Pythagore's Dilemma, Symbolic-Numeric Computation, and the Border Basis Method

Bernard Mourrain

Pythagore's Dilemma, Symbolic-Numeric Computation, and the Border Basis Method

Bernard Mourrain

Abstract. In this tutorial paper, we first discuss the motivation of doing symbolic-numeric computation, with the aim of developing efficient and certified polynomial solvers. We give a quick overview of fundamental algebraic properties, used to recover the roots of a polynomial system, when we know the multiplicative structure of its quotient algebra. Then, we describe the border basis method, justifying and illustrating the approach on several simple examples. In particular, we show its usefulness in the context of solving polynomial systems, with approximate coefficients. The main results are recalled and we prove a new result on the syzygies, naturally associated with commutation properties. Finally, we describe an algorithm and its implementation for computing such border bases.

1. Introduction

Polynomial system solving is ubiquitous in many applications such as geometric modeling, robotics, computer vision, computational biology, signal processing, ...

In CAGD (Computer-aided Geometric Design) for instance, the objects of a scene or a piece to be built are represented by piecewise-algebraic models (such as spline functions or NURBS), which are able to encode the geometry of an object in a compact way. Indeed this B-spline representations is heavily used in Computed Aided Geometric Design, being now a standard for the representation of shapes. From a practical point of view, critical operations such as computing intersection curves of parameterized surfaces, or analyzing their topology are performed on these geometric models, which require, in fine, to solve polynomial equations.

In robotics or molecular biology (rebuilding of a molecule starting from the matrix of the distances between its atoms obtained by NMR), we have to compute positions of solids, satisfying polynomial equations, deduced from distance constraints. In signal processing, computing high order statistics from signal observations lead to polynomial equations on the parameters that we want to identify.

Typical methods like minimization techniques, Newton-like methods, etc., are often used in these problems, but they do not always offer guarantees. They are local methods which do not provide global information on the set of solutions of the problems. However in many applications, it is important to detect all the possible solutions (usually all real solutions in a given domain).

In this paper, we will give a tutorial presentation of a symbolic-numeric method for solving a polynomial system $f_1 = \cdots = f_m = 0$, which yields such global information on the roots.

Our objective is to devise certified and output-sensitive methods, in order to combine control and efficiency. How can we realize this objective ?

First we have to make precise the context of our computation. The numbers that we can encode on a computer are integers, floating point numbers with fixed size. Such arithmetic is the basis of all symbolic and numeric methods in scientific computation. But there exists also dedicated efficient libraries to compute with integers, rational numbers or floating numbers of arbitrary size. Whereas in numerical computation, one uses fixed size floating point arithmetic, in symbolic computation, one is inclined to use large integer or rational numbers. But we should be aware that these number types, which are the basements of our computation, cannot represent all the numbers that we need in our modeling problems.

This is not a new problem. A long time ago, in the Ancient Greeks works, Geometry, the art of measuring the world, was already closely tied to arithmetic problems. Pythagore developed a complete model of computation, relating geometric constructions to (commensurable) numbers that we call today rational numbers. But Hyppase de Metaponte exhibited publically some weakness of this model (namely that $\sqrt{2}$ is not a rational number). The story says that this act of bravery had terrible consequences for him. Today, we want to deal with models of the real world on a computer. But this machine is able to compute efficiently only with fixed size or floating point numbers and we are facing again Pythagore's dilemma:

- *Should we consider that floating point arithmetic is sufficient to analyze all these problems?*
- *Should we accept to deal systematically with exact but implicit representation?*

The roots of symbolic-numeric computation can, somehow, be found in these questions. On one side, symbolic or exact computation allows to answer in a certified way, to many geometric questions such as counting the number of real roots in a domain, but suffers from a swell of complexity involving huge implicit representations. On the other hand, numerical computation is usually very efficient in approximating locally a given solution, but lacks for a global view on all the possible solutions. The objective of symbolic-numeric computation is to combine efficiency, doing approximate combination and certification, controlling the errors from the symbolic models.

The two main challenges, in this context, are to devise methods

- *which are numerically stable, when we perturb slightly the input coefficients,*
- *and which allow to control and improve the approximation level.*

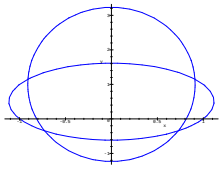
The first point is a prerequisite in the treatment of polynomial systems, for which the coefficients are known with some uncertainty. Such situations appear in many domains, such as CAGD, robotics, signal processing, where the models are not exact, due to measurement errors or to rounding operations, during the geometric processing steps.

Here is an example, which illustrates an instability in the algorithmic approach, which is not an instability of the problem. Consider a system of equations in two variables of the form

$$\begin{cases} p_1 := a\,x_1^2 + b\,x_2^2 + l_1(x_1, x_2) = 0 \\ p_2 := c\,x_1^2 + d\,x_2^2 + l_2(x_1, x_2) = 0 \end{cases}$$

where $a, b, c, d \in \mathbb{C}$ are complex numbers, $a\,d - b\,c \neq 0$ and l_1, l_2 are linear forms.

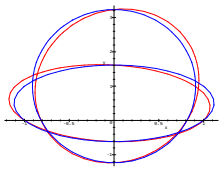
Let us compute a Gröbner basis [6] of these polynomials for a monomial order refining the degree order. The initial ideal is generated by (x_1^2, x_2^2) and the corresponding basis of $\mathfrak{A} = \mathbb{C}[x_1, x_2]/(p_1, p_2)$ is $\{1, x_1, x_2, x_1\,x_2\}$.



The two conics with horizontal and vertical axis and the basis $(1, x_1, x_2, x_1\,x_2)$ of $\mathfrak{A} = \mathbb{C}[x_1, x_2]/(p_1, p_2)$, deduced from a Gröbner basis computation for a monomial ordering refining the degree ordering. Consider now a small perturbation of this system

$$\begin{cases} \tilde{p}_1 = p_1 + \epsilon_1\,x_1\,x_2 \\ \tilde{p}_2 = p_2 + \epsilon_2\,x_1\,x_2, \end{cases}$$

where $\epsilon_1, \epsilon_2 \in \mathbb{C}$ are "small" parameters. The zero-set is also the points of intersection of two conics, which are slightly deformed but the initial ideal is now (x_1^2, x_1x_2, x_2^3) (if $x_1 \succ x_2$).



The basis of $\mathfrak{A} = \mathbb{C}[x_1, x_2]/(\tilde{p}_1, \tilde{p}_2)$, deduced from a Gröbner basis computation for the same monomial ordering, becomes $\{1, x_1, x_2, x_2^2\}$.

We see that, in the result of a small perturbation, basis may "jump" from one set of monomials to another, though the two situations are very closed to each

other from a geometric point of view. Moreover, some of the polynomials of the Gröbner basis have large coefficients, for we have to divide by coefficients of the order ϵ_1, ϵ_2. This computation illustrates an unstable algorithm in a geometrically stable configuration.

