Des ensembles triangulaires pour résoudre les systèmes polynomiaux : une comparaison de quatre méthodes

Philippe Aubry
Marc Moreno Maza
LIP6, Université Paris 6
4, Place Jussieu 75252 Paris Cedex 05
Tel : (33) 01 44 27 33 41
e-mail : aubry@posso.ibp.fr m3@posso.ibp.fr

résumé

Quatre méthodes de résolution de systèmes d'équations polynomiales sont présentées et implantées dans un cadre commun. Ces méthodes sont celles de Wu ([Wu87]), Lazard ([Laz91]), Kalkbrener ([Kal91]) et Wang ([Wan93b]). Elles sont comparées sur divers exemples avec une attention particulière portée à l'efficacité, la concision et la lisibilité des sorties.

Triangular Sets for Solving Polynomial Systems : a Comparison of Four Methods

Philippe Aubry
Marc Moreno Maza
LIP6, Université Paris 6
4, Place Jussieu 75252 Paris Cedex 05
Tel : (33) 01 44 27 33 41
e-mail : aubry@posso.ibp.fr m3@posso.ibp.fr

January 22, 1996

Abstract

Four methods for solving polynomial systems by means of triangular sets are presented and implemented in a unified way. These methods are those of Wu ([Wu87]), Lazard ([Laz91]), Kalkbrener ([Kal91]) and Wang ([Wan93b]). They are compared on various examples with emphasizing on efficiency, conciseness and legibility of the outputs.

Introduction

In this paper, we are concerned with the following problem : given a finite family F of multivariate polynomials over a field $\mathbf{k}$ and with ordered variables $X_1 < X_2 < \cdots < X_n$ to describe the affine variety $\mathbf{V}(F)$ (i.e. the common zeros of F over an algebraic closure of $\mathbf{k}$). Such a description is usually given by a finite family $\{T_1, \ldots, T_r\}$ of polynomial sets with particular properties, a *link* between the T_i and F, and an algorithm to compute the T_i from F. A well developed strategy since Buchberger's work ([Buc65]) is the following : given an ordering on the monomials, to choose for T_1 the Gröbner basis of the ideal generated by F and compute it by the Buchberger's algorithm.

Wu Wen-Tsün in [Wu87] introduced another way of solving algebraic systems which is the one we are concerned with in this paper. In that case each T_i is a polynomial set such that two distinct polynomials in T_i have distinct greatest variables. Such a T_i is called a triangular set. The points of $\mathbf{V}(T_i)$ where no leading coefficient of a polynomial in T_i, viewed as univariate in its greatest variable, vanishes are called the regular zeros of T_i. Then, in Wu's method, the variety $\mathbf{V}(F)$ is the union of the regular zeros of the T_i and this decomposition can be computed by the original Wu's CHRST-REM algorithm ([Wu87]). This method has been investigated in many papers. Among them : [Cho88, CG90, CG92, GM90, Wan92a, Wan92b]. Wu's method is efficient for geometric problems where the degenerate solutions are not interesting. For general problems it seems to be difficult to obtain an efficient implementation and this method may produce superfluous triangular sets. Wu's algorithm, like Buchberger's one, depends on many choices; moreover, its result is not uniquely defined.

A Wu's like decomposition of affine varieties can be obtained by Daniel Lazard's algorithm ([Laz91]). But in that case the definition of triangular sets has been strengthened (definition9) in order to guarantee irredundant and more canonical decompositions. Our paper reports a first implementation of this method and shows that it can be efficient. In [MR95], M. Moreno Maza and R. Rioboo report a very efficient implementation of another algorithm due to Daniel Lazard ([Laz92a]) and called *Lextriangular*. This last algorithm also computes decompositions similar to those of ([Laz91]) but the input must be a lexicographical Gröbner basis of a zero-dimensional ideal. Lazard's decompositions have at least two interesting properties. On one hand numeric solutions may be easily obtained from them because Lazard's triangular sets are normalized (definition 2). See also section 8 in [Laz92a] for more details. On the other hand Lazard's triangular sets are well suited for describing prime ideals (see section 3 in [Laz91]) whereas there is no bound on the minimal number of generators for a lexicographical Gröbner basis of a prime ideal.

In [Kal91] Michael Kalkbrener introduced another type of triangular sets called regular chains (definition 4) together with another link between F and the T_i. In that case $\mathbf{V}(F)$ is the union of the closures (w.r.t. Zarisky topology) of the regular zeros of the T_i. Let us mention an example to see the difference between Wu and Lazard's way of solving and Kalkbrener's one. We consider the system given by the following polynomials where the ordered variables are $c_2 > s_2 > c_1 > s_1 > b > a$ and where the coefficients lie in the field of rational numbers :

$$\left\{c_1\ c_2 - s_1\ s_2 + c_1 - a,\ s_1\ c_2 + c_1\ s_2 + s_1 - b,\ c_1^2 + s_1^2 - 1,\ c_2^2 + s_2^2 - 1\right\}$$

Our implementation of Lazard's algorithm ([Laz91]) produces the decomposition $\{T_1, T_2, T_3\}$ where :

$$\begin{array}{rcl}
T_1 & = & \{(4\ b^2 + 4\ a^2)\ s_1^2 + (-4\ b^3 - 4\ a^2\ b)\ s_1 + b^4 + 2\ a^2\ b^2 + a^4 - 4\ a^2, \\
 & & 2\ a\ c_1 + 2\ b\ s_1 - b^2 - a^2 \\
 & & 2\ a\ s_2 + (2\ b^2 + 2\ a^2)\ s_1 - b^3 - a^2\ b, \\
 & & 2\ c_2 - b^2 - a^2 + 2\} \\
T_2 & = & \{a,\ 2\ s_1 - b,\ 4\ c_1^2 + b^2 - 4,\ s_2 - b\ c_1,\ 2\ c_2 - b^2 + 2\} \\
T_3 & = & \{a,\ b,\ c_1^2 + s_1^2 - 1,\ s_2,\ c_2 + 1\}
\end{array}$$

How this solution has to be understood ? In T_1, one may arbitrarily choose a and b once $a(b^2 + a^2) \neq 0$, and obtain successively the values of the indeterminates s_1, c_1, s_2, c_2. The triangular sets T_2 and T_3 describe the case $a = 0$. Note that in T_2, one may choose arbitrarily b whereas $b = 0$ in T_3. So, where is the case $b^2 + a^2 = 0$? It is described by T_3. In fact, if we add this equation to the initial system the computed decomposition is only $\{T_3\}$. Now, our implementation of Kalkbrener's algorithm ([Kal91]) produces the decomposition $\{C\}$ where :

$$\begin{array}{rcl}
C & = & \{(4\ b^2 + 4\ a^2)\ s_1^2 + (-4\ b^3 - 4\ a^2\ b)\ s_1 + b^4 + 2\ a^2\ b^2 + a^4 - 4\ a^2, \\
 & & 2\ a\ c_1 + 2\ b\ s_1 - b^2 - a^2,\ s_2 - b\ c_1 + a\ s_1, \\
 & & s_1\ c_2 + b\ c_1^2 - a\ s_1\ c_1 + s_1 - b\}
\end{array}$$

In that case a point is a solution of the initial system if it lies in the closure of the regular zeros of the only triangular set above, denoted by C. Although C and T_1 are different, their regular zero sets have the same closure, which contains the regular zeros of the previous T_2 and T_3. Note that Kalkbrener's output is simpler but needs futher computations for the zeros satisfying $a(b^2 + a^2) = 0$.

Triangular sets can also be used to solve quasi-algebraic systems (definition 3). In [Wan93b] Dongming Wang proposed such a method by means of Wu's triangular sets. But Wang's process is different from Wu's one and seems to be more efficient.

In the conclusion of [Kal93] Michael Kalkbrener wrote : a comparison with the algorithms of Ritt, Wu and Lazard seems to be interesting. In the conclusion of [Wan93b] Dongming Wang wrote : a systematic analysis and comparison among them (the elimination methods of Lazard and Kalkbrener) both theoretically and practically remain interesting for future work. The purpose of this paper is to compare the methods of Wu ([Wu87]), Lazard ([Laz91]), Kalkbrener ([Kal91]) and Wang ([Wan93b]). In the first section, following [Laz92b], we introduce a coherent terminology to present their specifications. Then we study some properties of regular chains and their connection with towers of simple extensions (definition 5). We also look into the special case of Lazard sets. In the second section, we review the specifications of each method. Futhermore, we give a recursive adaptation of Wang's method which seems to us easier to read than the original iterative description. Our implementation of Lazard's method is based on an algorithm for gcd computations of univariate polynomials with coefficients in a separable tower of simple extensions. The idea is a generalisation of the one of [MR95]. This algorithm could not take place here and will be presented in a future paper. In the third section we discuss experimentations on those four methods. We think that a reasonable comparative implementation should satisfy the following requirements :

- the corresponding algorithms must be implemented and run with the same human, material and software conditions (using the same data structures and subroutines)
- to make sure that each computed solution is correct
- not to only focus on timings but also on the legibility of the ouputs and their suitability for further uses

A strongly typed and object-oriented language is convenient to satisfy the first requirement above. We used the AXIOM computer algebra system ([JS92]). We defined categories corresponding to the different properties of triangular sets, packages and domains for the common sub-routines. Futhermore, AXIOM is connected with GB, the very powerful Gröbner engine developed by J.C. Faugère ([Fau94]). This allowed us the non-trivial Gröbner basis computations which are needed in order to satisfy the second requirement above.

Our implementation of each method uses the same polynomial domain constructor. Thus, a method involving particular data-structures like the dynamic sets and dynamic polynomials ([Dia92]) could not enter within our experimentations. However we tested each of our examples with the dynamic evaluation. This method is only usable for *easy examples* and cannot compare with the methods of Wu, Wang, Kalkbrener and Lazard. But note that the goal of dynamic evaluation is not restricted to polynomial system solving. As we wanted to implement easily and completely each of the method we considered, we also discarded methods which depend on sophisticated techniques like Gröbner basis computations or factorizations.

In the last section, we report some experimental data on a set of test examples. Most of them can be found in the data base of the european research project PoSSo ([Com92]). They are also available by `ftp` on `posso.ibp.fr` in the directory `pub/papers/TriangularSets`. Finally, we investigate the computed decompositions for some relevant examples and point out some remarks suggested by our experimentations.

1 Triangular Sets and Towers of Simple Extensions

In this section we first recall the most general definition for *triangular sets* (definition 1). This is the one used in Wu's method ([Wu87]) and in Wang's method ([Wan93a, Wan93b]). Then we recall the definition of *regular chains* (definition 4) which are particular triangular sets used in Kalkbrener's method ([Kal91, Kal93]). In the third subsection, we give a definition for *towers of simple extensions* (definition 5) and we show that regular chains are suitable for encoding every tower of simple extensions. Finally, we study *Lazard sets* (definition 9), which are special regular chains. Their presentation is inspired by our adaptation of Lazard's method ([Laz91]) by means of polynomial gcd computations over tower of separable extensions (full details will appear in [Maz97]). Before dealing with triangular sets, we need some general notations. about rings, ideals and varieties.

