overline{x}] \longrightarrow K[\overline{x}]$ which is the identity on $\overline{x}$.

We can identify $\sigma_{\overline{a}_0}$ with $\overline{a}_0$, and extend σ for certain elements of $R(\overline{a})$ (more precisely to the local ring $R[\overline{a}]_{\ker(\sigma)}$). When we say that $\sigma(f)$ makes sense for $f \in R(\overline{a})[\overline{x}]$ it has to be understood that $f \in R[\overline{a}]_{\ker(\sigma)}[\overline{x}]$.

For obtaining k-quasi-canonical representations of the specification, as well as in other parts of the algorithms in this paper, we need to decompose polynomials into irreducible factors. We denote

$$\mathrm{FACVAR}(W, \succ_{\overline{a}}) = \{q_1, \ldots, q_s\}$$

the set of irreducible polynomials that are factors of the polynomials of W, normalized in a canonical form in order to be recognized when compared. CAS systems provide algorithms to do this.

2. Division algorithm and specialization

We now study division under specialization and prove the following

THEOREM 1: *Let $f, g_1, ..., g_s \in R[\overline{a}][\overline{x}]$, $W = \mathrm{FACVAR}(\mathrm{lc}(g_1), \ldots, \mathrm{lc}(g_s))$ and σ a specialization satisfying $\sigma(w) \neq 0$, for all $w \in W$, so that $\sigma(\mathrm{lc}(g_i)) \neq 0$, for $1 \leq i \leq s$. Let $q_1, \ldots, q_s \in R(\overline{a})[\overline{x}]$ and $r \in R(\overline{a})[\overline{x}]$ be the quotients and remainder of the division of f by $g_1, \ldots, g_s$ in $R(\overline{a})[\overline{x}]$. Then $\sigma(q_1), \ldots, \sigma(q_s)$ and $\sigma(r)$ make sense, and they are the corresponding quotients and remainders for the division of $\sigma(f)$ by $\sigma(g_1), \ldots, \sigma(g_s)$ in $K[\overline{x}]$.*

Proof: We prove the result by induction on the division step. Call $(p^{(j)}, {q_1}^{(j)}, \ldots, {q_s}^{(j)})$ and $(p'^{(k)}, {q'_1}^{(k)}, \ldots, {q'_s}^{(k)})$ the partial (remainders, quotients) at steps j and k respectively of the divisions in $R(\overline{a})[\overline{x}]$ and in $K[\overline{x}]$. We want to prove that for each j there is some k so that $\sigma(p^{(j)}) = p'^{(k)}$ and $\sigma(q_i^{(j)}) = q_i'^{(k)}$ for $i = 1, \ldots, s$. It is true, by construction, for $j = 0$ by picking $k = 0$. Suppose we are at step j in the division in $R(\overline{a})[\overline{x}]$, and let us assume the induction hypothesis. Let i_j be the first index for which $\mathrm{lpp}(g_{i_j})$ divides $\mathrm{lpp}(p^{(j)})$. The new partial quotient and the new partial remainder will be

$$\begin{aligned} \psi_{i_j} &= \frac{\mathrm{lm}(p^{(j)})}{\mathrm{lm}(g_{i_j})} \\ p^{(j+1)} &= p^{(j)} - \psi_{i_j} g_{i_j}. \end{aligned} \tag{4}$$

Observe that by hypothesis, σ makes sense on ψ_{i_j} and $p^{(j+1)}$. So, specializing, two cases can occur:

Case 1: $\sigma(\mathrm{lc}(p^{(j)})) = 0$. In this case $\sigma(\psi_{i_j}) = 0$ and $\sigma(p^{(j+1)}) = \sigma(p^{(j)})$, and the same is true for the partial quotients. So the same k applies to $j + 1$.

Case 2. $\sigma(\mathrm{lc}(p^{(j)})) \neq 0$. Then $\sigma(\psi_{i_j}) \neq 0$. As by hypothesis all $\mathrm{lpp}(g_i)$ remain stable under specialization, i_j is the first index for which $\mathrm{lpp}(\sigma(g_{i_j}))$ divides $\mathrm{lpp}(p'^{(k)})$, so

$$\begin{aligned} \psi'_{i_j} &= \frac{\mathrm{lm}(p'^{(k)})}{\mathrm{lm}(\sigma(g_{i_j}))} \\ p'^{(k+1)} &= p'^{(k)} - \psi'_{i_j} \sigma(g_{i_j}). \end{aligned}$$

Using the induction hypothesis for j and specializing equations (4) yields:

$$\begin{aligned}\sigma(\psi_{i_j}) &= \frac{\sigma(\mathrm{lm}(p^{(j)}))}{\sigma(\mathrm{lm}(g_{i_j}))} = \frac{\mathrm{lm}(p'^{(k)})}{\mathrm{lm}(\sigma(g_{i_j}))} = \psi'_{i_j} \\ \sigma(p^{(j+1)}) &= p'^{(k)} - \psi'_{i_j}\sigma(g_{i_j}) = p'^{(k+1)},\end{aligned}$$

so in this case we pick $k+1$. Note that the partial quotients also remain stable.

□

The division in $R(\overline{a})[\overline{x}]$ introduces coefficients with denominators in $R[\overline{a}]$. But these denominators arise only from leading coefficients of the divisors $g_1, \dots, g_s$, and are divisors of a product of powers of them. Under the hypothesis of theorem 1, we can still multiply f by a convenient factor μ in $R[\overline{a}]$ to eliminate non-vanishing denominators for the specification. This will only change the normalization of f.

Writing $q'_i = \mu\, q_i$ and $r' = \mu\, r$, we have

$$\mu\, f = \sum_{i=1}^{s} q'_i g_i + r'. \tag{5}$$

The result will be a division with quotients q'_i, $(1 \le i \le s)$ and remainder r' in $R[\overline{a}][\overline{x}]$. This algorithm is usually denoted as NORMALFORM.

If the condition on the leading monomials is satisfied, theorem (1) guarantees that NORMALFORM specializes automatically except for the not null factor $\sigma(\mu) \neq 0$.

3. Buchberger's algorithm and specialization

We denote GB0 the ordinary Buchberger's algorithm (see Cox, Little & O'Shea (1992)), that only add remainders of S-polynomials, acting on $k[\overline{x}]$, where k is any field (in particular it can be $R(\overline{a})$ or K).

We define PGB0 as the version of GB0 that uses equation (2) for computing S-polynomials and the normalization of the divisions described in section 2 to avoid denominators when computing in $R(\overline{a})[\overline{x}]$. We have:

THEOREM 2: *Let $F \subset K[\overline{a}][\overline{x}]$, $G_0 = \mathrm{PGB0}(F, \succ_{\overline{x}})$ and*

$$W = \mathrm{FACVAR}(\{\mathrm{lc}(g) : g \in G_0\}, \succ_{\overline{a}}).$$

Then

(i) *G_0 is a Gröbner basis of $\langle F\rangle R(\overline{a})[\overline{x}]$ whose polynomials belong to $R[\overline{a}][\overline{x}]$.*

(ii) *If $\sigma(w) \neq 0$, for all $w \in W$, then $\sigma(G_0)$ is a Gröbner basis of $\langle \sigma(F)\rangle K[\overline{x}]$.*

(iii) *G_0 can be minimized and reduced to a new basis G whose elements are in $R[\overline{a}][\overline{x}]$. Furthermore, under the above conditions on σ, $\sigma(G)$ is the reduced Gröbner basis of $\langle\sigma(F)\rangle K[\overline{x}]$ (apart from normalization).*

A precise description of the algorithm PGB that outlines theorem 2 is given in subsection 4.2.

Using definition 2 and theorem 1 the proof of the above proposition is easy.

So PGB has G and W as outputs. W correspond to the specification $\sigma \in \Sigma(\{0\}, W)$ for which G specializes to the reduced Gröbner basis of $\langle\sigma(F)\rangle K[\overline{x}]$. It outputs what we call the general case.

Nevertheless, the singular variety obtained by PGB is a sufficient one, but is not the minimal possible. It is important to obtain a better solution. This objective will be considered in the next section.

4. GGE: Generalized Gaussian Elimination algorithm

If we compute the singular variety of an ideal using PGB, it is clear, that the resulting reduced Gröbner basis is unique. But the computed singular variety depends on the development of PGB0 and is not uniquely determined by the ideal, as the algorithm can follow different paths. In any case, we obtain a sufficient condition for the Gröbner basis to specialize, but this condition is algorithm depending (see Example 4.3). It is very important to reduce the variety as much as possible.

For this purpose we introduce a Generalized Gaussian Elimination algorithm (GGE) to transform the initial basis F of the ideal into a more simple basis F'.

The idea of the algorithm comes from ordinary Gaussian elimination, and produces a triangulation of the leading terms with respect to the order $\succ_{\overline{xa}}$.

