EPiC Series in Computing
Volume 39, 2016, Pages 169–182
SCSS 2016. 7th International Symposium on Symbolic Computation in Software Science

Bit-size reduction of triangular sets in two and three variables

Yamashita Tetsuro[1] and Dahan Xavier[2*†]

[1] Fujitsu company
[2] Faculty of General Educational Research, Ochanomizu University, Tokyo, JAPAN.
xdahan@gmail.com

Abstract

At ISSAC 2004 [4] was introduced a transformation of a triangular lexicographic Gröbner basis generating a radical ideal of dimension zero, to a triangular family of polynomials generating the same ideal, which is no more a Gröbner basis but has significantly smaller coefficients in term of bit-size. We attempt in this article to extend this transformation to triangular sets that do not generate a radical ideal. We manage to treat the case of $n = 2$ variables, and in some extent the case of $n = 3$ variables. It resorts to an extra operation, the squarefree factorization; nevertheless this operation based on gcd benefits of efficient algorithms. When the number of variables n is greater than 2, more serious difficulties occur and are discussed for $n = 3$. An implementation in Maple in the case $n = 2$ confirms the expected reduction of the bit-size coefficients.

1 Introduction

A reduced lexicographic Gröbner basis for the monomial order $x_1 \prec x_2 \prec \cdots \prec x_n$ over a field K is *triangular* if all of its polynomials have a leading monomial that is a pure power of a variable, which are all pairwise distinct. It implies that the basis is a *regular sequence*. When there are n polynomials $T_1, T_2, \ldots, T_n$ in such a Gröbner basis, and they are assumed to be ordered so that $\mathrm{LM}(T_1) \prec \mathrm{LM}(T_2) \prec \cdots$ it can be written as:

$$\mathbf{T}\begin{cases} T_n(x_1, x_2, \ldots, x_{n-1}, x_n) = x_n^{d_n} + a_{n,n-1}(x_1, \ldots, x_n - 1)x_n^{d_{n-1}} \cdots \\ T_{n-1}(x_1, \ldots, x_{n-1}) = x_{n-1}^{d_{n-1}} + \cdots \\ \vdots \\ T_1(x_1) = x_1^{d_1} + \cdots \end{cases}, \tag{1}$$

where $\cdots$ stands for lower terms for $\prec$. It is common to call such a family *triangular sets*. Such simple data structures are at the core of the *triangular decomposition* method (see [8,

*Commissioned by "Strategic Information and Communications R&D Promotion Programme (SCOPE), No. 0159-0016" Ministry of Internal Affairs and Communications, JAPAN.

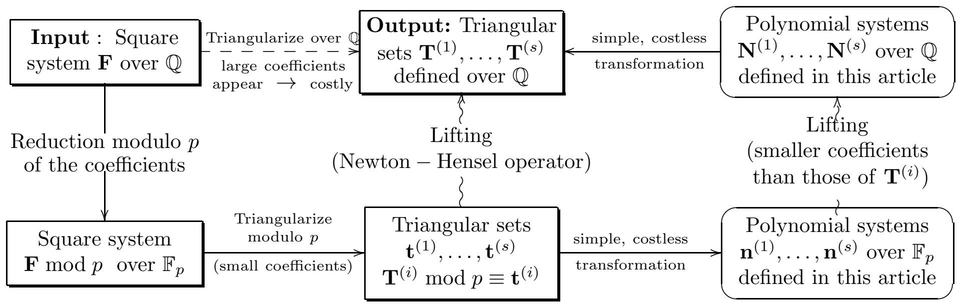
†And supported by Kakenhi research grant No.15K16003

1, 2, 7] among many others) to solve and manipulate polynomial systems. Note that within the assumptions made, *i.e.* a finite number of solutions, *monic* polynomials as those in (1) naturally appear, or, if not, a mere inversion in the base field K allows to reduce to monic polynomials. We focus mainly on the case of $K = \mathbb{Q}$ or $K = \mathbb{F}_p$ (see "Motivation" below). The transformation introduced in [4]

$$T_1 \to N_1, \qquad T_i \to N_i := \left(\prod_{\ell=1}^{i-1} \frac{\partial T_\ell}{\partial x_\ell}\right) T_i \bmod (T_1, \ldots, T_{i-1}), \tag{2}$$

allows to compute at a cheap cost polynomials with smaller coefficients. More precisely, the former has coefficients bit-size that grows at most in $O(d^{2n})$ where $d = \prod_i \deg_{x_i}(T_i)$, whereas the latter has coefficients bit-size that grows at most in $O(d^n)$. The two ideals $\langle T_1, \ldots, T_n\rangle$ and $\langle N_1, \ldots, N_n\rangle$ are equal but the second family is no longer a Gröbner basis. Such a transformation is implemented in Maple in the `RegularChains` library under the name `DahanSchostTransform`.

Motivation Despite not Gröbner bases, those families of polynomials find applications in *modular methods*, where instead of lifting the coefficients of T_i, we lift the coefficients of N_i which saves a significant amount of time. The figure below displays the concept of a modular method for a *triangular decomposition*. Instead of lifting directly the triangular sets modulo p denoted $\mathbf{t}^{(i)} = (t_1^{(i)}(x_1), t_2^{(i)}(x_1, x_2), \ldots, t_n^{(i)}(x_1, \ldots, x_n))$ in the figure below, it is more efficient to transform them into the polynomial families $\mathbf{n}^{(i)} = (\mathrm{n}_1^{(i)}(x_1), \ldots, \mathrm{n}_n^{(i)}(x_1, \ldots, x_n))$ modulo p by the operation (2), and then to lift them to polynomial systems $\mathbf{N}^{(i)}$ over $\mathbb{Q}$. This strategy is put into practice in [3] with precise complexity analysis, and implemented in the `RegularChains` library in Maple under the command name `EquiprojectableDecomposition`.



Input : Square system F over ℚ
Triangularize over ℚ
large coefficients appear → costly
Output: Triangular sets T^(1), ..., T^(s) defined over ℚ
simple, costless transformation
Polynomial systems N^(1), ..., N^(s) over ℚ defined in this article
Reduction modulo p of the coefficients
Lifting (Newton – Hensel operator)
Lifting (smaller coefficients than those of T^(i))
Square system F mod p over F_p
Triangularize modulo p
(small coefficients)
Triangular sets t^(1), ..., t^(s) T^(i) mod p ≡ t^(i)
simple, costless transformation
Polynomial systems n^(1), ..., n^(s) over F_p defined in this article

Figure 1: Prototype of a modular method modulo p to triangularize a polynomial system $\mathbf{F}$ having a finite number of solutions to a family of s triangular sets $\mathbf{T}^{(1)}, \ldots, \mathbf{T}^{(s)}$ (each of them is of the form $\mathbf{T}^{(i)} = (T_1^{(i)}(x_1), T_2^{(i)}(x_1, x_2), \ldots, T_n^{(i)}(x_1, \ldots, x_n))$). Instead of lifting directly the coefficients of the triangular sets $\mathbf{t}^{(i)}$ as shown in the second column, transformation to the family of systems $\mathbf{n}^{(i)}$ is performed before lifting the smaller coefficients, as shown in the third column (see [3] for details).

As we will see below, some "interpolation" formula appearing in Theorems 1, 2 and Propositions 1, 3 require that the field K is large enough: it is always enough to suppose that $|K|$ verifies, with notations introduced later, that $|K| > \deg_x(T_1)\deg_y(T_2)[\deg_z(T_3)] = d_1 d_2 [d_3]$.

When the polynomial system (1) does not generate a radical ideal, the formula (2) does not apply since $\prod_{\ell=1}^{i-1} \frac{\partial T_\ell}{\partial x_\ell}$ is no more necessarily invertible modulo $\langle T_1, \ldots, T_{i-1}\rangle$. In this paper, we extend those results to the simplest cases of $n = 2$ and to some extent $n = 3$ variables, as well as discussing the obstacles that prevent to consider a complete generalization to several variables.

Previous work We emphasize here that our aim is a simple, algorithmically efficient formula to transform a system of type (1) to another triangular family of polynomials generating the *same* ideal, but with smaller coefficients. We are thus not interested by removing multiplicities, and the present work is not related to squarefree decomposition algorithms modulo a triangular set that aim to remove multiplicity like in [5].