Notations 1 We denote by $\mathbb{N}$ the set of the non-negative integer numbers. Let $\mathbf{A}$ be a ring (all rings considered here are commutative noetherian rings with unit element) and E be a subset of $\mathbf{A}$. We denote by $\langle E \rangle_{\mathbf{A}}$ the ideal of $\mathbf{A}$ generated by E, and by $\mathbf{A}/E$ the quotient ring of $\mathbf{A}$ by $\langle E \rangle_{\mathbf{A}}$. For an element $a \in \mathbf{A}$, we denote by $\overline{a}^E$ the residue class of a in $\mathbf{A}/E$. If $E = \{a_1, \ldots, a_l\}$ we simply write $\langle a_1, \ldots, a_l \rangle_{\mathbf{A}}$ (or $\langle a_1, \ldots, a_l \rangle$) instead of $\langle E \rangle_{\mathbf{A}}$. If E is empty we state $\langle E \rangle_{\mathbf{A}} = \langle 0 \rangle$. We denote by $\mathbf{nz}(\mathbf{A})$ the multiplicatively closed subset of non-zero-divisors of $\mathbf{A}$ (this contains the group of invertible elements of $\mathbf{A}$) and by $\mathbf{q}(\mathbf{A})$ the ring of fractions with numerators in $\mathbf{A}$ and denominators in $\mathbf{nz}(\mathbf{A})$. Let I be an ideal of $\mathbf{A}$. We denote by $\mathsf{ap}(I)$ the associated prime ideals of I (i.e. the components of a minimal primary decomposition of $\sqrt{I}$). For an element $h \in \mathbf{A}$, the saturated ideal of I w.r.t. h (i.e. the set of the $b \in \mathbf{A}$ such that there exists a positive integer m with $h^m b \in I$) is denoted by $I : h^\infty$. The ideal generated in $\mathbf{A}[X]$ by I is denoted by $\mathbf{I}[X]$. For a polynomial $p \in \mathbf{A}[X]$ we denote by $\overline{p}^I$ the image of p in $\mathbf{q}(\mathbf{A}/I)[X]$ obtained by mapping the coefficients of p into $\mathbf{q}(\mathbf{A}/I)$. For a module $\mathbf{M}$ over $\mathbf{A}$ and a multiplicatively closed subset S of $\mathbf{nz}(\mathbf{A})$, we denote by $\mathrm{S}^{-1}\mathbf{M}$ the $\mathbf{A}$-module of fractions with numerator in $\mathbf{M}$ and denominator in S. Now assume that $\mathbf{A}$ is a polynomial ring with n variables and coefficients over a field $\mathbf{k}$. Let $\mathbf{K}$ be an algebraic closure of $\mathbf{k}$. For an ideal I of $\mathbf{A}$, we denote by $\mathbf{V}_K(I)$ (or simply $\mathbf{V}(I)$) the affine variety of $\mathbf{K}^n$ associated to I and if $I = \langle a_1, \ldots, a_l \rangle$ we simply write $\mathbf{V}(a_1, \ldots, a_l)$ instead of $\mathbf{V}(\langle a_1, \ldots, a_l \rangle)$. Finally, for $W \subseteq \mathbf{K}^n$, we denote by $\overline{W}$ the closure of W w.r.t. the Zarisky topology over $\mathbf{k}$ (whose closed sets are the $\mathbf{V}(I)$ for every ideal I of $\mathbf{A}$).

1.1 Triangular Sets

Notations 2 Let $\mathbf{R}$ be an integral domain. We denote by $\mathbf{k}$ the field of fractions of $\mathbf{R}$. Let $\mathbf{K}$ be an algebraic closure of $\mathbf{k}$. Let n be a positive integer and V a set of n ordered variables $X_1 < X_2 < \cdots < X_n$. For $1 \leq i \leq n$, let $\mathbf{R}_i = \mathbf{R}[X_1, \ldots, X_i]$ and $\mathbf{P}_i = \mathbf{k}[X_1, \ldots, X_i]$ be the rings of polynomials in i variables with coefficients in $\mathbf{R}$ and $\mathbf{k}$ respectively. We also define $\mathbf{R}_0 = \mathbf{R}$ and $\mathbf{P}_0 = \mathbf{k}$. Let $E \subseteq \mathbf{R}_n$ and $p, q \in \mathbf{R}_n$, with $p \neq 0$ and $q \notin \mathbf{R}$. For $v \in V$, we write $\mathsf{deg}(p, v)$ for the degree of p with respect to the variable v. We denote by $\mathsf{var}(E)$ the set of the variables $v \in V$ for which there exists $r \in E$ with $r \neq 0$ such that $\mathsf{deg}(r, v) > 0$. If $E = \{r\}$ we simply write $\mathsf{var}(r)$ intsead of $\mathsf{var}(E)$. We call the *main variable* of q, denoted by $\mathsf{mvar}(q)$, the greatest variable of q. When $E \not\subseteq \mathbf{R}$ we denote by $\mathsf{mvar}(E)$ the greatest variable of $\mathsf{var}(E)$. We call *initial* of q (denoted by $\mathsf{init}(q)$) the leading coefficient of q viewed as an univariate polynomial in $\mathsf{mvar}(q)$. We call *main degree* of q (denoted by $\mathsf{mdeg}(q)$) the degree $\mathsf{deg}(q, \mathsf{mvar}(q))$ and *tail* of q (denoted by $\mathsf{tail}(q)$) the polynomial $q - \mathsf{init}(q)\, \mathsf{mvar}(q)^{\mathsf{mdeg}(q)}$. We denote by $\mathsf{algVar}(E)$ the set of the variables $v \in V$ for which there exists $r \in E$ with $r \notin \mathbf{R}$ such that $\mathsf{mvar}(r) = v$. Let v be in V. We denote by E_v^-, E_v and E_v^+ the set of the non-constant polynomials $r \in E$ such that $\mathsf{mvar}(r) < v$, $\mathsf{mvar}(r) = v$ and $\mathsf{mvar}(r) > v$ respectively. If $E_v = \{r\}$ we simply write $E_v = r$.

Definition 1 *A subset T of $\mathbf{R}_n$ is called a* triangular set *if every polynomial of T is non-constant and if for all $p, q \in T$ with $p \neq q$ we have $\mathsf{mvar}(p) \neq \mathsf{mvar}(q)$.*

Example 1 Let $p \in \mathbf{R}_n$. Let $\mathsf{iter}(p)$ be the subset of $\mathbf{R}_n$ recursively defined as follows : if $p \in \mathbf{R}$ then $\mathsf{iter}(p) = \emptyset$ else $\mathsf{iter}(p) = \{p\} \cup \mathsf{iter}(\mathsf{init}(p))$. Then $\mathsf{iter}(p)$ is a triangular set of $\mathbf{R}_n$ whose elements are called the *iterated initials* of p.

Notations 3 Let E be a subset of $\mathbf{R}_n$ and $p, q \in \mathbf{R}_n$, with $p \neq 0$ and $q \notin \mathbf{R}$. We write $\mathsf{red?}(p, q)$ if $\mathsf{deg}(p, \mathsf{mvar}(q)) < \mathsf{mdeg}(q)$ holds. Then we write $\mathsf{red?}(p, E)$ if $\mathsf{red?}(p, r)$ holds for every $r \in E$. We write $\mathsf{iRed?}(p, q)$ if either $\mathsf{iter}(p)_v = \emptyset$ or $\mathsf{red?}(\mathsf{iter}(p)_v, q)$ holds where v is $\mathsf{mvar}(q)$. We write $\mathsf{normalized?}(p, q)$ if $\mathsf{iter}(p)_v = \emptyset$ holds where v is $\mathsf{mvar}(q)$. Then we write $\mathsf{iRed?}(p, E)$ if $\mathsf{iRed?}(p, r)$ holds for every $r \in E$. The same way we define $\mathsf{normalized?}(p, E)$. We denote by $\mathsf{prem}(p, q)$ and $\mathsf{pquo}(p, q)$ the pseudo-remainder and the pseudo-quotient of p by q when interpreting them as univariate in $\mathsf{mvar}(q)$. Let $T \subseteq \mathbf{R}_n$ be a triangular set. If $T = \emptyset$ we define $\mathsf{prem}(p, T) = p$ else we define $\mathsf{prem}(p, T) = \mathsf{prem}(\mathsf{prem}(p, T_v), T_v^-)$ where v is $\mathsf{mvar}(T)$. Then we denote by $\mathsf{prem}(E, T)$ the subset of $\mathbf{R}_n$ whose elements are the polynomials $\mathsf{prem}(r, T)$ for r in E. If $T = \{q\}$ we simply write $\mathsf{prem}(E, q)$ instead of $\mathsf{prem}(E, T)$.

Definition 2 *A triangular set T of $\mathbf{R}_n$ is called* reduced *(resp.* initially reduced *) (resp.* normalized*) if for every $t \in T$, denoting $\mathsf{mvar}(t)$ by v, we have $\mathsf{red?}(t, T_v^-)$ (resp. $\mathsf{iRed?}(t, T_v^-)$) (resp. $\mathsf{normalized?}(t, T_v^-)$).*

Example 2 In the introduction, the triangular sets T_1, T_2 and T_3 are reduced and normalized, whereas C is initially reduced but neither reduced nor normalized.

Notations 4 Let $p, q \in \mathbf{R}_n$, with $q \notin \mathbf{R}$. Let S be the multiplicatively closed subset of $\mathbf{R}_n$ generated by $h = \mathsf{init}(p)$. Let e be the minimal power of h by which p is multiplied in order to compute a polynomial r (by the peudo-division algorithm) such that $\mathsf{red?}(r, q)$ and $h^e p - r \in \langle q \rangle_{\mathbf{R}_n}$. In many cases, we have $e = \mathsf{deg}(p, \mathsf{mvar}(q)) - \mathsf{mdeg}(q) + 1$ and $r = \mathsf{prem}(p, q)$. If $\mathsf{red?}(p, q)$ we have $e = 0$ and $r = p$. Then we denote by $\mathsf{mod}(p, q)$ the element of $\mathrm{S}^{-1}\mathbf{R}_n$ defined by $\frac{r}{h^e}$. Let T be triangular set of $\mathbf{R}_n$. We denote by S the multiplicatively closed subset of $\mathbf{R}_n$ generated by 1 and the initials of the elements of T. If $p \notin \mathbf{R}$ then we write $v = \mathsf{mvar}(p)$, $i_p = \mathsf{init}(p)$, $d_p = \mathsf{mdeg}(p)$ and $t_p = \mathsf{tail}(p)$. Now, we define the element $\mathsf{mod}(p, T)$ of $\mathrm{S}^{-1}\mathbf{R}_n$ by iterating the following rules :

(1) $T = \emptyset$ or $p \in \mathbf{R} \implies \mathsf{mod}(p, T) = \frac{p}{1}$

(2) $\mathsf{red?}(p,\ T_v) \implies \mathsf{mod}(p, T) = \mathsf{mod}(i_p,\ T_v^-)\frac{v^{d_p}}{1} + \mathsf{mod}(t_p, T)$

(3) $\mathsf{mod}(v^{d_p}, T_v) = \frac{r}{s}$ and $\mathsf{mod}(i_p, T_v^-) = \frac{r'}{s'} \Longrightarrow \mathsf{mod}(p, T) = \dfrac{\mathsf{mod}(rr', T)}{ss'} + \mathsf{mod}(t_p, T)$

Thus for every $p \in \mathbf{R}_n$ there exist $r \in \mathbf{R}_n$ and $s \in S$ such that $\mathsf{mod}(p, T) = \frac{r}{s}$ with $\mathsf{red?}(r, T)$ and $sp - r \in \langle T \rangle_{\mathbf{R}_n}$. Finally, we define the element $\mathsf{iRed}(p, T)$ of $\mathbf{R}_n$ by iterating the following rules :

(a) $\mathsf{iRed?}(p, T) \implies \mathsf{iRed}(p, T) = p$

(b) $\mathsf{mod}(p,\ T_v) = \frac{r}{s} \implies \mathsf{iRed}(p, T) = \mathsf{iRed}(r, T)$

(c) $\mathsf{iRed}(i_p, T_v^-) = r$ and $\mathsf{mod}(s\, i_p - r,\ T_v^-) = 0 \implies \mathsf{iRed}(p, T) = \mathsf{iRed}(rv^{d_p} + st_p, T)$

Thus for every $p \in \mathbf{R}_n$ there exist $r \in \mathbf{R}_n$ and $s \in S$ such that $\mathsf{iRed}(p, T) = r$ with $\mathsf{iRed?}(r, T)$ and $sp - r \in \langle T \rangle_{\mathbf{R}_n}$.