Let us illustrate also the need to analyze approximation levels and to study convergence problems on the classical Wilkinson polynomial of degree 20:

$$p = \prod_{i=1}^{20} (x - i).$$

Expanding this product using large integer, and solving p with an appropriate solver (here we use `mpsolve` developed by D. Bini and G. Fiorentino [3]) yields the roots $1, 2, \ldots, 20$. Consider now the polynomial

$$q := p + 10^{-19}\, x^{19}.$$

If we solve with the same solver, we obtain the following approximations of the roots:

$$\left\{\begin{array}{l}
0.99999999999885647030 + \mathbf{i} * 0.780304076295825340952\, 10^{-30}, \\
1.99999999991869592542 - \mathbf{i} * 0.355997149308138207546\, 10^{-25}, \\
3.00000163363722549548 - \mathbf{i} * 0.193933463965337673953\, 10^{-21}, \\
3.99783995916073831012 - \mathbf{i} * 0.149038769767887986466\, 10^{-12} \\
4.92749594405462332247 \pm \mathbf{i} * 0.36215400924406582206, \\
5.46579204715263866632 \pm \mathbf{i} * 1.42160717840964156977, \\
6.02565971634962238568 \pm \mathbf{i} * 2.81462226694663675275, \\
6.67233216337412127217 \pm \mathbf{i} * 4.70785887169693640999, \\
7.57026709806031661287 \pm \mathbf{i} * 7.48404858744277845517, \\
9.34247692903625548411 \pm \mathbf{i} * 12.04455590692478139 66, \\
15.3308398638862630747 \pm \mathbf{i} * 20.56368618219690702 63, \\
44.1662154433454716695 \pm \mathbf{i} * 24.46550599127174407 95,
\end{array}\right.$$

Since the roots of the polynomial p are simple, applying the implicit function theorem, we can show that locally, they are continuous functions of the coefficients of the polynomial. In other words, for any $\epsilon > 0$, there exists a $\delta > 0$ such that a small perturbation of the input coefficients of at most δ induces a perturbation of at most ϵ on the roots. From a geometric point of view, we are in a stable situation.

However, the perturbed example shows that if we want a perturbation of order, say, 10^{-3} on the roots, we should consider approximation of the input coefficients at a precision much less than 10^{-19}. Here, a perturbation of 10^{-19} on the coefficients of p induces a perturbation of size > 20 on some of the roots.

A main challenge in symbolic-numeric methods is thus to analyze how the approximation on the input and output are related, and how to improve efficiently this level of approximation. In this tutorial paper, we are not going to elaborate on this difficult problem, but focus on the stability question in algebraic solvers.

2. From the Structure of $\mathfrak{A}$ to the Roots

In this section, we recall classical results, useful for solving effectively a polynomial system. We illustrate these on small examples.

We next define some notations which we will use hereafter. Let $\mathbb{K}$ be an effective field. The ring of n-variate polynomials over $\mathbb{K}$ will be denoted by R, $R = \mathbb{K}[\mathbf{x}] = \mathbb{K}[x_1, \ldots, x_n]$. We consider n-variate polynomials $f_1, \ldots, f_s \in R$. Our goal is to solve the system of equations $f_1 = 0, \ldots, f_s = 0$ over the algebraic closure $\overline{\mathbb{K}}$ of $\mathbb{K}$. These polynomials generate an ideal of $\mathbb{K}[\mathbf{x}]$ that we call I. From now on, we suppose that I *is zero dimensional* so that ist number of roots over $\overline{\mathbb{K}}$ is finite. The set of roots, with coordinates in the algebraic closure of $\mathbb{K}$, will be denoted by $\mathcal{Z}_{\overline{\mathbb{K}}^n}(I) = \{\zeta_1, \ldots, \zeta_d\}$, with $\zeta_i = (\zeta_{i,1}, \ldots, \zeta_{i,n}) \in \overline{\mathbb{K}}^n$.

2.1. The Quotient Algebra

We denote by $\mathfrak{A} = R/I$ the quotient algebra of R by I, that is the set of classes of polynomials in R modulo the ideal I. The class of an element $p \in R$, is denoted by $\overline{p} \in \mathfrak{A}$. Equality in $\mathfrak{A}$ is denoted by $\equiv$ and we have $a \equiv a'$ iff $a - a' \in I$.

The hypothesis that $\mathcal{Z}(I)$ is finite implies that the $\mathbb{K}$-vector space $\mathfrak{A}$ is of finite dimension (say D) over $\mathbb{K}$ [6, 8]. As we will see, we will transform the resolution of the non-linear system $\mathbf{f} = 0$, into linear algebra problems in the vector space $\mathfrak{A}$, which exploits its algebraic structure. Let us start with an example of computation in the quotient ring $\mathfrak{A}$.

Example 2.1. Let I be the ideal of $R = \mathbb{K}[x_1, x_2]$ generated by

$$\begin{aligned} f_1 &= 13\,x_1^2 + 8\,x_1\,x_2 + 4\,x_2^2 - 8\,x_1 - 8\,x_2 + 2 \\ f_2 &= x_1^2 + x_1\,x_2 - x_1 - \frac{1}{6}\,: \end{aligned}$$

The quotient ring $\mathfrak{A} = \mathbb{K}[x_1, x_2]/I$ is a vector space of dimension 4. A basis of $\mathfrak{A}$ is $1, x_1, x_2, x_1x_2$. We check that we have

$$\begin{aligned} x_1^2 &\equiv x_1^2 - f_2 = -x_1x_2 + x_1 + \frac{1}{6}. \\ x_1^2x_2 &\equiv x_1^2x_2 + \frac{1}{9}\,x_1\,f_1 - (\frac{5}{9} + \frac{13}{9}\,x_1 + \frac{4}{9}\,x_2)\,f_2 = -x_1x_2 + \frac{55}{54}\,x_1 + \frac{2}{27}\,x_2 + \frac{5}{54}. \end{aligned}$$

More generally, any polynomial in $\mathbb{K}[x_1, x_2]$ can be reduced, modulo the polynomials f_1, f_2, to a linear combination of the monomials $1, x_1, x_2, x_1\,x_2$, which as we will see form a basis of $\mathfrak{A}$.

Hereafter, $(\mathbf{x}^\alpha)_{\alpha \in E} = \mathbf{x}^E$ will denote a monomial basis of $\mathfrak{A}$. Any polynomial can be reduced modulo the polynomials $f_1, \ldots, f_s$, to a linear combination of the monomials of the basis $\mathbf{x}^E$ of $\mathfrak{A}$.

2.2. The Dual

An important ingredient of our methods is the dual space $\widehat{R}$ that is, the space of linear forms $\Lambda : R \to \mathbb{K}$. The *evaluation at a point* $\zeta \in \mathbb{K}^n$ is a well-known example of such linear forms: $\mathbf{1}_\zeta : R \to \mathbb{K}$ such that $\forall p \in R$, $\mathbf{1}_\zeta(p) = p(\zeta)$. Another class of linear forms is obtained by using differential operators. Namely, for any $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$, consider the map

$$\begin{array}{rcl} \mathbf{d}^\alpha : R & \to & \mathbb{K} \\ p & \mapsto & \dfrac{1}{\prod_{i=1}^n \alpha_i!} \, (d_{x_1})^{\alpha_1} \cdots (d_{x_n})^{\alpha_n} \, (p)(0), \end{array} \tag{1}$$

where d_{x_i} is the derivative with respect to the variable x_i. For a moment, we assume that $\mathbb{K}$ is of characteristic 0. We denote this linear form $\mathbf{d}^\alpha = (\mathbf{d}_1)^{\alpha_1} \cdots (\mathbf{d}_n)^{\alpha_n}$ and for any $(\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n, (\beta_1, \ldots, \beta_n) \in \mathbb{N}^n$ observe that

$$\mathbf{d}^\alpha \left(\prod_{i=1}^n x_i^{\beta_i} \right) (0) = \left\{ \begin{array}{l} 1 \text{ if } \forall i, \alpha_i = \beta_i, \\ 0 \text{ otherwise.} \end{array} \right.$$

It immediately follows that $(\mathbf{d}^\alpha)_{\alpha \in \mathbb{N}^n}$ is the dual basis of the primal monomial basis $(\mathbf{x}^\alpha)_{\alpha \in \mathbb{N}^n}$. Notice that $(\mathbf{d}^\alpha)_{\alpha \in \mathbb{N}^n}$ can be defined even in characteristic $\neq 0$. Hereafter, we will assume again that $\mathbb{K}$ is a field of arbitrary characteristic. By applying Taylor's expansion formula at 0, we decompose any linear form $\Lambda \in \widehat{R}$ as $\Lambda = \sum_{\alpha \in \mathbb{N}^n} \Lambda(\mathbf{x}^\alpha) \, \mathbf{d}^\alpha$. In particular, the evaluation $\mathbf{1}_\zeta$ is represented by

$$\mathbf{1}_\zeta = \sum_\alpha \zeta^\alpha \mathbf{d}^\alpha.$$

The map $\Lambda \to \sum_{\alpha \in \mathbb{N}^n} \Lambda(\mathbf{x}^\alpha) \, \mathbf{d}^\alpha$ defines a one-to-one correspondence between the set of linear forms Λ and the set $\mathbb{K}[[\mathbf{d}_1, \ldots \mathbf{d}_n]] = \mathbb{K}[[\mathbf{d}]] = \{\sum_{\alpha \in \mathbb{N}^n} \Lambda_\alpha \, \mathbf{d}_1^{\alpha_1} \cdots \mathbf{d}_n^{\alpha_n}\}$ of formal power series (*f.p.s.*) in the variables $\mathbf{d}_1, \ldots, \mathbf{d}_n$.

