Quadratic Newton Iteration for Systems with Multiplicity

G. Lecerf

Laboratoire GAGE, UMS MEDICIS,
École polytechnique, 91128 Palaiseau, France
lecerf@gage.polytechnique.fr

Preliminary version of 9th November 2001

Abstract

Newton's iterator is one of the most popular components of polynomial equation system solvers, either from the numeric or symbolic point of view. This iterator usually handles smooth situations only (when the Jacobian matrix associated to the system is invertible). This is often a restrictive factor. Generalizing Newton's iterator is still an open problem: how to design an efficient iterator with a quadratic convergence even in degenerate cases? We propose an answer for a $\mathfrak{m}$-adic topology when the ideal $\mathfrak{m}$ can be chosen generic enough: compared to a smooth case we prove quadratic convergence with a small overhead that grows with the square of the multiplicity of the root.

AMS Classification: 14-04, 14Q20, 14B05, 68W30.

1 Introduction

Let us consider the polynomial equation system $f_1 = f_2 = f_3 = 0$ in the unknowns x_1, x_2, x_3, where:

$$\begin{aligned} f_1 &:= 2\,x_1 + 2\,x_1^2 + 2\,x_2 + 2\,x_2^2 + x_3^2 - 1, \\ f_2 &:= (x_1 + x_2 - x_3 - 1)^3 - x_1^3, \\ f_3 &:= (2\,x_1^3 + 5\,x_2^2 + 10\,x_3 + 5\,x_3^2 + 5)^3 - 1000\,x_1^5. \end{aligned}$$

It is easy to check that $x^* := (0, 0, -1)$ is an isolated multiple root of this system. Assume that, for a given prime number p, we are given an approximate root $z := (z_1, z_2, z_3)$ in $\mathbb{Z}^3$, where $z_1 = 0 \bmod p$, $z_2 = 0 \bmod p$ and $z_3 = -1 \bmod p$. Our problem is to recover x^* from z. Let F denote the sequence of the above polynomials f_1, f_2, f_3 in $\mathbb{Z}[x_1, x_2, x_3]$.

If x^* were a simple root then the classical Newton iterator N would solve our problem:

$$N(z) := z - \left(\frac{dF}{dx}(z)\right)^{-1} F(z).$$

The iterator N is well-defined over the ring of the p-adic integers $\mathbb{Z}_p$ if the Jacobian matrix $\frac{dF}{dx}$ is invertible at z. Except for a finite number of primes p this condition is satisfied and the sequence $(N^\kappa(z))_{\kappa\geq 0}$ of the iterated of z converges to x^* quadratically in $\mathbb{Z}_p^3$, that is:

$$N^\kappa(z) - x^* \in p^{2^\kappa}\mathbb{Z}_p^3.$$

Then, using a rational reconstruction algorithm [Dix82], one can recover x^* from a p-adic precise enough approximation.

One can check that the multiplicity M of x^* is 18. This can be computed as the dimension of the $\mathbb{Q}$-algebra $\mathbb{Q}[[x_1, x_2, x_3+1]]/(F)$ thanks to the software Singular [GPS01]. Moreover the above system has 54 isolated solutions counted with multiplicity; x^* is the only multiple root.

The purpose of this article is the construction of an iterator $\widetilde{N}$ that generalizes N for multiple roots. The validity of $\widetilde{N}$ still depends on the choice of a lucky prime p. We show that there exists only a finite number of unlucky p. The number of operations in $\mathbb{Z}_p$ executed by $\widetilde{N}$ is linear in the evaluation complexity of the input system F and in the square of the multiplicity (up to logarithmic factors), but it is important to underline that the algorithm does not compute the multiplicity and does not need to know it.

1.1 Main Result

Let $\mathfrak{o}$ be a Noetherian domain and k its field of fractions. The reader may keep in mind that our cases of interest are $\mathfrak{o} = K[t]$, $\mathfrak{o} = K[t_1, \ldots, t_m]$ (where K is a field) and $\mathfrak{o} = \mathbb{Z}$ (as in the above situation). We denote by $\overline{k}$ the algebraic closure of k. We are given $f_1, \ldots, f_s$ polynomials in $\mathfrak{o}[x_1, \ldots, x_n]$. Let x^* be an isolated point (for the Zariski topology, see for instance [Mat86, §4]) of multiplicity M of the algebraic variety $\{x \in \overline{k}^n,\ f_1(x) = \cdots = f_s(x) = 0\}$. The multiplicity M is the dimension of the $\overline{k}$-algebra $\overline{k}[[x_1 - x_1^*, \ldots, x_n - x_n^*]]/(f_1, \ldots, f_s)$. Roughly speaking, we are concerned with the following lifting problem: can we recover x^* from one of its approximations modulo a maximal ideal $\mathfrak{m}$ of $\mathfrak{o}$?

We propose a partial answer to this question: our method works for some lucky ideals $\mathfrak{m}$ only. The first restrictions on $\mathfrak{m}$ are natural and concern the specialization of the root x^* modulo $\mathfrak{m}$.

Let Q be a *monic irreducible* polynomial of $k[T]$. We denote by u the image of T in the algebraic extension $k(u) := k[T]/(Q(T))$. We assume that $x^* \in k(u)^n$ and:

> (H_Q) There exists $\rho_Q \in \mathfrak{o}$ such that ρ_Q is a unit in $\mathfrak{o}/\mathfrak{m}$, $\rho_Q Q \in \mathfrak{o}[T]$ and the discriminant of $\rho_Q Q$ is a unit in $\mathfrak{o}/\mathfrak{m}$.