Taking F' as input for PGB0, the algorithm is simplified and we expect that less special cases will occur in the singular variety when an adequate order of the variables is chosen. In the examples it is shown that it is so. Nevertheless, the process is sensitive to the variable order chosen, and we cannot ensure, in general, complete minimality of the singular variety obtained. It is important to choose the variables in an order for which there are leading terms for the order $\succ_{\overline{xa}}$ not containing the parameters, and dividing other leading terms.

4.1. Generalized Gaussian Elimination, GGE

Input: $F = [f_1, \ldots, f_s] \subset R[\overline{x}, \overline{a}]$, ($\overline{x}$ and $\overline{a}$ as variables)
$\succ_{\overline{x}\,\overline{a}}$ the product elimination order.
Output: F' a new basis of $\langle F\rangle$ that specializes for any σ.

$G := F;\ G' = \phi$
WHILE $G \neq G'$
  FOR i from 1 to #G DO
    $G' := G$

$G := \phi$
FOR k from 1 to #G' DO
IF $k \neq i$ THEN
$f := \overline{g'_k}^{[g'_i]}$ respect to the order $\succ_{\overline{x}\,\overline{a}}$.
$G := G \cup [f]$
ELSE $G := G \cup [g'_i]$
$F' :=$ primpart(G).

Proposition 3: *Let $F \subset R[\overline{a}][\overline{x}]$, and $F' := \mathrm{GGE}\,(F, \succ_{\overline{xa}})$. Then*

(i) *F' is a new basis of $\langle F\rangle \mathrm{Quot}(R)[\overline{x}, \overline{a}]$ that has less or equal number of elements than F.*

(ii) *$\langle \mathrm{lpp}(F, \succ_{\overline{xa}})\rangle \subseteq \langle \mathrm{lpp}(F', \succ_{\overline{xa}})\rangle$, and also $\langle \mathrm{lpp}(F, \succ_{\overline{x}})\rangle \subseteq \langle \mathrm{lpp}(F', \succ_{\overline{x}})\rangle$.*

(iii) *F' specializes, i.e. $\sigma(F')$ is also a basis of $\langle \sigma(F)\rangle$.*

(iv) *The algorithm terminates.*

(v) *Each $p \in F'$, is the residue of dividing p by $F' - \{p\}$ with respect to the given order $\succ_{\overline{xa}}$. We say that F' is reduced for this order.*

Proof: (i) At each division in the algorithm we substitute, in a given F-basis, two polynomials in $R[\overline{a}][\overline{x}]$, say $\{f, g\}$ by $\{g, r\}$, where $f = g\,h + r$. As f and g can be expressed as a linear combination of g and r with coefficients in Quot(R), then $\{g, r\}$ is a new basis of $\langle f, g\rangle$ and, consequently, $\{g, r\}$ can replace $\{f, g\}$ to form a new basis of F. This is the only repeated action in computing the GGE. Observe also, that the final number of polynomials is less or equal than the number of polynomials in F, as we replace at each step two polynomials by two or one (when the residue is 0) polynomials.

(ii) The first part is obvious, as the new polynomials are remainders with respect to the order $\succ_{\overline{x}\,\overline{a}}$. When only the $\overline{x}$ are taken as variables, the proposition is a consequence of $\succ_{\overline{x}\,\overline{a}}$ being a compatible elimination order.

(iii) The argument given in (i) is also valid when the $\overline{a}$ are considered as parameters, as no denominators of $R[\overline{a}]$ appear.

(iv) We have to prove that the algorithm finishes. Let F_i be the base after the i^{th} loop. Define

$$d_i = \sum_{f \in F_i} \mathrm{multideg}(f, \succ_{\overline{xa}}).$$

As $\#F_i \leq \#F_{i-1}$ and for every division in the algorithm, whenever the leading term is divisible, the multideg of the remainder is strictly smaller then that of the dividend: necessarily $d_i \preceq_{\overline{xa}} d_{i-1}$. If $d_i = d_{i-1}$ then no leading monomial has changed in loop i. So, no monomial of a $f \in F_i$ is divisible by any leading monomial of an element of F_i (with constant leading coefficient, when we only consider the $\overline{x}$ as variables), as all possible divisions have been performed in loop i. The algorithm stops in this case. As $\succ_{\overline{xa}}$ is a well-order, d_i necessarily stabilize, and the algorithm effectively terminates. GGE works in the direction of the reduced Gröbner basis, even if it needs not reach it, as not all S-polynomials are constructed and tested.

(v) As all leading coefficients are constants of R, all polynomials in F' have been considered as divisors of the others, and no polynomial will be divisible by the remaining. This property is the triangulation property of GGE. □

We include GGE as a previous step of PGB algorithm, before calling PGB0, in order to transform the initial basis. The resulting singular variety will now be smaller, and if we choose a suitable order of the variables, in practice we are often able to obtain the minimal singular variety.

GGE algorithm is remarkable for linear systems with parameters. If we apply it dividing only with respect to the $\overline{x}$ variables, but not to the parameters, it reduces to the ordinary Gaussian elimination carried out with care, i.e. using only pivots not depending on the parameters. But if we apply it with respect to variables and parameters, it goes further, and provides a perfect first step for a reduced case discussion, as can be seen in the example below.

This algorithm is used in our general DISPGB algorithm. Nevertheless it is interesting by itself, as it provides a simplification of the basis that is useful in applications.

Let us give finally the PGB algorithm:

4.2. PGB algorithm

Input:	$F = \{f_1, \ldots, f_s\} \subset R[\overline{x}, \overline{a}]$, and $\succ_{\overline{xa}}$
Output:	G the reduced Gröbner basis of $\langle F \rangle$, with respect to $[\overline{x}]$.
	W: The singular variety outside of which the basis specializes.

$G_1 := \mathrm{GGE}(F, \succ_{\overline{xa}})$
$G_2 := \mathrm{PGB0}(G_1, \succ_{\overline{x}})$
$W := \mathrm{FACVAR}(\{\mathrm{lc}(g, \succ_{\overline{x}}) : g \in G_2\})$
$G := \mathrm{NORMALIZE}(\mathrm{REDUCE}(\mathrm{MINIMAL}(G_2, \succ_{\overline{x}}), \succ_{\overline{x}}), \succ_{\overline{xa}})$

4.3. Example

Consider the following linear system in the variables (x, y, z, u) with the parameters (a, b).

$$\begin{aligned} F \ := \ & [a\,x + 2\,y + 3\,z + u - 6, x + 3\,y - z + 2\,u - b, \\ & 3\,x - a\,y + z - 2, 5\,x + 4\,y + 3\,z + 3\,u - 9] \end{aligned}$$

We take F to be the basis of an ideal in $\mathbb{Z}[x, y, z, u, a, b]$ and compute a new basis with GGE algorithm, relative to the order $\mathrm{lex}(x, y, z, u, a, b)$. The result is:

$$\begin{array}{rcl} F' & := & \mathrm{GGE}(F, \mathrm{lex}(x,y,z,u,a,b)) \\ & = & [756\,x - 39\,a\,b - 4\,b - 155 - 117\,a + (117\,a + 51)\,u, \\ & & 189\,y + 6\,a\,b - 107 - 43\,b + 18\,a - (18\,a - 123)\,u, \\ & & 756\,z - 1439 + 236\,b + 99\,a + 33\,a\,b - (99\,a - 15)\,u, \\ & & (9\,a^2 - 30\,a + 21)\,u - 9\,a^2 - 3\,a^2\,b + 11\,a\,b + 22\,a - 49 + 28\,b,] \end{array}$$

As we see, the new basis is completely Gaussian reduced:

1. The new basis is triangular: each leading monomial is different when we consider as variables all x, y, z, u, a, b. In this case, it is also true if we consider only x, y, z, u as variables.
2. The three first pivots (leading coefficients) of the variables x, y, z are constants, and only the last leading coefficient in the variable u is a polynomial in the parameters. It is essentially the singular variety: $9\,a^2 - 30\,a + 21 = 3\,(a-1)\,(3\,a-7)$, precisely the value of the determinant.
3. Instead, the variable u is not eliminated in the equations giving x, y, z.
4. In this form, the system is very simple to discuss.
5. The result is, in this case, a minimal Gröbner basis, but not the reduced one.
6. Moreover, in this example, F' is itself a comprehensive Gröbner basis.

