Numerical Gröbner Bases and Syzygies: an Interval Approach

MARCO BODRATO and ALBERTO ZANONI
{bodrato, zanoni}@posso.dm.unipi.it

Dipartimento di Matematica
Università di Pisa
Via Buonarroti 2 – 56127 Pisa, Italy

Abstract. In Gröbner bases computation a general open question is how to guide calculations coping with numerical coefficients and/or not exact input data. It may happen that, due to error accumulation or insufficient working precision, the result is not one theoretically expects. The basis may have more or less polynomials, a different number of solutions, a zero set with wrong multiplicity, and so on. Augmenting precision we may overcome algorithmic errors, but we don't know in advance how much it should be, and a trial-and-error approach is often the only way. Coping with initial errors is an even more difficult task. In this work the combined use of syzygies and interval arithmetic is proposed as a technique to decide at each critical point of the algorithm what to do.

1 Introduction

For a general reference to Gröbner bases, we refer to [1], [4], [6] and [7]. The Gröbner bases bases computation is basically founded on polynomial head determination. The crucial point is then to test zero equality for leading coefficients, to decide if a monomial is the real head, or it has to be canceled.

If the wrong decision is made, the resulting basis may be completely wrong, not only in the sense that its coefficients may be far away from the expected values, but even that the leading term set (staircase) may be different. This latter aspect is the worst one.

The numerical stability problem for Gröbner bases computation has been studied for some years. Different approaches were proposed:

– Stetter [11], who takes into consideration the size of coefficients.
– Shirayanagy [10], who proves convergence with floating point computation for systems with exact coefficients with increasing precision. Unfortunately, no upper bound is given.
– Migheli [9] uses a combined numerical–modular approach (H).
– Zanoni [14] [15] uses coefficients with two different precisions to study perturbation effects (F2).
– Traverso and Zanoni [13] organise a general view in which they introduce the idea which is developed in the present work.

The weak point which seems common to many treatments is lack of flexibility. Infact, the ZT is usually tuned by some *fixed* parameters (coefficient sizes, number of correct digits, initial precision, etc.), and there is no automatic procedure to detect which are the best values (if they exist) for a system to be correctly treated. Some heuristics may be used, but trial–and–error is still the only general method to analyse all the cases with fixed–behaviour ZT.

What one looks for is an *adaptive* test, which defines a well–determined procedure to decide case by case if the coefficients under testing is zero or not. In other words, we would like that the system itself imposes the conditions that should be satisfied to fulfil, if possible, the ZT.

In [13] the use of syzygies is proposed. Giving relations expressing the "history" (trace) of the computations, they seem to be a good tool to analyse the situation when a ZT has to be applied. In this work we describe our experiments following this idea. A prototype implementation for first experiments is being realized with the C++ `PoSSoLib` library, result of the FRISCO [8] project.

2 Syzygies

Let $\mathbb{K}$ be a field and $\mathcal{F} = \{f_1, \ldots, f_s\} \subset \mathbb{K}[X] = \mathbb{K}[x_1, \ldots, x_n]$ a polynomial list representing the initial system. A *syzygy* for $\mathcal{F}$ is a s–uple of polynomials $\mathcal{H} = \{h_1, \ldots, h_s\} \subset \mathbb{K}[X]$ such that $\mathcal{H} \cdot \mathcal{F} = \sum_1^s h_i \cdot f_i = 0$.

Let $\mathcal{G} = \{g_1, \ldots, g_t\} \subset \mathbb{K}[X]$ be an equivalent system obtained from $\mathcal{F}$ during a Buchberger algorithm application. The idea is to keep track of all the passages that were done to obtain $\mathcal{G}$ from $\mathcal{F}$, as in the extended Euclid algorithm for Bezout's identity. In other words, we look for some polynomials k_{ij} such that

$$g_j(X) = \sum_{i=1}^{s} k_{ij}(X) \cdot f_i(X) \qquad j = 1, ..., t \tag{1}$$

It is possible to obtain syzygies and the k_{ij} by using a simple variant of Buchberger algorithm itself: its extended version (see [5]). Look at $f_1, ..., f_s \in \mathbb{K}[X]$ as vectors $(f_1, 1, 0, ..., 0), ..., (f_s, 0, ..., 0, 1) \in \mathbb{K}[X]^{s+1}$, considered as a $\mathbb{K}[X]$–module, with a term ordering in which comparisons are made on pairs (t, i) – where t is a term and i is an index indicating the position – such that any term in initial position is greater than whatever term in any other position.

3 Multi-component coefficients

The fundamental idea for the `MCoeff` type is to take benefit from different basic types, adding with the additional feature of interval arithmetic. We describe them with an eye to their their actual implementation in the C++ `PoSSoLib` library [8]. A `MCoeff` $m = (n, m_S, m_L, m_i, m_s)$ is an "enriched" representation of a real number. It has a mod p part n, two almost equal floats with different precisions, the short (m_S) and long (m_L) part, and an interval $[m_i, m_s] \ni m_S, m_L$.