Concerning the classical mathematical background we refer to [Mat86, §8]. We denote by $\hat{\mathfrak{o}}$ the completion of $\mathfrak{o}$ with respect to the $\mathfrak{m}$-adic topology. Any a in $\mathfrak{o}$ that is a unit in $\mathfrak{o}/\mathfrak{m}$ is a unit in $\mathfrak{o}/\mathfrak{m}^\kappa$ for any integer $\kappa \geq 1$ and is also a unit in $\hat{\mathfrak{o}}$. Under hypothesis (H_Q), the quotient ring $A := \hat{\mathfrak{o}}[T]/(Q(T))$ is well-defined as an $\hat{\mathfrak{o}}$-algebra of dimension the degree

of Q and inherits the complete and separated $(\mathfrak{m}A)$-adic topology. Multiplication in A is continuous: if $z_i \in \mathfrak{m}^{\kappa_i}A$, for $i = 1, 2$ and integers $\kappa_i \geq 0$, then $z_1 z_2 \in \mathfrak{m}^{\kappa_1+\kappa_2}A$.

From a practical point of view the situation is the following. We want to compute in A but knowing only an approximation $q \in \mathfrak{o}[T]$ of Q, that is $\rho_Q(Q - q)$ has all its coefficients in $\mathfrak{m}$. But it suffices to observe that A is isomorphic to $\hat{\mathfrak{o}}[T]/(q(T))$ (see Proposition 1 below) to make the computations in A effective.
In order to embed x^* in A we need the following hypothesis (H_{x^*}). We denote by p the canonical projection from $k[T]$ onto $k[u]$. Let $p^{-1} : k[u] \to k[T]$ be the linear map such that $p^{-1}(e)$ is the unique polynomial of degree less than the degree of Q and such that its projection in $k[u]$ is e.

> (H_{x^*}) There exists ρ_{x^*} in $\mathfrak{o}$ such that ρ_{x^*} is a unit in $\mathfrak{o}/\mathfrak{m}$ and the polynomial $\rho_{x^*} p^{-1}(x_i^*)$ is in $\mathfrak{o}[T]$, for $i = 1, \ldots, n$.

We still write x^* for the image of x^* in A when there is no danger of confusion. Let $z \in A^n$ be an approximation of x^* modulo $\mathfrak{m}$. In the case $M = 1$ (we say that x^* is a simple root) it is well-known that Newton's iterator answers our lifting problem ([Lan93, XII, §7] for instance). Let F denote the vector of polynomials $(f_1, \ldots, f_s)$ and $\frac{dF}{dx}$ the Jacobian matrix of F. If

> (H_J) The determinant of $\frac{dF}{dx}(x^*)$ in a unit in $A/\mathfrak{m}$,

then Newton's iterator N is well-defined in any neighborhood of x^*:

$$N(z) = z - \left(\frac{dF}{dx}(z)\right)^{-1} F(z).$$

Its convergence is **quadratic**, that is for all $\kappa \geq 1$:

$$z - x^* \in (\mathfrak{m}^{\kappa}A)^n \implies N(z) - x^* \in (\mathfrak{m}^{2\kappa}A)^n.$$

If the multiplicity M is greater than 1 then Newton's iterator is not defined anymore at x^*. Our aim is the generalization of this iterator N in order to handle multiple roots. We introduce an iterator $\widetilde{N}$ answering this problem and prove that the overhead is mainly M^2. Our algorithm works under some genericity conditions: the characteristic of k must be big enough, the coordinates must be generic enough and the maximal ideal $\mathfrak{m}$ must satisfy (H_Q), (H_{x^*}) and other conditions generalizing (H_J). The change of coordinates is represented by a matrix $\mathcal{M}$ of $GL_n(k)$ (the group of invertible $n \times n$ matrices over k). The complexity model we use is stated in §4.1. The constant Ω comes mainly from linear algebra complexity: $3 \leq \Omega < 4$. We refer to §4.1 for further details.

Theorem 1 *Let $\mathfrak{o}$ be a Noetherian domain and k its field of fractions. There exists a deterministic algorithm performing the following task. The inputs of the algorithm are:*

- *A sequence $f_1, \ldots, f_s$ of polynomials in $\mathfrak{o}[x_1, \ldots, x_n]$ given by a straight-line program of length L.*

- $q, v_1, \ldots, v_n$, *a sequence of polynomials in* $\mathfrak{o}[T]$;
- *A matrix* $\mathcal{M}$ *of* $GL_n(k)$.
- *A maximal ideal* $\mathfrak{m}$ *of* $\mathfrak{o}$.

We assume that the input satisfies the following hypotheses:

- q *is monic.*
- *There exists polynomials* $Q, V_1, \ldots, V_n$ *in* $k[T]$ *such that:*
  - Q *is monic and irreducible.*
  - *The point* $x^* = (V_1(u), \ldots, V_n(u))$ *in* $k(u) := k[T]/(Q(T))$ *is an isolated root of the system* $f_1 = \cdots = f_s = 0$ *with multiplicity* M.
  - *There exist* ρ_Q *and* ρ_{x^*} *in* $\mathfrak{o}$ *and being units in* $\mathfrak{o}/\mathfrak{m}$ *such that the polynomials* $\rho_Q Q$ *and* $\rho_{x^*} V_i$ *have all their coefficients in* $\mathfrak{o}$ *and* $\rho_Q(Q - q)$ *and* $\rho_{x^*}(V_i - v_i)$, *for* $i = 1, \ldots, n$, *have all their coefficients in* $\mathfrak{m}$.
- *The characteristic* $\mathrm{char}(k)$ *is either* 0 *or at least* $M + 1$.
- *The matrix* $\mathcal{M}$ *is outside of an algebraic hypersurface of* $GL_n(k)$ *depending on the input polynomials and the root* x^*.
- *The ideal* $\mathfrak{m}$ *does not contain an element* $a \neq 0$ *of* $\mathfrak{o}$ *that depends on the input polynomials, the root* x^* *and* $\mathcal{M}$.