Hereafter, we will identify $\widehat{R}$ with $\mathbb{K}[[\mathbf{d}_1, \ldots, \mathbf{d}_n]]$. The evaluation at 0 corresponds to the constant 1, under this definition. It will also be denoted $\mathbf{1}_0 = \mathbf{d}^0$.

Let us next examine the structure of the dual space. We can multiply a linear form by a polynomial ($\widehat{R}$ is an R-module) as follows. For any $p \in R$ and $\Lambda \in \widehat{R}$, we define $p \cdot \Lambda$ as the map $p \cdot \Lambda : R \to \mathbb{K}$ such that $\forall q \in R, p \cdot \Lambda(q) = \Lambda(p \, q)$. For any pair of elements $p \in R$ and for $\alpha_i \in \mathbb{N}$, $\alpha_i \geq 1$, we check that we have $\mathbf{d}_i^{\alpha_i}(x_i \, p)(0) = d_i^{\alpha_i - 1} p(0)$. Consequently, for any pair of elements $p \in R, \alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$, where $\alpha_i \neq 0$ for a fixed i, we obtain that

$$x_i \cdot \mathbf{d}^\alpha(p) = \mathbf{d}^\alpha(x_i \, p) = \mathbf{d}_1^{\alpha_1} \cdots \mathbf{d}_{i-1}^{\alpha_{i-1}} \mathbf{d}_i^{\alpha_i - 1} \mathbf{d}_{i+1}^{\alpha_{i+1}} \cdots \mathbf{d}_n^{\alpha_n}(p),$$

that is, x_i acts as the *inverse* of $\mathbf{d}_i$ in $\mathbb{K}[[\mathbf{d}]]$. This is the reason why in the literature such a representation is referred to as the *inverse system* (see, for instance, [13], [16], [9]).

2.3. The Multiplication Operators

The first operator that comes naturally in the study of $\mathfrak{A}$ is the operator of multiplication by an element of $a \in \mathfrak{A}$. For any element $a \in \mathfrak{A}$, we define the map

$$\begin{array}{rccc} M_a : \mathfrak{A} & \to & \mathfrak{A} \\ b & \mapsto & a\,b. \end{array}$$

We will also consider the transposed operator

$$\begin{array}{rccc} M_a^t : \widehat{\mathfrak{A}} & \to & \widehat{\mathfrak{A}} \\ \Lambda & \mapsto & M_a^{\mathbf{t}}(\Lambda) = \Lambda \circ M_a. \end{array}$$

The matrix associated to this operator in the dual basis of a basis of $\mathfrak{A}$ is the transposed of the matrix of M_a in this basis.

Example 2.2. Let us compute the matrix of multiplication by x_1 in the basis $(1, x_1, x_2, x_1x_2)$ of $\mathfrak{A} = \mathbb{K}[x_1, x_2]/(f_1, f_2)$, where f_1, f_2 are the polynomials of example 2.1. We multiply these monomials by x_1 and reduce them to a normal form. According to the computations of example 2.1, we have:

$$\begin{array}{rcl} 1 \times x_1 & \equiv & x_1, \\ x_1 \times x_1 & \equiv & -x_1x_2 + x_1 + \dfrac{1}{6}, \\ x_2 \times x_1 & \equiv & x_1x_2, \\ x_1x_2 \times x_1 & \equiv & -x_1x_2 + \dfrac{55}{54}\,x_1 + \dfrac{2}{27}\,x_2 + \dfrac{5}{54}. \end{array}$$

so that we have:

$$M_1 = \begin{bmatrix} 0 & \frac{1}{6} & 0 & \frac{5}{54} \\ 1 & 1 & 0 & \frac{55}{54} \\ 0 & 0 & 0 & \frac{2}{27} \\ 0 & -1 & 1 & -1 \end{bmatrix}.$$

The multiplication map can be computed, when a normal form algorithm is available. In the next section, we will describe how to compute such a normal form, in a symbolic-numeric setting.

The algebraic solver approach is based on the following fundamental theorem (see [2], [14], [23]):

Theorem 2.3. *Assume that* $\mathcal{Z}_{\overline{\mathbb{K}}^n}(I) = \{\zeta_1, \ldots, \zeta_d\}$.

1. *The eigenvalues of the linear operator* M_a *(resp.* $M_a^{\mathbf{t}}$*) are* $\{a(\zeta_1), \ldots, a(\zeta_d)\}$.
2. *The common eigenvectors of* $(M_a^{\mathbf{t}})_{a\in\mathfrak{A}}$ *are (up to a scalar)* $\mathbf{1}_{\zeta_1}, \ldots, \mathbf{1}_{\zeta_d}$.

Notice that if $(\mathbf{x}^\alpha)_{\alpha\in E}$ is a monomial basis of $\mathfrak{A}$, then the coordinates of the evaluation $\mathbf{1}_{\zeta_i}$ in the dual basis of $(\mathbf{x}^\alpha)_{\alpha\in E}$ are $(\zeta_i^\alpha)_{\alpha\in E}$ where $\zeta^\alpha = \mathbf{1}_\zeta(\mathbf{x}^\alpha)$. Thus, if the basis $(\mathbf{x}^\alpha)_{\alpha\in E}$ contains $1, x_1, \ldots, x_n$ (which is often the case), the coordinates $[v_\alpha]_{\alpha\in E}$ (in the dual basis) of the eigenvectors of $M_a^{\mathbf{t}}$ yield all the coordinates of the root: $\zeta = [\frac{v_{x_1}}{v_1}, \ldots, \frac{v_{x_n}}{v_1}]$. This leads to the following algorithm:

Algorithm 2.4. SOLVING IN THE CASE OF SIMPLE ROOTS.

Let $a \in R$ and $\mathtt{M}_a$ be the matrix of multiplication in a basis $\mathbf{x}^E = (1, x_1, \ldots, x_n, \ldots)$ of $\mathfrak{A}$.

1. Compute the eigenvectors $\Lambda = [\Lambda_1, \Lambda_{x_1}, \ldots, \Lambda_{x_n}, \ldots]$ of $\mathtt{M}_a^{\mathtt{t}}$.
2. For each eigenvector Λ with $\Lambda_1 \neq 0$, compute and output $\zeta = \left(\frac{\Lambda_{x_1}}{\Lambda_1}, \ldots, \frac{\Lambda_{x_n}}{\Lambda_1}\right)$.