Remark 1 Note that to apply rule (c) in the definition of $\mathsf{iRed}(p, T)$ it is necessary to store the intermediate denominators s which appear when applying rule (b). This notion of *iterated initials reduction* is the weakest notion of reduction which ensures the termination of Wu's algorithm ([Laz92b]). Futhermore, as it limits the number of reduction steps, it leads generally to an increase of efficiency in comparison with the *complete reduction* (i.e. the one based on the operation $(p, T) \longmapsto \mathsf{mod}(p, T)$).

Definition 3 *Every couple $\Sigma = (P, Q)$, where P and Q are two finite subsets of $\mathbf{R}_n$, is called a* quasi-algebraic system *in $\mathbf{R}_n$* (q.a.s. *for short*). *Let $\Sigma = (P, Q)$ be a q.a.s. in $\mathbf{R}_n$. The q.a.s. Σ is called* triangular *if P is a triangular set of $\mathbf{R}_n$. If $Q \neq \emptyset$ then we denote by $h(\Sigma)$ the product of the elements of Q, otherwise we define $h(\Sigma) = 1$. We call a* zero *of Σ every element of the subset of $\mathbf{K}^n$ denoted by $\mathbf{Z}(\Sigma)$ and defined by :*

$$\mathbf{Z}(\Sigma) = \mathbf{V}(P) \setminus \mathbf{V}(h(\Sigma))$$

The q.a.s. Σ is called inconsistent *if $\mathbf{Z}(\Sigma) = \emptyset$ else it is called* consistent. *The* saturated ideal *of the q.a.s. Σ is the saturated ideal of the ideal generated by P in $\mathbf{P}_n$ w.r.t. $h(\Sigma)$.*

Let $T \subseteq \mathbf{R}_n$ be a triangular set. We denote by $\Sigma(T)$ the triangular q.a.s defined by

$$\Sigma(T) = (T, \{\mathsf{init}(t) \mid t \in T\})$$

Then, we denote by $\mathsf{sat}_n(T)$ the saturated ideal of $\Sigma(T)$ and by $h(T)$ the product of the initials of the elements of T. Moreover, every zero of $\Sigma(T)$ is called a regular zero *of T and $\mathbf{Z}(\Sigma(T))$ is also written $\mathbf{W}(T)$ and called the* quasi-component *of T. Finally, following* [Wan93a, Wan93b]*, a triangular q.a.s. $\Sigma = (T, Q)$ is called* fine *if $\mathbf{V}(h(T)) \cap \mathbf{Z}(\Sigma) = \emptyset$ and $0 \notin \mathsf{prem}(Q, T)$.*

Remark 2 Let $T \subseteq \mathbf{R}_n$ be a triangular set. If T is a regular chain (definition 4) then $\mathbf{W}(T) \neq \emptyset$. This will result from theorems 1, and 2 and proposition 4. The converse is false as shown by the following example : $T = \{X_1^2, X_1X_2^2 + X_2 + 1\}$. Thus, if T is a normalized triangular set, then $\mathbf{W}(T) \neq \emptyset$. This will result from theorem 3 and proposition 5. If $\mathbf{W}(T) \neq \emptyset$ then $\Sigma(T)$ is fine but the converse is false, consider $T = \{X_1^2 - 1, X_1X_2^2 - X_1 + 1, X_2(X_1+1)X_3 + 1\}$.

Let Σ be a q.a.s. in $\mathbf{R}_n$. To decide whether Σ is consistent one can compute $\mathsf{sat}_n(\Sigma)$ by means of Gröbner bases techniques ([GTZ88, CLO91, Laz92b]). The answer is true iff $\mathsf{sat}_n(\Sigma) \neq \mathbf{P}_n$ (i.e. $h(T)$ does not lie in the radical of the ideal generated by T in $\mathbf{P}_n$). The following result shows more precisely the links between $\mathsf{sat}_n(\Sigma)$ and $\mathbf{Z}(\Sigma)$.

Theorem 1 *Let Σ be a q.a.s. in $\mathbf{R}_n$. Then we have :*

$$\overline{\mathbf{Z}(\Sigma)} = \mathbf{V}(\mathsf{sat}_n(\Sigma))$$

Proof. $\triangleright$ Let $\Sigma = (P, Q)$ be a q.a.s. in $\mathbf{R}_n$. We denote by H the principal ideal generated by $h(\Sigma)$ in $\mathbf{P}_n$ and by I the ideal generated by P in $\mathbf{P}_n$. It is clear that $\overline{\mathbf{Z}(\Sigma)} = \overline{\mathbf{V}(\sqrt{I}) \setminus \mathbf{V}(H)}$. Thus, by theorem 7 in [CLO91] p.193, we have $\overline{\mathbf{Z}(\Sigma)} = \mathbf{V}(\sqrt{I} : H)$. Finally, one can check that $\sqrt{I} : H = \sqrt{I} : h(\Sigma)^\infty = \sqrt{I : h(\Sigma)^\infty}$. $\triangleleft$

1.2 Regular Chains

The concept of regular chains in $\mathbf{P}_n$ is introduced by Kalkbrenner in [Kal91]. The definition below deals only with ideals and corresponds to a particular case of *system of representations* presented in [Kal95]. Let i be a positive integer and I an ideal in $\mathbf{P}_{i-1}$, recall that for $f \in \mathbf{P}_i$, we denote by $\overline{f}^I$ the canonical image of f in $\mathbf{q}(\mathbf{P}_{i-1}/I)[X_i]$.

Definition 4 *Let $i \in \mathbb{N}$ and T be a triangular set of $\mathbf{R}_i$. We say that T is a* regular chain *in $\mathbf{P}_i$ and that the ideal $\mathsf{Rep}_i(T)$ of $\mathbf{P}_i$ is its* representation *if either $i = 0$, $T = \emptyset$, and $\mathsf{Rep}_0(T) = \{0\}$, or $i > 0$ and one of the following assertions holds :*

(1) *$X_i \notin \mathsf{algVar}(T)$, the set T is a regular chain in $\mathbf{P}_{i-1}$ and*

$$\mathsf{Rep}_i(T) \;=\; \{f \in \mathbf{P}_i \mid (\forall \mathcal{P} \in \mathsf{ap}(\mathsf{Rep}_{i-1}(T)))\ \overline{f}^{\mathcal{P}} = 0\}$$

(2) *$X_i \in \mathsf{algVar}(T)$, the set $T_{X_i}^-$ is a regular chain in $\mathbf{P}_{i-1}$, for any associated prime ideal $\mathcal{P}$ of $\mathsf{Rep}_{i-1}(T_{X_i}^-)$ we have $\mathsf{init}(T_{X_i}) \notin \mathcal{P}$, and*

$$\mathsf{Rep}_i(T) \;=\; \{f \in \mathbf{P}_i \mid (\forall \mathcal{P} \in \mathsf{ap}(\mathsf{Rep}_{i-1}(T_{X_i}^-)))\ \overline{f}^{\mathcal{P}} \in \sqrt{\langle \overline{T_{X_i}}^{\mathcal{P}} \rangle_{\mathbf{q}(\mathbf{P}_{i-1}/\mathcal{P})[X_i]}}\}$$

Remark 3 With the notations of the above definition, if $X_i \in \mathsf{algVar}(T)$ then it follows from the condition $\mathsf{init}(T_{X_i}) \notin \mathcal{P}$ that $\mathsf{deg}(\overline{T_{X_i}}^{\mathcal{P}}, X_i) = \mathsf{mdeg}(T_{X_i})$. Thus if $r \in \mathbf{P}_i$ with $\mathsf{deg}(r, X_i) < \mathsf{mdeg}(T_{X_i})$, we have $\mathsf{deg}(\overline{r}^{\mathcal{P}}, X_i) < \mathsf{deg}(\overline{T_{X_i}}^{\mathcal{P}}, X_i)$.

Remark 4 The following results can be verified with general commutative algebra : let I an ideal in $\mathbf{A}$ and $h \in \mathbf{A}$, then $\sqrt{I}[X] = \sqrt{I[X]}$ and $(I : h^\infty)[X] = I[X] : h^\infty$. Thus, if T is a triangular set in $\mathbf{P}_{i-1}$, we have $\mathsf{sat}_i(T) = \langle \mathsf{sat}_{i-1}(T) \rangle_{\mathbf{P}_i}$.

Proposition 1 *Let i be a positive integer and T be a regular chain in $\mathbf{P}_i$.*

(i) if $X_i \notin \mathsf{algVar}(T)$ then $\mathsf{Rep}_i(T) \quad = \quad \langle \sqrt{\mathsf{Rep}_{i-1}(T)} \rangle_{\mathbf{P}_i}$

(ii) if $X_i \in \mathsf{algVar}(T)$ then

$$\mathsf{Rep}_i(T) \quad = \quad \{ f \in \mathbf{P}_i \;\mid\; (\exists m \in \mathbb{N}) \; \mathsf{prem}(f^m,\; T_{X_i}) \in \mathsf{Rep}_i(\; T_{X_i}^-) \}$$

Proof. $\triangleright$ Let $f \in \mathbf{P}_i$ and $\mathcal{P} \in \mathsf{ap}(\mathsf{Rep}_{i-1}(\; T_{X_i}^-))$. We first assume that $X_i \notin \mathsf{algVar}(T)$. We have $\overline{f}^{\mathcal{P}} = 0$ iff every coefficient of f, viewed as univariate in X_i, lies in $\mathcal{P}$. Thus, $f \in \mathsf{Rep}_i(T)$ iff every coefficient of f lies in $\sqrt{\mathsf{Rep}_{i-1}(T)}$, i.e. $f \in \langle \sqrt{\mathsf{Rep}_{i-1}(T)} \rangle_{\mathbf{P}_i}$.

Now we assume that $X_i \in \mathsf{algVar}(T)$ with $t = \; T_{X_i}$ and $h = \mathsf{init}(t)$. For $m \in \mathbb{N}$, we denote $\mathsf{prem}(f^m, t)$ by r_m. There exists $q \in \mathbf{P}_i$ and $\delta \in \mathbb{N}$ such that

$$h^\delta f^m = qt + r_m \tag{1}$$

First let us assume that $f \in \mathsf{Rep}_i(T)$. By point (2) of definition 4 there exists $m \in \mathbb{N}$ such that $\overline{f^m}^{\mathcal{P}} \in \langle \overline{t}^{\mathcal{P}} \rangle$. By choosing m big enough, we can take the same integer m for every prime ideal $\mathcal{P}$ in $\mathsf{ap}(\mathsf{Rep}_{i-1}(\; T_{X_i}^-))$. With the relation (1) we deduce that $\overline{t}^{\mathcal{P}}$ divides $\overline{r_m}^{\mathcal{P}}$. By remark 3 it follows that $\overline{r_m}^{\mathcal{P}} = 0$. Therefore $r_m \in \langle \sqrt{\mathsf{Rep}_{i-1}(\; T_{X_i}^-)} \rangle_{\mathbf{P}_i}$, and with (i) we obtain $r_m \in \mathsf{Rep}_i(\; T_{X_i}^-)$. Conversely, assume that there exists $m \in \mathbb{N}$ such that $r_m \in \mathsf{Rep}_i(\; T_{X_i}^-)$. We get $\overline{r_m}^{\mathcal{P}} = 0$ and thus $\overline{h^\delta f^m}^{\mathcal{P}} \in \langle \overline{t}^{\mathcal{P}} \rangle$. By definition $h \notin \mathcal{P}$, therefore $\overline{h}^{\mathcal{P}}$ is invertible. It follows that $\overline{f^m}^{\mathcal{P}} \in \langle \overline{t}^{\mathcal{P}} \rangle$, i.e. $f \in \mathsf{Rep}_i(T)$. $\triangleleft$

Proposition 2 *Let $i \in \mathbb{N}$ and T be a non-empty triangular set of $\mathbf{P}_i$ such that $X_i \in \mathsf{algVar}(T)$. Let us assume that for every $\mathcal{P} \in \mathsf{ap}(\sqrt{\mathsf{sat}_{i-1}(\; T_{X_i}^-)})$ we have $\mathsf{init}(\; T_{X_i}) \notin \mathcal{P}$. Let $r \in \mathbf{P}_i$ such that $r \in \mathsf{sat}_i(T)$. Then we have*

$$\mathsf{deg}(r, X_i) < \mathsf{mdeg}(\; T_{X_i}) \Rightarrow r \in \sqrt{\mathsf{sat}_i(\; T_{X_i}^-)}.$$