Let z *be the image of* $(v_1, \ldots, v_n)$ *in* $A := \hat{\mathfrak{o}}[T]/(q(T))$, *we still write* x^* *for its image in* A. *Let* κ *be a lower bound on the precision of* z *as an approximation of* x^* *in* A: $z - x^* \in (\mathfrak{m}^\kappa A)^n$. *Under the above conditions the algorithm computes polynomials* $\widetilde{v}_1, \ldots, \widetilde{v}_n$ *in* $\mathfrak{o}/(\mathfrak{m}^{2\kappa})[T]$ *such that the image* $\widetilde{N}(z)$ *of* $(\widetilde{v}_1, \ldots, \widetilde{v}_n)$ *in* A *approximates* x^* *at precision at least* 2κ, *that is:* $\widetilde{N}(z) - x^* \in (\mathfrak{m}^{2\kappa} A)^n$. *The algorithm performs*

$$\mathcal{O}\Big(n^3(nL + n^\Omega)M^2 \log(nM)\Big)$$

arithmetic operations in $A/(\mathfrak{m}^{2\kappa} A)$.

As an immediate consequence of the above theorem, the sequence $(\widetilde{N}^l(z))_{l \geq 0}$ converges quadratically to x^*: $\widetilde{N}^l(z) - x^* \in (\mathfrak{m}^{2^l \kappa} A)^n$.

It is important to notice that the theorem does not state neither the existence of a lucky $\mathfrak{m}$ nor generic enough changes of coordinates. If k has characteristic zero then almost all random choices of coordinates are generic enough. In the case when $\mathfrak{o} = K[t_1, \ldots, t_n]$ (where K is a field) the maximal ideals $\mathfrak{m} = (t_1 - p_1, \ldots, t_n - p_n)$ that are not lucky come from points $(p_1, \ldots, p_n)$ included in an algebraic hypersurface in K^n. Hence, if K has characteristic zero then almost all maximal ideals are lucky. In the case when $\mathfrak{o} = \mathbb{Z}$ only a finite number of maximal ideals are not lucky. We leave probability estimates for further work and focus only on the algorithm.

From [GLS01, Lemma 2] we recall that for the special case $M = 1$ the complexity of Newton's iterator is in $\mathcal{O}(nL + n^{\Omega})$. Focusing on the dependency on M, the overhead of $\widetilde{N}$ grows with M^2 up to logarithmic factors.

The complexity model and the algorithm are presented in §4. The basic tools necessary for the algorithm are given in §2. The mathematical idea is developed in §3. In §5 we use the classical dynamic evaluation framework to handle reducible sets of roots. In the next paragraphs we give some details about the important consequences in the field of polynomial systems solving. We conclude this introduction with the presentation of the new underlying ideas along with examples.

1.2 Motivation for Polynomial System Solving

In [GLS01] we present an algorithm to solve systems of polynomial equations and inequations: a practical variant of the *geometric resolution algorithm*. We recall the main framework of this algorithm. Let k be a field of characteristic zero. We are given polynomials $f_1, \ldots, f_n, g$ in $k[x_1, \ldots, x_n]$ and are interested in computing a description of the set of roots of the system $f_1 = \cdots = f_n = 0,\ g \neq 0$. Let $\mathcal{V}(f_1, \ldots, f_i)$ (resp. $\mathcal{V}(g)$) denote the algebraic variety solution of $f_1 = \cdots = f_i = 0$ (resp. $g = 0$). Let $\mathcal{V}_i$ be the Zariski closure of $\mathcal{V}(f_1, \ldots, f_i) \backslash \mathcal{V}(g)$, for $i = 1, \ldots, n$. Our algorithm is incremental in the number of equations to be solved, we compute a description of $\mathcal{V}_{i+1}$ from one of $\mathcal{V}_i$. Our method works under the following restrictive hypotheses:

(R_1) *Regularity hypothesis:* each $\mathcal{V}_i$ is equidimensional of dimension $n - i$;

(R_2) *Reduction hypothesis:* the Jacobian matrix of $f_1, \ldots, f_i$ has full rank when evaluated at $\mathcal{V}_i$.

We assume that the affine coordinates are generic enough (if this is not the case then replace them by a random affine transformation). Roughly speaking here is the incremental step of our solver: at step i, the variety $\mathcal{V}_i$ is represented by the finite set of solutions of the system $f_1 = \cdots = f_i = x_1 = \cdots = x_{n-i} = 0, g \neq 0$, called a *lifting fiber*. From this set of points we compute the curves solutions of the system $f_1 = \cdots = f_i = x_1 = \cdots = x_{n-i-1} = 0,\ g \neq 0$. This curve is parametrized by the variable x_{n-i}, it is called a *lifting curve*. This computation is called the *lifting step*. Then we compute the intersection of this curve with the next equation $f_{i+1} = 0$, this is the *intersection step*: according to hypothesis (R_1) this yields a finite set of points from which we remove the solutions of $g = 0$, this is the *cleaning step*. This way, we obtain a lifting fiber for $\mathcal{V}_{i+1}$, which represents the solutions of the system $f_1 = \cdots = f_{i+1} = x_1 = \cdots = x_{n-i-1} = 0,\ g \neq 0$.

We focus on the lifting step: the one of [GLS01] relies on Newton's iterator, the invertibility of the Jacobian matrix occurring in this iterator is equivalent to hypothesis (R_2). This is a restrictive factor of the method. Theorem 1 remedies this problem. Full details about the complete resulting solver are given in [Lec01]: we show how to get rid of hypothesis (R_1) as well and how to compute the equidimensional algebraic decompositions of the $\mathcal{V}_i$ in sequence.

1.3 Brief History

For centuries Newton's method has certainly been the most famous approach for solving equations and systems of equations numerically, but the idea of using it in a symbolic solver is more recent. We refer to Schost's thesis [Sch00, Chapitre 6] for a detailed historical presentation.