Now we compute $\mathrm{PGB}(F', \mathrm{lex}(x,y,z,u), \mathrm{lex}(a,b))$ to obtain the reduced Gröbner basis and the singular variety. We obtain:

$$\begin{array}{rcl} G & = & (9\,a^2 - 30\,a + 21)\,x - 6\,a\,b + 9\,a - 2\,b - 1, \\ & & (9\,a^2 - 30\,a + 21)\,y + 3\,a\,b + 20 - 23\,b, \\ & & (9\,a^2 - 30\,a + 21)\,z + 53\,a - 5\,a\,b - 18\,a^2 + 3\,a^2\,b - 39 + 6\,b, \\ & & (9\,a^2 - 30\,a + 21)\,u - 9\,a^2 - 3\,a^2\,b + 11\,a\,b + 22\,a - 49 + 28\,b] \\ & & \\ W & = & \{a - 1, 3\,a - 7\}. \end{array}$$

In this example, W determines the minimal singular variety, as it includes just the factors of the determinant of the system, that we know by linear algebra to be the singular condition. The general discussion of all cases can be done with DISPGB described in the next sections. The complete output provided by DISPGB for this example can be found in subsection 10.5

If we compute the Gröbner basis using PGB algorithm without previous reduction of F to F' by GGE, the basis G turns out to be the same, but the variety becomes $W_1 = \{a, a^2 + 35, 3\,a - 2, a - 1, 3\,a - 7\}$, showing the reduction obtained by GGE.

5. Specification of specializations

Let σ be a specialization. In the evolution of the algorithms different hypotheses about the specialization of some polynomials in $R[\overline{a}]$ are assumed.

DEFINITION 4: *Let σ be a specialization and $N = \{p_1, \dots, p_s\} \subset R[\overline{a}]$ and $W = \{q_1, \dots, q_r\} \subset R[\overline{a}]$ be sets of polynomials. We say that $\sigma \in \Sigma(N, W)$, or that (N, W) is the* actual *specification of σ if*

(i) *All the p_i's specialize to zero: $\sigma(p_i) = 0$, for $p_i \in N$.*

(ii) *All the q_i's specialize to non-zero: $\sigma(q_i) \neq 0$, for $q_i \in W$.*

We say that N are the null conditions and W the not-null conditions of the specification of all $\sigma \in \Sigma(N, W)$.

PROPOSITION 5: *Let $\sigma \in \Sigma(N, W)$ be a specialization. If $f \in \sqrt{\langle N \rangle}$, then $\sigma(f) = 0$. Equivalently:*

$$\sigma \in \Sigma(N, W) \Longrightarrow \sigma \in \Sigma(\sqrt{\langle N \rangle}, W)$$

Proof: As σ is an homomorphism if $f \in \langle N \rangle$ then $\sigma(f) = 0$. Moreover, as $\ker(\sigma)$ is a prime ideal this is also true if $f \in \sqrt{\langle N \rangle}$. □

PROPOSITION 6: *Let K be an algebraically closed field and $\Lambda = \Sigma(N, W)$. Set $h = \prod_{q \in W} q$. Then $\Lambda \neq \phi$ iff $\langle N \rangle \neq K[\overline{a}]$ and $h \notin \sqrt{\langle N \rangle}$.*

Proof: [$\Rightarrow$]: By proposition 5.

[$\Leftarrow$]: By Nullstellensatz, $\mathbb{I}\mathbb{V}(N) = \sqrt{\langle N \rangle}$, and if $h \notin \sqrt{\langle N \rangle}$, it exists $a_0 \in \mathbb{V}(N)$ for which $h(a_0) \neq 0$. Thus $\sigma_{a_0} \in \Lambda$, and $\Lambda \neq \phi$. □

The set of values of the parameters described by $\Sigma(N, W)$ is given by

$$\mathbb{V}(N) - [\mathbb{V}(N) \cap [\cup_{q \in W} \mathbb{V}(q)]]$$

PROPOSITION 7: *Let $W = \{q_1, \dots, q_r\}$, N be a Gröbner basis relative to the order $\succ_{\overline{a}}$ and $W' = \mathrm{FACVAR}(\{\overline{q_1}^N, \dots, \overline{q_r}^N\}$. Then*

$$\Sigma(N, W) = \Sigma(N, W')$$

Proof: We have $q_i = n_i + \overline{q_i}^N$, where $n_i \in \langle N \rangle$. As σ is an homomorphism,

$$\sigma(q_i) = \sigma(n_i) + \sigma(\overline{q_i}^N) = \sigma(\overline{q_i}^N)$$

So $\sigma(q_i) \neq 0$ iff $\sigma(\overline{q_i}^N) \neq 0$, and we can finally decompose into irreducible factors. □

DEFINITION 8 (QUASI-CANONICAL REPRESENTATION OF A SPECIFICATION): *A representation (N, W) is said to be k-quasi-canonical, if it verifies*

(i) *N is the reduced Gröbner basis in* $\mathrm{Quot}(R)[\overline{a}]$ *for the order* $\succ_{\overline{a}}$ *of the set of polynomials that specialize to zero.*

(ii) *The polynomials in W spezializing to non-zero are reduced modulo N and are irreducible over $k[\overline{a}]$, where k is some intermediate field between* $\mathrm{Quot}(R)$ *and the algebraic closure $\overline{K}$. They are normalized in a canonical form in order to be recognized when compared. (Normalization depends on the field k, but is easy to define for* $\mathbb{Q}, \mathbb{R}$ *and* $\mathbb{C}$.

(iii) $\displaystyle\prod_{q\in W} q \notin \sqrt{\langle N\rangle}$.

(iv) *The polynomials in N are square-free.*

(v) *If some $p \in N$ factors, then no factor of p belongs to W.*

PROPOSITION 9: *Any specification that is non-empty in the algebraic closure has a k-*quasi-canonical *representation (N, W). The representation is unique, if we impose N to be a basis of $\sqrt{\langle N\rangle}$.*

Nevertheless, for practical computation reasons, we do not impose N to be radical.

Proof: (i) We can choose a Gröbner basis to define the ideal $\langle N\rangle$. This will report benefits in the algorithms.

(ii) By proposition 7 we can reduce the polynomials in W modulo N, decompose into irreducible factors and normalize them (depending on the field).

(iii) By proposition 6, $\prod_{q\in W} q \notin \sqrt{\langle N\rangle}$ is a necessary and sufficient condition for N and W to be compatible conditions in the algebraic closure, i.e. to determine a non-empty $\Sigma(N, W)$.

(iv) We can drop any multiple factor in N as a consequence of proposition 5.

(v) We can also drop any factor of a polynomial $p \in N$ belonging to W as $p = gh$ and $g \in W$ imply $\sigma(p) = 0$ and $\sigma(g) \neq 0$, which leads to $\sigma(h) = 0$.

□

If in the development of the algorithms a polynomial $p' \in R[\overline{a}]$ such that $p' \in \sqrt{\langle N\rangle}$ and $p' \notin \langle N\rangle$ is detected, we can refine the representation of the specification adding p' to the base N, and re-computing the new Gröbner basis N'. We then have:

$$\langle N\rangle \subseteq \langle N'\rangle \subseteq \sqrt{\langle N\rangle}.$$

Then (N', W), (N, W) and $(\sqrt{\langle N\rangle}, W)$ are representations of the same specification $\Sigma(\sqrt{\langle N\rangle}, W)$, but (N', W) is a finer representation than (N, W). Even if some $p \in N'$ has a factor in W, it can also be dropped, as it specializes to non-zero.

DEFINITION 10: *Let* $f_1, f_2 \in R[\overline{a}][\overline{x}]$. *We say that* $f_1 \approx_\sigma f_2$ *(are* σ*-equivalents) iff* $\sigma(f_1) = \sigma(f_2)$.

When a specification is assumed, then we can substitute the polynomials in a set by σ-equivalent polynomials. This will be done all along the algorithm. It must be noticed, that doing so, the new σ-equivalent set is no more a basis of the original $F \subset R[\overline{a}][\overline{x}]$. This is an important observation for the understanding of the algorithm. Doing so, we abandon to compute a CGB, but the algorithm is simplified towards our objective. Observe that by doing this, unnecessary polynomials specializing to zero in the actual specification are also dropped.

DEFINITION 11: *Given two specifications* (N_1, W_1) *and* (N_2, W_2) *we say that* $(N_2, W_2) \geq (N_1, W_1)$, *if* $\sqrt{\langle N_1 \rangle} \subseteq \sqrt{\langle N_2 \rangle}$ *and* $W_1 \subseteq W_2$.

If $(N_2, W_2) \geq (N_1, W_1)$ *and* $\sqrt{\langle N_1 \rangle} \neq \sqrt{\langle N_2 \rangle}$ *or* $W_1 \neq W_2$, *then we say that* $(N_2, W_2) > (N_1, W_1)$.

DEFINITION 12: *The specifications where* $N = \{0\}$ *are denoted* general, *in the sense that* $\Sigma(\{0\}, W)$ *contains all specializations except a set in the* $m-1$ *dimensional variety* $\cup_{q \in W} \mathbb{V}(q)$ *defined by* W.

5.1. CANSPEC algorithm

The algorithms will only use quasi-canonical representations of specifications.