Proof. $\triangleright$ Define $h = \mathsf{init}(\; T_{X_i})$. First we assume that $\; T_{X_i}^- = \emptyset$. Then there exists $\delta \in \mathbb{N}$ such that $\; T_{X_i}$ divides $h^\delta r$. The hypothesis on the degree implies $r = 0$, which proves the assertion. Now let us assume that $\; T_{X_i}^- \neq \emptyset$ and denote $(\prod_{t \in \; T_{X_i}^-} \mathsf{init}(t))$ by h'. Since $r \in \mathsf{sat}_i(T)$ there exists $\delta \in \mathbb{N}$ and $q \in \mathbf{P}_i$ such that $(hh')^\delta r + q \; T_{X_i} \in \langle \; T_{X_i}^- \rangle_{\mathbf{P}_i}$. Let $\mathcal{P}$ be a prime ideal associated to $\sqrt{\mathsf{sat}_{i-1}(\; T_{X_i}^-)}$. It is a classical result that $h' \notin \mathcal{P}$. Since $h \notin \mathcal{P}$ by hypothesis, we have $\overline{hh'}^{\mathcal{P}}$ invertible. Therefore $\overline{\; T_{X_i}}^{\mathcal{P}}$ divides $\overline{r}^{\mathcal{P}}$. As we have $\mathsf{deg}(\overline{r}^{\mathcal{P}}, X_i) < \mathsf{deg}(\overline{\; T_{X_i}}^{\mathcal{P}}, X_i)$, we get $\overline{r}^{\mathcal{P}} = 0$, and the statement follows. $\triangleleft$

Proposition 3 *Let $i \in \mathbb{N}$ and T be a non-empty triangular set of $\mathbf{P}_i$ such that $X_i \in \mathsf{algVar}(T)$. Let us assume that for every $\mathcal{P} \in \mathsf{ap}(\sqrt{\mathsf{sat}_{i-1}(\; T_{X_i}^-)})$ we have $\mathsf{init}(\; T_{X_i}) \notin \mathcal{P}$. Let $f \in \mathbf{P}_i$. Then we have*

$$f \in \sqrt{\mathsf{sat}_i(T)} \quad \Longleftrightarrow \quad (\exists m \in \mathbb{N}) \;\mid\; \mathsf{prem}(f^m,\; T_{X_i}) \in \sqrt{\mathsf{sat}_i(\; T_{X_i}^-)}$$

$\mathsf{Proof.}$ $\triangleright$ We first consider $f \in \sqrt{\mathsf{sat}_i(T)}$. Let $m \in \mathbb{N}$ such that $f^m \in \mathsf{sat}_i(T)$. Then we clearly have $\mathsf{prem}(f^m,\ T_{X_i}) \in \mathsf{sat}_i(T)$, and the result immediately follows from proposition 2. Conversely, let m be an integer such that $\mathsf{prem}(f^m,\ T_{X_i}) \in \sqrt{\mathsf{sat}_i(\ T^-_{X_i})}$. We assume $T^-_{X_i} \neq \emptyset$, else the result is obvious. Let $h' = \prod_{t \in T^-_{X_i}} \mathsf{init}(t)$. There exists $\delta \in \mathbb{N}$ and $q \in \mathbf{P}_i$ such that $h^\delta f^m = q\ T_{X_i} + \mathsf{prem}(f^m,\ T_{X_i})$. We easily obtain from this equality that $h^\delta f^m \in \sqrt{\langle T \rangle_{\mathbf{P}_i}} : h'^\infty$. Thus we have $f^m \in \sqrt{\langle T \rangle_{\mathbf{P}_i}} : (hh')^\infty$, i.e. $f^m \in \sqrt{\mathsf{sat}_i(T)}$. It follows that $f \in \sqrt{\mathsf{sat}_i(T)}$. $\triangleleft$

Theorem 2 *Let $i \in \mathbb{N}$ and T be a regular chain in $\mathbf{P}_i$. Then we have*

$$\mathsf{Rep}_i(T) \ = \ \sqrt{\mathsf{sat}_i(T)}$$

$\mathsf{Proof.}$ $\triangleright$ For $i = 0$ the result is obvious. Let $i > 0$ and let us assume that the equality holds for $i-1$. If $X_i \notin \mathsf{algVar}(T)$, the equality easily follows from proposition 1 and remark 4. Now we assume that $X_i \in \mathsf{algVar}(T)$. From proposition 1 again, we have

$$\mathsf{Rep}_i(T) \ = \ \{f \in \mathbf{P}_i \ \mid \ (\exists m \in \mathbb{N})\ \mathsf{prem}(f^m,\ T_{X_i}) \in \mathsf{Rep}_i(\ T^-_{X_i})\}$$

Since $X_i \notin \mathsf{algVar}(\ T^-_{X_i})$ we know that $\mathsf{Rep}_i(\ T^-_{X_i}) \ = \ \sqrt{\mathsf{sat}_i(\ T^-_{X_i})}$ from the previous *transcendental case*. Finally we obtain the result with the proposition 3. $\triangleleft$

Proposition 4 *Let $i \in \mathbb{N}$ and T be a regular chain in $\mathbf{P}_i$. Then we have* $\mathsf{Rep}_i(T) \neq \mathbf{P}_i$.

$\mathsf{Proof.}$ $\triangleright$ It follows from both relations of proposition 1 that $1 \in \mathsf{Rep}_i(T)$ iff $1 \in \mathsf{Rep}_{i-1}(\ T^-_{X_i})$. Thus, since the statement is clear for $i = 0$, it also holds for any i. $\triangleleft$

1.3 Towers of Simple Extensions

From now on, $i \in \{0, \ldots, n\}$ is a integer, $\mathbf{k} = \mathbf{A}_0 \subseteq \mathbf{A}_1 \subseteq \cdots \subseteq \mathbf{A}_i$ are rings, $T \subseteq \mathbf{R}_i$ is a triangular set, and F_i is an algebra homomorphism of $\mathbf{P}_{i+1}$ into $\mathbf{A}_i[X_{i+1}]$.

Definition 5 *The set T is a* regular set *of $\mathbf{R}_i$ whose* associated map *is F_i and whose* associated tower of simple extensions *is $(\mathbf{A}_0, \ldots, \mathbf{A}_i)$ if one of the both assertions holds :*

(1) *$i = 0$, the set T is empty and F_i is the identity-map of $\mathbf{P}_1$*

(2) *$i > 0$, the set $T^-_{X_i}$ is a regular set of $\mathbf{R}_{i-1}$ whose associated tower of simple extensions is $(\mathbf{A}_0, \ldots, \mathbf{A}_{i-1})$ and whose associated map is denoted by F_{i-1} such that one of the both assertions holds :*

   (*i*) *$X_i \notin \mathsf{algVar}(T)$ and we have*

$$\mathbf{A}_i = \mathbf{q}(\mathbf{A}_{i-1}[X_i]) \ \text{ and } \ (\forall p \in \mathbf{P}_i)\ F_i(p) = \frac{F_{i-1}(p)}{1}$$

   (*ii*) *$X_i \in \mathsf{algVar}(T)$, the element $F_{i-1}(\mathsf{init}(\ T_{X_i}))$ is a unit in $\mathbf{A}_{i-1}$ and we have*

$$\mathbf{A}_i = \mathbf{q}(\ \mathbf{A}_{i-1}[X_i]/\langle F_{i-1}(\ T_{X_i})\rangle) \text{ and } (\forall p \in \mathbf{P}_i)\ F_i(p) = \frac{\overline{F_{i-1}(p)}^{\langle F_{i-1}(\ T_{X_i})\rangle}}{1}$$

More, in cases (i) and (ii), we state : $F_i(X_{i+1}) = X_{i+1}$.

Definition 6 *The sequence* $(\mathbf{A}_0, \ldots, \mathbf{A}_i)$ *is called a* tower of simple extensions *of* $\mathbf{k}$ *(t.o.s.e. for short) if there exists a regular set of* $\mathbf{R}_i$ *whose associated tower of simple extensions is* $(\mathbf{A}_0, \ldots, \mathbf{A}_i)$*. If* T *is a regular set of* $\mathbf{R}_i$ *whose associated t.o.s.e. is* $(\mathbf{A}_0, \ldots, \mathbf{A}_i)$ *the ring* $\mathbf{A}_i$ *is called the* top-extension of T.

Remark 5 Let $T \subseteq \mathbf{R}_i$ be a regular set whose associated t.o.s.e. is $(\mathbf{A}_0, \ldots, \mathbf{A}_i)$. For $0 \leq j \leq i$ and $x \in \mathbf{A}_j$, note that x is either a unit in $\mathbf{A}_j$ or a zero-divisor in $\mathbf{A}_j$. More, if $j < i$ and if x is a unit in $\mathbf{A}_j$ then it is also a unit in $\mathbf{A}_{j+1}$. Proposition 5 gives an important example of regular sets and proposition 6 characterizes the zero-divisors and units in the T's associated t.o.s.e.

Proposition 5 *Let* $T \subseteq \mathbf{R}_i$ *be a normalized triangular set. Then* T *is a regular set.*

Proof. $\triangleright$ If $i = 0$, the statement is clear. Thus, we can assume that $i > 0$ and that $T^-_{X_i}$ is a regular set. If $X_i \notin \mathsf{algVar}(T)$, the statement is clear again. If $X_i \in \mathsf{algVar}(T)$, we have $\mathsf{normalized?}(\mathsf{init}(T_{X_i}), T^-_{X_i})$. In order to show that $\mathsf{init}(T_{X_i})$ cannot be a zero-divisor in $T^-_{X_i}$'s associated t.o.s.e., it suffices to use remark 5 together with the following classical remark : for a ring $\mathbf{A}$, a polynomial $p \in \mathbf{A}[X]$ is a zero-divisor in $\mathbf{A}[X]$ iff there exists an element $a \in \mathbf{A}$ such that $a\,p = 0$. $\triangleleft$

Proposition 6 *Assume that* T *is a regular set of* $\mathbf{R}_i$ *whose associated map is* F_i *and whose associated t.o.s.e. is* $(\mathbf{A}_0, \ldots, \mathbf{A}_i)$*. Then, for every* $p \in \mathbf{P}_i$ *we have :*

(1) $F_i(p) = 0 \iff \mathsf{mod}(p, T) = 0 \iff \mathsf{prem}(p, T) = 0 \iff p \in \mathsf{sat}_n(T)$

(2) $F_i(p)$ *is a unit in* $\mathbf{A}_i$ *iff for every prime ideal* $\mathcal{P} \in \mathsf{ap}(\mathsf{sat}_n(T))$ *we have* $p \notin \mathcal{P}$.

Proof. $\triangleright$ The proof is based on the following classical remark. For an ideal I in a noetherian ring $\mathbf{A}$, for $x \in \mathbf{A}$, the element $\overline{x}^I$ is a zero-divisor in $\mathbf{A}/I$ iff there exists a prime ideal $\mathcal{P}$ associated to I such that x belongs to $\mathcal{P}$ ([SZ67], volume 1, p.214). $\triangleleft$

Theorem 3 *The triangular set* T *is a regular chain iff* T *is a regular set.*

Proof. $\triangleright$ The statement results easily from proposition 6 and theorem 2. $\triangleleft$

1.4 Lazard Sets

Remark 6 In [Laz91], Lazard introduced what we call *Lazard sets* (definition 9). A Lazard set is a particular regular set whose top-extension is a product of fields. The use of field products is motivated by definition 7 and proposition 7. Lazard sets are built by means of gcd computations (in the sense of definition 7) together with definition 8 and theorem 8. Full details will appear in [Maz97] and in a future paper.