In the non-archimedian case of $\mathfrak{I}$-adic topologies, where $\mathfrak{I}$ is an ideal of a ring R, we attribute the introduction of Newton's method in computer algebra to Zassenhaus [Zas69] for greatest common divisor computations (known as Hensel's lemma). In the field of polynomial equation system solving, the earliest occurrence of Newton's iterator seems to be due to Trinks [Tri85] in 1985: he proposed to lift to the rational numbers of a shape-lemma [GM89] Gröbner basis which is only known modulo a lucky prime number.

In 1988, Winkler [Win88] generalized Trinks' approach for the computations of p-adic approximations of Gröbner bases. The choice of a lucky p is then discussed in several papers: Gianni [Gia87], Kalkbrener [Kal87, Kal97], Pauer [Pau92], Gräbe [Grä93], Assi [Ass94], and more recently Gianni, Fortuna and Trager [GFT00]. In [Nau98] Nauheim proposes a method to handle lifting with an unlucky p.

The *geometric resolution algorithm* we are concerned with has been introduced by Giusti, Heintz, Morais, Morgenstern and Pardo [GHM$^+$98] in the early 1990s: Newton's iterator is used to compress the straight-line programs encoding the resolutions of the intermediate varieties $\mathcal{V}_i$ in the incremental solving process. The representation of each $\mathcal{V}_i$ has a size polynomial in its degree. The resulting solver has a complexity mainly polynomial in the maximum of the degrees of the intermediate varieties $\mathcal{V}_i$. This was a breakthrough in theoretical complexity. This idea has been developed in a series of papers [Mor97, GHH$^+$97, GHMP97, Häg98, HKP$^+$00]. From a practical point of view, the geometric resolution algorithm has been turned into an efficient software called Kronecker [Lec99]. It has been designed by Giusti, Lecerf and Salvy [GLS01] and implemented in the Magma computer algebra system [BC89, BC90, BCM94, BC95, CP96, BCP97]: the theoretical algorithm has been completely redesigned and simplified. We improved its complexity dramatically. We introduced, independently of Heintz, Matera and Waissbein [HMW01], the notion of *lifting curves*.

I propose in [Lec00] a theoretical generalization of the geometric resolution algorithm for computing an equidimensional decomposition of the solution set of any polynomial equation system (removing the above regular reduced restrictive hypotheses (R_1) and (R_2)). My approach is based on Bertini's first theorem as initiated in [KP96, Mor97]: the input system is replaced with generic linear combinations of the given equations. I prove that the only lifting to perform concerns the smooth components and can be done using the classical Newton iterator. But my algorithm presents two main drawbacks. On the one hand, the substitution of the original system by linear combinations of the original equations spoils the evaluation complexity of the intermediate systems. On the other hand the theoretical description does not lead to an easy implementation as in [GLS01] when applying the *deforestation* idea introduced in [GHL$^+$00] to eliminate straight-line programs in the intermediate computations. Jeronimo and Sabia also

propose a generalization of the algorithm in [JS00]. Their approach and purposes are different: they provide an idealistic description of each equidimensional component instead of a geometric resolution. This is a less convenient output for numerical solving.

The purpose of this article is the generalization of the algorithm presented in [GLS01], but without mixing the equations of the input system and keeping the natural incremental resolution process. The main problem is to deal with situations featuring multiple components.

From a numerical point of view the one dimensional case is now well understood as demonstrated in Yakoubsohn's recent work [Yak00]. But in several variables there exists no satisfying generic Newton iterator handling multiplicities. In [MS95] Möller and Stetter postprocess Gröbner bases numerically in order to compute all the roots of a zero-dimensional polynomial equation system. In [Ste96] Stetter exploits some Gröbner bases in order to obtain local information about a cluster of roots. A generic numerical iterator has been proposed by Ojika, Watanabe and Mitsui in [Oji82, OWM83, Oji87]. Their idea consists in replacing the original system by another one for which the considered singular root has smaller multiplicity. This is done by differentiating well chosen equations. After a finite number of steps they obtain a system for which the considered root is simple. The computations are done mixing numerical and symbolical manipulations. It is a pity that their study lacks stability and complexity analyses. Since they call their algorithm the *Modified Deflation Algorithm*, we will refer to our method as a **deflation algorithm**.

1.4 Presentation of the Method

Before entering the mathematical framework of our algorithm we introduce the basic ideas along examples.

Example 1 We start with the easiest case, with one variable only: $n = 1$, $\mathfrak{o} = \mathbb{Q}[t]$, $k = \mathbb{Q}(t)$ and one polynomial $f(x)$ in $\mathfrak{o}[x]$. Let $p \in \mathbb{Q}$ and $\mathfrak{m} = (t - p)$, we are given an algebra $A = \hat{\mathfrak{o}}[T]/(q(T))$ as defined above in §1.1 and an approximation $z \in A$ of a root x^* of f of multiplicity M. If M is known and is greater than 1 we can replace f by its $(M-1)$st derivative $\widetilde{f} := \frac{\partial^{M-1} f}{\partial x^{M-1}}$ and then use the classical Newton iterator (if $\widetilde{f}'(x^*)$ is invertible in A). Practically we proceed this way: to evaluate $\widetilde{f}$ and its first order derivative at the point z, we take a new variable dx and evaluate f in the power series ring $A[[dx]]$ at $z + dx$ and precision $\mathcal{O}(dx^{M+1})$:

$$f(z + dx) = \sum_{i=0}^{M} \frac{1}{i!} \frac{d^i f}{dx^i}(z) dx^i + \mathcal{O}(dx^{M+1}),$$

so that introducing the function **coeff** to extract the coefficient of its first argument with respect to its second one, we deduce the iterator:

$$\widetilde{N}(z) := z - \widetilde{f}/\widetilde{f}'(z) = z - \frac{1}{M} \frac{\text{coeff}(f(z + dx), dx^{M-1})}{\text{coeff}(f(z + dx), dx^M)}. \tag{1}$$

Now let us assume that M is *a priori* unknown. Our strategy is to determine M and then to use the iterator $\widetilde{N}$. The success of the algorithm relies on the choice of $\mathfrak{m}$.