Definition 1. *Let $x \in \mathbb{R} \setminus \{0\}$. The natural number $s = \mathrm{size}(x)$ such that*

$$x = a \cdot 2^s \qquad \textit{with } \frac{1}{2} \leq |a| < 1.$$

is the size *of x.*

ZERO TEST : A `MCoeff` m is considered to be zero

- when its short or long part is exactly zero, or
- when it is the result of an addition or subtraction in which the size drops by more than *cancellation*, or
- when $\mathrm{size}(m_S) - \mathrm{size}(m_L) >$ *discrepancy* (indicating that all the meaningful digits disappeared, and only garbage remained).

Definition 2. *A multi component coefficient m is* dangerous *when zero is contained in its interval part,* not dangerous *otherwise. A polynomial with multi component coefficients involved in the Buchberger algorithm is* dangerous *when its leading coefficient is and it is no more head–reducible with respect to the current basis,* not dangerous *otherwise.*

A not dangerous coefficient is surely different from zero.

For our purposes, we choosed *not* to use directly the interval part in the zero detection, as, for example, is done in [10]. There the author, with a slightly different but substantially equivalent notation, considers as zero a coefficient having negative *inf* and positive *sup.* In the paper it is proved that, if sufficiently high precision is given, this leads to a correct rewriting rule, but up to now there is no way to detect an upper bound for such precision, so trial and error has to be used for each example. Moreover, his approach is valuable only when initial data are exact, so that augmenting the working precision makes sense.

4 The technique

We show here how one can proceed to detect and solve critical zero/not zero detection during a classical Buchberger algorithm application. As is well known, there are many degrees of freedom in it, regarding to divisors ordering, critical pairs choice, pre– and post–processing of a S-polynomial, etc.

Numerical Buchberger Algorithm

```
I   Construct the $\overline{\mathcal{F}}$ system with MCoefficients, and start the Buchberger algorithm.
II  If there is a remaining S-polynomial, compute $r$, its complete reduction with respect
    to the current basis, otherwise go to step V.
III If $r = 0$ or its head coefficient $c$ is not dangerous, update the data structures and
    go to step II, otherwise to IV.
IV  Decide if $c$ can really be or is surely different from 0. In the first case delete the
    head in $r$ and go to III, otherwise update data structures and continue from II.
V   Extract the final polynomials $g_i$ from the obtained basis, and output them.
```

Let $\mathbb{K}$ be the field of reals, and $\mathcal{F}$ be given, with the finite precision determining the width of the initial intervals I_i for its coefficients. Any system $\mathcal{F}'$ obtained from $\mathcal{F}$ slightly moving the coefficients in the corresponding I_i is a valid representation of the problem, indistinguishable from $\mathcal{F}$ (we say it is *near* to $\mathcal{F}$). This is the freedom in looking for the most interesting representative, that is the most *instable* one, with more interesting properties, such as positive root multiplicity, etc. The key step in the Numerical Buchberger Algorithm is the ZT in (IV).

5 The zero test

Let $\alpha, \beta, \gamma, \delta, \dots \in \mathbb{N}^n$ be multindexes, $T = \{X^\delta \mid |\delta| = 0, 1, ...\}$ the term basis and $t(r) = X^\rho$ the head term of r. We look at the relations (1) concerning r

$$r(X) = \sum_\gamma r_\gamma X^\gamma = \sum_{i=1}^s k_i(X) \cdot f_i(X)$$

Let K_i^α be the not zero coefficient of k_i in the monomial containing the term X^α, and F_i^β similarly for f_i. Abusing notation, we also introduce new variables $F = \{F_i^\beta \mid \beta \in B_i\}$ and $K = \{K_i^\alpha \mid \alpha \in A_i\}$ corresponding to these coefficients. Thanks to the `MCoeff` approach, interval limits for the unknowns are available. If we perform the computations on the right hand side of this relation and then equate coefficients, we obtain the following system

$$S_{\rho,c} = \left\{ 0 = \sum_{\substack{i=1 \\ \alpha+\beta=\gamma}}^s K_i^\alpha F_i^\beta \qquad \gamma > \rho \quad ; \quad r_\gamma = \sum_{\substack{i=1 \\ \alpha+\beta=\gamma}}^s K_i^\alpha F_i^\beta \qquad \gamma \le \rho \right\} \tag{2}$$

The zero test asks if:

$\boxed{\mathcal{P}_\rho}$: are there values for K_i^α, F_i^β inside their intervals with $r_\rho = c = 0$?

This means to choose an initial system near to $\mathcal{F}$ and a set of syzygy values letting the computation trace still be valid, but such that (iterating the process) we can put to zero as many coefficients as possible. We introduce some vector having appropriate dimension for easiness of reference in the following:

1. i_1, s_1 : whose entries are the extremes of the intervals for F_i^β, for all i, β.
2. i_2, s_2 : similarly for K_i^α, for all possible i and α.