The set of output points ζ contains the set of simple roots of $\mathcal{Z}(I)$, since for such roots the eigenspace is one-dimensional. But as we will see on the next example, it can also yield in some cases[1] the multiple roots :

Example 2.1 **continued.** We compute the eigenvalues, their multiplicity, and the corresponding normalized eigenvector of the transposed of the matrix of multiplication by x_1:

Eigenvector	Eigenvalue	Multiplicity
$[1, -\frac{1}{3}, \frac{5}{6}, -\frac{5}{18}]$	$-\frac{1}{3}$	2
$[1, \frac{1}{3}, \frac{7}{6}, \frac{7}{18}]$	$\frac{1}{3}$	2

As the basis chosen for the computation is $(1, x_1, x_2, x_1x_2)$, the previous theorem tells us that the solutions of the system can be read off, from the 2^{nd} and the 3^{rd} coordinates of the normalized eigenvectors: $\zeta_1 = (-\frac{1}{3}, \frac{5}{6})$ and $\zeta_2 = (\frac{1}{3}, \frac{7}{6})$. Moreover, the 4^{th} coordinate of these vectors is the product of the 2^{nd} by the 3^{rd} coordinates.

In order to compute exactly the set of roots, counted with their multiplicity, we employ the following theorem. It is based on the fact that commuting matrices share common eigenspaces.

Theorem 2.5. [14, 16, 5] *There exists a basis of* $\mathfrak{A}$ *such that* $\forall a \in R$, *the matrix* M_a *is, in this basis, of the form*

$$\mathtt{M}_a = \begin{bmatrix} \mathtt{N}_a^1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathtt{N}_a^d \end{bmatrix} \text{ with } \mathtt{N}_a^i = \begin{bmatrix} a(\zeta_i) & & \star \\ & \ddots & \\ \mathbf{0} & & a(\zeta_i) \end{bmatrix}.$$

Here again, it leads to an algorithm:

[1] depending on the type of multiplicity

Algorithm 2.6. SOLVING BY SIMULTANEOUS TRIANGULATION.

INPUT: `The matrices of multiplication` $\mathtt{M}_{x_i}$ $(i = 1, \ldots, n)$ `in a basis of` $\mathfrak{A}$.

1. `Compute a (Schur) decomposition` P `such that all matrices` $\mathtt{T}_{x_i} = \mathtt{P}\mathtt{M}_{x_i}\mathtt{P}^{-1}$ $(i = 1, \ldots, n)$ `are upper-triangular.`
2. `Compute the diagonal vectors` $\mathbf{t}_i = (t^1_{i,i}, \ldots, t^n_{i,i})$ `of the triangular matrices` $\mathtt{T}_{x_i} = (t^i_{i,k})$ `(for` $i = 1, \ldots, D$`).`

OUTPUT: $\mathbf{t}_i, i = 0, \ldots, D$ `the solutions of the input polynomial system, repeated with their multiplicity.`

The first step is performed by computing a ordered Schur decomposition of M_l (where l is a generic linear form) which yields a matrix P of change of basis. Next, we compute the matrices $\mathtt{T}_{x_i} = \mathtt{P}\mathtt{M}_{x_i}\mathtt{P}^{-1}$ $(i = 1, \ldots, n)$ which are triangular, since they commute with M_l. The decomposition of the multiplication operators in theorem 2.5 is in fact induced by a decomposition of the algebra

$$\mathfrak{A} = \mathfrak{A}_1 \oplus \cdots \oplus \mathfrak{A}_d,$$

where $\mathfrak{A}_i$ is the local algebra associated with the root ζ_i. More precisely, there exists elements $\mathbf{e}_1, \ldots, \mathbf{e}_d \in \mathfrak{A}$, such that $i, j = 1, \ldots, d$

$$\begin{cases} \mathbf{e}_i^2 \equiv \mathbf{e}_i, \\ \mathbf{e}_i \mathbf{e}_j \equiv 0, i \neq j \\ \mathbf{e}_1 + \cdots + \mathbf{e}_d \equiv 1 \end{cases}$$

These polynomials, which generalize the univariate Lagrange polynomials, are called the fundamental idempotents of $\mathfrak{A}$. They are such that $\mathfrak{A}_i = \mathbf{e}_i\,\mathfrak{A}$ and $\mathbf{e}_i(\zeta_j) = 1$ if $i = j$ and 0 otherwise. The dimension of the $\mathbb{K}$-vector space $\mathfrak{A}_i$ is the *multiplicity* μ_{ζ_i} of ζ_i. See [26, 14, 8].

2.4. An Exact Representation of the Roots

In some problems, it is important to have an exact representation of the roots, with which we can effectively compute. Hereafter, we recall how to represent them as the image, by a rational map, of the roots of a univariate polynomial. The Chow form of $\mathfrak{A}$ is the homogeneous polynomial in $\mathbf{u} = (u_0, \ldots, u_n)$ of degree D, defined by:

$$\mathcal{C}_I(\mathbf{u}) = \det(u_0 + u_1\,\mathtt{M}_{x_1} + \cdots + u_n\,\mathtt{M}_{x_n}).$$

According to theorem 2.5, we have

Theorem 2.7. *The Chow form of* $\mathfrak{A}$ *is*

$$\mathcal{C}_I(\mathbf{u}) = \prod_{\zeta \in \mathcal{Z}(I)} (u_0 + u_1\zeta_1 + \cdots + u_n\zeta_n)^{\mu_\zeta}.$$

where μ_ζ *is the multiplicity of* ζ.

Example 2.1 **continued.** We compute the Chow form of the variety $I = (f_1, f_2)$, using the matrices of multiplication by x_1 and x_2, computed previously.

$$\det(u_0 + u_1 M_1 + u_2 M_2) = \left(u_0 + \frac{1}{3}u_1 + \frac{7}{6}u_2\right)^2 \left(u_0 - \frac{1}{3}u_1 + \frac{5}{6}u_2\right)^2$$

We check that it is a product of linear forms, whose coefficients yield the roots $\zeta_1 = (-\frac{1}{3}, \frac{5}{6})$ and $\zeta_2 = (\frac{1}{3}, \frac{7}{6})$. The exponents yield the multiplicity of the roots (here 2).

From this Chow form, it is possible to deduce a rational representation of the points of $\mathcal{Z}(I)$, as describe in the following algorithm. See [21, 1, 22, 12] for more details.

Algorithm 2.8. Univariate Rational Representation

Input: a *multiple* $\Delta(\mathbf{u})$ of the Chow form $I \subset R$.

1. Compute the square free part of $\Delta(u)$.
2. Choose a generic $\mathbf{t} \in \mathbb{K}^{n+1}$ and compute the first terms of
$$d(\mathbf{t} + \mathbf{u}) = d_0(u_0) + u_1\, d_1(u_0) + \cdots + u_n\, d_n(u_0) + \cdots$$
3. Compute the redundant rational representation $\zeta_1 = \frac{d_1(u_0)}{d_0'(u_0)}$, ..., $\zeta_n = \frac{d_n(u_0)}{d_0'(u_0)}$, $d_0(u_0) = 0$.
4. Factor $d_0(u_0)$, keep the good prime factors and output the corresponding simplified rational univariate representations of the roots $\mathcal{Z}(I)$.

This result describes the coordinates of the roots of the polynomial system $f_1 = 0, \ldots, f_s = 0$, as the image by an explicit rational map of some of the roots of $d_0(u_0)$. Since we start with a multiple of $\Delta(\mathbf{u})$, in order to have exactly the roots we can remove the redundant factor of d_0, by substituting the rational representation back into the equations $f_1, \ldots, f_n$.

This completes our description of the quotient algebra $\mathfrak{A} = \mathbb{K}[\mathbf{x}]/I$, and shows how knowing its multiplicative structure yields a representation of the roots of I. We are now going to consider how to compute effectively the multiplication structure of $\mathfrak{A}$. For this purpose, we will consider normal form methods, which given a polynomial $p \in \mathbb{K}[\mathbf{x}]$, compute a canonical element for its class in $\mathfrak{A} = \mathbb{K}[\mathbf{x}]/I$ (or modulo I).

3. Border Basis Method

Gröbner basis computation yields, by reduction, the normal form of any element modulo I. As shown in the introduction, however by computing a Gröbner basis on a perturbed system, we obtain a completely different representation of $\mathfrak{A}$. In this section, we are going to detail an alternative approach, known as the border

basis method, or generalized normal form method [15, 18, 25, 24, 19, 11, 10], and adapted to symbolic-numeric computation.