Definition 7 *Let* $\mathbf{A}$ *be a ring and* p_1, p_2, g *be polynomials in* $\mathbf{A}[X]$*. We say that* g *is a gcd of* p_1 *and* p_2 *if the following holds :*

$$\langle p_1, p_2 \rangle_{\mathbf{q}(\mathbf{A})[X]} = \langle g \rangle_{\mathbf{q}(\mathbf{A})[X]}$$

Remark 7 If $\mathbf{q}(\mathbf{A})[X]$ is not a principal ideal domain, the polynomials p_1 and p_2 do not necessarily have gcd in the sense of the previous definition. They may also have several gcds. But if their leading coefficients are not zero-divisors in $\mathbf{A}$ then there exist $e, e' \in \mathbf{nz}(\mathbf{A})$ such that $e\,g = e'\,g'$. We chose this definition to generalize usual gcd algorithms which give a Bezout relation together with a pseudo-divisor (see [MR95]).

Proposition 7 ([MR95]) *Let $\mathbf{A}_1, \ldots, \mathbf{A}_i$ be integral domains and let $\mathbf{A}$ be their direct product (thus, sums and products in $\mathbf{A}$ are computed componentwise). Then for every p_1 and p_2 in $\mathbf{A}[X]$ there exists $g \in \mathbf{A}[X]$ which is a gcd of p_1 and p_2.*

Definition 8 *Let $\mathbf{A}$ be a ring of characteristic 0 and $p \in \mathbf{A}[X]$ with positive degree. We say that p is :*

(i) primitive *if the ideal of $\mathbf{A}$ generated by the coefficients of p is the unit-ideal.*

(ii) square-free *if p and its derivative generate the unit-ideal of $\mathbf{q}(\mathbf{A})[X]$.*

Proposition 8 *Let $\mathbf{A}$ be a noetherian ring of characteristic 0 and $p \in \mathbf{A}[X]$ with positive degree. Assume that $\mathbf{A}$ is a field or a product of fields and that p is monic and square-free (in the sense of the previous definition). Then, we have :*

(i) *the ideal generated by p in $\mathbf{A}[X]$ is a radical ideal.*

(ii) *each one of the rings $\mathbf{q}(\mathbf{A}[X])$ and $\mathbf{A}[X]/\langle p \rangle$ is a field or a product of fields.*

proof $\triangleright$ Property (i) is clear if $\mathbf{A}$ is a field. Assume now that $\mathbf{A}$ is a product of fields $\mathbf{k}_1 \times \cdots \times \mathbf{k}_n$. We denote by $\pi_i p$ the i-th component of p in $\mathbf{k}_1[X] \times \cdots \times \mathbf{k}_n[X]$ and by R_i the ideal of $\mathbf{A}[X]$ generated by $(1_1, \ldots, 1_{i-1}, \pi_i p, 1_{i+1}, \ldots, 1_n)$. Note that the R_i are relatively prime ideals and that their product is $\langle p \rangle_{\mathbf{A}[X]}$. Thus we have :

$$\langle p \rangle_{\mathbf{A}[X]} = \cap_1^n R_i$$

and property (i) follows from the fact that the R_i are radical ideals. Property (ii) results from the following remark of D. Lazard : if $\mathbf{A}$ is a noetherian ring where every element is either a unit or non-nilpotent zero-divisor then $\mathbf{A}$ is product of fields. This can be derived from the theory of Lazard rings (see [Maz97]).

Definition 9 *Let $T \subseteq \mathbf{R}_n$ be a regular set. The set T is called :*

(i) square-free *if for $1 \leq i \leq n$ we have : $X_i \in \mathsf{algVar}(T) \Longrightarrow F_{i-1}(\ T_{X_i})$ square-free,*

(ii) primitive *if for $1 \leq i \leq n$ we have : if $X_i \in \mathsf{algVar}(T)$ then for $1 \leq j < i$ the coefficients of the polynomial T_{X_i} viewed as a multivariate polynomial in $\mathbf{A}_j[X_{j+1}, \cdots, X_i]$. generate the unit-ideal of $\mathbf{A}_j$.*

A triangular set of $\mathbf{R}_n$ is called a Lazard set *if it is normalized, square-free and primitive. A t.o.s.e. is called* separable *if it is associated to a square-free regular set.*

Theorem 4 *Let $T \subseteq \mathbf{R}_n$ be a Lazard set and let $\mathbf{A}$ be its top-extension. Then the following assertions hold :*

(i) *$\mathbf{A}$ is a product of fields*

(ii) *$\mathsf{sat}_n(T)$ is a radical ideal*

(iii) *for every $p, q \in \mathbf{A}[X]$ there exists $g \in \mathbf{A}[X]$ such that g is a gcd of p and q*

proof $\triangleright$ Property (i), (ii), (iii) follow respectively from propositions 8, 6, and 7. $\triangleleft$

Remark 8 Let $T \subseteq \mathbf{R}_n$ be a Lazard set, F its associated map and $\mathbf{A}$ its top-extension. Assume that $\mathbf{A}$ is a product of m fields $\mathbf{k}_1 \times \cdots \times \mathbf{k}_m$. Let $p, q \in \mathbf{A}[X]$. To compute a gcd of p and q one may apply a standard algorithm in each $\mathbf{k}_i[X]$. But in practise the $\mathbf{k}_i$ are not known. So we perform in $\mathbf{A}[X]$ the variation of *subresultant gcd algorithm* proposed in [MR95] as if $\mathbf{A}$ was an integral domain. Then we use a *D5-like process* ([DDD85]) to split the computations when a zero-divisor is discovered. Let $r \in \mathbf{R}_n$ with $\mathsf{red?}(r, T)$. To decide whether the element $F(r)$ is a unit in $\mathbf{A}$ we proceed from the following way. If $r = 0$ the answer is false. Else, if $r \in \mathbf{k}$ the answer is true. Else, if $\mathsf{mvar}(r) \notin \mathsf{algVar}(T)$ the answer is given by checking the invertibility of $F(\mathsf{init}(r))$. Else the answer is given by checking the invertibility of the resultant of r and T_v w.r.t. $v = \mathsf{mvar}(r)$ where the coefficients of those polynomials are interpreted in the top-extension of T_v^-. This process is analogous to the one described in [MR95].

2 A review of the four Methods

In this section we first recall the specifications of each method together with the main properties of the decompositions that they compute. A complete review of the algorithms could not take place here. For Wu's method one can refer to [Wu87] or [Wan91]. However we summarize the main features of the methods of Lazard and Kalkbrener which both involve gcd computations over towers of simple extensions. Moreover, we give a recursive presentation of the first method proposed by D. Wang in [Wan93b] This adaptation appeared to us more concise than the original presentation.

2.1 Specifications

Let $F \subseteq \mathbf{R}_n$ be a finite set of polynomials. The algorithms of Wu and Lazard compute a finite family $\{T_1, \ldots, T_r\}$ of initially reduced triangular sets such that

$$\mathbf{V}(F) = \bigcup_{i=1}^{r} \mathbf{W}(T_i)$$

In the case of Wu's method, one of the T_i, say C, satisfies the following :

(i) $\mathbf{W}(C) \subseteq \mathbf{V}(F) \subseteq \mathbf{V}(C)$

(ii) $\mathbf{V}(F) = \mathbf{W}(C) \cup \bigcup_{p \in C} \mathbf{V}(F \cup \{\mathsf{init}(p)\})$

Such a triangular set is called a *characteristic set* for F ([Wu87]). In the case of Lazard's method, each T_i is a Lazard set. Lazard's decompositions (but not Wu's ones) are *irredundant* in the following sense :

$$\bigcup_{j \neq i} \mathbf{W}(T_j) \neq \bigcup_{j} \mathbf{W}(T_j)$$

Kalkbrener's method computes a finite family $\{T_1, \ldots, T_r\}$ of regular chains but deals rather with variety than regular zeros. The decomposition is such that

$$\mathbf{V}(F) = \bigcup_{i=1}^{r} \mathbf{V}(\mathsf{Rep}_n(T_i))$$

Thus by theorems 1 and 2 we also have

$$\mathbf{V}(F) = \bigcup_{i=1}^{r} \overline{\mathbf{W}(T_i)}$$

The proposition 4 guarantees that for every T_i we have $\mathbf{W}(T_i) \neq \emptyset$ but we may have $\bigcup_{j \neq i} \mathbf{W}(T_j) = \bigcup_j \mathbf{W}(T_j)$ for some i.

Wang's method computes a finite family $\{(T_1, Q_1), \ldots, (T_r, Q_r)\}$ of fine triangular q.a.s. such that

$$\mathbf{V}(F) = \bigcup_{i=1}^{r} \mathbf{Z}(T_i, Q_i).$$

Such a decomposition is produced by $\mathsf{trianguler}(F, \emptyset, \emptyset)$ (theorem 5). There is a no reason for a fine triangular system produced by the method of Wang described below (called *elimination without projection* in [Wan93b]) to be necessarily consistent. But, may be due to our optimizations, we never encountered inconsistent fine triangular system during our experiences. Note that Wang proposes also a method called *elimination with projection* in [Wan93b]) to produce necessarily consistent outputs.

2.2 Lazard's Method

The main procedure of Lazard's method is called $\mathsf{intersect}$. Given $T \subseteq \mathbf{R}_n$ and $p \in \mathbf{R}_n$ the operation $\mathsf{intersect}(p, T)$ returns a finite family of Lazard sets $\{S_1, \ldots, S_l\}$ such that

$$\mathbf{V}(p) \cap \mathbf{W}(T) \ \subseteq\ \cup_1^l \mathbf{W}(S_i) \ \subseteq\ \overline{\mathbf{V}(p) \cap \mathbf{W}(T)}$$

Given $\{T_1, \ldots, T_s\}$, a finite family of Lazard sets, we define $\mathsf{intersect}(p, \{T_1, \ldots, T_s\})$ as the union of the $\mathsf{intersect}(p, T_i)$. Then, given a finite subset $F = \{f_1, \ldots, f_m\}$ of $\mathbf{R}_n$ we define $\mathsf{intersect}(F, T) = \mathsf{intersect}(f_1, \mathsf{intersect}(\ldots, \mathsf{intersect}(f_m, T)))$. Thus $\mathsf{intersect}(F, \emptyset)$ produces a finite family of Lazard sets $\{S_1, \ldots, S_l\}$ such that

$$\mathbf{V}(F) = \bigcup_{i=1}^{r} \mathbf{W}(S_i)$$

We will not describe here how to produce irredundant decompositions. The operation $\mathsf{intersect}(p, T)$ proceeds in the following way.

(l_1) If $\mathsf{normalized?}(p, T)$ holds then go to step (l_2) with $r = p$ else go to next step.

($l_1{}'$) If $\mathsf{normalized?}(p, T)$ does not hold, compute two polynomials $q, r \in \mathbf{R}_{i+1}$ such that $\mathsf{normalized?}(r, T)$ and $\mathsf{mod}(pq - r, T) = 0$ and $\mathsf{mod}(p, T) = 0 \iff \mathsf{mod}(r, T) = 0$. Polynomials q and r are computed by means of an extended (i.e. with Bezout coefficients) version of the gcd algorithm sketched in remark 8. Here the computations may be split if $\mathsf{mod}(p, T)$ is a zero-divisor. The polynomial r is also denoted by $\mathsf{normalize}(p, T)$. Now, go to next step.

(l_2) If $r = 0$ then returns $\{T\}$. Else, if $r \in \mathbf{k}$ then returns $\{\,\}$. Else go to next step.

(l_3) Return $\mathsf{intersect}(\mathsf{tail}(r), \mathsf{intersect}(\mathsf{init}(r), T))$ and go to to next step.

(l_4) Remove the content of r viewed as univariate in $\mathsf{mvar}(r)$ and go to next step.

(l_5) If $T^{-}_{\mathsf{mvar}(r)} \cup \{r\}$ is a square-free regular set then go to step (l_7)

(l_6) Let $v = \mathsf{mvar}(r)$. Compute a (normalized w.r.t. T_v^-) gcd of r and its derivative w.r.t. v while interpreting their coefficients in the top-extension of T_v^- (here computations may be split). Let g be this gcd, replace r by $\mathsf{pquo}(r, g)$. Thus $T_v^- \cup \{r\}$ is now a square-free regular set. Go to step (l_3).