Evidently, $i_l \le s_l$ for $l = 1, 2$. For the way i_1 and s_1 entries are defined, 0 is never contained in the initial intervals for $\mathcal{F}$. This prevents the trivial null solution to be admissible. One could be tempted to write and solve

$$\mathcal{P}_\rho : \begin{cases} \textbf{min} \;\; c = \left| \displaystyle\sum_{\substack{i=1 \\ \alpha+\beta=\rho}}^s K_i^\alpha F_i^\beta \right| & (O) \\ 0 = \displaystyle\sum_{\substack{i=1 \\ \alpha+\beta=\gamma}}^s K_i^\alpha F_i^\beta \qquad \gamma > \rho & (P_1) \\ \left.\begin{array}{l} i_{1,i}^\beta \le F_i^\beta \le s_{1,i}^\beta \\ i_{2,i}^\alpha \le K_i^\alpha \le s_{2,i}^\alpha \end{array}\right\} \text{for all possible } i, \alpha, \beta \end{cases} \tag{3}$$

One can distinguish the two cases $\mathcal{P}_\rho^+$, $\mathcal{P}_\rho^-$ if the approximate value we obtain for c is greater or smaller than zero, respectively.

From now on we will call equation (O) the *objective function* (o.f). The following sections throw some light on the drawbacks of such an approach, introducing a possible alternative.

6 Considerations about system writing

When performing a Gröbner basis computation, many degrees of freedom are present, not only because of the customisable strategies of the algorithm, but also under a strict mathematical point of view. Infact, the two equations

$$f(X) = 0 \qquad\qquad \alpha \cdot f(X) = 0 \tag{4}$$

are perfectly equivalent when $\alpha \neq 0$. This implies that all the obtained polynomials – and therefore their coefficients – are determined up to a constant. In particular, this is also true for the coefficient c, and the decision we must take is in fact a “binary” one: if it is zero or not. We can't *really* consider its absolute value. If we really want to do so, we have some way to remove the extra degree(s) of freedom.

Let $t(f_i) = X^{\delta_i}$ the leading term of f_i: looking for a simplification of the system shape,

1. we make the initial polynomials f_i monic. This gives a double advantage. First, no more initial degrees of freedom. Second, all leading coefficients being now *exactly* equal to 1, the $F_i^{\delta_i}$ variables are now fixed, and there is then no reason at all even to introduce and consider them. We therefore have *once and for all* reduced in a trivial way the research space dimension.
2. if some f_i has exactly one exact coefficient (with trivial interval $[v, v]$) the above step can be viewed as a “shift” of the exactness to the new head coefficient (one). If f_i has more than one exact coefficient and the head one is not, it may seem that we loose precision, but we can easily recover the information keeping some relation(s) for the exact F's. If, e.g., $F_{1,3} = 2$ and $F_{1,5} = 3$ we should put apart the relation $3 \cdot F_{1,3} - 2 \cdot F_{1,5} = 0$ and use it when needed.

For a uniform treating, we will consider the most general formulation, supposing all the coefficients as not precisely known. For what concerns notation, in the following we will

– indicate with $K_O \subset K$ the subset of K variables appearing in the o.f. O. The single variable F_i^β will also be indicated with F_{ij}, where j is the position (starting from 0 for the head term) of the coefficient in the i^{th} polynomial sparse representation. Similarly for K_i^α.

– introduce the specialisation function $r_{\mathcal{F}} : \mathbb{K}[K, F] \to$ `MCoeff` evaluating a polynomial expression to its `MCoeff` value obtained substituting the values for K and F variables and using `MCoeff` arithmetic.

7 Solving the system

We have essentially a quadratic optimisation problem with quadratic restrictions. Theoretically, the problem could be solved by using Lagrange multipliers, considering also the available limitations on F and K variables. We instead propose another way to treat it, without introducing new variables, but directly applying as soon as possible the available information on data.

We can in general proceed following two main directions to nullify the o.f.:

Symbolic : Extract the relations (polynomials in F) such that the o.f. becomes zero. This means the ones that were used by the algorithm that must be fulfilled for the trace to remain valid up to the current point, and the o.f. itself, rewritten only in terms of F variables.

Numeric : Find numerically only some particular values for the initial coefficients satisfying these relations.

The former gives global information, the latter only punctual.

Obtaining exact, symbolic relations in F is extremely useful to set appropriately the initial values for $\mathcal{F}$. It may infact happen that some relations were obtained thanks to the `MCoeff` ZT, that is, forcing explicitly the result to zero. Tuning the F such that these relations are satisfied *exactly*, we refine the computation, introducing less corrections and therefore less garbage. Moreover, each time that new F–relations are determined, they must be still verified in all of the following computations. Coherence must be preserved. In general, however, symbolic analysis is very difficult in practice, and a numerical treatment of $\mathcal{P}_\rho$ becomes necessary. However, we develop here the symbolic aspect, in order to have a general idea of the problems and results that may be encountered.