Example 3.1. Consider the system:

$$f_1 := x_1^2 + x_2^2 - x_1 + x_2 - 2; f_2 := x_1^2 - x_2^2 + 2\,x_2 - 3;$$

The computation of a Gröbner basis for the degree-lexicographic ordering yields

$$2\,{x_2}^2 - x_1 - x_2 + 1, 2\,{x_1}^2 - x_1 + 3\,x_2 - 5.$$

The leading monomials are x_1^2, x_2^2 and the corresponding monomial basis of $\mathfrak{A}$ is $\{1, x_1, x_2, x_1\,x_2\}$ Consider now a small perturbation:

$$f_1, f_2 + 10^{-7}\,x_1\,x_2$$

We obtain:

$$\begin{array}{rcl}
2\,{x_2}^2 & \equiv & +x_1 + x_2 - 1 + 0.0000001\,x_1x_2, \\
{x_1}^2 & \equiv & -{x_2}^2 + x_1 - x_2 + 2, \\
x_1\,x_2 & \equiv & 10000000.99999999999999500000000000000125\,{x_2}^2 \\
 & & -5000000.25000001249999937499996875000156250000781250\,x_1 \\
 & & -5000000.75000003749999312499990625001718750023437 50\,x_2 \\
 & & +5000000.25000006249999937499984375000156250039062 50
\end{array}$$

The leading monomials are now $x_1\,x_2, x_1^2, x_2^3$ and the corresponding basis of $\mathfrak{A}$ is $B = \{1, x_1, x_2, x_2^2\}$.

Notice however that $\{1, x_1, x_2, x_1x_2\}$ is still a basis of $\mathfrak{A}$ and that we have the following equivalences in $\mathfrak{A}$:

$$\begin{array}{rcl}
x_1^2 & \equiv & -0.00000005\,x_1x_2 + \frac{1}{2}\,x_1 - \frac{3}{2}\,x_2 + \frac{5}{2} \\
x_2^2 & \equiv & +0.00000005\,x_1x_2 + \frac{1}{2}\,x_1 + \frac{1}{2}\,x_2 - \frac{1}{2} \\
x_2{x_1}^2 & \equiv & 0.49999999\,x_1x_2 - 0.74999998\,x_1 + 1.75000003\,x_2 + 0.74999994, \\
x_1{x_2}^2 & \equiv & 0.49999999\,x_1x_2 - 0.25000004\,x_1 - 0.74999991\,x_2 + 1.25000004]
\end{array}$$

In this representation, we observe that the perturbation on the latter rewriting rules is of the same order as on the input coefficients. We also notice that this set of linear relations between the monomials on the border of B and B yields directly the matrices of multiplication by x_1, x_2 in $\mathfrak{A}$, since we are able to compute the product of any element of the basis $B = \{1, x_1, x_2, x_1x_2\}$ by x_1 or x_2.

The key observations, that we deduce from this example, are the following:

- *If we are in a geometrically stable situation, a basis of the quotient algebra by a polynomial system will remain a basis for a small perturbation of this system.*
- *To compute the structure of* $\mathfrak{A}$*, we only have to known to which combination of the basis monomials, the monomials on the* border *are equivalent to.*

We are going to detail now, how these remarks can be turned into an algorithm. For more details on the method that we describe, see [15], [17], [25], [18], [19].

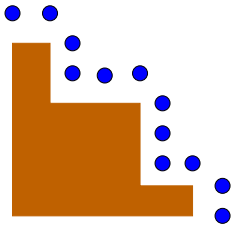
3.1. Border Monomials

The support $\mathrm{supp}(p)$ of a polynomial $p \in \mathbb{K}[\mathbf{x}]$ is the set of monomials appearing with non-zero coefficients in p. Given a set S of elements of $\mathbb{K}[\mathbf{x}]$, we denote by $\langle S \rangle$ the $\mathbb{K}$-vector space spanned by the elements of S. We denote the set of all the monomials in the variables $\mathbf{x} = (x_1, \ldots, x_n)$ by $\mathcal{M}$. The usual degree of a monomial $m \in \mathcal{M}$, will be denoted by $|m|$.

A set of monomials B is said to be *connected to* 1 if and only if, for every monomial m in B, there exists a finite sequence of variables $(x_{i_j})_{j\in[1,l]}$ such that $1 \in B$, $\Pi_{j=1\ldots l'} x_{i_j} \in B$, $\forall l' \in [1, l]$ and $\Pi_{j\in[1,l]}\, x_{i_j} = m$. A set of monomials B is said to be *stable by division* if $m = m'm''$ in B implies that m' and m'' are in B. Notice that if B is stable by division, it is connected to 1.

For any subset S of R, we denote by S^+ the set $S^+ = S \cup x_1\, S \cup \cdots \cup x_n\, S$, $\partial S = S^+ \backslash S$. If S is a set of monomials, ∂S with be called the set of *border monomials* of S.

In the following example, with two variables, B is the set of monomials under the staircase. It is stable by division. The set ∂B is formed by the dotted monomials.



In order to compute the structure of the quotient algebra $\mathfrak{A}$, we have

- to compute a (monomial) basis B of $\mathfrak{A}$,
- and for each border monomial m in ∂B, to compute the linear combination of monomials in B equal to it modulo the ideal I.

From this construction, we deduce directly the tables of multiplication by the variables and thus the roots, as described in section 2.

3.2. Reduction

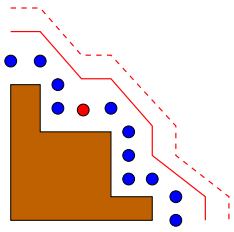
Suppose that we have a monomial set B containing 1, which might be a basis of $\mathfrak{A}$, and for each monomial $m \in \partial B$ we have computed $b_m \in \langle B \rangle$ such that $\rho_m = m - b_m \in I$. Let $F = (\rho_m)_{\partial B}$. F is called *a rewriting family* for B, if it has the following property: $\forall f, f' \in F$,

- f has exactly **one** monomial that we denoted by $\gamma(f)$ (also called the *leading monomial* of f) in ∂B,
- $\mathrm{supp}(f) \subset B^+$, $\mathrm{supp}(f - \gamma(f)) \subset B$,
- if $\gamma(f) = \gamma(f')$ then $f = f'$,

For $B = \{1, x, y, xy\}$, the set of polynomials $F = \{x^2 - 1, y^2 - x_1, x^2y - x_1, y^2x - y\}$ is a reducing family of degree 3.

This notion of rewriting family can be refined by the degree, as described in [19].

Once we have a rewriting family, we have a way to project any monomial onto $\langle B \rangle$ modulo the ideal I. For this purpose, we introduce the notion of B-index: let $B^{[i]} = (\dots(B^+)\dots)^+$ be the application of i times the operator $^+$ on B. Since B contains 1, for any monomial $m \in \mathcal{M}$, there exists k such that $m \in B^{[k]}$. We say that a monomial m is of B-index k if $m \in B^{[k]} - B^{[k-1]}$, and we denote it by $\delta_B(m)$. By convention, $B^{[0]}$ is B, which is the set of monomials of B-index 0. We denote by $B^{[<k]} = B^{[0]} \cup \dots \cup B^{[k-1]}$. We represent here the monomials of B-index 1 (the points), those of B-index 2 (first polygon) and 3 (dashed polygon):



Using the polynomials in F, any monomial in $B^{[1]} = \partial B$ can be rewritten in $\langle B \rangle$. By induction, we prove that any monomial $B^{[k]}$ can be rewritten in $\langle B^{[<k]} \rangle$ modulo the ideal generated by F. Thus iterating this reduction, any polynomial of $\mathbb{K}[\mathbf{x}]$ can be projected modulo (F) onto $\langle B \rangle$. For details and algorithms, see [15], [19]. Here also, we can refine this reduction, with respect to the degree or any graduation.