(l_7) Let $v = \mathsf{mvar}(r)$. Define $T_v^+ = \{t_k, \ldots, t_l\}$ with $\mathsf{mvar}(t_k) < \cdots < \mathsf{mvar}(t_l)$. Compute $\mathcal{D} = \mathsf{intersect}(t_l, \mathsf{intersect}(\ldots, \mathsf{intersect}(t_k,\ T_v^- \cup \{r\})))$. Then remove from $\mathcal{D}$ any triangular set U such that $\mathsf{normalize}(\mathsf{init}(t_i),\ U^-_{\mathsf{mvar}(t_i)}) = 0$ for some $i \in \{k, \ldots, l\}$. Now, go to next and last step.

(l_8) return $\mathsf{intersect}(p, \mathcal{D})$ where p is the input polynomial.

2.3 Kalkbrener's Method

Kalkbrener's Method is not so incremental as Lazard's one. We think that a good way to sketch this method is to give the algorithm of decomposition with the specifications of Kalkbrener's algorithm for computing gcd over towers of extensions ([Kal95]).

- algorithm $\mathsf{gcd}_n(C, F)$

Input: C a regular chain in $\mathbf{P}_{n-1}$ and F a finite subset of $\mathbf{R}_n$.

Output: $\{(C_1, g_1), \ldots, (C_s, g_s)\}$ where every C_k is a regular chain in $\mathbf{P}_{n-1}$ and every g_k is a polynomial in $\mathbf{R}_n$ such that

- $\bigcup_1^s \mathsf{ap}(\mathsf{Rep}_{n-1}(C_k)) = \mathsf{ap}(\mathsf{Rep}_{n-1}(C))$
- for all $\mathcal{P} \in \mathsf{ap}(\mathsf{Rep}_{n-1}(C_k))$,
  1. $F = \emptyset \ \Rightarrow s = 1$ and $g_1 = 0$,
     $F \neq \emptyset \ \Rightarrow\ \overline{g_k}^{\mathcal{P}}$ is the gcd of $\overline{F}^{\mathcal{P}}$ in $\mathbf{q}(\ \mathbf{P}_{n-1}/\mathcal{P})[X_n]$ for each k
  2. if $g_k \notin \mathbf{k}$ and $\mathsf{mvar}(g_k) = X_n$ then $\mathsf{init}(g_k) \notin \mathcal{P}$
     if $g_k \notin \mathbf{k}$ and $\mathsf{mvar}(g_k) < X_n$ then $g_k \notin \mathcal{P}$
  3. $g_k \in \langle \mathsf{Rep}_n(C_k) \cup F \rangle_{\mathbf{P}_n}$.

- algorithm $\mathsf{decompose}_n(F)$

Input: F a finite subset of $\mathbf{R}_n$

Output: regular chains $T_1, \ldots, T_r$ of $\mathbf{P}_n$ such that $\sqrt{\langle F \rangle_{\mathbf{P}_n}} = \cap_i \sqrt{\mathsf{sat}_n(T_i)}$

$\mathsf{decompose}_n(F)$ ==
    $F := F \setminus \{0\}$
    empty? F => $\{\emptyset\}$
    $F \cap \mathbf{R} \neq \emptyset$ => $\{\ \}$
    $\Theta := \emptyset$
    $F' := F \cap \mathbf{R}_{n-1}$
    $\Delta := \mathsf{decompose}_{n-1}(F')$
    for $C \in \Delta$ **repeat**
        $\Gamma := \mathsf{gcd}_n(C, F \setminus F')$
        for $(C_i, g_i) \in \Gamma$ **repeat**
            $g_i = 0$ => $\Theta := \Theta \cup \{C_i\}$
            $\mathsf{mvar}(g_i) < X_n$ => $\Theta := \Theta \cup \mathsf{decompose}_n(F \cup \{g_i\})$
            $\Theta := \Theta \cup \{C_i \cup \{g_i\}\} \cup \mathsf{decompose}_n(F \cup \mathsf{init}(g_i))$
    return Θ

2.4 Wang's Method

Let $\Sigma = (P, Q)$ a q.a.s. in $\mathbf{R}_n$ such that $\mathsf{mvar}(P) = X_i$. The algorithm $\mathsf{eliminer}$ presented below (proposition 9) splits the q.a.s. Σ into several q.a.s. which contain at most one equation with X_i as main variable (see definition 10). Its proof is based on the following lemma 1 ([Wan93a]) and lemma 2 (which is a practical remark whose proof is left to the reader).

Definition 10 *Let $1 \leq i \leq n$ and $\Sigma = (P, Q)$ a q.a.s. in $\mathbf{R}_n$ such that $P \subseteq \mathbf{R}_i$ and $Q \subseteq \mathbf{R}_n$. We call* elimination *of the variable X_i in Σ a set Λ of triplets (P_k, Q_k, τ_k) such that for any k, P_k, Q_k and τ_k are finite subsets of $\mathbf{R}_n$ and verify the following conditions :*

- *(i)* $P_k \neq \emptyset \Rightarrow \mathsf{mvar}(P_k) < X_i$
- *(ii)* $\tau_k \neq \emptyset \;\Rightarrow\; (\exists t \in \mathbf{R}_i \setminus \mathbf{R}_{i-1}) \;\mid\; \tau_k = \{t\}$
- *(iii)* $\mathbf{Z}(P, Q) = \bigcup_{(P_j, Q_j, \tau_j) \in \Lambda} \mathbf{Z}(P_j \cup \{\tau_j\}, Q_j)$.

Lemma 1 *Let f a non constant polynomial in $\mathbf{R}_n$ and (P, Q) a q.a.s. in $\mathbf{R}_n$. Then*

$$\mathbf{Z}(P \cup \{f\}, Q) = \mathbf{Z}(\mathsf{prem}(P, f) \cup \{f\}, Q \cup \{\mathsf{init}(f)\}) \;\cup\; \mathbf{Z}(P \cup \{\mathsf{init}(f), \mathsf{tail}(f)\}, Q).$$

Lemma 2 *Let (P, Q) be a q.a.s. in $\mathbf{R}_n$ and $f \in \mathbf{R}_n \setminus \mathbf{R}$. Then*

$$\mathsf{init}(f) \in Q \Rightarrow \mathbf{Z}(P \cup \{f\}, Q) = \mathbf{Z}(P \cup \{f\}, \mathsf{prem}(Q, f))$$

Proposition 9 *Let v be a variable in V and (P, Q) a q.a.s. in $\mathbf{R}_n$ such that* $\mathsf{mvar}(P) \leq v$*. Then the algorithm* $\mathsf{eliminer}(v, P, Q)$ *below computes an elimination of the variable v in the q.a.s. (P, Q). In particular, if the output of the algoritm is the empty set, then* $\mathbf{Z}(P, Q) = \emptyset$.

- $\mathsf{eliminer}(v, P, Q) ==$
  - $P := P \setminus \{0\}$
  - $(0 \in Q)$ **or** $(P \cap \mathbf{R} \neq \emptyset)$ => **return**$(\{\,\})$
  - $P_v = \emptyset$ => **return** $(\{(P, Q, \emptyset)\})$
  - $f :=$ a polynomial in P_v with minimal degree in v
  - $P_1 := (\,P_v \setminus \{f\}) \,\cup\, \{\mathsf{init}(f), \mathsf{tail}(f)\} \,\cup\, P_v^-$
  - $Q_2 := Q \cup \{\mathsf{init}(f)\}$
  - **empty?** $(\,P_v \setminus \{f\})$ => **return** $(\,\{(\,P_v^-, \mathsf{prem}(Q_2, f), \{f\})\} \,\cup\, \mathsf{eliminer}(v, P_1, Q)\,)$
  - $P_2 := \mathsf{prem}(\,P_v \setminus \{f\}, f) \,\cup\, \{f\} \,\cup\, P_v^-$
  - **return** $(\,\mathsf{eliminer}(v, P_2, Q_2) \,\cup\, \mathsf{eliminer}(v, P_1, Q)\,)$

Proof. $\triangleright$ We will prove termination and correctness by induction on $s(P) = \sum_{p \in P \setminus \{0\}} \mathsf{deg}(p, v)$.

If a constant occurs in P or $0 \in Q$, the result is obvious. Else, if $s(P) = 0$, then $P_v = \emptyset$ and the algorithm terminates. The correctness is obvious. Now we assume that $s(P) > 0$, i.e. P_v is not empty. First we remark that $s(P_1) < s(P)$ since $\mathsf{deg}(\mathsf{init}(f), v) = 0$ and $\mathsf{deg}(\mathsf{tail}(f), v) < \mathsf{deg}(f, v)$. Then two cases can be distinguished :

- $P_v = \{f\}$. By induction, $\mathsf{eliminer}(v, P_1, Q)$ terminates and is correct. Therefore $\mathsf{eliminer}(v, P, Q)$ terminates. The correction follows from application of lemma 1 and lemma 2.

- $P_v \setminus \{f\} \neq \emptyset$. Let us denote $P_v \setminus \{f\}$ by P'. For any $p \in P'$, we have $\mathsf{deg}(\mathsf{prem}(p, f), v) < \mathsf{deg}(f, v) \leq \mathsf{deg}(p, v)$. Since P' is not empty, we thus obtain $s(\mathsf{prem}(P', f)) < s(P')$, and consequently $s(P_2) < s(P)$. Then termination and correctness follow by application of lemma 1 and induction hypothesis. $\triangleleft$

By *decreasing* use of the algorithm eliminer, we easily obtain a triangulation of any q.a.s. as we can see now with the following algorithm.

Theorem 5 *Let $1 \leq i \leq n$ and (P, Q) a q.a.s. in $\mathbf{R}_n$ such that $P \subseteq \mathbf{R}_i$. Let T a triangular set of $\mathbf{R}_n$ such that $T \cap \mathbf{R}_i = \emptyset$. Then the following algorithm* triangular(P, Q, T) *computes a finite family $\{(T_1, Q_1), \ldots, (T_r, Q_r)\}$ of triangular q.a.s. such that*

$$\mathbf{Z}(P \cup T, Q) = \bigcup_{k=1}^{r} \mathbf{Z}(T_k, Q_k).$$

- triangular(P, Q, T) ==
  - $P := P \setminus \{0\}$
  - $(0 \in Q)$ **or** $(P \cap \mathbf{R} \neq \emptyset)$ => **return** $(\{\})$
  - **empty?** P => **return** $(\{(T, Q)\})$
  - $v :=$ mvar(P)
  - $\Lambda :=$ eliminer(v, P, Q)
  - **return** $(\bigcup_{(P_j, Q_j, \tau_j) \in \Lambda}$ triangular$(P_j, Q_j, \tau_j \cup T))$

Proof. $\triangleright$ The proof of the algorithm is obtained by induction on the smallest integer such that $P \subseteq \mathbf{R}_i$, which we will denote by $i(P)$. For $i(P) = 0$, i.e. $P \subseteq R$, the result is obvious. Now assume that $i(P) > 0$. We can eliminate the cases $0 \in Q$, $P \cap \mathbf{R} \neq \emptyset$, and $\Lambda = \emptyset$, which terminate immediately and are clearly correct. Then by specifications of the algorithm eliminer, we obtain

$$\mathbf{Z}(P \cup T, Q) = \mathbf{Z}(P, Q) \cap \mathbf{V}_K(T) = \bigcup_{(P_j, Q_j, \tau_j) \in \Lambda} \mathbf{Z}(P_j \cup (\{\tau_j\} \cup T), Q_j).$$

Now we state $T_j = \{\tau_j\} \cup T$. The triplets (P_j, Q_j, T_j) satisfy the input conditions of triangular. And since $i(P_j) < i(P)$, the result follows from induction. $\triangleleft$

3 Implementation

3.1 General Requirements

In the introduction we gave three requirements in order to make a fair comparison of the methods for polynomial system solving. The most important is to implement and run the corresponding algorithms with the same human, material and software conditions. More generally, given a system of equations to be solved, we want that the difference between the corresponding computations only depend on the difference between the corresponding algorithms. In particular, we want our implementations of those methods to use the same data structures and sub-routines. We will describe this last point below.