We propose to consider the system composed by (P_1) equations – see (3) – as living in $\mathbb{K}[F][K]$ instead than $\mathbb{K}[K, F]$, that is, with F variables considered as parameters. The system becomes then a sparse parametric linear one.

$$\mathcal{M}_{\mathcal{F}} \cdot K = \begin{pmatrix} \cdot & \cdot & \cdot \\ \cdot & F_{ij} & \cdot \\ \cdot & \cdot & \cdot \end{pmatrix} \begin{pmatrix} \cdot \\ K_{ij} \\ \cdot \end{pmatrix} = 0$$

We note that $\mathcal{M}_{\mathcal{F}}$ entries are monic monomials in F. For what concerns symbolic manipulations, it is possible to consider the system as temporarily living in $\mathbb{Z}[F][K]$, and pass to $\mathbb{K}$ only when necessary. Moreover, there is no predefined term ordering for F and K variables. This freedom will be extremely fruitful.

Our idea is trying to use (P_1) to eliminate as many K variables as possible, trying to focus mainly on F variables:

1. Using Gauss' method, we can reduce (at least partially, see below) the system, and eventually express some of the K variables, say the K_v subset, in terms of the remaining "free parameters" $K_p = K \setminus K_v$ and (usually complicated) rational functions $R_i^\alpha(F)$.
2. Substituting the obtained expressions for K_O in the objective function, we obtain an alternate expression c' depending essentially only on F variables. Because of this, only the forward phase of Gauss' method is really necessary, to delay as much as possible the computation complexity.

See section 10 for some examples.

8 Ordering, preprocessing and pivot management

We explain how to manage the system in order to reduce it as much as possible.

8.1 Ordering and preprocessing

Before reducing the system, we should choose the ordering for K variables. We note that the truly interesting part of $\overline{\mathcal{M}_{\mathcal{F}}}$ is limited to the lines corresponding to $K_O \cap K_v$. To minimise successive substitutions and reduce the amount of computations, it seems reasonable to

(A) consider K_O variables as the smallest ones.

In our experiments, we noticed that some equation $p(F) \cdot K_{ij} = 0$ may appear. If we know that K_{ij} is not dangerous, we immediately obtain a relation concerning F variables which must be forced to zero. In other words, we are told that the algorithm considered $p(F) = 0$ for this particular trace. This can be a new information usable to tune appropriately the coefficient of $\mathcal{F}$, moving thus to a nearby $\mathcal{F}'$ for which $p(F) = 0$ is *exactly* satisfied. The suggestion is then to

(B) consider K variables with dangerous values as big as possible.

These criteria may be (partially) incompatible. Anyway, (B) is more important, aiming to avoid to express non dangerous variables in terms of dangerous ones.

Definition 3. *A polynomial $p \in \mathbb{Z}[K, F] \subset \mathbb{K}[K, F]$ (an equation $p = 0$) is* mute *if p is a linear binomial in K variables with its two coefficients equal to 1.*

Due to their nothing more than “renaming” function, we can preprocess the system performing the substitutions they indicate (we use here only one index for simplicity)

$$K_i + K_j = 0 \qquad \Longrightarrow \qquad K_i = -K_j$$

deleting some K_i once and for all from the system.

This reduction helps from the numerical point of view, too. Frequently the intervals of the two variables K_i and K_j are widely different, may be one is dangerous and the other is not. The simple equality relation allows us to consider the *intersection* of the intervals, and use it in the forthcoming computations.

8.2 Pivot management

A fundamental question in each of Gauss' reduction steps is the choice of a pivot. We follow here an approach similar to the ”total maximum pivot” adopted in the numerical case, when looking for the pivot with maximum absolute value in the remaining part of the system, in order to minimise error propagation. In our case, working with parametric coefficients, the problem is essentially to decide if a coefficient *can* be used as pivot, that is, if it is surely different from zero, and eventually choose the most simple one, such that symbolic expansions are as limited as possible. Finding a good pivot may influence K variables order.

Definition 4. *Let $p \in \mathbb{K}[F][K]$, and $hc(p) \in \mathbb{K}[F]$ be its leading coefficient. We say that p is* trustable *if $r_{\mathcal{F}}(hc(p))$ is not dangerous,* not trustable *otherwise. p is TH* (term–headed) *if $hc(p)$ is a monomial $c \cdot t_F = c \cdot \prod_{ij} F_{ij} \in \mathbb{Z}[F] \subset \mathbb{K}[F]$.*

As a possible heuristic to find the best not dangerous pivot, we classify the coefficients in order of preference. If there is a constant coefficient, we use it, otherwise we choose according to the length and eventually the degree, taking the minimum. If a K–linear polynomial p obtained during the process is trustable, its head coefficient can be used as pivot, and p itself as a simplifying equation.