Triangular decomposition algorithms indeed are not ideal-theoretic, but set-theoretic ones, therefore they do not necessarily represent the same ideal as the input system; the set of solutions is the same (or just *almost* the same in the case of regular chains). A first obstacle to represent non-radical ideals by triangular sets is that it is not always possible. But even when the ideal can be represented, in a broad term, by triangular sets, we are not aware of algorithms that allow to find these sets. As drawn in the conclusion, the reason of this difficulty can be explained by the lack of gcd algorithm over non-reduced rings of type $K[x]/\langle x^2\rangle$. The results presented can contribute to understand better what is going on in this situation.

2 Case of two variables

We consider two variables $x \prec_{lex} y$ and two polynomials $T_1(x) \in K[x]$ and $T_2(x,y) \in K[x,y] \setminus K[x]$. The fact that $\{T_1, T_2\}$ is a reduced Gröbner basis of the zero-dimensional ideal that it generates implies the following elementary facts: T_1 is monic in x and T_2 is monic in y. Let $d_1 = \deg_x(T_1)$ and $d_2 = \deg_y(T_2)$, and write $T_2(x,y) = y^{d_2}+c_1(x)y^{d_2-1}+\cdots+c_{d_2-1}(x)y+c_{d_2}(x)$. Each polynomial $c_i \in K[x]$ verifies $\deg_x(c_i) < d_1$.

2.1 Review of the radical case

In this paragraph, are briefly recalled the results of [4], valid for a radical ideal only, in the case of two variables. The radical assumption made there implies the following assumption:

Assumption 1. All the roots of T_1 are simple: $T_1(x) = \prod_{i=1}^{d_1}(x-\alpha_i)$, with $\alpha_i \neq \alpha_j$ if $i \neq j$.

Theorem 1. *Under Assumption 1, define* $p_i(y) := T_2(\alpha_i, y)$ *and* $M_i(x) := \prod_{\substack{0<j<d_1 \\ j\neq i}}(x-\alpha_j)$.

1. *T_2 can be written as a Lagrange interpolation polynomial:*

$$T_2(x,y) = \sum\nolimits_{i=1}^{d_1} p_i(y)u_iM_i(x), \quad \textit{where} \quad u_i = \sum\nolimits_{j\neq i}\frac{1}{\alpha_i-\alpha_j}. \tag{3}$$

2. *Let* $N_2(x,y) \equiv T_1'(x)\cdot T_2(x,y) \bmod \langle T_1\rangle$ *then*

$$N_2 = \sum\nolimits_{i=1}^{d_1} p_i(y)M_i(x). \tag{4}$$

3. *The polynomial N_2 verifies:* $\langle T_1, N_2\rangle = \langle T_1, T_2\rangle$.

Remark 1. 1. It should be noted that $\{T_1, N_2\}$ is not a Gröbner basis.

2. The simplification yielding N_2 consists in suppressing the factors u_i from the "Lagrange idempotent" $u_iM_i(x)$ in Equality (3). This is the reason why coefficients in N_2 are usually smaller than those of T_2.

Example 1. Let $T_1(x) = (x-a)(x-b)$ $(a \neq b \in K)$. Then $T_2(x,y) = T_2(a,y)\frac{x-a}{b-a} + T_2(b,y)\frac{x-b}{a-b}$ holds. As for N_2:

$$\begin{aligned} T_1'T_2 &= \{(x-a)+(x-b)\}\{\ T_2(a,y)\frac{x-a}{b-a} + T_2(b,y)\frac{x-b}{a-b}\ \} \\ T_1'T_2 &\equiv T_2(a,y)\frac{(x-a)^2}{b-a} + T_2(b,y)\frac{(x-b)^2}{a-b} \equiv T_2(a,y)(x-a) + T_2(b,y)(x-b) \mod \langle T_1 \rangle \end{aligned}$$

and according to Statement 2 of Theorem 1 $\qquad N_2 = T_2(a,y)(x-a) + T_2(b,y)(x-b).$

2.2 Preliminary toward a generalization to multiple roots

We remove Assumption 1 and consider in this subsection a polynomial T_1 having multiple roots: $T_1(x) = \prod_{i=1}^r (x-\alpha_i)^{e_i}$ (where $\alpha_i \neq \alpha_j$ if $i \neq j$ and $e_i \in \mathbb{N}_{>0}$). The following proposition is a basic, straightforward application of the Chinese Remaindering Theorem.

Proposition 1. *Let* $p_i(x,y) \equiv T_2(x,y) \bmod \langle (x-\alpha_i)^{e_i} \rangle$ *and* $M_i(x) := \frac{T_1(x)}{(x-\alpha_i)^{e_i}} \in \bar{K}[x]$. *Note that* $\deg_y(p_i(y)) = \deg_y(T_2(x,y)) = d_2$. *The following assertions are satisfied:*

1. *There exist* $u_i \in \bar{K}[x]$ *such that:* $\quad \sum_{i=1}^r u_i M_i = 1, \quad u_i \in \bar{K}[x]$.
2. $T_2 \equiv \sum_{i=1}^r p_i u_i M_i \bmod \langle T_1 \rangle$ *(or equivalently* $\quad T_2 = \mathrm{NF}_{\langle T_1 \rangle}\left(\sum_{i=1}^r p_i u_i M_i\right)$ *).*

Example 2. Given $T_1 := (x-1)^2(x-2)^3$, let $p_1 = y+x,\ p_2 = y+x^2$. To compute T_2 from the previous Proposition, we need $M_1 = (x-2)^3,\ M_2 = (x-1)^2$. The Extended Euclid Algorithm provides the equality $(2-3x)(x-2)^3 + (3x^2-14x+17)(x-1)^2 = 1$, so that with $u_1 = 2-3x,\ u_2 = 3x^2-14x+17$, holds $u_1M_1 + u_2M_2 = 1$.

$$\begin{aligned} T_2 &\equiv p_1u_1M_1 + p_2u_2M_2 \mod \langle T_1 \rangle \\ &\equiv (2-3x)(x-2)^3(y+x) + (3x^2-14x+17)(x-1)^2(y+x^2) \mod \langle T_1 \rangle \\ &\equiv y + 3x^6 - 23x^5 + 68x^4 - 96x^3 + 65x^2 - 16x \mod \langle T_1 \rangle \\ &\equiv y + x^4 - 7x^3 + 19x^2 - 20x + 8 \mod \langle T_1 \rangle \end{aligned}$$

We can verify that $T_2 \equiv p_1 \mod \langle (x-1)^2 \rangle$, $T_2 \equiv p_2 \mod \langle (x-2)^3 \rangle$.

Proposition 2. *Let* $F(x) = \sum_{i=1}^r M_i(x) \in \bar{K}[x]$, *we have:*

$$F(x) \cdot T_2(x,y) \equiv \sum_{i=1}^r p_i(y) M_i(x) \bmod \langle T_1(x) \rangle.$$

Proof. By Proposition 1, $T_2(x,y) \equiv \sum_{i=1}^r p_i(y)u_i(x)M_i(x) \bmod \langle T_1(x) \rangle$ holds. Regarding that $\sum_{i=1}^r u_i(x)M_i(x) = 1$, we have:

$$Fu_iM_i = \left(\sum_{i=1}^r M_i\right) u_iM_i \equiv u_iM_i^2 \equiv (1 - \sum_{j\neq i} M_j)M_i \equiv M_i \bmod \langle T_1 \rangle.$$

Therefore, $\quad F \cdot T_2 \equiv F\left(\sum_{i=1}^r p_iu_iM_i\right) \equiv \sum_{i=1}^r p_i(Fu_iM_i) \equiv \sum_{i=1}^r p_iM_i \bmod \langle T_1 \rangle.$ □

Corollary 1. *Defining* $N_2(x,y) := \mathrm{NF}_{\langle T_1 \rangle}(F(x) \cdot T_2(x,y))$, *holds:* $\quad \langle T_1, T_2 \rangle = \langle T_1, N_2 \rangle$.