We compute the multiplicity M_p of z as a root of f in $A/\mathfrak{m}A$. For this purpose we evaluate f at $z+dx$ in the power series ring $A/\mathfrak{m}A[[dx]]$:

$$M_p := \mathrm{val}_{dx} f(z+dx),$$

where val denotes the valuation function. Assuming that $M_p = M$ and that $\frac{d^M f}{dx^M}(x^*)$ is a unit in $A/\mathfrak{m}A$ then the sequence $(\widetilde{N}^\kappa(z))_\kappa$ converges quadratically to x^*.

As for the justification of this algorithm we observe that $M_p = M$ if and only if $\frac{\partial^M f}{\partial x^M}(z)$ is a unit in $A/\mathfrak{m}A$. This condition generalizes (H_J). In particular this proves that, except for a finite set of choices, almost all p yield a correct multiplicity.

Example 2 We now take $n=2$, $\mathfrak{o}=\mathbb{Q}[t]$, $f_1=(x_1-t)^3-x_2^2$, $f_2=x_2^4+6t^5x_1-t^6$. One can check that the set of roots of

$$x_1=0,\ x_2^2+t^3=0 \tag{2}$$

is an isolated component of the variety defined by $f_1=f_2=0$ of multiplicity $M=2$. This is confirmed by the following computations.

Let p be a point in $\mathbb{Q}$ and $\mathfrak{m}=(t-p)$. We are given $q(T)=T^2+p^3$, the corresponding algebra $A=\mathbb{Q}[[t-p]][T]/(q(T))$ and the approximate root $z=(0,u)$, where u denotes the image of T in A. In order to satisfy (H_Q) and (H_{x^*}) it suffices that $p\neq 0$.

Let dx_2 be a new variable. First we compute an approximation of $y_1\in A[[dx_2]]$, the power series solution of $f_1(y_1,u+dx_2)=0$ using an effective version of the implicit function theorem:

$$y_1=(t-p)+\frac{2}{3p^2}udx_2-\frac{1}{9p^2}dx_2^2+\mathcal{O}((t-p)^2,dx_2^3).$$

Substituting x_1 by the above approximation of y_1 in the other equation $f_2=0$ we get:

$$0=20p^2u(t-p)dx_2-\frac{10}{3}p^2(2p+(t-p))dx_2^2+\mathcal{O}((t-p)^2,dx_2^3).$$

Differentiating the above equation with respect to dx_2 yields a linear equation in dx_2:

$$0=20p^2u(t-p)-\frac{20}{3}p^2(2p+(t-p))dx_2+\mathcal{O}((t-p)^2,dx_2^2).$$

This equation admits a unique solution of valuation 1 and precision $\mathcal{O}((t-p)^2)$:

$$dx_2=\frac{3}{2p}u(t-p)+\mathcal{O}((t-p)^2).$$

We deduce that $(0,u(1+\frac{3}{2p}(t-p)))$ is an approximate root at precision $\mathcal{O}((t-p)^2)$. From $x_2^*=u(1+\frac{3}{2p}(t-p))+\mathcal{O}((t-p)^2)$ we deduce that $u=x_2^*(1-\frac{3}{2p}(t-p))+\mathcal{O}((t-p)^2)$. Substituting u by this value in $u^2+p^3=0$ we recover an approximation of (2):

$$x_1=0,\quad x_2^2+p^3+3p^2(t-p)=0+\mathcal{O}((t-p)^2).$$

We could repeat this process once more and reach the precision $\mathcal{O}((t-p)^4)$. In this way we recover (2) completely.

Example 3 We are coming back to the example of the beginning: $n = 3$, $\mathfrak{o} = \mathbb{Z}$, $k = \mathbb{Q}$, and $F = f_1, f_2, f_3$, where

$$\begin{aligned} f_1 &:= 2\,x_1 + 2\,x_1^2 + 2\,x_2 + 2\,x_2^2 + x_3^2 - 1, \\ f_2 &:= (x_1 + x_2 - x_3 - 1)^3 - x_1^3, \\ f_3 &:= (2\,x_1^3 + 5\,x_2^2 + 10\,x_3 + 5\,x_3^2 + 5)^3 - 1000\,x_1^5. \end{aligned}$$

The root $x^* = (0, 0, -1)$ lies in k^3 and has multiplicity 18. For the sake of simplicity we chose an example without algebraic extension. The Jacobian matrix of F at x^* has rank 1. We can not treat this example now, it will serve to illustrate our algorithm later in §3.2 and §4.5.

2 Preliminaries

In this section we present the foundations of our generalized Newton iterator.

2.1 Local and Global Point of Views

We explain in [GLS01, §4] how to lift an algebraic eliminant polynomial in a global way. Informally speaking we recall that the local point of view corresponds to situations when the parameterization of the variables are known with greater precision than the minimal polynomial defining the algebra in which the computations are done. The global point of view is when the minimal polynomial is updated at each improvement of the precision. The following proposition enlightens a bit the results of [GLS01, §4]. It says that both computations are equivalent.