PROPOSITION 13: *Given the specification* (N, W), *the algorithm* CANSPEC *produces a* k*-quasi-canonical representation* (N', W') *if* test = true. *If* test = false *then* (N, W) *are incompatible, i.e.* $\Sigma(N, W) = \phi$

CANSPEC

Input:	(N, W):	the specification of σ.
	$\succ_{\overline{a}}$:	monomial order for the parameters
Output:	*test*:	has the value *true* if $\Sigma(N, W) \neq \phi$ in the algebraic closure and *false* if they are incompatible.
	(N', W'):	A k-quasi-canonical representation of the specification.

$W' :=$ FACVAR($\{\overline{q}^N : q \in W\}$
$test := true$; $t := true$; $N' := N$
$h := \prod_{q \in W} q$
IF $h \in \sqrt{\langle N' \rangle}$ THEN $test := false$; $N' := \{1\}$; STOP
WHILE t DO
  $t := false$
  $N'' :=$ Drop any factor of a polynomial in N' that belongs
    to W', as well as multiple factors.

IF $N'' \neq N'$ THEN
  $t := true$
  $N' := \text{GB}(N'', \succ_{\overline{a}})$
  $W' := \text{FACVAR}(\{\overline{q}^N : q \in W'\})$

Proof: The first step reduces W' modulo N using proposition 7. Then one uses proposition 6 to test compatibility. It is sufficient to test it only once as no transformation to a new representation (N', W') will alter it.

Then the loop intends to simplify N' eliminating any factor in N' belonging to W'. If one factor is found then one recomputes the Gröbner basis for N' and reduces W' modulo N'. The loop is finite by the ascending chain condition (ACC).

Thus all the conditions for (N', W') to be a quasi-canonical representation are met. □

6. General description of DISPGB algorithm

We are now ready to describe the general DISPGB algorithm (Discussing Parametric Gröbner Basis). DISPGB builds up a binary tree structure beginning at the root. At a the vertex, bifurcation takes place making the dichotomic decision about some leading coefficient of a polynomial in $R[\overline{a}][\overline{x}]$ to become zero or not-zero in the specialization, like in dynamical evaluation (Duval, 1995). As DISPGB progresses in depth, the specification of the specialization increases in the branches. At each vertex v the following objects are stored in the table variable

$$T(v) = (c_c, B_v, N_v, W_v)$$

1. The new condition c_v of the form $p(\overline{a}) = 0$ (for type 0 vertices), or $p(\overline{a}) \neq 0$ (for type 1 vertices), that is assumed when the algorithm gains access to vertex v in order to increase the specification of σ.
2. The pair (N_v, W_v) determining a quasi-canonical representation of the specification of σ at this vertex (see definition 8). (N_v, W_v) accumulates all decisions assumed from the root till vertex v.
3. $B_v \subset R[\overline{a}][\overline{x}]$, a set of polynomials that specializes to a basis of the ideal $\langle\sigma(F)\rangle$ for any $\sigma \in \Sigma(N, W)$. (It is no more a basis of $\langle F\rangle R[\overline{a}][\overline{x}]$).

A vertex is terminal whenever B_v becomes a set of polynomials specializing to a Gröbner basis of $\langle\sigma(F)\rangle$ for $\sigma \in \Sigma(\langle N_v\rangle, W_v)$. Then specialization of σ is finished. The algorithm finishes when all branches arrive to terminal vertices.

Terminal vertices contain the complete information about the distinct specialized Gröbner basis and specifications. Tracing the tree provides a dichotomic discussion of the decisions leading to the corresponding cases.

Even if it does not exist a unique canonical discussion, as it depends on the order in which the decisions are taken, a minimum disjoint set of cases is obtained.

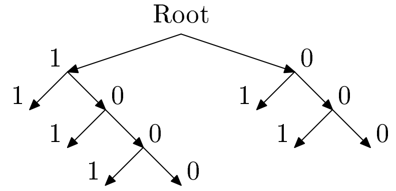

Figure 1: Tree structure of the algorithm in the example of subsection 10.4

DISPGB labels a vertex v adjoining 0 or 1 to the label of its just upper-vertex. It adjoins a 0 if the vertex corresponds to a null-condition and a 1 if it corresponds to a not-null-condition. So the label of a vertex characterize the path of decisions that have been taken so far.

At vertex v, σ is supposed to belong to $\Sigma(N_v, W_v)$, and the algorithm determines a B_v that specializes to a basis of $\langle\sigma(F)\rangle$ for which $\langle \mathrm{lpp}(B_v, \succ_{\overline{x}})\rangle$ is as large as possible, i.e. B_v is as near as possible to the specializing Gröbner basis.

To clarify the description, we give in fig. 1 the tree corresponding to the example in subsection 10.4, with 4 variables, 4 parameters and 4 equations (second degree in the variables). It is a typical discussion and gives 7 distinct final cases.

An important observation is that DISPGB can be completely parallelized. † When a decision is taken at some vertex producing a branching of the algorithm, only the information of the actual vertex is needed to follow the branches below it, and no other information about the upper or lateral vertices is needed. So the algorithm can be parallelized.

In the example, 4 processors would be useful that is the maximal width of the tree.

In order to arrive at a terminal vertex following the direct path, only some more lateral computations than for the general case are needed. This is the reason while the parallelized algorithm has the same time-complexity.

7. Conditional algorithms

DISPGB uses two conditional algorithms: NEWCOND and CONDPGB. The objective of NEWCOND is, given $f \in R[\overline{a}][\overline{x}]$ and a specification (N, W) of σ, to test if a new undecidable condition must be assumed in order to be able to decide about the specialization of the leading coefficient to zero or not. The objective of CONDPGB is, given the specification and $B \subset R[\overline{a}][\overline{x}]$, where $\sigma(B)$ is a basis of $\langle\sigma(F)\rangle$, to advance in the direction of a specializing Gröbner basis as much as possible.

†We implemented it in *Maple* (where parallel computations are not yet possible). In order to be supple, the implementation allows the user to choose visiting first 0 or 1 vertices when building up the tree. It also allows to begin at a known vertex giving the informations (N_v, W_v, B_v) in it, so well as other useful facilities.

7.1. NEWCOND algorithm

PROPOSITION 14: *Let* (N, W) *be the specification of* σ, *and* $f \in R[\overline{a}][\overline{x}]$. *Then the following algorithm* NEWCOND *determines the triplet* (cd, f', N') *where* $f' \approx_\sigma f$.

If $cd = \phi$, *then* $\sigma(\mathrm{lc}(f', \succ_{\overline{x}})) \neq 0$ *for* $\sigma \in \Sigma(N, W)$. *If* $cd \neq \phi$, *set* $c = \prod_{w \in cd} w$. *Then* $\sigma(\mathrm{lc}(f', \succ_{\overline{x}}))$ *vanishes or not depending on the nullity or not of* $\sigma(c)$.

(N', W') *is a refinement of* (N, W), *when some polynomial in* $\sqrt{\langle N \rangle}$ *is detected.*

Input:	f:	$\in R[\overline{a}][\overline{x}]$
	(N, W):	Null and not-null conditions of the specification of σ.
Output:	cd:	The new conditional set.
	f':	The σ-equivalent to f with deciding leading coefficient.
	(N', W'):	A refinement of the specialization (N, W).

```
f' := f;
test := true
N' := N
WHILE test DO
    IF lc(f') ∈ √⟨N'⟩ THEN
        f' := f' − lm(f')
        N' := GB(N' ∪ lc(f'), ≻_ā)
        W' := { w̄^{N'} : w ∈ W }
    ELSE test := false
f' := f̄'^{N'}
cd := FACVAR(lc(f')) − W'
```

Proof: If the specification of σ implies that $\sigma(\mathrm{lc}(f)) = 0$, NEWCOND begins eliminating iteratively the leading term of f. In this process, some polynomial in $\sqrt{\langle N \rangle}$ can be detected. This is used to improve N and a new Gröbner basis N' is recomputed. This is then used to improve also W. When no more leading monomials of f' can be dropped, then f' is reduced with respect to N' dividing by it. (N', W') is now not sufficient to decide if the actual $\mathrm{lc}(f')$ specialize to zero as a consequence of the null condition N'.

The algorithm now computes $cd := \mathrm{FACVAR}(\mathrm{lc}(f')) - W'$. Obviously, if $cd = \phi$ then $\sigma(\mathrm{lc}(f')) \neq 0$ for $\sigma \in \Sigma(N', W')$. If not, $\sigma(\mathrm{lc}(f', \succ_{\overline{x}}))$ vanishes or not depending on the nullity or not of $\sigma(c)$. □

When NEWCOND has concluded and has detected a new condition, in order to continue the specification of σ, DISPGB will have to decide either

(0) The product of all polynomials in cd become zero by specialization ($c = 0$).

(1) All polynomials in cd become different from zero by specialization ($c \neq 0$).

7.2. CONDPGB: Conditional Parametric Gröbner Basis algorithm

At a given point DISPGB needs to use a Buchberger type algorithm that takes into account the specification and intends to determine a specializing Gröbner basis. We denote it CONDPGB (Conditional Parametric Gröbner Basis).