Proof. The inclusion $\langle T_1, T_2 \rangle \supset \langle T_1, N_2 \rangle$ is clear.

Since T_1 and F are relatively prime, by the extended Euclid algorithm there exists $u, v \in \bar{K}[x]$ such that $uT_1 + vF = 1$. Moreover by definition, there exists $A \in \bar{K}[x,y]$ such that $N_2 = FT_2 + AT_1$. As a result, $T_2 = uT_1T_2 + vFT_2 = uT_1T_2 + v(N_2 - AT_1) \in \langle T_1, N_2 \rangle$. Thus $\langle T_1, T_2 \rangle \subset \langle T_1, N_2 \rangle$. □

Remark 2. As in the case of Assumption 1, $\{T_1, N_2\}$ is not a Gröbner basis. The fact that $F \in K[x]$ is proved in Lemma 2, showing that $N_2 \in K[x, y]$ thanks to Corollary 1.

In Example 2 the equation $N_2 \equiv p_1 M_1 + p_2 M_2 \bmod \langle T_1 \rangle$ yields $N_2 = x^3y - 5x^2y + 10xy - 7y + 2x^4 - 8x^3 + 13x^2 - 8x$. In that toy example, it cannot be told that N_2 is simpler than T_2. But if the exponents e_i of the factors of T_1 are sufficiently large it becomes clear that N_2 has smaller coefficients than those of T_2 as shown below.

Example 3. Set $T_1 = (x-1)^4(x-2)^5$ and $p_1 = y + x,\ p_2 = y + x^2$. Let us compute T_2 and N_2.

$$T_2 = y + 20x^8 - 254x^7 + 1389x^6 - 4265x^5 + 8030x^4 - 9480x^3 + 6849x^2 - 2768x + 480$$
$$N_2 = x^5y - 9x^4y + 36x^3y - 74x^2y + 76xy - 31y + 2x^6 - 14x^5 + 46x^4 - 84x^3 + 81x^2 - 32x$$

Remark 3. As under Assumption 1, this is explained by the fact that the factor u_i is removed from T_2 in the formula 2. of Proposition 1 to obtain N_2.

2.3 Computation of N_2 in a special case

According to the previous subsection it is easy to compute N_2 if we know the polynomials $M_i = \frac{T_1}{(x-\alpha_i)^{e_i}}$. It suffices to compute the polynomial F of Proposition 2 and $\mathrm{NF}_{\langle T_1 \rangle}(F \cdot T_2)$ to obtain N_2. Unfortunately, to get the polynomials M_is, we are not aware of a simpler method than factorizing T_1, process which can be prohibitive especially over $\mathbb{Q}$. What we have in mind is a simple formula as in 2. of Theorem 1 to compute N_2 from T_1, T_2. To this end, we first consider a simpler assumption:

Assumption 2. The roots α_i below are pairwise distinct and have same multiplicity e:

$$T_1(x) = \prod_{i=1}^{r} (x - \alpha_i)^e \in K[x].$$

Lemma 1. *Under Assumption 2, let $S(x) := \mathrm{sqf}(T_1(x)) := \prod_{i=1}^{r}(x - \alpha_i)$ be the squarefree part of T_1 and let $F_{e,j}(x) := \sum_{i=1}^{r}[M_i(x)]^{j/e}$. The following recurrence relation holds:*

$$F_{e,1}(x) = S'(x), \qquad F_{e,\ell+1} = F_{e,1} \cdot F_{e,\ell} - \frac{S}{\ell} \cdot F'_{e,\ell} \quad (1 \le \ell \le e-1) \tag{5}$$

In particular, all polynomials $F_{e,j} \in K[x]$.

Proof. By assumption $S = \prod_{i=1}^{r}(x - \alpha_i)$, $F_{e,\ell} = \sum_{i=1}^{r} \prod_{j \neq i} (x - \alpha_j)^\ell$. Moreover $F'_{e,\ell} = \ell \sum_{i=1}^{r} \sum_{k \neq i} (\prod_{j \neq i} (x - \alpha_j)^\ell / x - \alpha_k)$ holds, yielding:

$$\begin{aligned} F_{e,1} \cdot F_{e,\ell} &= \sum_{i_1, i_2} \Big(\prod_{j \neq i_1} (x - \alpha_j) \cdot \prod_{j \neq i_2} (x - \alpha_j)^\ell \Big) = \sum_{i_1 = i_2} \prod_{j \neq i_1} (x - \alpha_j)^{\ell+1} \\ &+ \sum_{i_1 \neq i_2} (x - \alpha_1)^{\ell+1} \cdots (x - \alpha_{i_1})^\ell \cdots (x - \alpha_{i_2}) \cdots (x - \alpha_r)^{\ell+1} = F_{e,\ell+1} + \frac{S}{\ell} \cdot F'_{e,\ell} \end{aligned}$$

which is what we wanted to prove. □

Remark 4. This recurrence relation is used to compute $F = F_{e,e}$ of Proposition 2 from $S = \mathrm{sqf}(T_1)$. The latter polynomial is obtained under Assumption 2 simply by taking the e-th root of T_1, and the former polynomial F requires less than $O(A \log(A) \log(\log(A)))$ operations in K to be computed, where $A = (e+1)(d_1 - e)$ (see Appendix).

2.4 General case: need of squarefree factorization

The same multiplicity Assumption 2 is here lifted, and we consider the general case $T_1 = \prod_{i=1}^{r}(x-\alpha_i)^{e_i}$ where the roots α_i are pairwise distinct. Let

$$T_1 = S_1 S_2^2 \cdots S_n^n. \tag{6}$$

be the squarefree decomposition of T_1. Recall that this factorization is straightforward to obtain by *e.g.* Yun's squarefree factorization [6, Algorithm 14.21, p. 385] valid in characteristic zero. Since in the case of $\mathrm{char}(K) = p > 0$ we have assumed $|K|$ large enough, it is also valid in this case. The squarefree decomposition (6) can be obtained in $O(d_1 \log(d_1)^2 \log(\log(d_1)))$ operations in K by [6, Theorem 14.23], which is more efficient than computing a complete factorization of T_1 (and *far* more efficient if $\mathrm{char}(K) = 0$).

Besides, it should be said that the squarefree decomposition has a good behavior under reduction modulo a prime p. If p is large enough, the squarefree factors over $\mathbb{Q}$ correspond to the one obtained modulo p. This contrasts with the factorization into irreducibles where Chebotarev's density theorem [6, § 15.5, page 429] restrict to a given proportion the primes that yield a good behavior for the reduction modulo p.

Lemma 1 can be applied to each factor S_1, $S_2^2, \ldots, S_n^n$, for which Assumption 2 holds, in order to compute respective polynomials $F_{1,1}$, $F_{2,2}, \ldots, F_{n,n}$.

Lemma 2. *The polynomial F of Proposition 2 can be computed from the polynomials S_js and the polynomials $F_{j,j}$ and by the following formula*

$$F = \sum_{j=1}^{n} F_{j,j} T_1 / S_j^j \quad \text{and} \quad F \in K[x]. \tag{7}$$

Proof. Writing $S_j^j = \prod_{k_j}(x-\alpha_{k_j})^j$,

$$\sum_{j=1}^{n} F_{j,j}\frac{T_1}{S_j^j} = \sum_{j=1}^{n}\left(\sum_{k_j}\frac{S_j^j}{(x-\alpha_{k_j})^j}\right)\frac{T_1}{S_j^j} = \sum_{j=1}^{n}\left(\sum_{k_j}\frac{T_1}{(x-\alpha_{k_j})^j}\right) = F.$$

Therefore Equation (7) is satisfied. Additionally, by Lemma 2 $F_{j,j} \in K[x]$, and therefore $F_{j,j}\frac{T_1}{S_j{}^j} = F_{j,j}\prod_{\ell\neq j} S_\ell^\ell \in K[x]$. Thus $F \in K[x]$. □

Computing F requires (i) to compute the squarefree decomposition of T_1, (ii) for each squarefree factor S_j, compute the polynomial $F_{j,j}$ and (iii) use formula of Lemma 2 to obtain F. The computation of N_2 is summarized in Algorithm 1. The complexity estimates on the right correspond to upper bounds on the number of operations over K, where n the number of squarefree factors of T_1. Details are written in the appendix. The estimates in Steps 7 or 9 dominate the overall cost. Correctness follows from the lemma hereafter:

Lemma 3. *Let $(S_1, S_2, \cdots, S_n)$ be the squarefree decomposition of $T_1 = \prod_{i=1}^{r}(x-\alpha_i)^{e_i} \in K[x]$. If we apply Lemma 1 successively to $S_1, S_2^2, \cdots, S_n^n$, (instead of T_1) then we obtain $F_{1,1}$, $F_{2,2}, \cdots$, $F_{n,n}$ respectively. Regarding the polynomial T_2, we have:*

$$N_2 \equiv \sum_{j=1}^{n}(F_{j,j}T_2 \mod \langle S_j^j\rangle)\prod_{\ell\neq j} S_\ell^\ell \mod \langle T_1\rangle.$$

Proof. This is clear from the definition of $N_2 \equiv FT_2 \bmod \langle T_1\rangle$ (Cf. Proposition 2, Corollary 1) and Lemma 2. □

Algorithm 1: Computation of N_2

```
   Data: $T_1 = \prod_{i=1}^{r}(x-\alpha_i)^{e_i} \in K[x],\ T_2 \in K[x,y]$
   Result: $N_2 \in K[x,y]$.
1  Compute the squarefree decomposition $(S_1, S_2, \cdots, S_n)$ of $T_1$ // $O(d_1 \log(d_1)^2 \log(\log(d_1)))$ ;
2  for $1 \le i \le n$ do
3      $m_i \longleftarrow T_1/S_i^i,\ p_i \longleftarrow T_2 \mod S_i^i$   // $O((n-1)d_2 d_1 \log(nd_1)\log(\log(nd_1)))$;
4  end
5  for $1 \le j \le n$ do
6      Use recursive formula (5) to find $F_{j,j}$   // $O(nd_1 \log(nd_1)\log(\log(nd_1)))$;
7      $q_j \longleftarrow F_{j,j} p_j \mod S_j^j$   // $O(nd_1 d_2 B \log(nd_1)\log(\log(nd_1)))$;
8  end
9  $N_2 \longleftarrow \sum_{k=1}^{n} q_k m_k$   // $O(nd_1 d_2 \log(nd_1)\log(\log(nd_1)))$;
10 return $N_2$
```

3 Attempt of generalizations: case of three variables

The treatment of the case of $n = 2$ variables relies crucially on the fact that $T_1 \in K[x]$ admits a (univariate) squarefree decomposition. This is more complicated for $T_2 \in K[x,y]$ and we treat only special cases in this section.

Consider three variables x, y, z ordered as $x \prec_{lex} y \prec_{lex} z$. We suppose that $T = \{T_1(x)\,, T_2(x,y)\,, T_3(x,y,z)\}$ is lexicographic Gröbner basis of the 0-dimensional ideal that it generates. Thus we have $\mathrm{LT}(T_2) = y^{d_2}, \mathrm{LT}(T_3) = z^{d_3}$ for integers $d_2 > 0, d_3 > 0$.

3.1 Review on the case of a radical ideal in three variables

Assumption 3. Let $T_1 = \prod_{i=1}^{d_1}(x-\alpha_i)$ ($\alpha_i \in \bar{K}$, if $i \ne i'$ then $\alpha_i \ne \alpha_{i'}$) and $T_2 \in K[x,y]$ be such that for $1 \le i \le d_1$ $T_2(\alpha_i, y) = \prod_{j=1}^{d_2}(y - \beta_{ij})$ ($\beta_{ij} \in \bar{K}$, if $j \ne j'$then $\beta_{ij} \ne \beta_{ij'}$) is satisfied. Then the ideal generated by T_1 and T_2 is radical.

Theorem 2. *1) Under Assumption 3, we have:*

$$T_3(x,y,z) = \sum_{i=1}^{d_1}\left(\sum_{j=1}^{d_2} T_3(\alpha_i, \beta_{ij}, z)\prod_{j' \ne j}\frac{y-\beta_{ij'}}{\beta_{ij}-\beta_{ij'}}\right)\prod_{i' \ne i}\frac{x-\alpha_i'}{\alpha_i - \alpha_i'}. \tag{8}$$

2) Define $F = \frac{dT_1}{dx},\ G = \frac{\partial T_2}{\partial y}$. Then the following holds:

$$F \cdot G \cdot T_3 \equiv \sum_{i=1}^{d_1}\left(\sum_{j=1}^{d_2} T_3(\alpha_i, \beta_{ij}, z)\prod_{j' \ne j}(y-\beta_{ij'})\right)\prod_{i' \ne i}(x-\alpha_i') \mod \langle T_1, T_2\rangle. \tag{9}$$

3) Defining $N_3 := \mathrm{NF}_{\langle T_1,T_2\rangle}(F\,G\,T_3)$, the ideal equality holds: $\langle T_1, T_2, T_3\rangle = \langle T_1, N_2, N_3\rangle$.

Remark 5. Since $F \in K[x],\ G \in K[x,y]$ we also have $N_3 \in K[x,y]$.
Moreover, as already noted in the case of two variables in Remarks 1, 3, by comparing T_3 and N_3 in (8) and (9) the terms $\prod_{j' \ne j}\frac{1}{\beta_{ij}-\beta_{ij'}}\prod_{i' \ne i}\frac{1}{\alpha_i - \alpha_i'}$ have been removed.

3.2 Toward generalization

We do not assume that T_1, T_2, T_3 generate a radical ideal anymore. More precisely:

Assumption 4. Let us write $T_1 = \prod_{i=1}^{r_1}(x-\alpha_i)^e$ ($\alpha_i \in \bar{K}$, if $i \neq i'$ then $\alpha_i \neq \alpha_{i'}, e > 0$) in $\bar{K}[x]$. Concerning T_2, for $1 \leq i \leq r_1$, we assume that $T_2 \bmod \langle (x-\alpha_i)^e \rangle$ is factorized over $\bar{K}[x,y]$, and that each factor comes with the same multiplicity: $T_2 \equiv$

$$\prod_{j=1}^{r_2}(y-g_{i,j}(x))^\ell \bmod \langle (x-\alpha_i)^e \rangle,\ (g_{i,j} \in \bar{K}[x], \text{ and } g_{i,j}(x) \not\equiv g_{i,j'}(x) \bmod \langle (x-\alpha_i)^e \rangle \text{ if} j \neq j').$$

Remark 6. Under Assumption 4, for any $1 \leq i \leq r_1$, define $M_i = T_1/(x-\alpha_i)^e \in \bar{K}[x]$. By Proposition 2, there exists $u_i \in \bar{K}[x]$ such that $\sum_{i=1}^{r_1} u_i M_i = 1$ and

$$T_2 \equiv \sum_{i=1}^{r_1} \left(\prod_{j=1}^{r_2} (y-g_{ij}(x))^\ell \right) u_i M_i \mod \langle T_1 \rangle \quad (\text{in } \bar{K}[x,y]).$$

We need another hypothesis, which was not necessary in the case of two variables (see Concluding Remarks "Discussion").

Assumption 5. On top of Assumption 4, suppose additionally

$$\text{for all } 1 \leq i \leq r_1, \quad \text{when } j \neq j', \quad \gcd((x-\alpha_i)^e,\ g_{ij}(x) - g_{ij'}(x)) = 1 \ (\text{in } \bar{K}[x]).$$

Remark 7. This is equivalent to $\gcd(x-\alpha_i,\ g_{ij}(x)-g_{ij'}(x)) = 1$, or $g_{ij}(\alpha_i) \neq g_{ij'}(\alpha_i)$.