Proposition 1 *Let Q and q be monic polynomials in $\hat{\mathfrak{o}}[T]$ such that $Q - q$ has all its coefficients in $\mathfrak{m}$ and the discriminant of Q is a unit in $\hat{\mathfrak{o}}/\mathfrak{m}$. Then $\hat{\mathfrak{o}}[T]/(Q(T))$ is homeomorphic to $\hat{\mathfrak{o}}[T]/(q(T))$. For any integer $\kappa \geq 1$, $(\hat{\mathfrak{o}}/\mathfrak{m}^\kappa)[T]/(Q(T))$ is homeomorphic to $(\hat{\mathfrak{o}}/\mathfrak{m}^\kappa)[T]/(q(T))$.*

Proof. The idea is to construct a root of q in $\hat{\mathfrak{o}}[T]/(Q(T))$ and a root of Q in $\hat{\mathfrak{o}}[T]/(q(T))$ and use these roots to construct the homeomorphisms. We denote by U (resp. u) the image of T in $\hat{\mathfrak{o}}[T]/(Q(T))$ (resp. $\hat{\mathfrak{o}}[T]/(q(T))$). Since $Q - q$ has all its coefficients in $\mathfrak{m}$ the discriminant of q equals the discriminant of Q in $\hat{\mathfrak{o}}/\mathfrak{m}$. Hence the derivative $q'(U)$ is invertible in $\hat{\mathfrak{o}}[T]/(Q(T))$. In $\hat{\mathfrak{o}}[T]/(Q(T))$ we build the sequence $(a_\kappa)_{\kappa \geq 0}$:

$$a_0 = U, \quad a_{\kappa+1} = a_\kappa - \frac{q(a_\kappa)}{q'(a_\kappa)}, \ \kappa \geq 0.$$

Since $q(U) = 0$ modulo $\mathfrak{m}$ the sequence (a_κ) converges quadratically to a root a of q in $\hat{\mathfrak{o}}[T]/(Q(T))$:

$$a - a_\kappa \in \mathfrak{m}^{2^\kappa}\Big(\hat{\mathfrak{o}}[T]/(Q(T))\Big).$$

Since a is a root of q the following map g is well-defined as a continuous $\hat{\mathfrak{o}}$-algebra morphism:

$$\begin{array}{rccc} g: & \hat{\mathfrak{o}}[T]/(q(T)) & \to & \hat{\mathfrak{o}}[T]/(Q(T)) \\ & u & \mapsto & a \end{array}$$

Exchanging the roles of Q and q we construct b as a root of Q in $\hat{\mathfrak{o}}[T]/(q(T))$ and define h:

$$\begin{array}{rccc} h: & \hat{\mathfrak{o}}[T]/(Q(T)) & \to & \hat{\mathfrak{o}}[T]/(q(T)) \\ & U & \mapsto & b \end{array}$$

Since $q(u) = 0$ in $\hat{\mathfrak{o}}[T]/(q(T))$ then $h(g(q(u))) = q(h(g(u))) = 0$. Since $h(g(u)) = 0$ modulo $\mathfrak{m}$, it follows that $h(g(u)) = u$ in $\hat{\mathfrak{o}}[T]/(q(T))$ and in a similar way that $g(h(U)) = U$ in $\hat{\mathfrak{o}}[T]/(Q(T))$. This proves that g is an isomorphism and that $g^{-1} = h$. By construction g and h are both continuous and can be restricted modulo $\mathfrak{m}^\kappa$ for any $\kappa \geq 1$. □

2.2 Basic Notations and Definitions

Let k be a field and S denote $k[[x_1, \ldots, x_n]]$, the power series ring in n variables over k. We denote by $\mathrm{val}(\phi)$ the **valuation** of ϕ in S. By convention the valuation of 0 is $+\infty$. For any subset Φ of S we define $\mathrm{val}(\Phi)$ as the minimum of the valuations of its elements. If Φ is empty, it has valuation $+\infty$. The **support** of a polynomial or a series is its set of monomials with a nonzero coefficient.

The first tool we need is a local counterpart of the classical Noether normalization lemma for algebraic varieties (see for instance [Mat86, §33]). Let Ψ be an ideal of S of valuation m, we say that a variable x_i is in **Weierstraß position** if there exists an element of Ψ of valuation m and having x_i^m in its support.

Like Noether positions, Weierstraß positions are easy to obtain: if m is the valuation of the ideal Ψ (assume that $\Psi \neq (0)$), then there exists an element ψ in Ψ of valuation m. Let ψ_m be the homogeneous component of valuation m of ψ and $a_1, \ldots, a_{n-1}$ in k such that $\psi_m(a_1, \ldots, a_{n-1}, 1)$ is not zero (assume that such a point exists), then the following change of variables puts x_n into Weierstraß position: replace x_i by $x_i + a_i x_n$ for $1 \leq i \leq n-1$. In particular, if k has characteristic zero it is always possible to find such a_i.

Lemma 1 *Let Ψ be an ideal of S, there exists an algebraic hypersurface of k^{n-1} such that for any element $(a_1, \ldots, a_{n-1})$ outside of it the following change of variables yields a Weierstraß position for x_n: replace x_i by $x_i + a_i x_n$ for $1 \leq i \leq n-1$.*

Let Φ be a subset of S, we define its **first partial derivative** $\frac{\partial \Phi}{\partial x_i}$ with respect to the variable x_i as the set $\Phi \cup \{\frac{\partial \phi}{\partial x_i},\ \phi \in \Phi\}$. If Ψ is an ideal of S then so is $\frac{\partial \Psi}{\partial x_i}$. Moreover the derivative of the ideal generated by Φ is generated by the derivative of Φ.

Lemma 2 *Let $\alpha_i \geq 0$, $i = 1, \ldots, n$. If $\alpha_l \geq 1$ then*

$$\frac{\partial}{\partial x_l}(x_1^{\alpha_1}, \ldots, x_n^{\alpha_n}) = (x_1^{\alpha_1}, \ldots, x_{l-1}^{\alpha_{l-1}}, x_l^{\alpha_l - 1}, x_{l+1}^{\alpha_{l+1}}, \ldots, x_n^{\alpha_n}).$$

Corollary 1 *Let $\alpha_i \geq 0$, $\beta_i \geq 0$, for $i = 1, \ldots, n$. Let $\pi_1 = (x_1^{\alpha_1}, \ldots, x_n^{\alpha_n})$ and $\pi_2 = (x_1^{\beta_1}, \ldots, x_n^{\beta_n})$, if $\alpha_l \geq 1$ and $\beta_l \geq 1$ then*

$$\frac{\partial}{\partial x_l}(\pi_1 \cap \pi_2) = \frac{\partial \pi_1}{\partial x_l} \cap \frac{\partial \pi_2}{\partial x_l}.$$

2.3 Gradient of an Ideal

In order to compute effectively in S we need to fix the precision of the series. The precisions used by our algorithm are built on what we call the *gradient* of an ideal. This construction is motivated by the following situation.