Suppose now that k reduction steps were successfully done for $\mathcal{M}_{\mathcal{F}}$ leading to $\mathcal{M}_{\mathcal{F}}^{(k)}$, and no trustable polynomial is found for the current reduction step. We can (indeed, we must) look a bit further. We search for a trustable entry that could play as pivot in the tails of the remaining polynomials. If $m_{ij}^{(k)} \in \mathcal{M}_{\mathcal{F}}^{(k)}$ is such, we exchange the k^{th} and j^{th} columns of the matrix, that is change the positions of the corresponding K variables in the ordering, and proceed using the (reordered) i^{th} polynomial as current simplifying equation.

Non–trustable polynomials may be used as symbolic simplifiers, but only for special cases. The typical case is given by $p_1 = p_{1,1}(F)K_i^{\alpha} + t_1(F,K)$ and $p_2 = p_{2,1}(F)K_i^{\alpha} + t_2(F,K)$,with the division theorem giving $p_{2,1} = p_{1,1} \cdot q + r_{2,1}$ with $q \neq 0$. Then p_1 can be used to (partially) simplify p_2 as follows

$$p_2' = r_{2,1}(F)K_i^{\alpha} + \big(t_2(F,K) - q(F) \cdot t_1(F,K)\big)$$

If the content of an intermediate polynomial p has a factor c_1 such that $r_{\mathcal{F}}(c_1)$ is not dangerous, then c_1 can be deleted (being surely not zero).

All these considerations permit to simplify as much as possible the system, *adding no extra hypotheses on coefficients.*

9 Final shape

Avoiding denominators, the final result is the reduced matrix $\overline{\mathcal{M}}_{\mathcal{F}}$ corresponding to the (partially) reduced system.

$$\overline{\mathcal{M}}_{\mathcal{F}} = \left(\begin{array}{c|c} \mathcal{D}(F) & -\mathcal{N}_1(F) \\ \hline 0 & \mathcal{N}_2(F) \end{array}\right)$$

with $\mathcal{D}(F)$ diagonal matrix with diagonal entries $d_1(F), \dots, d_t(F)$, and $\mathcal{N}_1(F)$, $\mathcal{N}_2(F)$ are $t \times u$, $v \times u$ sub–matrices, respectively, with $d_i, n_{hk}^{(1)}, n_{hk}^{(2)} \in \mathbb{K}[F]$.

Expliciting everything we have

$$\overline{\mathcal{M}}_{\mathcal{F}} \cdot K = $$

$$= \begin{cases} \mathcal{D}(F) \cdot K_v = \mathcal{N}_1(F) \cdot K_p \\ \mathcal{N}_2(F) \cdot K_p = 0 \end{cases} = \left(\begin{array}{ccc|ccc} d_1(F) & & 0 & -n_{11}^{(1)}(F) & \cdots & -n_{1u}^{(1)}(F) \\ & \ddots & & \vdots & & \vdots \\ 0 & & d_t(F) & -n_{t1}^{(1)}(F) & \cdots & -n_{tu}^{(1)}(F) \\ \hline & & & n_{11}^{(2)}(F) & \cdots & n_{1u}^{(2)}(F) \\ & 0 & & \vdots & & \vdots \\ & & & n_{v1}^{(2)}(F) & \cdots & n_{vu}^{(2)}(F) \end{array}\right) \begin{pmatrix} K_v \\ \hline K_p \end{pmatrix}$$

We underline here again that from the practical point of view it is sufficient to compute only the *forward* phase of Gauss' reduction. The backward part may be done when reducing the o.f. modulo the new equations.

Suppose that $\mathcal{N}_2 = 0$, that is no conditions concerning the free parameters are present. In this case it is possible to solve the equations with respect to K_v and obtain

$$K_i^{\alpha} = \sum_{j=1}^{u} \frac{n_{ij}^{(1)}(F)}{d_j(F)} K_{p,j} = \sum_{j=1}^{u} R_{ij}(F) K_{p,j} \tag{5}$$

Focusing on K_O and substituting their expressions in terms of F, K_p, O becomes

$$c = \sum_{\substack{i=1 \\ \alpha+\beta=\rho}}^{s} F_i^{\beta} \cdot \left(\sum_{j=1}^{u} R_{ij}(F) K_{p,j} \right) = \sum_{j=1}^{u} R'_{ij}(F) K_{p,j} = R(F, K_p)$$

with $R'_{ij}(F) = \displaystyle\sum_{\substack{i=1 \\ \alpha+\beta=\rho}}^{s} F_i^{\beta} \cdot R_{ij}(F)$ and $R(F, K_p) = \dfrac{N(F, K_p)}{D(F)}$

It is clear that $D(F) = lcm(d_1(F) \dots d_t(F))$ is surely not dangerous. The case $K_p = \{\overline{K}\}$ is the best one: we have $c = R(F, K_p) = \frac{N(F)}{D(F)} \cdot \overline{K}$, indicating explicitly that and how c depends on initial coefficients, remembering the always present degree of freedom, signed by the surviving $\overline{K}$ variable – see relations (4).