CONDPGB

Input:	B:	The actual specializing basis.
	(N, W):	The actual specification of σ.
Output:	$test$:	if $test = true$ then $\sigma(B')$ is yet the Gröbner basis.
	B':	The new completed specializing basis.
	(N', W'):	The specification (N, W) can be refined in the process.

$test :=$ true; $s := \#B$; $J := \{(i,j) : 1 \le i < j \le s\}$
$B' := B$; $N' := N$; $W' := W$; $t := s$
WHILE $J \neq \phi$ and $test$ DO
  Select $(i,j) \in J$
  $J := J - \{(i,j)\}$
  IF Buchberger's reductibility criterions are false THEN
    $S := \overline{\mathrm{SPOL}(B'[i], B'[j], \succ_{\overline{x}})}^{B'}_{\succ_{\overline{x}}}$
    $S := \overline{S}^{N'}_{\succ_{\overline{x}\overline{a}}}$
    IF $S \neq 0$ THEN
      $(c_{dec}, S, N', W') :=$ NEWCOND(S, N', W')
      IF $c_{dec} = \phi$ THEN
        IF $S \neq 0$ THEN
          $t := t + 1$
          $B' := B' \cup \{S\}$
          $J := J \cup \{(i,t) : 1 \le i < t\}$
      ELSE
        $test := false$
        $B' := B' \cup \{S\}$
IF $test$ THEN
  $B' :=$ REDGB(MINGB$(B', \succ_{\overline{x}}), \succ_{\overline{x}}, \succ_{\overline{a}})$

PROPOSITION 15: *Let (N, W) be the specification of σ and let $B \subset R[\overline{a}][\overline{x}]$ be a set of polynomials for which:*

(i) *$\sigma(B)$ is a basis of $\langle\sigma(F)\rangle$.*

(ii) *$\sigma(\mathrm{lc}(g, \succ_{\overline{x}})) \neq 0$ for $g \in B$ and $\sigma \in \Sigma(N, W)$.*

The algorithm CONGPGB *determines the quadruple $(test, B', N', W')$.*

If test=true, *then $\sigma(B')$ is (except for normalization), the reduced Gröbner basis of $\langle\sigma(F)\rangle$ for $\sigma \in \Sigma(N'W')$.*

If test $=$ false*, then* B' *is an extended set of* B *for which* $\langle \mathrm{lpp}(B, \succ_{\overline{x}})\rangle \subsetneq \langle \mathrm{lpp}(B', \succ_{\overline{x}})\rangle$*, and* B' *contains at least one polynomial for which the actual specification* (N', W') *cannot decide if its leading coefficient specializes to zero or not.*

Proof: CONDPGB is essentially Buchberger's algorithm (PGB0) with the following differences:

(i) Step $S := \overline{S}^{N'}_{\succ_{\overline{xa}}}$.
After computing the usual remainder of the S-polynomial, the result is reduced with respect to N' for the order $\succ_{\overline{xa}}$. Obviously, both polynomials are σ-equivalents, as the difference is a polynomial with coefficients in $\langle N'\rangle$ that specializes to 0.

(ii) Step $(c_{dec}, S, N', W') :=$ NEWCOND(S, N', W').
If $S \neq 0$ then, before continuing PGB0, CONDPGB applies NEWCOND to test if $\mathrm{lc}(S, \succ_{\overline{x}})$ specializes to zero or not. This can reduce S to a σ-equivalent polynomial, say S', that can eventually be 0. Three cases can occur:

a) If $S' = 0$ then there is nothing new to do and the algorithm continues with PGB0.
b) If $S' \neq 0$ and $\sigma(\mathrm{lc}(S, \overline{x})) \neq 0$ then S' is adjoined to the base and PGB0 continues.
c) If $S' \neq 0$ and $c_{dec} \neq \phi$ then S' is adjoined to the base and CONDPGB stops, returning the new base B'. It is important to note that, in that case, at least one new polynomial has been adjoined to the base, whose leading power product cannot be divided by the others in B'. Thus the ideal of leading power products is strictly greater than before.

If CONDPGB never passes through the subcase c), then it continues until a Gröbner basis with non-zero specializing leading coefficients is reached. In this case, CONDPGB determines first the minimal and then the reduced Gröbner base. Then $c_{dec} = \phi$ is returned.

$\square$

8. Control Algorithms

Here we describe the central algorithm DISPGB and two other recursive algorithms BRANCH and NEWVERTEX that manage the control flow.

8.1. DISPGB algorithm

DISPGB is initialized computing the GGE(F). As discussed in section 4 this reduces the number of cases to be discussed, as it reduces the singular variety. Then it calls the general discussion and branching algorithm BRANCH. This is not the only instance where GGE is used. GGE is called again whenever the discussion needs a new branching as a consequence of a new leading coefficient whose specialization cannot be decided without increasing the specification. This can reduce further the singular variety.

DISPGB

Input:	$F \subset R[\overline{x}][\overline{a}]$, and the orders $\succ_{\overline{x}}$ and $\succ_{\overline{a}}$.
Output:	The table $T(v)$ having the tree structure described in section 6 containing at each vertex v the quadruple $T(v) = (c_v, B_v, N_v, W_v)$.

$B := \mathrm{GGE}(F, \succ_{\overline{xa}})$
$T(\phi) := (\phi, B, \phi, \phi)$
$\mathrm{BRANCH}(\phi, B, \phi, \phi)$

We give here the algorithms BRANCH and NEWVERTEX that are recursive procedures calling one another. The control flow is governed by them.

8.2. BRANCH algorithm

Input:	v:	Label of the new vertex.
	B:	Specializing basis at the upper vertex.
	(N, W):	Specification of σ at vertex v (not necessarily canonical).
Output:		Recursive algorithm. It stores the new (B', N', W') (quasi-canonical representation) at the vertex. It manages the control flow.

```
cd := φ
FOR i TO #B WHILE cd = φ DO
    f := B[i]
    (cd, f', N', W') := NEWCOND(f, N, W)
    Substitute the i-th element f of B' by f'
pivot := i − 1
T(v) := (−, B, N', W') (cond is already stored in T(v))
IF cd = φ THEN
    (test, B', N', W') := CONDPGB(B, N', W')
    IF test THEN
        T(v) = (−, B', N', W', terminal vertex)
        STOP (B' is already the Gröbner basis)
    ELSE
        BRANCH(v, B', N', W')
ELSE
    NEWVERTEX(1, v, cd, B', N', W', pivot)
    NEWVERTEX(0, v, cd, B', N', W', pivot)
```

BRANCH begins testing the polynomials in B using NEWCOND. If all have leading coefficients specializing to non-zero ($cd = \phi$), then CONDPGB is called in order to find the specializing Gröbner basis. If this objective is reached, the results are stored at the vertex and the procedure stops. If not, at least some new polynomial with non-decidable leading coefficient has been adjoined by CONDPGB (see proposition 15), and then BRANCH is recursively called in order to continue the discussion.

If some polynomial in B with non-decidable leading coefficient is detected ($cd \neq 0$), then it is taken as pivot and is used to bifurcate the tree (NEWVERTEX) with two new branches of type 0 and 1, increasing the specification.

8.3. NEWVERTEX algorithm

Input:	n	0 or 1. If $n = 1$, the new condition is taken not-null. If $n = 0$ then it is taken null.
	u:	Label of previous vertex. Current vertex becomes $v := (u, n)$.
	cd:	Irreducible factors of the new condition.
	B:	The current base at previous vertex.
	(N, W):	Specification of σ at previous vertex.
	$pivot$:	The index of pivot polynomial of B whose leading coefficient is responsible of the new condition.
Output:		NEWVERTEX creates the new vertex with label v and stores the new values of $(cond_v, B_v, N_v, W_v)$ in $T(v)$. Then it call BRANCH to continue the process.