Let $\mathfrak{o}$ be a Noetherian domain, k be its field of fractions, f be a polynomial function in $\mathfrak{o}[x_1, \ldots, x_n]$ given by a *straight-line program* (see §4.1 for precise considerations) and π is a zero-dimensional monomial ideal of $S := k[[x_1, \ldots, x_n]]$ (an ideal generated by monomials). For short, we write monomials using multi-indices: if $\alpha = (\alpha_1, \ldots, \alpha_n)$, then x^α denotes $x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ and $|\alpha|$ the sum of the α_i: $|\alpha| := \alpha_1 + \cdots + \alpha_n$. For each monomial x^α which does not belong to π we want to compute the corresponding partial derivative of f at a given point $a := (a_1, \ldots, a_n)$: $\frac{\partial^{|\alpha|} f}{\partial x^\alpha}(a) := \frac{\partial^n f}{\partial x_1^{\alpha_1} \cdots \partial x_n^{\alpha_n}}(a)$. It is classical to handle this situation by evaluating f at the point $(a_1 + x_1, \ldots, a_n + x_n)$ modulo the ideal π and picking up the right coefficients. Introducing the function $\mathbf{coeff}(f, x^\alpha)$ that returns the coefficient of x^α in f, we know that for all $x^\alpha \notin \pi$:

$$\frac{\partial^{|\alpha|} f}{\partial x^\alpha}(a) = (\alpha_1! \cdots \alpha_n!)\mathrm{coeff}\Big(f(a_1 + x_1, \ldots, a_n + x_n), x^\alpha\Big). \tag{3}$$

We are interested in extending this computation in order to compute the values of the gradients of the $\frac{\partial^{|\alpha|} f}{\partial x^\alpha}$. For this purpose we construct an ideal denoted by $\nabla_{\mathcal{L}}\pi$ (where $\mathcal{L}$ stands for the set $\{1, \ldots, n\}$, for the moment) such that the evaluation modulo $\nabla_{\mathcal{L}}\pi$ instead of π yields the gradients of $\frac{\partial^{|\alpha|} f}{\partial x^\alpha}$ for all x^α at point a via (3). It suffices to take $\nabla_{\mathcal{L}}\pi$ as the biggest monomial ideal not containing the $x_i x^\alpha$, for all $i \in \mathcal{L}$ and all x^α not in π.

We first prove that this construction is optimal and then give the properties used in the proof of the correctness of our algorithm presented in §4.

Let us now formalize the above construction. We consider the set $\mathcal{M}$ of monomials generated by n indeterminates $x_1, \ldots, x_n$ with nonnegative exponents:

$$\mathcal{M} = \{x_1^{\alpha_1} \cdots x_n^{\alpha_n},\ \alpha_i \geq 0,\ i = 1, \ldots, n\},$$

it is a semigroup (a group without an inverse operation) for the multiplication of monomials. A subset π of $\mathcal{M}$ stable under multiplication by any element of $\mathcal{M}$, that is $\mathcal{M}\pi = \pi$, is called a **stable subset**. We denote by $(x^{\beta_1}, \ldots, x^{\beta_s})$ the stable subset $\cup_{i=1}^s \mathcal{M}x^{\beta_i}$ generated by the monomials $x^{\beta_1}, \ldots, x^{\beta_s}$.

Let π be a stable subset of $\mathcal{M}$, $\overline{\pi}$ its complement (in $\mathcal{M}$) and $\mathcal{L}$ a subset of $\{1, \ldots, n\}$, we define $\nabla_{\mathcal{L}}\pi$, the **gradient** of π with respect to the variables in $\mathcal{L}$, as the complement set of $\overline{\pi} \cup \cup_{i \in \mathcal{L}} x_i \overline{\pi}$. For example, if $n = 2$, $\pi = (x_1^{\alpha_1}, x_2^{\alpha_2})$ and $\mathcal{L} = \{1, 2\}$ then $\nabla_{\mathcal{L}}\pi = (x_1^{\alpha_1+1}, x_2^{\alpha_2+1}, x_1^{\alpha_1} x_2^{\alpha_2})$.

Lemma 3 *With the above notations, $\nabla_{\mathcal{L}}\pi$ is a stable subset of $\mathcal{M}$.*

Proof. Let x^α be a monomial of $\nabla_{\mathcal{L}}\pi$, it suffices to prove that for each i the monomial $x_i x^\alpha$ is in $\nabla_{\mathcal{L}}\pi$. If it were not the case then $x_i x^\alpha$ would be either in $\overline{\pi}$ or one of the $x_j x^\beta$ with $j \in \mathcal{L}$ and x^β in $\overline{\pi}$. The first situation immediately leads to a contradiction. As for the second situation, if i were equal to j we could deduce that $x^\alpha = x^\beta \in \overline{\pi}$, hence i would be different from j. From $x_i x^\alpha = x_j x^\beta$ we could construct γ such that $x^\beta = x_i x^\gamma$ and $x^\alpha = x_j x^\gamma$. Hence, x^γ would belong to $\overline{\pi}$ and x^α would be in $x_j\overline{\pi}$, which is a contradiction again. □

We extend this construction to monomial ideals. If π is a monomial ideal of S (an ideal generated by monomials) and $\mathcal{L}$ a subset of $\{1,\ldots,n\}$ we define the **gradient** of π with respect to the variables in $\mathcal{L}$, denoted by $\nabla_{\mathcal{L}}\pi$, as the monomial ideal generated by the gradient with respect to $\mathcal{L}$ of the stable subset spanned by a set of monomials generating π (this construction is independent of the choice of the generating set).