Another important requirement is to make sure that each computed solution (by one of the implementations of the four methods) is correct. We concentrated on this last point instead of the search of very optimized implementations. We think that only checking *by hand* some computations (necessarily simple) produced by an implementation is not sufficient to make sure that this implementation is correct, especially for mixed dimensional problems. We had a wrong implementation of Wu's method during three years (solving Liu's example in 147 sec) due to a programming mistake in the management of the elimination of the redundant branches. Thanks to our checking process (to be described below) we discovered this bug. However our current implementation of Wu's method does not solve Liu's example any more.

This checking process

- has been intensively tested for more than one year,
- is based on simple and well known algorithms and
- is implemented in a direct way in AXIOM as an over-level of the GB software ([Fau94])

Thus it will be considered as *certainly reliable.*

In our analysis of the computed solutions we also look for other informations than timings or correctness. Given a solution, we want to know if some of the computed triangular sets are inconsistent or if some quasi-components are contained in another quasi-component (or in the closure of another quasi-component). This could also be done as we will see.

3.2 Description of the implementation

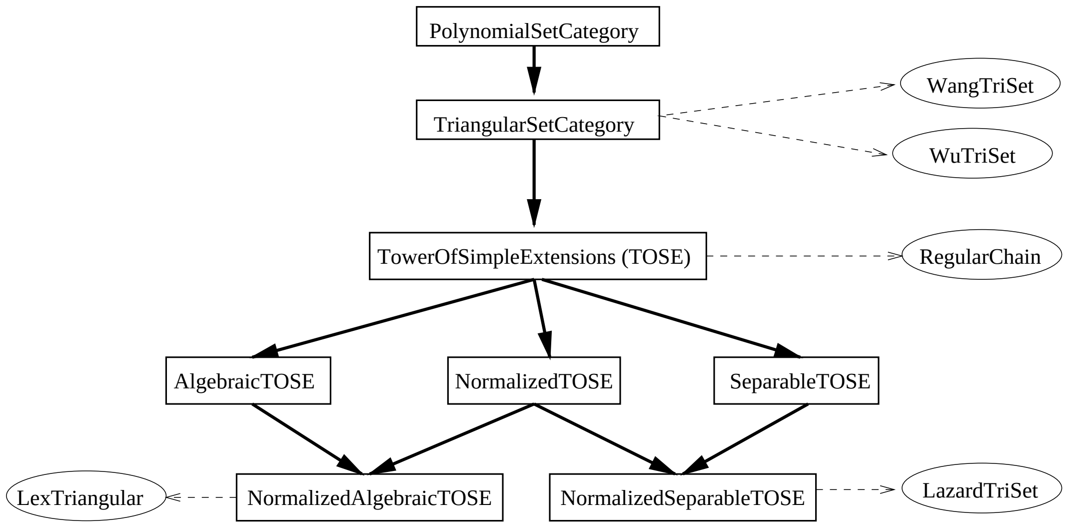
Each implementation of the four methods uses the same AXIOM domain for polynomials (with a sparse and recursive representation). We first defined an AXIOM category for finite subsets of $\mathbf{R}_n$. This category exports and implements operations on sets, ideals and varieties like $(I, J) \longmapsto I \cap J$ and $(I, p) \longmapsto I : p^{\infty}$ where $I, J \subseteq \mathbf{R}_n$ denote ideals and $p \in \mathbf{R}_n$ is a polynomial. We implement these operations by means of Gröbner bases techniques ([CLO91]) in an AXIOM package using the connection between AXIOM and GB, the very powerful Gröbner engine developed by J.C. Faugère ([Fau94]). Then we wrote an AXIOM category for (general) triangular sets of $\mathbf{R}_n$. This category exports and implements basic operations like $(T, v) \longmapsto T_v$ and $(p, T) \longmapsto \mathsf{prem}(p, T)$ and $(p, T) \longmapsto \mathsf{iRed}(p, T)$ (notation 3 and notation 4) where v is a variable and $T \subseteq \mathbf{R}_n$ is a triangular set. It also exports and implements more sophisticated operations like :

- $T \longmapsto \mathsf{sat}_n(T)$ in order to check consistency of a triangular set,
- $(F \subseteq \mathbf{R}_n, \{T_1, \ldots, T_r \subseteq \mathbf{R}_n\}) \longmapsto \mathbf{V}(F) \stackrel{?}{=} \cup_i \overline{\mathbf{W}(T_i)}$ in order to check the correctness of a computed decomposition.

Moreover this category exports (but does not implement) an operation $F \subseteq \mathbf{R}_n \longmapsto \mathsf{zeroSetSplit}(F)$ which represents any method for solving polynomial system by means of triangular sets. From the category of general triangular sets we derived a category for towers of simple extensions. It exports the associated map of a t.o.s.e. implemented with the operation $(p, T) \longmapsto \mathsf{mod}(p, T)$ (notation 4). It also exports operations like $(p, T) \longmapsto$ is-$\mathsf{mod}(p, T)$-a-unit ?. Finally, from the category of t.o.s.e. we derived three categories corresponding to particular properties of regular sets :

- a category for the regular sets $T \subseteq \mathbf{R}_n$ such that $\mathsf{algVar}(T) = \{X_1, X_2, \ldots, X_n\}$. called *algebraic t.o.s.e.*,
- a category for the normalized regular sets called *normalized t.o.s.e.* and
- a category for the square-free regular sets called *separable t.o.s.e.*

Now, each method is an implementation of the operation $F \subseteq \mathbf{R}_n \longmapsto \mathsf{zeroSetSplit}(F)$ in an AXIOM domain of the suitable category. For instance, Kalkbrener's method is implemented in an domain which belongs to the t.o.s.e category and which is called *RegularChains* (see the picture below). Note that we implemented the lexTriangular algorithm ([MR95]) in an AXIOM domain called *LexTriangular* and which belongs to both categories of normalized t.o.s.e. and algebraic t.o.s.e.



4 Examples

We now present two tables of results of our experiments. They are respectively dedicated to dimension zero and positive dimension. We give below the sources of our examples. For every example F and every method which decomposes $\mathbf{V}(F)$ into triangular systems $\sigma_1, \ldots, \sigma_r$ we give two informations. The first one is the computing time (evaluation and garbage collector). The second one :

- is $n(\sigma_1), \ldots, n(\sigma_r)$ where $n(\sigma_i)$ denotes the number of solutions of σ_i, if $\mathbf{V}(F)$ has dimension 0
- else $d(\sigma_1), \ldots, d(\sigma_r)$ where $d(\sigma_i)$ denote the dimension of $\mathsf{sat}_n(\sigma_i)$

In order to make more concise these sequences of numbers we use some notations. Let us take the example 11 with Wang's method in the first table. The sequence $2^2, 4, 16^2$ means that the decomposition contains two triangular sets with 16 solutions, one triangular set with 4 solutions and two triangular sets with 2 solutions. The same kind of notation applies for sequences of dimensions. Futhermore in that case, we precise the inclusions between the saturated ideals of the components (when these inclusions could be computed).

0-dim Exp.	Wang	Wu	Kalkbrener	Lazard
1	0.48	0.33	0.27	0.20
	4	4	2^2	2^2
2	0.48	0.30	0.93	0.98
	8	8	8	8
3	0.65	2.97	3.45	0.45
	2	2	2	2
4	0.92	103	1.68	429
	10	10	10	10
5	1.27	7.10	3.02	88
	10	10	1,7	1,7
6	0.12	0.13	0.22	0.25
	$2,3$	$2,4$	$1^3,2$	$1^3,2$
7	36.3	?	99.6	124.5
	$1,2,6,8,18,32^2$		$1^5,2^4,8,12,16,32$	$1^5,2^4,8,12,16,32$
8	0.33	?	1.08	0.47
	$1,120$		$1,120$	$1,120$
9	0.57	?	24.13	0.73
	1,720		1,720	1,720
10	0.73	?	11000	0.95
	1,5040		1,5040	1,5040
11	43.50	?	6.02	17.55
	$2^2,4,16^2$		$2,4,6,8,12$	$4^2,8,16$
12	1.50	?	2.23	0.62
	$4,6$		$1,3,4$	$1^2,2,4$
13	?	?	781	22
			$10^5,20$	$10^5,20$

pos.dim exp.	Wang	Wu	Kalkbrener	Lazard
14	1.30	4.88	4.6	0.87
	$2_a \subset 3_a, 2_b \subset 3_b$	$2_a \subset 3_a, 2_b \subset 3_b$	$2 \subset 3_a, 3_b$	3^2
15	0.28	0.35	0.17	0.23
	$1^2 \subset 2$	$1^2 \subset 2$	1	1
16	0.45	0.47	1.33	0.70
	$3^3 \subset 4$	$3^3 \subset 4$	$4,4,4$	$3^3 \subset 4$
17	0.37	0.90	0.67	0.60
	$0^2 \subset 1$	$0^2 \subset 1$	1	$0^2 \subset 1$
18	0.77	0.40	0.25	0.40
	$2^2 \subset 3$	$2^2 \subset 3$	3	$2^2 \subset 3$
19	0.70	4.98	1.18	0.94
	$0 \subset 2, 1^3 \subset 2$	$0^4 \subset 2, 1^3 \subset 2$	2	$1^2 \subset 2$
20	1.30	1.25	2.18	0.77
	$2^3,3$	$-1,2^3,3$	$1^5 \subset 2^2 \subset 3, 2^3$	$2^3,3$
21	2.15	9.05	4.40	8.25
	1^3	1^2	1^3	1^4
22	0.20	0.13	0.50	7
	$0 \subset 1$	$0 \subset 1$	1	$0^5 \subset 1$
23	5.77	140	24.80	10.83
	$1,2^4,3^4 \subset 4$	$0^4,1^{12},2^{14},3^5 \subset 4$	4	$2^2 \subset 4, 3^3 \subset 4_a$
24	9.6	?	38	5.75
	$0,2^2,3^4$		$0,1^2,2^8,3^3$	$0,2^2,3^3$
25	4.97	27.47	76.37	68
	$2^2,3^8,4^5,5$	$2^{12},3^{13},4^5,5$	5	$2,3^7,4^5,5$
26	8.68	?	13.90	142
	$1^3,2$		1,2	$0,1^4,2$
27	?	?	50	?
			1^2	
28	≈ 2000	?	?	?