When $\overline{K}$ is not dangerous, the problem is reduced to check if $N(F)$ – which is nothing more than the determinant of a certain sub–matrix of $\mathcal{M}_{\mathcal{F}}$ – has a zero inside the region defined by inequalities concerning F variables. If this happens, then the answer to $\mathcal{P}_\rho$ is "yes", otherwise "no". Instead of determining roots, we may consider

$$T^+ : \begin{cases} \mathbf{min} \quad c = N(F) \\ i_1 \le F \le s_1 \end{cases} \qquad\qquad T^- : \begin{cases} \mathbf{max} \quad c = N(F) \\ i_1 \le F \le s_1 \end{cases}$$

and not computing effectively the optimum value, but only an admissible assignment giving the *opposite* sign – with respect to the one we have obtained in our particular computation – for the new c within interval tolerances. This is a much easier task, because the admissible region is now convex.

– If we find a point π_1 such that $sign(N(\pi_1)) = -sign(N(\pi_0))$, where π_0 is the particular one we have, we can consider the segment connecting them, reducing thus to the univariate case. Because of the zero theorem for continue functions, there surely exist $\pi_R = \pi_0 + t \cdot (\pi_1 - \pi_0)$ solving our problem, with $t \in (0, 1)$. We found that $N(F)$ is a new relation for F variables that must be considered in all the following computations, and found one root of hers.
– If such a F_1 point does not exist – that is, the polynomial does not change sign – then we conclude that c is surely different from zero. We may restrict its interval part excluding zero and continue the computation.

10 Examples

We present here some easy system examples, to show what may arise in practical cases. For simplicity, we indicate only the floating point values v of the coefficients. Intervals are $[v(1 - sign(v) \cdot 2^{\omega}), v(1 + sign(v) \cdot 2^{\omega})]$. For each example we indicate the value of ω and the used term ordering: DRL for DegRevLex, L for Lex.

$\boxed{\boxed{1}}$ $[3, DRL]$ $\quad \mathcal{F}_1 = (z - 1000\ ,\ x^2y + zx + x\ ,\ xy^2 + zy)$

In this simple example we have that the S-poly $s = S(x^2y+zx+x, xy^2+zy) = xy$. So where is the problem ? The algorithm added and subtracted $xy(z-1000)$, and therefore the head coefficient *seems* to be zero. The resulting system is:

$$O = (F_{1,2})K_{1,0} \qquad ; \qquad P_1 = \begin{cases} K_{1,0} + K_{2,0} & = 0 \\ (F_{1,1})K_{1,0} + (F_{2,1})K_{2,0} & = 0 \end{cases}$$

From this we can see that the o.f. can not be zero (the interval for $K_{1,0}$ does not contain zero). Simplifying it, we obtain a relation $F_{1,1} - F_{2,1} = 0$ which was quite easy to detect looking at the initial system. Finding it here, it means that this relation was used to obtain the current polynomial.

$\boxed{\boxed{2}}$ $[3, DRL]$ $\quad \mathcal{F}_2 = (z - 10\ , z^5 + 20x^2y + 21xy\ ,\ z^5 + 21xy^2 + 20y^2)$

Buchberger algorithm simplifies z^5 to $10z^4$ and so on till we obtain 10^5. This process gives us all the $K_{0,j}$ variables, which are completely independent from computation of critical heads. Then the first S-polynomial

$$s = S(20x^2y + 21xy + 10^5, 21xy^2 + 20y^2 + 10^5) = (20 \cdot 20 - 21 \cdot 21)xy^2 + \ldots$$

Since these values are really intervals, it is possible to have that the obtained head coefficient equal equals zero. This is not the relation we see as an o.f., because the algorithm does continue, simplifying xy^2. That's why we have to study a function of the form $(F)K$, where K is dangerous.

More precisely, we have that for 6th critical pair we find a dangerous situation:

$$\begin{array}{l} O = (F_{2,2})K_{2,1} \\ \\ \\ P_1 = \end{array} \begin{cases} K_{0,0} + K_{2,0} = 0 \ (1) & (F_{0,1})K_{0,2} + K_{0,5} = 0 \ (7) \\ K_{0,1} + K_{1,0} = 0 \ (2) & (F_{0,1})K_{0,3} + K_{0,6} = 0 \ (8) \\ (F_{0,1})K_{0,0} + K_{0,2} = 0 \ (3) & (F_{0,1})K_{0,4} + K_{0,7} = 0 \ (9) \\ (F_{0,1})K_{0,1} + K_{0,3} = 0 \ (4) & (F_{0,1})K_{0,5} + K_{0,8} = 0 \ (10) \\ K_{0,4} + K_{2,1} = 0 \ (5) & (F_{0,1})K_{0,6} + K_{0,9} = 0 \ (11) \\ (F_{1,1})K_{1,0} + (F_{2,1})K_{2,0} = 0 \ (6) & (F_{0,1})K_{0,7} + K_{0,10} = 0 \ (12) \\ (F_{1,2})K_{1,0} + (F_{2,2})K_{2,0} + (F_{2,1})K_{2,1} & = 0 \ (13) \end{cases}$$