Proposition 3. *Under Assumptions 4,5, consider for $1 \leq i \leq r_1$, $1 \leq j \leq r_2$ the notations g_{ij}, M_i and u_i of Remark 6. Define additionally $M_{ij},\ u_{ij} \in \bar{K}[x,y]$ as follows:*

$$M_{ij} \equiv \frac{T_2}{(y-g_{ij})^\ell} \bmod \langle (x-\alpha_i)^e \rangle, \ \textit{and} \ \sum_j u_{ij} M_{ij} \equiv 1 \mod \langle (x-\alpha_i)^e \rangle, \ \textit{over } \bar{K}[x,y] \quad (10)$$

Then, by denoting $p_{ij} \equiv T_3 \bmod \langle (x-\alpha_i)^e, (y-g_{ij})^\ell \rangle \in \bar{K}[x,y,z]$, T_3 satisfies:

$$T_3(x,y,z) \equiv \sum_{i=1}^{r_1} \left(\sum_{j=1}^{r_2} p_{ij} u_{ij} M_{ij} \right) u_i M_i \mod \langle T_1, T_2 \rangle, \quad (11)$$

Equivalently $T_3 = \mathrm{NF}_{\langle T_1,T_2\rangle}\Big(\sum_{i=1}^{r_1} (\sum_{j=1}^{r_2} p_{ij} u_{ij} M_{ij}) u_i M_i \Big)$.

Proof. Conditions related to M_i and u_i were already proved for the case of two variables in the previous section. Let us construct the polynomials $u_{ij} \in \bar{K}[x,y]$ satisfying Condition (10).

Denote $A = \bar{K}[x]/\langle (x-\alpha_i)^e \rangle$. By Assumption 5, for all $1 \leq i \leq r_1, 1 \leq j' < j \leq r_2$, polynomials $g_{ij}(x) - g_{ij'}(x)$ possess inverses in A,

$$y - g_{ij'} = y - g_{ij} + g_{ij} - g_{ij'} \quad \text{in } A[y]$$
$$(g_{ij} - g_{ij'})^{-1}(y-g_{ij'}) - (g_{ij}-g_{ij'})^{-1}(y-g_{ij}) = 1 \ \text{ in } A[y]$$

from which we get: $\langle y - g_{ij}(x) \rangle + \langle y - g_{ij'}(x) \rangle = \langle 1 \rangle$ in $A[y]$. Therefore polynomials u_{ij} verifying Condition (10) exist.

Let us show that Equation (11) holds. For $1 \leq i \leq r_1$, according to the Chinese Remaindering Theorem (CRT) applied to Condition (10) gives: $T_3 \equiv \sum_{j=1}^{r_2} p_{ij} u_{ij} M_{ij} \mod \langle (x-\alpha_i)^e, T_2 \rangle$. Similarly, the definition of M_i, u_i yields again by the CRT: $T_3 \equiv \sum_{i=1}^{r_1} (\sum_{j=1}^{r_2} p_{ij} u_{ij} M_{ij}) u_i M_i \mod \langle T_1, T_2 \rangle$ which is Equation (11). □

Example 4. Defining $T_1 = x^2(x-2)^2,\ T_2 = (y-x^2)^2(y-1)^2$, we have:

$$\begin{cases} T_2 \equiv y^2(y-1)^2 \mod \langle x^2\rangle \\ T_2 \equiv (y-4x+4)^2(y-1)^2 \mod \langle (x-2)^2\rangle \end{cases}$$

so that Assumption 5 is fulfilled. Define $M_2 = x^2$ and $M_1 = (x-2)^2$, as well as:

$$\begin{cases} M_{12} = y^2 \\ M_{11} = (y-1)^2 \\ M_{22} = (y-4x+4)^2 \\ M_{21} = (y-1)^2 \end{cases} \quad \begin{cases} p_{11} \equiv z \bmod \langle M_2, M_{12}\rangle \\ p_{12} \equiv z+1 \bmod \langle M_2, M_{11}\rangle \\ p_{21} \equiv z+x \bmod \langle M_1, M_{22}\rangle \\ p_{22} \equiv z+y \bmod \langle M_1, M_{21}\rangle \end{cases} \quad \begin{cases} u_{11} = 1+2y \\ u_{12} = 3-2y \\ u_{21} = \frac{1}{27}(8xy-18y-32x+75) \\ u_{22} = \frac{1}{27}(-8xy+18y+1) \end{cases}$$

Thus, for $1 \le i \le 2$, $\sum_{1\le j\le 2} u_{ij}M_{ij} \equiv 1 \bmod \langle M_i\rangle$ hold. From these data, Proposition 3 gives:

$$\begin{aligned} T_3 &\equiv \sum_{i=1}^{2} u_i M_i \big(\sum_{j=1}^{2} p_{ij}u_{ij}M_{ij}\big) \mod \langle T_1, T_2\rangle \\ &\equiv u_1M_2(p_{11}u_{11}M_{12} + p_{12}u_{12}M_{11}) + u_2M_1(p_{21}u_{21}M_{22} + p_{22}u_{22}M_{21}) \mod \langle T_1,T_2\rangle \end{aligned}$$

Therefore, after taking normal form modulo T_1, T_2 we get:

$$T_3 = z - \frac{19}{36}x^3y^3 + \frac{169}{108}x^2y^3 - 2y^3 + x^3y^2 - \frac{103}{36}x^2y^2 + 3y^2 - \frac{2}{3}x^3y + \frac{16}{9}x^2y + \frac{7}{36}x^3 - \frac{13}{27}x^2$$

We can check that the polynomial T_3 satisfies expressly $T_3 \equiv p_{ij} \bmod \langle M_i, M_{ij}\rangle$ for $1 \le i \le 2$, $1 \le j \le 2$.

The following proposition follows from Proposition 3. The notations are the same. For $1 \le i \le r$ define $G_i \in \bar{K}[x,y]$ such that $G_i \equiv \sum_{j=1}^{r_2} M_{ij} \mod \langle (x-\alpha_i)^e\rangle$ in $\bar{K}[x,y]$.

Proposition 4. *By defining* $F = \sum_{i=1}^{r_1} M_i,\ G = \sum_{i=1}^{r_1} G_iu_iM_i$ *the following hold:*

$$F \in K[x], \quad G \in K[x,y], \qquad F\cdot G\cdot T_3 \equiv \sum_{i=1}^{r_1}\Big(\sum_{j=1}^{r_2} p_{ij}M_{ij}\Big)M_i \mod \langle T_1, T_2\rangle.$$

Proof. The fact that $F \in K[x]$ and $G \in K[x,y]$ is proved in Lemma 6. From Proposition 3:

$$T_3 \equiv \sum_{j=1}^{r_2} p_{ij}u_{ij}M_{ij} \mod \langle (x-\alpha_i)^e, T_2\rangle.$$

The fact that $M_{ij}M_{ij'} \equiv 0 \bmod \langle (x-\alpha_i)^\ell, T_2\rangle$ if $j \ne j'$ implies that:

$$\begin{aligned} G_i\cdot \sum_{j=1}^{r_2} p_{ij}u_{ij}M_{ij} &\equiv \Big(\sum_{j=1}^{r_2} M_{ij}\Big)\Big(\sum_{j=1}^{r_2} p_{ij}u_{ij}M_{ij}\Big) \mod \langle (x-\alpha_i)^e, T_2\rangle \\ \equiv \sum_{j=1}^{r_2} p_{ij}u_{ij}M_{ij}^2 &\equiv \sum_{j=1}^{r_2} p_{ij}M_{ij}\Big(1 - \sum_{t\ne j} u_{it}M_{it}\Big) \bmod \langle (x-\alpha_i)^e, T_2\rangle \\ &\equiv \sum_{j=1}^{r_2} p_{ij}M_{ij} \mod \langle (x-\alpha_i)^e, T_2\rangle. \end{aligned}$$

By the Chinese Remaindering Theorem,

$$\sum_{i=1}^{r_1} u_iM_i\Big(G_i\sum_{j=1}^{r_2} p_{ij}u_{ij}M_{ij}\Big) \equiv \sum_{i=1}^{r_1} u_iM_i\Big(\sum_{j=1}^{r_2} p_{ij}M_{ij}\Big) \mod \langle T_1, T_2\rangle.$$

Therefore by denoting $G = \sum_{i=1}^{r_1} G_iu_iM_i$,

$$G\cdot T_3 \equiv \sum_{i=1}^{r_1} u_iM_i\big(\sum_{j=1}^{r_2} p_{ij}M_{ij}\big) \mod \langle T_1, T_2\rangle$$

holds. Thanks to Proposition 2 $F\cdot G\cdot T_3 \equiv \sum_{i=1}^{r_1} M_i(\sum_{j=1}^{r_2} p_{ij}M_{ij}) \bmod \langle T_1, T_2\rangle$. □

The following Corollary is proved in a similar way as Corollary 1

Corollary 2. *Defining* $N_3 := \mathrm{NF}_{\langle T_1,T_2\rangle}(F\cdot G\cdot T_3)$*, the following equality of ideals is satisfied:*

$$\langle T_1,T_2,T_3\rangle = \langle T_1,T_2,N_3\rangle = \langle T_1,N_2,N_3\rangle.$$

Remark 8. 1. The fact that $N_3 \in K[x,y,z]$ is proved in Lemma 6.