Let us denote by R_F such a linear projection from $\mathbb{K}[\mathbf{x}]$ to $\langle B \rangle$, deduced from the rewriting family F for a set B connected to 1. By definition, we have the following properties:

$$\forall m \in B, R_F(m) = m,$$
$$\forall m \in \partial B, R_F(m) = m - \rho_m,$$

where $\rho_m \in F$ is the unique member of F, which monomial $\in \partial B$ is m. More generally, for $p \in \langle B^+ \rangle$, we have

$$R_F(p) = p - \sum_{m \in \mathrm{supp}(p) \cap \partial B} \lambda_m \rho_m,$$

where λ_m is the coefficient of m in p.

This allows us to define the following operator:

$$\begin{aligned} M_i : \langle B \rangle &\to \langle B \rangle \\ b &\mapsto R_F(x_i b). \end{aligned}$$

It has the following properties: $\forall m \in B$,

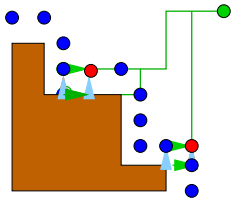
- if $x_i\, m \in B$, then $M_i(m) = x_i\, m$,
- otherwise $m' := x_i\, m \in \partial B$ is a border monomial, so that $M_i(m) = x_i m - \rho_{x_i\, m}$.

3.3. Normal Form Criterion

A question, which at this point is not solved, is how to check that B is really a basis of $\mathfrak{A}$. In other words, how can we check

- that $\mathbb{K}[\mathbf{x}] = \langle B \rangle \oplus (F)$, or equivalently
- that R_F is a normal form, modulo the ideal (F), or equivalently
- that the reduction by the rewriting family F is unique.

Let us illustrate the situation on a bivariate example:



In this picture, the isolated monomial can, a priori, be reduced in many different ways to a linear combination of elements in B. Each path down from this monomial to a monomial of B yields a distinct way to reduce it. In such a path, a horizontal step corresponds to the multiplication by x_1 and a vertical step to the multiplication by x_2. In order to ensure that the reduction is unique, we thus have to check that the multiplication by x_1 first, then the reduction R_F, then the multiplication by x_2 and again the reduction by R_F yield the same result than when we exchange the role of x_1 and x_2. In other words, we have to check that

$$M_1 \circ M_2 = M_2 \circ M_1$$

on B. It turns out that such commutation condition is sufficient to guaranty that we have a normal form, assuming that the basis B is connected to 1:

Theorem 3.2. [15] *Let $B \subset \mathcal{M}$ connected to 1. Let $R_F : B^+ \to B$ be a projection from B^+ to B, with kernel F and let $I = (F)$ be the ideal generated by F. Let*

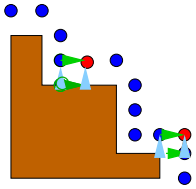
$$\begin{array}{rcl} M_i : B & \to & B \\ b & \mapsto & R_F(x_i b). \end{array}$$

Then, the two properties are equivalent:

1. *For all $1 \leq i, j \leq n$, $M_i \circ M_j = M_j \circ M_i$.*
2. *$\mathbb{K}[\mathbf{x}] = \langle B \rangle \oplus I$.*

If this holds, the reduction induced by R_F is a normal form modulo I onto $\langle B \rangle$.

Effectively, it means that we have to check the commutation only for the monomials on the border of B, as it is discussed below and illustrated in this figure:



To simplify the presentation, let us assume now that B is *stable by division*. Let $m \in B$ and consider two variables x_i, x_j, with $i \neq j$.

- If $m' := x_i\, x_j\, m \in B$, then since B is stable by division, $x_i\, m, x_j\, m \in B$ and we have $M_i \circ M_j(m) = x_i\, x_j\, m = M_j \circ M_i(m)$.
- If $x_i\, m \in B$ but $x_j\, m \notin B$, we have

$$\begin{aligned} M_i(m) &= x_i\, m, \\ M_j(m) &= x_j\, m - \phi(x_j\, m), \\ M_j \circ M_i(m) &= x_j\, x_i\, m - \rho_{x_i\, x_j\, m} \\ M_i \circ M_j(m) &= x_i M_j(m) + \sum_{\omega \in \partial B} \lambda_\omega \rho_\omega. \end{aligned}$$

  so that $M_j \circ M_i(m) = M_i \circ M_j(m)$ implies that

$$\rho_{x_i\, m}\, x_i - \rho_{x_i\, x_j\, m} = \sum_{\omega \in \partial B} \lambda_\omega \rho_\omega. \tag{2}$$

- If $m\, x_i \notin B$ and $m\, x_j \notin B$, then

$$\begin{aligned} M_i(m) &= m\, x_i - \rho_{m\, x_i}, \\ M_j(m) &= m\, x_j - \rho_{m\, x_j}, \\ M_j \circ M_i(m) &= x_j M_i(m) - \sum_{\omega \in \partial B} \lambda_{j,\omega} \rho_\omega \\ M_i \circ M_j(m) &= x_i M_j(m) - \sum_{\omega \in \partial B} \lambda_{j,\omega} \rho_\omega \end{aligned}$$

  and $M_j \circ M_i(m) = M_i \circ M_j(m)$ implies that

$$x_j \rho_{x_i\, m} - x_i \rho_{x_j\, m} = \sum_{\omega \in \partial B} (\lambda_{j,\omega} - \lambda_{i,\omega}) \rho_\omega \tag{3}$$

The relations (2) and (3) are syzygies between the polynomials of F. They are called *next-door* and *across-the-street relations* in [10]. They are special forms of the C-polynomials (also called commutation polynomials in [19]) that we define as follows: For any polynomials $f_1, f_2 \in F$, let the C-polynomial relative to γ be

$$C(f_1, f_2) = \frac{\operatorname{lcm}(\gamma(f_1), \gamma(f_2))}{\gamma(f_1)} f_1 - \frac{\operatorname{lcm}(\gamma(f_1), \gamma(f_2))}{\gamma(f_2)} f_2.$$

Theorem 3.2 is equivalent to the following proposition:

Proposition 3.3. [19] *Assume that B is connected to 1 that F is a rewriting family on B. Then if for all $f, f' \in F$ such that $C(f, f') \in B^+$, we have*

$$C(f, f') \in \langle F \rangle,$$

then B is a basis of $\mathfrak{A} = \mathbb{K}[\mathbf{x}]/(F)$.

This property generalizes the well known property of the S-polynomials in the computation of Gröbner bases [4].

3.4. Syzygies and Commutation Relations

We are going to analyze more precisely, the relations between the polynomials $F = (f_\omega)_{\omega \in \partial B}$. These relations or syzygies form a module that we denote by

$$Syz(F) = \{\sum_\omega h_\omega e_\omega \in \mathbb{K}[\mathbf{x}]^{\partial B}; \sum_\omega h_\omega \rho_\omega = 0\},$$

where $(e_\omega)_{\omega \in \partial B}$ is the canonical basis of $\mathbb{K}[\mathbf{x}]^{\partial B}$.

We denote by Ξ the module of $\mathbb{K}[\mathbf{x}]^{\partial B}$ generated by the relations (2) and (3).

Lemma 3.4. *$\forall m \in \mathcal{M}$, $\forall \theta \in \partial B$,*

$$m\, e_\theta \equiv \sum_{\omega \in \partial B} m_\omega e_\omega \quad \text{modulo } \Xi.$$

with $\delta_B(m_\omega) = |m_\omega| + 1$.