$v := (u, n)$
$c := \prod cd$
IF $n = 0$ THEN
  $cond := (c = 0)$
  $W' := W$
  $N' := \mathrm{GB}(cd \cup N, \succ_{\overline{a}})$
ELSE $(n = 1)$
  $cond := (c \neq 0)$
  $W' := W \cup cd$
  $N' := N$
  $B' :=$ Substitute every polynomial g in B whose $\mathrm{lpp}(g, \succ_{\overline{x}})$ can be divided by $\mathrm{lpp}(g_{pivot}, \succ_{\overline{x}})$, by the S-polynomial of boths.
$(test, N', W') := \mathrm{CANSPEC}(N', W')$
IF $test$ THEN
  $B' := \mathrm{GGE}(\{\overline{g}^{N'} : g \in B'\} - \{0\}, \succ_{\overline{xa}})$
  $T(v) := (cond, B', N', W')$ (Create and store)
  $\mathrm{BRANCH}(v, B', N', W')$
ELSE STOP

THEOREM 16: *Given $F \subset R[\overline{a}][\overline{x}]$ and the monomial orders $\succ_{\overline{x}}$, $\succ_{\overline{a}}$,* DISPGB *constructs a table T with binary tree structure that contains at each terminal vertex the quadruple*

$$T_v = (cd, B, N, W)$$

where, either the vertex is marked as incompatible conditions, or

(i) *(N, W) is a quasi-canonical specification of a specialization.*

(ii) *For $\sigma \in \Sigma(N, W)$, $\sigma(B)$ is the reduced Gröbner basis of $\sigma(F)$ (except for normalization).*

(iii) *Any specialization corresponds to one and only one of the specifications of the terminal vertices.*

(iv) *The algorithm terminates.*

Moreover, tracing the tree from the root and considering the condition cd *at the vertices, a dichotomic discussion of the cases is provided.*

Proof: (i) Whenever an extended specialization is computed (in NEWVERTEX), the algorithm CANSPEC is called, producing either a quasi-canonical representation of the specification, or detecting incompatible conditions. In that case it marks the vertex as terminal and with incompatible conditions. Thus, only quasi-canonical representations of specializations are used.

(ii) A vertex is marked as terminal only if conditions are incompatible or if CONDPGB has concluded with a basis specializing to the reduced Gröbner basis of $\sigma(F)$ (except for normalization).

(iii) Is an immediate consequence of the dichotomic character of the decisions taken at each vertex ($\sigma(c) = 0$ or $\sigma(c) \neq 0$).

(iv) Is a consequence of the ascending chain condition (ACC) of the ideals in Noetherian rings. When BRANCH is called, the leading coefficients in B are analyzed.

If all these leading coefficients specialize to non-zero in the specification, then CONDPGB is called. Proposition 15 ensures that $\langle \text{lpp}(B, \succ_{ox}) \rangle$ strictly increases in the call.

If the specialization of some leading coefficient is not decidable, then bifurcation takes place calling NEWVERTEX twice. The branch for which the null-condition is assumed, increases the ideal $\langle N \rangle$. The branch for which the not-null condition is assumed increases the number of assumed not-null leading coefficients under specialization, without modifying the base. If in the next decisions none of the above cases occur, then it will arrive a moment where all leading coefficients will be assumed not-null in the specification, and then, CONDPGB will be called, increasing $\langle \text{lpp}(B, \succ_{ox}) \rangle$.

So in any case we have ascending chains of ideals that stabilize. Consequently the algorithm terminates. □

9. Minimal Singular Variety: GENCASE algorithm.

In section 3, we proved the existence of a singular variety determined by PGB algorithm, outside of which, Gröbner bases specialize properly. This case is the unique m-dimensional case. All other cases are contained in a $(m-1)$-dimensional variety V, determined as a union of irreducible varieties. A specification given by $(N = \phi, W)$, where W is provided by PGB is sufficient to ensure a proper specialization of the Gröbner basis. We also mentioned in section 4 that using GGE reduces the computed singular variety, and the same is done by DISPGB.

Let us describe a case C obtained by DISPGB by the 5-tuple $C = [l, p, B, N, W]$, where l is the label, p is the the list of inequalities and equalities in the order that they have been taken to produce the case (i.e. its path), B is the reduced Gröbner basis, and (N, W) the quasi-canonical representation of the specification of the case. Let L be the list of cases obtained by DISPGB and described in the above form.

Except for the trivial problems specializing everywhere, DISPGB always obtains a particular case, say C_0, labeled $[1, \ldots, 1]$, whose specification (N_0, W_0) only contains inequalities, $(N_0 = \phi)$ and for which a proper specialization is assured. The inequalities W_0 are sufficient for making sure that we are outside of the singular variety. But not all the conditions in W_0 are necessary to assure proper specialization. i.e., some other cases can also specialize properly.

A case $C = [l, p, B, N, W]$ in L is said to be *special* if $\mathrm{lpp}(B) \neq \mathrm{lpp}(B_0)$ and *normal* if $\mathrm{lpp}(B) = \mathrm{lpp}(B_0)$. Let L_s denote the set of special cases in L and L_n the set of normal cases in L (including C_0). Using these notations we can define the minimal variety and state a theorem about it.

DEFINITION 17: *Given a set of polynomials $F \subset R[\overline{a}][\overline{x}]$, the* minimal singular variety $V_{\min}$ *is the minimal variety of K^m containing all special cases L_s provided by* DISPGB.

THEOREM 18: *Given $F \subset R[\overline{a}][\overline{x}]$, let L be the list of cases determined by* DISPGB*, C_0 the case corresponding to the label $[1, \ldots, 1]$, $L_n \subseteq L$ the set of normal cases, $L_s \subseteq L$ the set of special cases and*

$$W_{0m} = \mathrm{FACVAR}(\mathrm{lc}(B_0, \succ_{\overline{x}})).$$

Then

(i) *The minimal singular variety $V_{\min} = \bigcup_{w \in W_{\min}} \mathbb{V}(w)$, determined by its irreducible components, verifies: $W_{0m} \subseteq W_{\min} \subseteq W_0$.*

(ii) *For all normal cases, the Gröbner basis specialize properly.*

Proof: (i) Conditions W_0 are sufficient for properly specialization of the algorithms. But some of them can be unnecessary. On the other hand, the conditions in W_{0m} are obviously necessary to have a normal case. Thus the components of $V_{\min}$ are those of W_{0m} plus at most some of the irreducible components in W_0 not in W_{0m}.

(ii) Let G be the reduced Gröbner basis of the ideal $I = \langle F \rangle R(\overline{a})[\overline{x}]$. Under the hypothesis W_0, we have $\sigma_0(G) = B_0$ (except for normalization) as all steps of the computation of G specialize properly to B_0. So

$$\langle \mathrm{lpp}(I) \rangle = \langle \mathrm{lpp}(G) \rangle = \langle \mathrm{lpp}(B_0) \rangle = \langle \mathrm{lpp}(\langle \sigma_0(F) \rangle) \rangle.$$

But by definition of L_n, for all cases in L_n

$$\langle \mathrm{lpp}(B_{L_{ni}}) \rangle = \langle \mathrm{lpp}(B_0) \rangle$$

As $B_{L_{ni}}$ is the reduced Gröbner basis for the case L_{ni}, we have

$$\langle \mathrm{lpp}(B_{L_{ni}}) \rangle = \langle \mathrm{lpp}(\langle \sigma_{L_{ni}}(F) \rangle) \rangle$$

Thus $\langle \mathrm{lpp}(I) \rangle = \langle \mathrm{lpp}(\langle \sigma_{L_{ni}}(F) \rangle) \rangle$, so that, for all cases in L_n, a proper specialization holds.

□

Using part (ii) of theorem 18 we define the *general case* C_g by the specification $(N_g, W_g) = (\phi, W_{\min})$. It contains all the normal cases L_n that are completely outside $V_{\min}$. There can exist cases in L_n that have a non empty intersection with $V_{\min}$. The parts of these cases outside $V_{\min}$ are in the general case, and the parts inside $V_{\min}$ are considered as special cases, even if they specialize properly.

This theorem does not determine completely W_g, even if we observe that in most examples $W_{\min} = W_{0m}$. But the theorem limits very strictly the candidates. Thus GENCASE begins testing if all special cases obtained by DISPGB are inside W_{0m}. If so, the general case substitutes all cases in L_n that are completely outside $V_{\min}$ and those having a non empty intersection with $V_{\min}$, will be restricted to their intersection with $V_{\min}$, and consequently this condition must be added to their specification. If not, GENCASE gives a warning and adds convenient polynomials of $W_0 - W_{0m}$ to $V_{\min}$ to reach the objective.

For linear systems the system determinant gives the general case condition. The corresponding generalized value for general polynomial systems is thus

$$\Delta = \prod_{w \in W_g} w$$

Theorem 18 induces GENCASE algorithm, improving the output of DISPGB.

GENCASE algorithm

Input:	T	The table constructed by DISPGB.
	$\succ_{\overline{x}}$:	$\overline{x}$-termorder.
	$\succ_{\overline{a}}$:	$\overline{a}$-termorder
Output:	L'	The list of cases: $C_g = [\text{'General case'}, p_g, B_g, N_g = \phi, W_g]$, and the normal and special cases inside V_g.