2. From the definition of N_3, we see that $N_3 \equiv \sum_{i=1}^{r_1}(\sum_{j=1}^{r_2} p_{ij}M_{ij})M_i \mod \langle T_1,T_2\rangle$. By comparing with T_3 thanks to Proposition 3, the terms $u_{ij}\cdot u_i$ have been removed.

3.3 Computation of N_3

In this paragraph, is explained how to compute efficiently the polynomial G of Proposition 4 under Assumption 3. The notations are the same as in Propositions 3,4.

Lemma 4. *In Assumption 4 when* $\ell = 1$ *we have:* $G \equiv \frac{\partial T_2}{\partial y} \mod \langle T_1\rangle$.

Proof. From Remark 6 and the fact that $\ell = 1$:

$$\frac{\partial T_2}{\partial y} \equiv \sum_{j=1}^{r_2}\prod_{k\neq j}(y-g_{ik}(x)) \equiv \sum_{j=1}^{r_2} M_{ij} \equiv G_i \mod \langle (x-\alpha_i)^e\rangle. \tag{12}$$

By the Chinese Remaindering theorem, it follows that:

$$\sum_{i=1}^{r_1} u_iM_i(\sum_{j=1}^{r_2}\prod_{k\neq j}(y-g_{ik}(x))) \equiv \sum_{i=1}^{r_1} u_iM_iG_i \equiv G \mod \langle T_1\rangle. \tag{13}$$

Putting these equalities altogether, we obtain the equality stated. □

Lemma 4 supplies with a straightforward method to compute G and thus, according to the definition of N_3 in Corollary 2, to compute N_3 when $\ell = 1$. We focus next on the case $\ell > 1$ and the computation of G is addressed in Lemma 7. Recall that from Proposition 4

$$G = \sum_{i=1}^{r_1} u_iM_iG_i \equiv \sum_{i=1}^{r_1} u_iM_i(\sum_{j=1}^{r_2}\prod_{k\neq j}(y-g_{ik}(x))^{\ell}) \mod \langle T_1\rangle$$

Since the term $\sum_{i=1}^{r_1} u_iM_i$ has been treated in the section dealing with the case of two variables, we only need to address the computation of G_i. The Lemma below is "local", *i.e* modulo $\langle (x-\alpha_i)^e\rangle$, and allows to compute G_i. The proof is similar to as the one of Lemma 1. Lemma 7 is a global version to compute G.

Lemma 5. *Denote* $S_i = \prod_{j=1}^{r_2}(y-g_{ij}(x))$, $G_{i,k} = \sum_{j=1}^{r_2}\prod_{j'\neq j}(y-g_{ij'}(x))^k$. *Note that* $G_i = G_{i,\ell}$. *Starting from* S_i, $G_{i,\ell}$ *can be computed by the following recurrence relation:*

$$G_{i,1} = \frac{\partial S_i}{\partial y}, \quad G_{i,k+1} = G_{i,1}\cdot G_{i,k} - \frac{S_i}{k}\cdot\frac{\partial G_{i,k}}{\partial y} \quad (1 \le k \le \ell - 1).$$

Remark 9. Under Assumption 4, this Lemma is not of practical use, since we only need the "global" version of it stated in Lemma 7.

Lemma 6. *The polynomial* N_3 *defined in Corollary 2 is in* $K[x,y,z]$.

Proof. By Lemma 2 $F \in K[x]$ therefore we focus on proving that $G \in K[x, y]$. According to the definition of N_3, this is indeed sufficient to prove that $N_3 \in K[x, y, z]$. Proposition 4 gives $G = \sum_{i=1}^{r_1} u_i M_i (\sum_{j=1}^{r_2} \prod_{k \neq j} (y - g_{ik})^{\ell}) \bmod \langle T_1 \rangle$. Consider the following isomorphism of algebras derived from the Chinese Remaindering Theorem:

$$\begin{array}{rcl} \varphi : \oplus_{i=1}^{r_1} (\bar{K}[x]/\langle (x - \alpha_i)^e \rangle)[y] & \to & (\bar{K}[x]/\langle T_1 \rangle)[y] \\ (a_1, a_2, \cdots, a_{r_1}) & \mapsto & \sum_{i=1}^{r_1} u_i M_i a_i. \end{array}$$

Here $S_i = \prod_{j=1}^{r_2} (y - g_{i,j}(x)) \in \bar{K}[x, y]$ and $T_2 \equiv S_i^{\ell} \mod \langle (x - \alpha_i)^e \rangle$. Let us define $S = \varphi(S_1, S_2, \cdots, S_{r_1})$. We have:

$$\begin{array}{rcl} T_2 & \equiv & \sum_{i=1}^{r_1} u_i M_i S_i^{\ell} \;\equiv\; \varphi(S_1^{\ell}, S_2^{\ell}, \cdots, S_{r_1}^{\ell}) \mod \langle T_1 \rangle \\ & \equiv & \varphi(S_1, S_2, \cdots, S_{r_1})^{\ell} \;\equiv\; S^{\ell} \mod \langle T_1 \rangle \end{array}$$

By assumption, $T_2 \in K[x, y]$ hence $S^{\ell} \in K[x, y]$ since that $T_1 \in K[x]$. Combined with the fact that $S \in \bar{K}[x, y]$, it follows that $S \in K[x, y]$. Hence $\frac{\partial S}{\partial y} \in K[x, y]$, and by the next Lemma 7 $H_1 \in K[x, y]$. In this way it is proved that $G (= H_{\ell}$ in the next lemma$) \in K[x, y]$. □

Lemma 7. *For $t = 1, \ldots, \ell$ let $H_t := \varphi(G_{1,t}, G_{2,t}, \ldots, G_{r_1,t})$, so that $H_{\ell} = G$. Let $S = T_2^{1/\ell} \bmod \langle T_1 \rangle$ be as in the proof of Lemma 6. These polynomials verify the recurrence relation:*

$$H_1 = \frac{\partial S}{\partial y}, \quad \textit{and} \quad H_{k+1} = H_1 H_k - \frac{S}{k} \frac{\partial H_k}{\partial y} \quad (k \geq 1)$$

Proof. Let $G_{i,j} = \sum_{j=1}^{r_2} \prod_{k \neq j} (y - g_{ik})^j$ (as defined in Lemma 5). First let us handle H_1

$$H_1 = \sum_{i=1}^{r_1} u_i M_i G_{i,1} = \sum_{i=1}^{r_1} u_i M_i \frac{\partial S_i}{\partial y}$$

Since $u_i M_i \in \bar{K}[x]$, this polynomial does not depend on y:

$$H_1 = \frac{\partial}{\partial y} (\sum_{i=1}^{r_1} u_i M_i S_i) = \frac{\partial \varphi(S_1, \cdots, S_{r_1})}{\partial y} = \frac{\partial S}{\partial y}.$$

Next let us treat the case $k \geq 1$:

$$\begin{array}{rcl} H_{k+1} & = & \sum_{i=1}^{r_1} u_i M_i G_{i,k+1} \;=\; \sum_{i=1}^{r_1} u_i M_i (G_{i,1} G_{i,k} - \frac{S_i}{k} \frac{\partial G_{i,k}}{\partial y}) \\ & = & \varphi(G_{1,1} G_{1,k} - \frac{S_1}{k} \frac{\partial}{\partial y} G_{1,k}, \cdots, G_{r_1,1} G_{r_1,k} - \frac{S_{r_1}}{k} \frac{\partial G_{r_1,k}}{\partial y} \\ & = & \varphi(G_{1,1}, \cdots, G_{r_1,1}) \varphi(G_{1,k}, \cdots, G_{r_1,k}) - \frac{\varphi(S_1, \cdots, S_{r_1})}{k} \frac{\partial}{\partial y} \varphi(G_{1,k}, \cdots, G_{r_1,k}) \end{array}$$

This latter equation is equal to $H_1 H_k - \frac{S}{k} \frac{\partial H_k}{\partial y}$ which is what we wanted to show. □

4 Concluding remarks

Summary of contributions We have extended the definitions of polynomials N_2, and to some extent N_3 of [4] to non-radical ideals that have a triangular set $\{T_1, T_2, T_3\} \subset K[x, y, z]$ as lexicographic Gröbner basis. A comparable decrease of the bit-size of coefficient has been observed in a Maple implementation. A major difference though lies in the use of squarefree decomposition, which, despite enjoying of fast algorithms in order to be computed, induces some complications that prevent to generalize these formulas in full generality to more than two variables. A detailed complexity analysis of the bit-size as done in [4] is planed in a later work, the present work focusing more on *feasibility* in two or three variables.