Proof. By definition, we have $\theta \in \partial B$, so that $\delta_B(\theta) = 1$. If $\delta_B(m\,\theta) = |m| + 1$, the property is true. Otherwise, $\delta_B(m\,\theta) < |m| + 1$, which implies that there exists $i_0 \in [1, \ldots, n]$ such that $m = x_{i_0}\, m'$ with $x_{i_0}\theta \in \partial B$. Using the relations (2), we have

$$m\, e_\theta = m'\, x_{i_0} e_\theta \equiv m'(e_{x_{i_0}\theta} + \sum_{\omega \in \partial B} \lambda_\omega e_\omega),$$

with $|m'| < |m|$. By iterating this reduction (which will eventually stop), we prove that modulo the relations (2), we have $m\, e_\theta \equiv \sum_{\omega \in \partial B} m_\omega e_\omega$ with $\delta_B(m_\omega) = |m_\omega| + 1$. □

Lemma 3.5. *$\forall m, m' \in \mathcal{M}$, $\forall \theta, \theta' \in \partial B$ such that $m\theta = m'\theta'$, $\theta \neq \theta'$ and $\delta_B(m\,\theta) = |m| + 1 = k$, $\delta_B(m'\,\theta') = |m'| + 1 = k$, we have*

$$m\, e_\theta - m' e_{\theta'} \equiv \tilde{m}\, e_{\tilde{\theta}} - m' e_{\theta'} + \sum_\omega h_\omega\, e_\omega \quad \text{modulo } \Xi.$$

with $\delta_B(h_\omega\, \rho_\omega) < k$ and $|\mathrm{lcm}(\tilde{\theta}, \theta')| < |\mathrm{lcm}(\theta, \theta')|$.

Proof. Let $\nu = \mathrm{lcm}(\theta, \theta')/\theta$, $\nu = \mathrm{lcm}(\theta, \theta')/\theta'$. As $\theta \neq \theta'$, $|\nu| \neq 0$ or $|\nu'| \neq 0$. If $|\nu| = 0$, then $|\nu'| > 0$ and $\theta = \nu'\theta'$ so that $\delta_B(m'\theta') \leq |m'| - |\nu'| + 1$, which is a contradiction. Similarly, we cannot have $|\nu'| = 0$.

This implies that there exists $i_0 \neq i_1$ such that x_{i_0} divides θ (and m') and such that x_{i_1} divides θ' (and $m = x_{i_1}\, t$). Let us consider $\tilde{\theta} \in \partial B$ such that $x_{i_1}\theta = x_{i_0}\tilde{\theta}$. Using the relations (3), we have

$$x_{i_1} e_\theta \equiv x_{i_0} e_{\tilde{\theta}} + \sum_{\omega \in \partial B} \lambda_\omega\, e_\omega.$$

Therefore, we deduce that modulo the relations (3),

$$m e_\theta - m' e_{\theta'} \equiv \tilde{m} e_{\tilde{\theta}} - m' e_{\theta'} + \sum_{\omega} \lambda_\omega\, t\, e_\omega,$$

with $\tilde{m} = x_{i_0}\, t$, $\delta_B(te_\omega) \leq |t| + 1 < |m| + 1$ and $t x_{i_0}\theta = m'\theta'$, so that $|\mathrm{lcm}(\tilde{\theta}, \theta')| < |\mathrm{lcm}(\theta, \theta')|$. □

This yields the following, conjectured in [10]:

Theorem 3.6. *Assume that B is stable by division and that we have a rewriting family $F = (\rho_\omega)_{\omega \in \partial B}$, satisfying the conditions of theorem* 3.2. *Then $Syz(F)$ is generated by the relations* (2) *and* (3).

Proof. Let $\sigma = \sum_\omega p_\omega\, e_\omega \in Syz(F)$. Applying lemma 3.4, by reduction modulo Ξ, we may assume that $\delta_B(p_\omega \omega) = |p_\omega| + 1$. Let us consider the maximum of such B-indices.

As $\sum_\omega p_\omega\, \rho_\omega = 0$, there exist $\theta \neq \theta' \in \partial B$ and monomials $m \in \mathrm{supp}(p_\theta)$, $m' \in \mathrm{supp}(p_{\theta'})$ such that $m\,\theta = m'\,\theta'$. By lemma 3.5, we can replace this pair $(m\, e_\theta, m'\, e_{\theta'})$ by a new pair where either the degree of $\mathrm{lcm}(\theta, \theta')$ or the B-index of $m\,\theta$ (resp. $m'\,\theta'$) is smaller.

Since we cannot iterate infinitely these reductions steps, we deduce that σ is in the module generated by the relations (2) and (3). □

4. Algorithm and Software

This analysis leads to the following scheme of algorithm, for computing a normal form, modulo the ideal generated by the polynomials $F = (f_1, \ldots, f_s)$.

Since we want to use rewriting rules on monomials, at some point of our computation, we will need to choose a specific monomial in a given polynomial. For that purpose, we introduce a choice function $\gamma : \mathbb{K}[\mathbf{x}] \to \mathcal{M}$ satisfying the following properties:

- $\gamma(p) \in \mathrm{supp}(p)$,
- if $m \in \mathrm{supp}(p)$, $m \neq \gamma(p)$ then $\gamma(p)$ does not divide m,
- and $\Lambda(\gamma(p)) = \max\{\Lambda(m),\ m \in \mathrm{supp}(p)\}$,

where Λ is a "degree" or a graduation of $\mathbb{K}[\mathbf{x}]$.

This choice function has some similarity with the leading term function in Gröbner basis constructions, but offers much more freedom. For instance, on can use the *Macaulay* choice function γ, such that for all $p \in \mathbb{K}[\mathbf{x}]$, $\gamma(p) = x_1^{\alpha_1} \cdots$

$x_n^{\alpha_n}$ satisfies, $\deg_{\mathbb{N}}(\gamma(p)) = \max\{\deg_{\mathbb{N}}(m); m \in \mathrm{supp}(p)\} = d$, and $\exists i_0$ st. $\alpha_{i_0} = \max\{\deg_{x_i}(m), m \in \mathrm{supp}(p) \text{ and } , \deg_{\mathbb{N}}(m) = d; i = 1, \ldots, n\}$.

Or one can take, among the monomials of largest degree, the one with the coefficients of largest modulus, if we are working over $\mathbb{R}$ or $\mathbb{C}$.

Algorithm 4.1. COMPUTE THE GENERALIZED NORMAL FORM

INPUT: $f_1, \ldots, f_s \in \mathbb{K}[\mathbf{x}]$ such that $I = (f_1, \ldots, f_s)$ defines a complex variety of dimension 0.

- Initialize P with the polynomials of F of minimal degree k and B with the set of monomials outside $\gamma(P)$.
- repeat
  1. Compute $\tilde{P}$ the set of polynomials of P^+ and of C-polynomials of P, which are in B^+.
  2. Apply linear transformation of the coefficient matrix of P' in order to rewrite monomials of ∂B in terms of the monomials in B.
  3. Update B,
     -- either by removing the monomial multiples of $\gamma(p)$, for $p \in \langle \tilde{P} \rangle \cap \langle B \rangle$,
     -- or by adding the monomials of ∂B, not in $\gamma(\tilde{P})$.
  4. Update P by inserting the new polynomials obtained from step 2, with one monomial in ∂B and the other in B.

  until $\partial B = \gamma(P)$.

OUTPUT: The rewriting family P on the basis B of $\mathfrak{A} = \mathbb{K}[\mathbf{x}]/I$.

The algorithm described here has been implemented by Ph. Trebuchet [25] in the library SYNAPS [2] (see `solve(L, Newmac<C>())`). It corresponds to about 50 000 lines of C++-code.