We numbered P_1 equations for clarity. Equations (1), (2) and (5) are mute. Performing all the simplifications, the really interesting part of the system is

$$P_1' = \begin{cases} (F_{1,1})K_{1,0} + (F_{2,1})K_{2,0} & = 0 \\ (F_{1,2})K_{1,0} + (F_{2,2})K_{2,0} + (F_{2,1})K_{2,1} & = 0 \end{cases}$$

$K_{1,0}$ and $K_{2,0}$ are not dangerous, while $K_{2,1}$ is. We therefore change ordering, putting $K_{2,1}$ as the greatest variable. We finally have

$$\overline{\mathcal{M}}_{\mathcal{F}} \cdot K = \left(\begin{array}{cc|c} F_{1,1}F_{2,1} & 0 & F_{1,1}F_{2,2} - F_{1,2}F_{2,1} \\ 0 & F_{1,1} & F_{2,1} \end{array}\right) \begin{pmatrix} K_{2,1} \\ K_{1,0} \\ K_{2,0} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

and the o.f. becomes $O = \dfrac{N(F)}{D(F)} K_{2,0} = \dfrac{F_{1,1}F_{2,2} - F_{1,2}F_{2,1}}{F_{1,1}F_{2,1}} K_{2,0}$

As expected, $N(F)$ represents a determinant. We translated the uncertainty problem from K space ($K_{2,1}$ is dangerous) to a convex F subspace.

$\boxed{\mathbf{3}}$ $[5, L]$ $\mathcal{F}_3 = \begin{cases} 2xyt - 2x + y^2z - z & = 0 \\ x^3z - 4x^2yt - 4x^2 - 4xy^2z - 4xz - 2y^3t + 10y^2 + 10yt - 2 & = 0 \\ xt^2 - x + 2yzt - 2z & = 0 \\ xz^3 - 4xzt^2 - 4xz - 4yz^2t - 2yt^3 + 10yt - 4z^2 + 10t^2 - 2 & = 0 \end{cases}$

Processing the 28^{th} critical pair we find a dangerous situation:

$$O = (F_{2,3})K_{2,1} + (F_{2,2})K_{2,6} + (F_{3,6})K_{3,0} + (F_{3,3})K_{3,2}$$

$$P_2 = \begin{cases} (F_{0,1})K_{0,2} + (F_{2,1})K_{2,6} + K_{2,7} + (F_{3,2})K_{3,2} + (F_{3,1})K_{3,3} = 0 \\ K_{0,2} + (F_{2,1})K_{2,1} + K_{2,2} + (F_{3,2})K_{3,0} + (F_{3,1})K_{3,1} = 0 \\ (F_{0,2})K_{0,4} + (F_{2,2})K_{2,4} + (F_{3,5})K_{3,0} + (F_{3,4})K_{3,1} = 0 \\ (F_{2,1})K_{2,0} + K_{3,1} = 0 \qquad K_{0,1} + K_{2,1} + (F_{3,1})K_{3,0} = 0 \\ (F_{2,1})K_{2,2} + (F_{3,2})K_{3,1} = 0 \qquad (F_{0,2})K_{0,5} + (F_{3,5})K_{3,1} = 0 \\ K_{0,3} + K_{2,3} = 0 \qquad K_{0,4} + (F_{2,1})K_{2,3} + K_{2,4} = 0 \\ K_{0,5} + (F_{2,1})K_{2,4} = 0 \qquad (F_{0,1})K_{0,0} + K_{2,5} + K_{3,2} = 0 \\ (F_{2,1})K_{2,5} + K_{3,3} = 0 \qquad (F_{0,1})K_{0,1} + K_{2,6} + (F_{3,1})K_{3,2} = 0 \\ (F_{2,1})K_{2,7} + (F_{3,2})K_{3,3} = 0 \qquad K_{0,0} + K_{2,0} + K_{3,0} = 0 \\ (F_{0,1})K_{0,3} + K_{2,8} = 0 \qquad (F_{0,1})K_{0,4} + (F_{2,1})K_{2,8} + K_{2,9} = 0 \\ (F_{0,1})K_{0,5} + (F_{2,1})K_{2,9} = 0 \qquad (F_{0,2})K_{0,1} + (F_{2,2})K_{2,1} + (F_{3,3})K_{3,0} = 0 \\ (F_{0,2})K_{0,0} + (F_{2,2})K_{2,0} = 0 \qquad (F_{0,2})K_{0,2} + (F_{2,2})K_{2,2} + (F_{3,3})K_{3,1} = 0 \\ (F_{2,3})K_{2,0} + (F_{2,2})K_{2,5} = 0 \qquad (F_{0,2})K_{0,3} + (F_{2,2})K_{2,3} + (F_{3,4})K_{3,0} = 0 \end{cases}$$