Implementation in Maple Algorithm 1 to compute the polynomial N_2 from T_1 and T_2 has been implemented in Maple. The squarefree decomposition algorithm is Yun's one taken from [6, Algorithm 14.21, p. 385] (see Section 2.4). The code occupies less than 40 lines. The table below shows benchmarks in the case where $T_1 = (x-a)^{e_1}(x-b)^{e_2}$ for: (i) some random values $a, b \in \mathbb{Z}$ having 15 digits (ii) $e_1 + e_2 = 22$, and e_1 is ranging from 1 to 11 (iii) moduli of $T_2 \bmod \langle (x-a)^{e_1} \rangle$ and $T_2 \bmod \langle (x-b)^{e_2} \rangle$ are chosen randomly of degree 15. It is indeed *not* necessary to consider more than two factors since this case was already treated in the previous work [4] which showed the decrease in the bit-size of coefficients. As we can see, the bit-size

	e_1	1	2	3	4	5	6	7	8	9	10	11
T_2	total	232139	242837	234372	230948	222758	223658	225067	216909	219667	212003	191825
	max	575	612	583	550	544	544	543	512	512	494	471
N_2	total	61865	60738	60669	54793	58166	56831	54069	51230	49423	46876	44330
	max	328	311	300	260	269	255	239	225	211	197	179

Table 1: Line **total** displays the *sum* of the number of digits over *all* the coefficients in T_2 (upper) or N_2 (lower). Line **max** displays the *maximal* number of digits (≈bit-size) among all the coefficients in T_2 or N_2.

decrease observed get more important as *both* degrees e_1 and e_2 get higher: when $e_1 = 1$ and $e_2 = 21$, the ratio between the maximal bit-size of the coefficients of N_2 and the one of T_2 is $\frac{328}{575} \approx 0.57$ whereas the ratio becomes $\frac{179}{471} \approx 0.38$ for $e_1 = e_2 = 11$. However better ratio than $1/3$ where not observed even for higher values of e_1 and e_2.

Discussion The algorithm to compute N_2 treated in Section 2 takes as input the polynomials T_1, T_2 and by means of efficient gcd-based subroutines output N_2. For N_3, Assumptions 5,4 are added. To be removed the following obstacles must be overcome:

Example 5. Let $T_1 = x^2(x+1)^2 \in \mathbb{Q}[x],\ T_2 = (y-x^2)(y-x) \in \mathbb{Q}[x,y]$.

From $T_2 \equiv y(y-x) \mod \langle x^2 \rangle$, we see that $\gcd(x^2, 0-x) = \gcd(x^2, -x) \neq 1$. If we set $A = K[x]/\langle x^2 \rangle$, then:

$$A[y]/\langle y(y-x)\rangle \not\simeq A[y]/\langle y\rangle \oplus A[y]/\langle y-x\rangle. \tag{14}$$

On the other hand, since $T_2 \equiv (y+2x+1)(y-x) \mod \langle (x+1)^2 \rangle$, we have $\gcd((x+1)^2, -2x-1-x) = \gcd((x+1)^2, -3x-1) = 1$. If we write $B = K[x]/\langle (x+1)^2 \rangle$, then:

$$B[y]/\langle (y+2x+1)(y-x)\rangle \simeq B[y]/\langle y+2x+1\rangle \oplus B[y]/\langle y-x\rangle. \tag{15}$$

In this example, Assumption 5 is verified in Equation (15) but not in Equation (14). Actually, the ring $A[y]/\langle y(y-x)\rangle$ is primary so we cannot decompose it furthermore. We have no algorithmic solution for this kind of input.

As for the restriction implied by the "same multiplicity" assumption 4, it can be loosened up to: denoting $T_1 = \prod_i S_i^{e_i}$ the squarefree decomposition of T_1, then $T_2 \equiv \prod_j (y - g_{ij}(x))^{\ell_i} \bmod \langle S_i^{e_i} \rangle$, with $g_{ij} \in \bar{K}[x]$ and verifying Assumption 5. Note that Assumption 4 states that $i = 1$, and thus $\ell = \ell_1$. Example 5 above does not fulfill this generalized condition on T_2.

References

[1] P. Aubry, D. Lazard, and M. Moreno Maza. On the theories of triangular sets. *J. of Symbolic Computation*, 28(1,2):45–124, 1999.

[2] C. Chen and M. Moreno Maza. Algorithms for computing triangular decompositions of polynomial systems. In *Proceedings of the 36th International Symposium on Symbolic and Algebraic Computation*, ISSAC '11, pages 83–90, New York, NY, USA, 2011. ACM.

[3] X. Dahan, M. Moreno Maza, É. Schost, W. Wu, and Y. Xie. Lifting techniques for triangular decompositions. In *ISSAC'05*, pages 108–115. ACM, 2005.

[4] X. Dahan and É. Schost. Sharp estimates for triangular sets. In *ISSAC '04: Proceedings of the 2004 International Symposium on Symbolic and Algebraic Computation*, pages 103–110. ACM Press, 2004.

[5] X. Li, C. Mou, and D. Wang. Decomposing polynomial sets into simple sets over finite fields: The zero-dimensional case. *Computers & Mathematics with Applications*, 60(11):2983 – 2997, 2010.

[6] J. von zur Gathen and J. Gerhard. *Modern computer algebra*. Cambridge University Press, New York, NY, USA, 2003. Second Eddition.

[7] D. Wang. *Elimination Methods*. Texts & Monographs in Symbolic Computation. Springer Vienna, 2012.

[8] W. T. Wu. On zeros of algebraic equations - an application of Ritt principle. *Kexue Tongbao*, 5:1–5, 1986.

Appendix

We detail here the complexity estimates written in Algorithm 1. Let us recall some notations; $d_1 := \deg(T_1)$, $d_2 = \deg_y(T_2)$, and $d_{S_j} = \deg(S_j)$ where $T_1 = \prod_{j=1}^n S_j^j$ is the squarefree decomposition of T_1. In particular, the degree equality $d_1 = \sum_j j d_{S_j}$ holds.

Following [6, p. 242], we will use *multiplication time* function denoted $\mathsf{M}(d)$. This is an upper bound on the maximal number of operations in the base field K to perform the multiplication of two polynomials of degree at most d. Shönhage Strassen's version of Fast Fourier Transform provides the estimate $\mathsf{M}(d) = O(d\log(d)\log(\log(d)))$. $\mathsf{M}(\,.\,)$ is thus *super-linear*: $\mathsf{M}(a)+\mathsf{M}(b) \leq \mathsf{M}(a+b)$. We start by establishing the complexity stated in Remark 4.

Lemma 8. *To compute $F_{e,e}$ from $S = \mathrm{sqf}(T_1)$ using the recurrence relation of Lemma 1, less than $O(A\log(A)\log(\log(A)))$ operations over K is required (where $A = (e+1)(d_1-e)$).*

Proof. Let $\delta_{e,\ell} := \deg(F_{e,\ell})$. Starting from $\delta_{e,1} = \deg(F_{e,1}) = \deg(S) = r-1$ (where $r = d_1/e$) the recurrence relation gives $\delta_{e,\ell+1} = \deg(F_{e,1}) + \deg(F_{e,\ell} - 1) = \delta_{e,1} + \delta_{e,\ell} = r - 1 + \delta e, \ell$, yielding $\delta_{e,\ell} = (r-1)\ell = \ell d_1/e - \ell$.