The main part is the computation of the linear algebra part, which consists in computing a triangular form of the coefficient matrix of the polynomials $\tilde{P}$. As this coefficient matrix may be sparse, it uses sparse LU decomposition algorithm, which avoid to produce too dense rows when performing column pivoting operations on the matrix. More precisely, the sparse matrix data structure `MatSps<double, sparse::rep2d<double> >` based on the container `sparse::rep2d<T>` provides the vector space operations and the matrix-vector multiplication, for sparse matrices. The container type `sparse::rep2d<>` corresponds to the NCFORMAT type of SUPERLU [7], so that no copy of objects are needed to call the external routines. The LU decomposition routines of SUPERLU have been ported in C++ to be able to use generic coefficients.

Once this triangular form is obtained, intersecting $\langle \tilde{P} \rangle$ and $\langle B \rangle$ consists just in extracting the rows of the matrix corresponding to polynomials with supports in $\langle B \rangle$.

[2] `http://www-sop.inria.fr/galaad/software/synaps/`

The numerical approximation of the roots are obtained by eigenvalues computation, using the library Lapack (the routine `dgegv`) and the strategy described in [5].

Several experiments have been performed to analyse the behavior of the methods, depending on the choice functions such as the choice function associated to the Degree Reverse Lexicographical order, or to the degree lexicographical order, a choice function that returns randomly any of the monomials of maximum degree of the polynomial given as its input, the Macaulay's choice function, the choice function over the rational that minimize the memory needed in the reduction loop, ... We also experiment with different type of coefficients such as rational numbers, or extended floating point numbers, to analyze the size and the precision of the computation. See [19] or [20], for detailed results.

It turns out that in many situations, the Macaulay choice function produces representation of the quotient algebra $\mathfrak{A}$, which is more compact than the others, faster to compute and which requires less accuracy. This is not always the case, and understanding how to compute the optimal representation of $\mathfrak{A}$ from a numerical and algebraic point of view is still an open and challenging problem.

5. Open problems

This new approach for constructing border basis is a generalisation of Gröbner basis computation. Not tied to a monomial order, it provides more freedom to perform polynomial reductions. In linear algebra, matrix triangulation with column pivoting is much better from a numerical point of view, than triangulation without column pivoting. Similarly in this new approach, we can choose pivots according to numerical criterion. However, many open problems remain to be solved:

- What is the optimal strategy to obtain a compact description of the quotient algebra, when dealing with exact coefficients ?
- When we are computing with approximate coefficients, which reduction strategy should we adopt to obtain a minimal numerical error on the description of this quotient algebra?
- How to determine tuned thresholds in zero-tests, when performing the numerical matrix reductions?
- How can we estimate the condition number of the approximation of the quotient algebra?
- Can we connect this condition number with the error on the solutions ?
- Is there a Newton-like method to improve efficiently the level of approximation of the quotient algebra?

As we said, the interaction between symbolic and numeric computation is a fascinating area, where many important problems are waiting for solutions.

References

[1] M.E. Alonso, E. Becker, M.F. Roy, and T. Wörmann. Zeros, multiplicities and idempotents for zero dimensional systems. In L. González-Vega and T. Recio, editors, *Algorithms in Algebraic Geometry and Applications*, volume 143 of *Prog. in Math.*, pages 1–15. Birkhäuser, Basel, 1996.

[2] W. Auzinger and H. J. Stetter. An elimination algorithm for the computation of all zeros of a system of multivariate polynomial equations. In *Proc. Intern. Conf. on Numerical Math.*, volume 86 of *Int. Series of Numerical Math*, pages 12–30. Birkhäuser Verlag, 1988.

[3] D. Bini. Numerical computation of polynomial zeros by means of Aberth's method. *Numerical Algorithms*, 13, 1996.

[4] B. Buchberger. Gröbner bases: An algebraic method in ideal theory. In N.K. Bose, editor, *Multidimensional System Theory*, pages 184–232. Reidel Publishing Co., 1985.

[5] R.M. Corless, P.M. Gianni, and B.M. Trager. A reordered Schur factorization method for zero-dimensional polynomial systems with multiple roots. In W.W. Küchlin, editor, *Proc. ISSAC*, pages 133–140, 1997.

[6] D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Undergraduate Texts in Mathematics. Springer Verlag, New York, 1992.

[7] J. Demmel, J. Gilbert, and Xiaoye S. Li. An asynchronous parallel supernodal algorithm for sparse gaussian elimination. *SIAM J. Matrix Anal. Appl.*, 20(4):915–952, 1999.

[8] M. Elkadi and B. Mourrain. *Introduction à la résolution des systèmes d'équations algébriques*, 2003. Notes de cours, Univ. de Nice (310 p.).

[9] P.A. Fuhrmann. *A polynomial approach to linear algebra*. Springer-Verlag, 1996.

[10] A. Kehrein and Kreuzer. A characterisation of border bases. *J. Pure and Applied Algebra*, 196:251–270, 2005.

[11] A. Kehrein, M. Kreuzer, and L. Robbiano. An algebraist's view on border bases. In A. Dickenstein and I. Emiris, editors, *Solving Polynomial Equations: Foundations, Algorithms, and Applications.*, volume 14 of *Algorithms and Computation in Mathematics*, pages 169–202. Springer, 2005.

[12] G. Lecerf. Computing an equidimensional decomposition of an algebraic varety bymeans of geometric resolutions. *Proc. ISSAC*, pages 209–216, 2000.

[13] F.S. Macaulay. *The Algebraic Theory of Modular Systems*. Cambridge Univ. Press, 1916.

[14] B. Mourrain. Computing isolated polynomial roots by matrix methods. *J. of Symbolic Computation, Special Issue on Symbolic-Numeric Algebra for Polynomials*, 26(6):715–738, Dec. 1998.

[15] B. Mourrain. A new criterion for normal form algorithms. In M. Fossorier, H. Imai, Shu Lin, and A. Poli, editors, *Proc. AAECC*, volume 1719 of *LNCS*, pages 430–443. Springer, Berlin, 1999.

[16] B. Mourrain and V. Y. Pan. Multivariate polynomials, duality and structured matrices. *J. of Complexity*, 16(1):110–180, 2000.

[17] B. Mourrain and Ph. Trébuchet. Solving projective complete intersection faster. In C. Traverso, editor, *Proc. Intern. Symp. on Symbolic and Algebraic Computation*, pages 231–238. New-York, ACM Press., 2000.

[18] B. Mourrain and Ph. Trébuchet. Algebraic methods for numerical solving. In *Proc. of the 3rd International Workshop on Symbolic and Numeric Algorithms for Scientific Computing'01 (Timisoara, Romania)*, pages 42–57, 2002.

[19] B. Mourrain and Ph. Trébuchet. Generalised normal forms and polynomial system solving. In M. Kauers, editor, *Proc. Intern. Symp. on Symbolic and Algebraic Computation*, pages 253–260. New-York, ACM Press., 2005.

[20] B. Mourrain and Ph. Trébuchet. Generalised normal forms and polynomial system solving. Technical Report 5471, INRIA Sophia-Antipolis, 2005.

[21] J. Renegar. On the computational complexity and geometry of the first order theory of reals (I, II, III). *J. Symbolic Computation*, 13(3):255–352, 1992.

[22] F. Rouillier. Solving zero-dimensional polynomial systems throuhg Rational Univariate Representation. *App. Alg. in Eng. Com. Comp.*, 9(5):433–461, 1999.

[23] H. J. Stetter. Eigenproblems are at the heart of polynomial system solving. *SIGSAM Bulletin*, 30(4):22–25, 1996.

[24] Hans J. Stetter. *Numerical polynomial algebra*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2004.

[25] Ph. Trébuchet. *Vers une résolution stable et rapide des équations algébriques*. PhD thesis, Université Pierre et Marie Curie, 2002.

[26] W.V. Vasconcelos. *Computational Methods in Commutative Algebra and Algebraic Geometry*, volume 2 of *Algorithms and Computation in Mathematics*. Springer-Verlag, 1998.

Bernard Mourrain
GALAAD INRIA
BP 93 06902 Sophia Antipolis
France
e-mail: `mourrain@sophia.inria.fr`