Which, once simplified, permits to rewrite the o.f. – $K_{2,0}$ is not dangerous – in a surprisingly simple way, with smaller degrees than expected

$$O = \frac{N(F)}{D(F)} K_{2,0} = \frac{-F_{0,2}F_{2,2}F_{3,6} + F_{0,2}F_{2,3}F_{3,3} + F_{2,2}^2F_{3,6} - F_{2,2}F_{2,3}F_{3,3}}{F_{0,2}F_{2,2}} K_{2,0}$$

$\boxed{\mathbf{4}}$ $[3, DRL]$ $\mathcal{F}_4 = (x^2yz + 3x^2y + 2xz \ , \ xy^2z + xy^2 + yz \ , \ x^2y^2 + 1)$

This is a less lucky system, because two K variables remain in the o.f., and numerical analysis is compulsory.

$$O = (F_{2,1})K_{2,1} \quad ; \quad P_4 = \begin{cases} K_{0,0} + K_{1,0} + K_{2,0} & = 0 \\ (F_{0,1})K_{0,0} + (F_{1,1})K_{1,0} & = 0 \\ (F_{0,2})K_{0,0} + K_{0,1} + (F_{1,2})K_{1,0} + K_{1,1} & = 0 \\ (F_{0,1})K_{0,1} + (F_{1,1})K_{1,1} + K_{2,1} & = 0 \\ (F_{0,2})K_{0,1} + (F_{1,2})K_{1,1} + (F_{2,1})K_{2,0} & = 0 \end{cases}$$

The simplified system has only 5 equations with 7 variables: we then obtain

$$O = (F_{0,1} - F_{1,1})K_{1,0} + (F_{0,1})K_{2,0}$$

11 Conclusions

We presented an approach to zero testing in numerical Gröbner bases computations with not exact initial coefficients. The use of syzygies and the ad–hoc introduced *multi–component* coefficients permitted to obtain equations to be fulfilled by the coefficients of the initial polynomials and of the syzygies. If, within interval tolerances, the doubtful coefficient can be zero, a new equation must be taken care of, otherwise intervals can be refined and computations proceed.

We tried to adopt mostly a symbolic approach, making use of numerical information whenever available. A deeper analysis of the numeric behaviour on real–life examples is planned.

References

1. W. W. Adams, P. Loustaunau. *An Introduction to Grobner Bases*, Graduate Studies in Mathematics, Volume 3, AMS, 1994.
2. G. Alefeld, J. Herzberger *Introduction to Interval Computations*, Academic Press, New York (1983)
3. C. Bonini, K.–P. Nischke, C. Traverso *Computing Gröbner bases numerically: some experiments*, proceedings SIMAI (1998)
4. T. Becker, V. Weispfenning *Gröbner Bases: A Computational Approach to Commutative Algebra*, Graduate Studies in Mathematics, Volume 141, Springer Verlag, 1993 (second edition, 1998).
5. M. Caboara, C. Traverso *Efficient Algorithms for ideal operations*, ISSAC 98, ACM Press (1998)
6. D. Cox, J. Little, D. O'Shea. *Ideals, Varieties, and Algorithms*, Springer–Verlag, 1991 (second corrected edition, 1998).
7. D. Cox, J. Little, D. O'Shea *Using algebraic geometry*, Springer-Verlag (1998)
8. FRISCO *A Framework for Integrated Symbolic/Numeric Computation*, ESPRIT Project LTR 21024, 1996–1999, European Union.
9. L. Migheli *Basi di Gröbner e aritmetiche approssimate*, Tesi di Laurea, Universitá di Pisa (In italian) (1999)
10. K. Shirayanagi *Floating Point Gröbner Bases*, Journal of Mathematics and Computers in Simulation 42, pp. 509–528 (1996)
11. H.J. Stetter *Stabilization of polynomial system solving with Gröebner bases*, proceedings ISSAC, pp. 117–124 (1997)
12. C. Traverso *Hilbert functions and the Buchberger algorithm*, Journal of Symbolic Computation 22, n. 4, pp. 355–376 (1996)
13. C. Traverso, A. Zanoni *Numerical Stability and Stabilization of Groebner Basis Computation*, Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation, Universit de Lille, France, ACM Press, Teo Mora ed., pp. 262–269 (2002)
14. A. Zanoni *Numerical stability in Gröbner bases computation*, Proceedings of the 8^{th} Rhine Workshop on Computer Algebra. H. Kredel, W. K. Seidler, ed. pp. 207–216 (2002)
15. A. Zanoni *Numerical Gröbner bases*, PhD thesis, Università di Firenze (2003)