Ex.	Source or description
1	Solotare [Com92]
2	Moeller 4 [Com92]
3	Trinks 2 [BGK86]
4	Trinks 1 [BGK86]
5	Katsura 3 [BGK86]
6	system $L_2 = \{x_1^2 + x_2 + x_3 - 1, x_1 + x_2^2 + x_3 - 1, x_1 + x_2 + x_3^2 - 1\}$ with $x_1 < x_2 < x_3$.
7	system $L_3 = \{x_1^3 + x_2 + x_3 + x_4 - 1, x_1 + x_2^3 + x_3 + x_4 - 1, x_1 + x_2 + x_3^3 + x_4 - 1, x_1 + x_2 + x_3 + x_4^3 - 1\}$ with $x_1 < \cdots < x_4$.
8	system $R_5 = \{x_1(x_1+1), (x_2^2 + x_2 + 1)x_1 + x_2, p_3, p_4, p_5\}$ where $p_i = x_i x_{i-1}^{i-1} + (x_i^i + 1)x_{i-1} + x_i$ and $x_1 > \cdots > x_5$.
9	system $R_6 = \{x_1(x_1+1), (x_2^2 + x_2 + 1)x_1 + x_2, p_3, p_4, p_5, p_6\}$ with $x_1 > \cdots > x_6$.
10	system $R_7 = \{x_1(x_1+1), (x_2^2 + x_2 + 1)x_1 + x_2, p_3, p_4, p_5, p_6, p_7\}$ with $x_1 > \cdots > x_7$.
11	Caprasse [Com92]
12	Singular Points : $F = \{f, \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\}$ where $f = (y-x)(y^2+x^2-1)(y^2-x)$.
13	Cyclic 5 [Laz92a]
14	Discriminant degree 4 (ex. 49 in [Wan91])
15	Cyclic-4 [Laz92a]
16	Buchberger [Com92]
17	Donati-Traverso [Wan93b]
18	Alonso [Com92]
19	Robot Plano fácil (see introduction)
20	Euler theorem [Dia92]
21	Wang [Wan92a]
22	Wu [Wan92a]
23	Robot Plano difícil $F = \{s_1^2 + c_1^2 - 1, s_2^2 + c_2^2 - 1, (l_3c_2 + l_2)s_1 + l_3s_2c_1 - b, -1 - l_3s_2s_1 + (l_3c_2 + l_2)c_1 - a\}$ with $b < a < l_3 < l_2 < c_2 < s_2 < c_1 < s_1$.
24	Butcher [BGK86]
25	Robot Romin $F = \{-ds_1 - a, dc_1 - b, l_2c_2 + l_3c_3 - d, 2s_2 + l_3s_3 - c, s_1^2 + c_1^2 - 1, s_2^2 + c_2^2 - 1, s_3^2 + c_3^2 - 1\}$ with $d < c < b < a < l_3 < l_2 < c_3 < s_3 < c_2 < s_2 < c_1 < s_1$
26	f633 [FdSMR96]
27	Neural Network [Kal91]
28	Liu [Liu89]

Let us examinate the outputs of two examples from the tables. Alonso example(18) corresponds to a prime ideal of dimension 3 whose Kalkbrener's method describes with only one regular chain C. The other algorithms extract points which are in the closure of the regular zeros of C and provide similar results. We effectively verified in these cases that both the outputs of dimension 2 where contained in $\overline{\mathbf{W}(C)}$.

- Wang's method :
  $\{(\{r, t^2u+1, v, z+t, (2t^2+1)x + t^2u + 2t^6 - t^2\}, [2t^2+1, t]), (\{v - tr, tuz - 1, ry - t^2u - 1, (u - r - 2)x - u - 2t^4 + 1\}, [u, r, t, u - r - 2]), (\{r + 2t^4 + 1, u + 2t^4 - 1, v + 2t^5 + t, (2t^5 - t)z + 1, (2t^4+1)y - 2t^6 + t^2 + 1\}, [2t^4+1, 2t^4 - 1, t])\}$
- Wu's method :
  $\{\{r + 2t^4 + 1, u - r - 2, v - tr, (2t^5 - t)z + 1, (2t^4+1)y + t^2u + 1\}, \{r, t^2u + 1, v - tr, tz + t^2, (2t^2+1)x + t^2u + 2t^6 - t^2\}, \{v - tr, tuz - 1, ry - t^2u - 1, (u - r - 2)x - u - 2t^4 + 1\}\}$
- Kalkbrener's method :
  $\{\{v - tr, tuz - 1, ry - t^2u - 1, (u - r - 2)x - u - 2t^4 + 1\}\}$
- Lazard's method :
  $\{\{r + 2t^4 + 1, u + 2t^4 - 1, v + 2t^5 + t, (2t^5 - t)z + 1, (2t^4+1)y - 2t^6 + t^2 + 1\}, \{v - tr, tuz - 1, ry - t^2u - 1, (u - r - 2)x - u - 2t^4 + 1\}, \{r, t^2u + 1, v, z + t, (2t^2+1)x + 2t^6 - t^2 - 1\}\}$

The following example (Singular points on a curve, example 12) of dimension zero is slightly different. This example seems not too difficult but Wu's method failed. Here the methods give different results and we can note that Lazard's decomposition is efficient for timing and legibility.

- Wang's method :
$\{(\{x^2+x-1, xy^2+x-1\}, \emptyset), (\{2x^4-2x^3-x^2+x,$
$(46x^5+48x^4-64x^3-24x^2+18x+2)\,y-48x^5-51x^4+70x^3+28x^2-25x\}, \emptyset)\}$
- Kalkbrener's method :
$\{\{2x^3-2x^2-x+1, (3448x^2-17x-1711)\,y-3431x^2-13x+1724\}$
$\{x^2+x-1, (426x-265)\,y^2+691x-426\}, \{x, y\}\}$
- Lazard's method :
$\{\{x^2+x-1, y^2-x\}, \{2x^2-1, y-x\}, \{x-1, y-1\}, \{x, y\}\}$

Conclusions

For easy examples, we remark that all methods generally have good computing times and that the legibility of the outputs they produce is satisfactory. Nevertheless Wu's method failed in some rather easy zero-dimensionnal examples, namely Caprasse, R5. Futhermore, for both cases of dimension 0 and positive dimension Wu's method solves clearly less problems than the other methods. And for the most difficult examples Wu's method can solve, the outputs are hard to read (see Robot Romin). In our opinion, the reason is the following. Wu's method cannot split the computations (in order to obtain several triangular sets) before computing a characteristic set of F (which is sometimes hard to compute, especially for zero-dimensionnal problems) whereas the other methods may split their computations earlier. More generally, it seems that methods based on gcd computations over tower of simple extensions, namely those of Lazard and Kalkbrener, may discover factorizations that other methods cannot find (Cyclic-5).

Let us now concentrate on Wang's method. This method may be very efficient for difficult examples. But, the produced outputs are generally less legible than the ones of Kalkbrener and Lazard. Futhermore, as Wu's method, the method of Wang is disappointing in not too difficult zero-dimensional examples, namely Caprasse and Cyclic-5. Note that whereas our implementation of Wu's method produced some inconsistent triangular sets, this never happened with our implementation of Wang's method.

Kalkbrener's method is the only method which solves every example except `Liu` and often produces the most concise outputs (except for Cyclic-5). Futhermore, this method has the best timings for difficult problems lile f633, Robot Romin, Neural Network. But one has to keep in mind that this method solves polynomial systems in a *more lazy way* than the other three. This method is also inefficient for some zero-dimensional examples (Cyclic-5, R_7) whereas Lazard's method succeeds with these examples. The reason seems to be the use of normalization (in Lazard's method) which can replace big algebraic expressions by a single integer number in zero-dimensional examples .

However, normalization and square-free factorization over tower of separable extensions are time consuming. This is the reason why Lazard's method may also be inefficient in some not difficult examples (Katsura 3, Trinks difficult). For describing affine varieties by means of regular zeros of triangular sets, Lazard's method gives the best outputs. Moreover, this is the only method which produces irredundant decompositions. We think that the methods based on computation of gcd over tower of extensions are promising. The experiments show that they must be further investigated for more efficiency. A future challenge consists in using our practice of the algorithms of Kalkbrener and Lazard to take advantage of both methods and resolve more difficult problems.

References

[BGK86] W. Boege, R. Gebauer, and H. Kredel. Some examples for solving systems of algebraic equations by calculating groebner bases. *J. Symb. Comp.*, 2:83–98, 1986.

[Buc65] B. Buchberger. *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal.* PhD thesis, Innsbruck, 1965.

[CG90] S.C. Chou and X.S. Gao. Ritt-Wu's decomposition algorithm and geometry theorem proving. In *Proc. CADE-10*, pages 202–220, Kaiserslautern, Germany, 1990.

[CG92] S.C. Chou and X.S. Gao. Solving parametric algebraic systems. In *Proc. ISAAC'92*, pages 335–341, Berkeley, California, 1992.

[Cho88] S.C. Chou. *Mechanical Geometry Theorem Proving.* D. Reidel Publ. Comp., Dordrecht, 1988.

[CLO91] D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms.* Spinger-Verlag, 1991.

[Com92] European Commission. *PoSSo - Polynomial System Solving Research Project.* Esprit Scheme Project No. 6846, 1992. Has been extended to the FRISCO project.

[DDD85] J. Della Dora, C. Discrescenzo, and D. Duval. About a new method method for computing in algebraic number fields. In *Proc. EUROCAL 85 Vol. 2*, volume 204 of *Lect. Notes in Comp. Sci.*, pages 289–290. Springer-Verlag, 1985.

[Dia92] Teresa Gomez Diaz. *Quelques applications de l'évaluation dynamique.* Université de Limoges, 1992. Thèse de Doctorat.

[Fau94] J.C. Faugère. *Résolution des systèmes d'équations algébriques.* Université Paris 6, 1994. Thèse de Doctorat.

[FdSMR96] J.C. Faugère, F. Moreau de Saint Martin, and F. Rouiller. Design of nonseparable bidimensional wavelets and filter banks using Gröbner bases techniques. *IEEE Trans. in Signal Processing*, 1996. preprint.

[GM90] G. Gallo and B. Mishra. Efficient algorithms and bounds for Wu-Ritt characteristic sets. In *Proc. MEGA'90*, pages 119–142, 1990.

[GTZ88] P. Gianni, B. Trager, and G. Zacharias. Gröbner Bases and Primary Decomposition Of Polynomial Ideals. *J. Symb. Comp.*, 6:149–167, 1988.

[JS92] Richard D. Jenks and Robert S. Stutor. *AXIOM, The Scientific Computation System*. Springer-Verlag, 1992. AXIOM is a trade mark of NAG Ltd, Oxford UK.

[Kal91] M. Kalkbrener. *Three contributions to elimination theory*. PhD thesis, Johannes Kepler University, Linz, 1991.

[Kal93] M. Kalkbrener. A generalized euclidean algorithm for computing triangular representations of algebraic varieties. *J. Symb. Comp.*, 15:143–167, 1993.

[Kal95] M. Kalkbrener. Algorithmic properties of polynomial rings. Master's thesis, Dep. of math., Swiss Federal Institute of Technology,Zurich, 1995.

[Laz91] D. Lazard. A new method for solving algebraic systems of positive dimension. *Discr. App. Math*, 33:147–160, 1991.

[Laz92a] D. Lazard. Solving zero-dimensional algebraic systems. *J. Symb. Comp.*, 15:117–132, 1992.

[Laz92b] D. Lazard. Terminology for triangular and characteristic sets. Technical report, Université Paris 6, 1992. PoSSo report.

[Liu89] Z.J. Liu. An algorithm on finding all isolated zeros of polynomial equations. *MM Research Preprints*, 4(63-76), 1989.

[Maz97] M. Moreno Maza. *Calculs de Pgcd au-dessus des Tours d'Extensions Simples et Résolution des Systèmes d'Équations Algébriques*. PhD thesis, Université Paris 6, 1997. Preprint.

[MR95] M. Moreno Maza and R. Rioboo. Polynomial gcd computations over towers of algebraic extensions. In *Proc. AAECC-11*, pages 365–382. Springer, 1995.

[SZ67] P. Samuel and O. Zariski. *Commutative algebra*. D. Van Nostrand Company, INC., 1967.

[Wan91] D. M. Wang. On Wu's method for solving systems of algebraic equations. Technical Report RISC-LINZ Series no 91-52.0, Johannes Kepler University, Austria, 1991.

[Wan92a] D. M. Wang. An implementation of the characteristic set method in Maple. In *Proc. DISCO'92*, Bath, England, 1992.

[Wan92b] D. M. Wang. Some improvements on Wu's method for solving systems of algebraic equations. In Wu Wen-Tsün and Cheng Min-De, editors, *Proc. of the Int. Workshop on Math. Mechanisation*, Beijing, China, 1992. Institute of Systems Science, Academia Sinica.

[Wan93a] D. M. Wang. An elimination method based on Siedenberg's theory and its applications. In F. Eysette and A. Galligo, editors, *Comptutational Algebraic Geometry*, pages 301–328. Birkhäuser Boston, Inc., 1993.

[Wan93b] D. M. Wang. An elimination method for polynomial systems. *J. Symb. Comp.*, 16:83–114, 1993.

[Wu87] W. T. Wu. A zero structure theorem for polynomial equations solving. *MM Research Preprints*, 1:2–12, 1987.