The addition in the recurrence formula has indeed a negligible cost. For each $\ell = 0, \dots, e-1$, we need to perform one differentiation of a polynomial of degree $\delta_{e,\ell} = \ell d_1/e - \ell$; one multiplication of a polynomial of degree $\delta_{e,\ell}-1$ by one of degree $r = d_1/e$; and one multiplication by two polynomials of respective degree $r-1 = d_1/e-1$ and $\delta_{e,\ell}$. The differentiation is negligible, remains to evaluate the cost of the two multiplications: $\mathsf{M}(\delta_{e,\ell}-1) + \mathsf{M}(\delta_{e,\ell}) \leq 2\mathsf{M}(\delta_{e,\ell})$. This is repeated e times yielding $\sum_{\ell=1}^e 2\mathsf{M}(\delta_{e,\ell}) \leq 2\mathsf{M}(\sum_{\ell=1}^e \ell(d_1/e-1)) = 2\mathsf{M}(\frac{e(e+1)}{2}(\frac{d_1}{e}-1)) \leq \mathsf{M}((e+1)(d_1-e))$ by super-linearity of $\mathsf{M}(\,.\,)$. The complexity of the lemma follows. □

Theorem 3. *(a) Step 1 requires less than $O(d_1\log(d_1)^2\log(\log(d_1)))$ operations in K.*

(b) The cost of the for loop at Steps 2-4 is less than $O((n-1)d_1d_2\log(nd_1)\log(\log(nd_1)))$.

(c) Step 6 requires up to $\mathsf{M}((j+1)(jd_{S_j}-j))$ operations in K.

(d) Step 7 can be performed within $d_2\left[\mathsf{M}(C_j)+4\mathsf{M}(B_j)+\mathsf{M}(jd_{S_j})+O(jd_{S_j})\right]$ *operations in* K*, where* $C_j=\max\{d_{S_j}-j, d_1-jd_{S_j}\}$ *and* $B_j=d_1-j(d_{S_j}+1)$*.*

(e) The cost of the for loop of Steps 5-8 is less than $O(nd_2d_1\log(nd_1)\log(\log(nd_1)))$*.*

(f) Step 9 requires at most $O(nd_2d_1\log(nd_1)\log(\log(nd_1)))$ *operations in* K*.*

(g) the total cost, in term of number operations in K*, of Algorithm 1 is roughly upper bounded by* $O(nd_2d_1\log(nd_1)\log(\log(nd_1)))$*.*

Proof. (a) is explained at the beginning of Section 2.4.

Fast Newton's iteration based Euclid algorithm [6, Theorem 9.6] allows to perform the division T_1/S_i^i in $4\mathsf{M}(\deg(T_1)-\deg(S_i^i))+\mathsf{M}(\deg(S_i^i))+O(\deg(S_i^i))$. The loop runs over the number of squarefree factors n of T_1, yielding the following estimate using the notations stated before the theorem, deduced from the super-linearity of $\mathsf{M}(\,.\,)$,

$$\sum_{i=1}^{n} 4\mathsf{M}(d_1-id_{S_i})+\mathsf{M}(id_{S_i})+O(id_{S_i})\le 4\mathsf{M}((n-1)d_1)+\mathsf{M}(d_1)+O(d_1).$$

The dominating term of the above function can be bounded by $O((n-1)d_1\log(nd_1)\log(\log(nd_1)))$.

The second operation performed during the loop of Steps 2-4 is $T_2 \bmod S_i^i$, which amounts to execute d_2 Euclidean divisions, one for each coefficient of $T_2\in(K[x])[y]$. A similar analysis done for the first operation T_1/S_i^i shows that this requires up to $O((n-1)d_1d_2\log(nd_1)\log(\log(nd_1)))$ operations over K. This shows (b).

The estimate of Step 6 stated in (c) is explained in Lemma 8.

Note that $\deg(F_{j,j})=\delta_{j,j}=j(d_{S_j}/j-1)=d_{S_j}-j$ and that each coefficient in x of $p_j\in(K[x])[y]$ has degree at most $d_1-jd_{S_j}$. The multiplication $F_{j,j}p_j$ thus costs at most $d_2\mathsf{M}(C_j)$, $C_j:=\max\{d_{S_j}-j, d_1-jd_{S_j}\}$. Since each coefficient in x of $F_{j,j}p_j(x,y)$ is thereby of degree in x less than $B_j:=d_1-j(d_{S_j}+1)$ it follows that the reduction $\bmod S_j^j$ of each of these coefficients can be estimated to cost less than $d_2\left[4\mathsf{M}(d_1-j(d_{S_j}+1))+\mathsf{M}(jd_{S_j})+O(jd_{S_j})\right]$. All in all, we can see that Step 7 can be performed within the number of operations over K stated in (d).

To estimate the overall cost of the for loop at steps 5-8, it suffices to sum up the complexities of Steps 6 and 7 above. For Step 6: $\sum_{j=1}^{n}\mathsf{M}((j+1)(d_{S_j}-j))\le\mathsf{M}(\sum_{j=1}^{n}(j+1)(d_{S_j}-j))$. We observe that $\sum_{j=1}^{n}(j+1)(d_{S_j}-j)=d_1+(\sum_j d_{S_j})-(n(n+1)/2)$, yielding $\mathsf{M}(d_1+(\sum_j d_{S_j})-(n(n+1)/2))$.

For Step 7: after summing up over j the three terms $\mathsf{M}(C_j)$, $4\mathsf{M}(B_j)$ and $\mathsf{M}(jd_{S_j})$ involved in the estimate of (d), we obtain $\mathsf{M}((n-1)d_1+\tilde{C}_k)$, $4\mathsf{M}((n-1)d_1-(\sum_j d_{S_j}))$ and $\mathsf{M}(d_1)$ respectively (here $\tilde{C}_k=0$ if $C_j=d_1-jd_{S_j}$ for all j, or $\tilde{C}_k=d_{S_k}-k-(d_1-kd_{S_k})$ if for some, necessarily unique k, $C_k=d_{S_k}-k$). As one can see, the dominating term is $\mathsf{M}(n(d_1-1)+\tilde{C}_k)\le\mathsf{M}(nd_1)$. It is multiplied by d_2, largely outclassing the cost of Step 6. Therefore, the overall cost of Step 6 and Step 7 is dominated by a term is $d_2\mathsf{M}(nd_1)$ yielding the upper bound of (e).

Finally Step 9 consists in multiplying the d_2 coefficients in x of $q_k\in(K[x])[y]$ by m_k. The degrees at play are respectively kd_{S_k} and $d_1-kd_{S_k}$, yielding a $d_2\mathsf{M}(C_k)$ where $C_k=\max\{d_1-kd_{S_k}, kd_{S_k}\}$. Let $\mathcal{I}:=\{1\le k\le n \mid kd_{S_k}=\max_j\{jd_{S_j}\}\}$. If $C_k=kd_k$ for some k, then $d_1-kd_{S_k}=\sum_{j\neq k} jd_j\le kd_k$ showing that $k\in\mathcal{I}$ and $\mathcal{I}=\{k\}$.

It remains to sum over $j=1,\ldots,n$. As for the multiplication, in the case where $C_k=d_1-kd_{S_k}$ for all k, we obtain the upper bound $d_2\mathsf{M}((n-1)d_1+\tilde{C}_k)$, where $\tilde{C}_k=0$ if for all j, $C_j=d_1-jd_{S_j}$, and $\tilde{C}_k=2kd_{S_k}-d_1$ if there is a (unique) k for which $C_k=kd_k$. This is upper bounded by $d_2\mathsf{M}(nd_1)$ and thus by $O(nd_2d_1\log(nd_1)\log(\log(nd_1)))$ as stated.

We can see that Step 7 or Step 9 in worst case outclasses other steps, yielding the result (g). □