$L :=$ [$C(l) : l \in$ terminal vertices of T]
$C_0 :=$ Select from L the case $C([1..1])$ specified by $(N_0 = \phi, W_0)$
$L_1 := L - C_0$
$lp_0 := \{\mathrm{lpp}(f) : f \in B(C_0)\}$
$L_0 :=$ Select from L_1 the cases having lp_0 as set of $\mathrm{lpp}(f)$
$L_s := L_1 - L_0$
$W_g :=$ FACVAR(lc($B_0, \succ_{\overline{x}}$)
$V_g = \cup_{w \in W_g} \mathbb{V}(w)$
WHILE not all cases in L_s belong to V_g DO
    Warning!
    Add convenient elements of $W_0 - W_g$ to V_g
$C_g :=$ ['General case' , $\{w \neq 0 : w \in W_g\}, B_0, \phi, W_g$]
$L_{ns} :=$ Select all cases in L_1 with non-empty intersection with V_g and
    intersect them with V_g
$L' := C_g \cup L_{ns} \cup L_s$

10. Applications

Many interesting applications can be examined using DISPGB and GENCASE with very satisfactory results. In the following, we illustrate the power of the method with five different examples.

10.1. Linear systems

A particularity of DISPGB is its exceptional behaviour for linear systems. Let us give an example. Consider the following linear system with the variables (x, y, z) and the parameters (a, b, c):

$$\begin{aligned} x + cy + bz + a &= 0 \\ cx + y + az + b &= 0 \\ bx + ay + z + c &= 0 \end{aligned}$$

Applying DISPGB and GENCASE with respect to the monomial orders lex(x, y, z) and lex(a, b, c), the following polynomial appears in the discussion:

$$\Delta \equiv a^2 + b^2 + c^2 - 2\,a\,b\,c - 1,$$

that is nothing else than the value of the determinant of the system. Table 1 resumes the discussion of the cases.
GENCASE summarizes the [1,1] case and the [0,1] cases obtained by DISPGB into the general case $\Delta \neq 0$, as it corresponds to the discussion by determinants. The [1,1] case corresponds to a larger singular variety, namely $\mathbb{V}(\Delta) \cup \mathbb{V}(c^2 - 1)$, and the [0,1] case corresponds to $c^2 - 1 = 0, a - bc \neq 0$. But the final use

Table 1: Discussion of the linear system

Case	Conditions	Gröbner basis
General case	$\Delta \neq 0$	$[\ \Delta\, z + c^3 - b^2 c - a^2 c + 2\, a\, b - c,$ $\Delta\, y + b^3 - c^2 b - a^2 b + 2\, c\, a - b,$ $\Delta\, x + a^3 - c^2 a - b^2 a + 2\, b\, c - a\ ]$
[1,0,1]	$c^2 - 1 \neq 0,$ $\Delta = 0,$ $a\, b - c \neq 0$	$[\ 1\ ]$
[1,0,0]	$c^2 - 1 \neq 0,$ $\Delta = 0,$ $a\, b - c = 0$	$[\ (c^2-1)y + (b\, c - a)z - b^3 + c^2 b,$ $(c^2-1)x + (c\, a - b)z + cb^3 - a + c^2 a - c^3 b\ ]$
[0,0,1]	$c^2 - 1 = 0,$ $a - b\, c = 0,$ $b^2 - 1 \neq 0$	$[\ z + c,$ $x + c\, y\ ]$
[0,0,0]	$c^2 - 1 = 0,$ $a - b\, c = 0,$ $b^2 - 1 = 0$	$[\ x + cy + bz + b\, c\ ]$

of GENCASE, reduces both cases to only one, and the singular variety to the minimal one. All other are special cases corresponding to $\Delta \neq 0$.

One can see that the discussion provided by the algorithm cannot be improved: only strictly different cases corresponding to different sets of leading power products of the reduced Gröbner basis appear. It is hard to obtain such a reduced discussion using only determinants.

10.2. Singular points of a conic

We study the singular points of the conic

$$x^2 + b\, y^2 + 2\, c\, x\, y + 2\, d\, x + 2\, e\, y + f = 0$$

The polynomial system is:

$$[\ x^2 + b\, y^2 + 2\, c\, x\, y + 2\, d\, x + 2\, e\, y + f,\ \ x + c\, y + d,\ \ b\, y + c\, x + e\ ]$$

The following discriminant appears in the discussion

$$\Delta = b\, d^2 - b\, f + c^2\, f - 2\, e\, c\, d + e^2$$

Table 2 describes the discussion. The general case corresponds to non-degenerate conics. Case [0,1] corresponds to two parallel lines. This is an interesting case, as even if the Gröbner basis specializes properly, it is inside the singular variety, corresponding to a special case (degenerate conic without singular points). Finally, cases [1,0] and [0,0,1] correspond to two incident lines, and case [0,0,0] to two coincident lines.

Table 2: Discussion of singular points of a conic

Case	Conditions	Gröbner basis
General case	$\Delta \neq 0$	$[\ 1\ ]$
Special cases	$\Delta = 0$	
$[0,1]$	$cd-e=0, d^2-f\neq 0$	$[\ 1\ ]$
$[1,0]$	$cd-e\neq 0$	$[\ d^2-f+(cd-e)y, -de+cf+(cd-e)x\ ]$
$[0,0,1]$	$cd-e=0, d^2-f=0,$ $b-c^2\neq 0$	$[\ y, x+d\ ]$
$[0,0,0]$	$cd-e=0, d^2-f=0,$ $b-c^2=0$	$[\ x+cy+d\ ]$

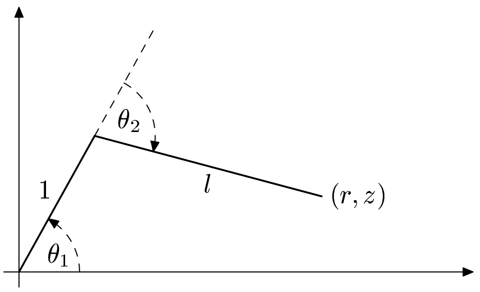

Figure 2: Simple two arms robot.

10.3. The inverse kinematics problem for a simple robot

We consider now the simple plane robot of fig. 2 with two arms of lengths 1 and l respectively.

The inverse kinematics problem is provided by the solution of the following system:

$$\begin{aligned}
&r - c_1 + l(s_1\, s_2 - c_1\, c_2) = 0\\
&z - s_1 - l(s_1\, c_2 + s_2\, c_2) = 0\\
&s_1^2 + c_1^2 - 1 = 0\\
&s_2^2 + c_2^2 - 1 = 0
\end{aligned}$$

where (s_1, c_1, s_2, c_2) are the sines and cosines of the two angles θ_1 and θ_2 defined in fig. 2. In the inverse kinematics problem these are the variables to be determined for the parameter values (r, z, l). Applying DISPGB plus GENCASE to this system, using $\mathrm{lex}(s_1, c_1, s_2, c_2)$ and $\mathrm{lex}(r, z, l)$ orders, the discussion provided in table 3 is obtained.

DISPGB provides $V = \mathbb{V}(l) \cup \mathbb{V}(r) \cup \mathbb{V}(r^2+z^2) \cup \mathbb{V}(z) \cup \mathbb{V}(-l^2+r^2+1)$ as singular variety, plus 4 other cases that are in fact special cases of the general case. GENCASE obtains the minimal singular variety $V_g = \mathbb{V}(l) \cup \mathbb{V}(r^2+z^2)$ in only one general case containing all previous five cases.

Table 3: Discussion of the two arms robot

Case	Conditions	Gröbner basis
General case	$l \neq 0$, $r^2+z^2 \neq 0$,	$[\ 2lc_2+l^2-r^2-z^2+1, l^4-2r^2l^2-2z^2l^2-2l^2$ $+4l^2s_2^2-2r^2+r^4-2z^2+2r^2z^2+z^4+1,$ $rl^2-r-r^3-z^2r+(2r^2+2z^2)c_1-2lzs_2,$ $zl^2-zr^2-z^3-z+(2r^2+2z^2)s_1+2rls_2\ ]$
[1,1,0,1,1]	$l \neq 0,\ r \neq 0,$ $r^2+z^2=0, z\neq 0$ $l^2-1\neq 0$	$[\ 2lc_2+l^2+1,\ 2zls_2+r-rl^2,$ $4r(l^2-1)c_1+l^4+1-4z^2-2l^2,\ -4z^4+z^2+2r^2$ $+4zr^2(l^2-1)s_1+l^4z^2+2r^2l^4-4r^2l^2-2l^2z^2\ ]$
[1,1,0,1,0]	$l \neq 0,\ r \neq 0,$ $r^2+z^2=0, z\neq 0$ $l^2-1=0$	[1]
[1,0,0,1]	$l\neq 0, r=0,$ $z=0, l^2-1\neq 0$	[1]
[1,0,0,0]	$l\neq 0, r=0, z=0,$ $l^2-1=0$	$[\ lc_2+1, s2, c_1^2+s_1^2-1\ ]$
[0,1]	$l=0, r^2+z^2-1\neq 0$	[1]
[0,0]	$l=0, r^2+z^2-1=0$	$[\ {c_2}^2+{s_2}^2-1, c_1-r, s_1-z\ ]$

Moreover, the minimal singular variety has a simple geometrical interpretation: the special cases correspond either to the end of the robot being at the origin, or the length of the second arm being 0.

The special cases [1,1,0,1,1] and [1,1,0,1,0] are only compatible for complex values of the parameters ($r=\pm i\, z\neq 0$) and have no interest for the real configuration.

For the cases [1,0,0,1] and [1,0,0,0] the end of the robot is at the origin. For [1,0,0,1] the system is inconsistent as the length of the second arm is different from ± 1. For [1,0,0,0] the angle θ_1 is free whether θ_2 is π for $l=1$ or 0 for $l=1$, corresponding to the same physical solution.

The cases [0,1] and [0,0] correspond to the degenerate robot of length $l=0$. For [0,1] the system is inconsistent as the position of the end of the robot arm is not at distance 1 of the origin. For [0,0] the angle θ_1 is determined, whether θ_2 is free, corresponding to the degeneration of the configuration.

GENCASE reveals itself to be very useful for this problem.

10.4. Load-flow problem for a 3 nodes Electrical Network

In Montes (1995, 1998) we studied the load-flow problem for electrical networks. The equations are polynomial systems with parameters. The following system concerning a three nodes electrical network is considered:

$$\begin{aligned}
&14-12\,e_2-110\,f_2-2\,e_3-10\,f_3-P_1=0,\\
&2397-2200\,e_2+240\,f_2-200\,e_3+40\,f_3-20\,Q_1=0,\\
&16\,e_2^2-4\,e_2\,e_3-20\,e_2\,f_3+20\,e_3\,f_2+16\,f_2^2-4\,f_2\,f_3-12\,e_2+110\,f_2-P_2=0,\\
&2599\,e_2^2-400\,e_2\,e_3+80\,e_2\,f_3-80\,e_3\,f_2+2599\,f_2^2-400\,f_2\,f_3
\end{aligned}$$

$$- 2200\, e_2 - 240\, f_2 - 20\, Q_2 = 0.$$

Here e_2, f_2, e_3, f_3 are the variables, representing real and imaginary parts of voltages, and P_1, Q_1, P_2, Q_2 are the parameters representing the real and imaginary components of powers. We use the orders $\mathrm{lex}(e_2, f_2, e_3, f_3)$ and $\mathrm{lex}(P_1, Q_1, P_2, Q_2)$. As a result of the application of DISPGB the following polynomials appear:

$$\begin{aligned}
h_1 &= P_1 - 20\\
h_2 &= 20\, Q_1 - 3497,\\
h_3 &= 6999\, P_2 - 800\, Q_2\\
g_1 &= 7786876\, h_1 - 79955\, h_2,\\
g_2 &= 400\, (h_1^2 + h_2^2),\\
g_3 &= 7995500\, h_1 + 1946719\, h_2\\
g_4 &= h_2\, Q_2\, (1814947407168\, h_2 + 20\,778249588225\, h_1) + 1392556035\, h_3^2
\end{aligned}$$

The discussion provided by DISPGB and GENCASE is given in table 4.

Table 4: Discussion of the load-flow problem

Case	Conditions	Leading power products
[1,1] = General case	$g_1 \neq 0,\ g_2 \neq 0$	f_3^2, e_3, f_2, e_2
[1,0,1]	$g_1 \neq 0,\ g_2 = 0,\ g_3\, h_3 \neq 0$	f_3, e_3, f_2, e_2
[1,0,0,1]	$g_1 \neq 0,\ g_2 = 0,\ g_3\, h_3 = 0,\ g_4 \neq 0$	1
[1,0,0,0]	$g_1 \neq 0,\ g_2 = 0,\ g_3\, h_3 = 0,\ g_4 = 0$	e_3, f_2, e_2
[0,1]	$g_1 = 0,\ h_2 \neq 0$	f_3, e_3^2, f_2, e_2
[0,0,1]	$g_1 = 0,\ h_2 = 0,\ h_3 \neq 0$	1
[0,0,0]	$g_1 = 0,\ h_2 = 0, h_3 = 0$	e_3^2, f_2, e_2

In this example, DISPGB gives directly the general case, and no use of GENCASE is needed. We only give the sets of leading power products of the different bases, in order to illustrate how the initial objective has been reached (the bases themselves have large coefficients and would confuse the results). Except for the inconsistency cases [1,0,0,1] and [0,0,1], all other have distinct sets of leading power products and cannot been reduced. Obviously the minimal singular variety in this example is $\mathbb{V}(g_1, g_2)$

10.5. Application to Example 4.3

We give here the complete discussion for the example 4.3 (see table 5) given by DISPGB, where

$$\Delta = (a-1)(3a-7) = 3a^2 - 10a + 7.$$

and G is the basis of the general case given there.

Table 5: Discussion of example 4.3

Case	Conditions	Gröbner basis
[1]	$\Delta \neq 0$	G
[0,1]	$\Delta = 0$, $ba - 8a - 28 + 35b \neq 0$	$[\ 1\ ]$
[0,0]	$\Delta = 0$, $ba - 8a - 28 + 35b = 0$	$[\ -696 + 339b + (-176b + 148)u + 252z$, $-75 + 33b + 63y + (-32b + 67)u$, $-309b + 204 + 252x + (208b - 152)u\ ]$

Acknowledgements

I am very grateful to Volker Weispfenning for his remarks about the convenience of using his algorithm for the load-flow problem Montes (1995), and for the fruitful discussions and suggestions. His comments induced the present work and helped me in the design. I am also grateful to his student Michael Pesch for his patience and interest in the use of his software.

I am also highly indebted to Rafael Sendra for his critical and detailed lecture of the preprint and for his accurate comments that I have carefully incorporated.

References

W.W. Adams, A.K. Boyle, (1992). Some Results on Gröbner Bases over Commutative Rings, *Jour. Symb. Comp.*, **13**, 473–484.

A. Assi, (1994). On Flatness of Generic Projections, *Jour. Symb. Comp.*, **18**, 447–462.

D. Bayer, M. Stillman, (1987). A theorem on refining division orders by the reverse lexicographic order, *Duke J. Math.*, **55**, 321–328.

T. Becker, V. Weispfenning, *Gröbner Bases. A Computational Approach to Commutative Algebra*, (1993). Springer, New-York,

T. Becker, (1994). On Gröbner bases under specialization, *Appl. Algebra Engrg. Comm. Comput.*, **5**, num. 1, 1–8.

D. Cox, J. Little, D. O'Shea, *Ideals, Varieties and Algorithms*, (1992). Springer, New-York,

D. Duval, (1995). Évaluation dynamique et clôture algébrique en Axiom, *J. Pure Appl. Algebra*, **99**, num. 3, 267–295.

R. Gebauer, H.M. Möller, *On an installation of Buchberger's Algorithm*, (1987). Springer, New-York,

P. Gianni, T. Mora, Algebraic Solutions of Systems of Polynomial Equations using Gröbner Bases, *Applied algebra, algebraic algorithms and error-correcting*

Codes, (1987). Menorca, *Lecture Notes in Comut. Sci.*, (1989). **356**, 247–257. Springer,

P. Gianni, Properties of Gröbner Bases under Specializations, *EUROCAL'87*, (1989). J.H. Davenport, **378**, 293–297. Springer LNCS,

P. Gianni, B. Trager, G. Zacharias, (1988). Gröbner Bases and Primary Decomposition of Polynomial Ideals, *Jour. Symb. Comp.*, **6**, 149–167.

H.G. Gräb, (1993). On Lucky Primes, *Jour. Symb. Comp.*, **15**, 199–209.

M. Kalkbrenner, Solving systems of algebraic equations by using Gröbner Bases, *EUROCAL'87, European Conference on Computer Algebra. Lecture Notes in Computer Science*, J.H. Davenport, ed., (1987). **378**, 282–292.

M. Kalkbrenner, (1997). On the stability of Gröbner Bases under specializations, *Jour. Symb. Comp.*, **24**, num. 1, 51–58.

H.M. Möller, (1988). On the construction of Gröbner Bases using syzygies, *Jour. Symb. Comp.*, **6**, 345–359.

A. Montes, (1995). Solving the load flow problem using Gröbner bases, *SIGSAM Bull.*, **29**, 1–13.

A. Montes, (1998). Algebraic solution of the load-flow problem for a 4-nodes electrical network, *Math. and Comp. in Simul.*, **45**, 163–174.

A. Montes, Basic algorithm for Specialization in Gröbner basis, *Actas de EACA-99*, I. Bermejo, ed., (1999). Universidad de La Laguna, Tenerife, 215–228.

W.Y. Sit, (1992). An Algorithm for Solving Parametric Linear Systems, *Jour. Symb. Comp.*, **13**, 353–394.

F. Pauer, (1992). On Lucky Ideals for Gröbner Basis Computations, *Jour. Symb. Comp.*, **14**, 471–482.

V. Weispfenning, (1992). Comprehensive Gröbner Bases, *Jour. Symb. Comp.*, **14**, 1–29.