If π is a zero-dimensional ideal of S (in the sense that the quotient ring S/π has Krull dimension zero) we denote by $\deg(\pi)$ the **degree** of π that is the dimension of the finite dimensional k-algebra S/π. For a zero-dimensional ideal π the complement in $\mathcal{M}$ of $\pi \cap \mathcal{M}$ is finite: we call it the **support** of π. The support of π is a basis of the quotient ring S/π as a k-linear space. The cardinal of the support is finite and equals the degree of π. For example, if $n = 2$, $\mathrm{supp}(x_1^{\alpha_1}, x_2^{\alpha_2}) = \{x_1^i x_2^j,\ 0 \leq i \leq \alpha_1 - 1,\ 0 \leq j \leq \alpha_2 - 1\}$.

Proposition 2 *For two subsets $\mathcal{L}_1$ and $\mathcal{L}_2$ of $\{1,\ldots,n\}$ and for any monomial ideal π:*

$$\nabla_{\mathcal{L}_1 \cup \mathcal{L}_2}\pi = \nabla_{\mathcal{L}_1}\pi \cap \nabla_{\mathcal{L}_2}\pi.$$

Proof.

$$\begin{aligned}
&\mathrm{supp}(\nabla_{\mathcal{L}_1 \cup \mathcal{L}_2}\pi) \\
&\quad = \mathrm{supp}(\pi) \cup \cup_{i \in \mathcal{L}_1 \cup \mathcal{L}_2} x_i \mathrm{supp}(\pi) \\
&\quad = \mathrm{supp}(\pi) \cup \Big(\cup_{i \in \mathcal{L}_1}\ x_i \mathrm{supp}(\pi) \Big) \cup \Big(\cup_{i \in \mathcal{L}_2}\ x_i \mathrm{supp}(\pi) \Big) \\
&\quad = \mathrm{supp}(\nabla_{\mathcal{L}_1}\pi) \cup \mathrm{supp}(\nabla_{\mathcal{L}_2}\pi).
\end{aligned}$$

□

Let $\pi = (x_1^{\alpha_1}, \ldots, x_n^{\alpha_n})$, for $\alpha_i \geq 0$, $i = 1,\ldots,n$. Let $l \in \{1,\ldots,n\}$ then

$$\nabla_{\{l\}}\pi = (x_1^{\alpha_1}, \ldots, x_{l-1}^{\alpha_{l-1}}, x_l^{\alpha_l + 1}, x_{l+1}^{\alpha_{l+1}}, \ldots, x_n^{\alpha_n}),$$

we deduce:

Corollary 2 *Let $\alpha_i \geq 0$, $i = 1,\ldots,n$ and $\mathcal{L}$ be a subset of $\{1,\ldots,n\}$. We have the following formula:*

$$\nabla_{\mathcal{L}}(x_1^{\alpha_1}, \ldots, x_n^{\alpha_n}) = \cap_{i \in \mathcal{L}}(x_1^{\alpha_1}, \ldots, x_{i-1}^{\alpha_{i-1}}, x_i^{\alpha_i + 1}, x_{i+1}^{\alpha_{i+1}}, \ldots, x_n^{\alpha_n}).$$

Combined with Corollary 1 we deduce:

Corollary 3 *Let $\alpha_i \geq 0$, $i = 1, \ldots, n$, $\mathcal{L}$ be a subset of $\{1, \ldots, n\}$ and l such that $\alpha_l \geq 1$. We have the following equality:*

$$\frac{\partial}{\partial x_l} \nabla_{\mathcal{L}}(x_1^{\alpha_1}, \ldots, x_n^{\alpha_n}) = \nabla_{\mathcal{L}}\Big(\frac{\partial}{\partial x_l}(x_1^{\alpha_1}, \ldots, x_n^{\alpha_n})\Big).$$

From Corollary 2 we also deduce:

Corollary 4 *If $\alpha_i \geq 0$, for all $i = 1, \ldots, n$, then*

$$\nabla_{\{1,\ldots,n\}}(x_1^{\alpha_1}, \ldots, x_n^{\alpha_n}) \cap k[[x_2, \ldots, x_n]] = \nabla_{\{2,\ldots,n\}}(x_2^{\alpha_2}, \ldots, x_n^{\alpha_n}).$$

We give two useful bounds on the degree of the gradients:

Proposition 3 *According to the above notations, if π is zero-dimensional of degree M then $\nabla_{\mathcal{L}}\pi$ is zero-dimensional and*

$$\deg(\nabla_{\mathcal{L}}\pi) \leq (1 + \#\mathcal{L})\deg(\pi),$$

where $\#\mathcal{L}$ denotes the cardinal of the set $\mathcal{L}$.
If π is generated by $x_1^{\alpha_1}, \ldots, x_n^{\alpha_n}$, with $\alpha_i \geq 1$, for $i = 1, \ldots, n$, we have:

$$\deg(\pi) = \alpha_1 \cdots \alpha_n, \quad \deg(\nabla_{\mathcal{L}}\pi) = \deg(\pi)\Big(1 + \sum_{i \in \mathcal{L}} \frac{1}{\alpha_i}\Big).$$

Proof. The proof is straightforward from the definition of the gradient. □

In our complexity estimates we use the first bound only. But one has to keep in mind that it is not sharp at all as soon as $\pi \subset (x_1, \ldots, x_n)$.

Proposition 4 *For any monomial ideal π and two subsets $\mathcal{L}_1 \subseteq \mathcal{L}_2$ of the set $\{1, \ldots, n\}$ the following inclusions hold:*

$$\pi^2 \subseteq \nabla_{\mathcal{L}_2}\pi \subseteq \nabla_{\mathcal{L}_1}\pi \subseteq \pi.$$

Proof. The inclusions $\nabla_{\mathcal{L}_2}\pi \subseteq \nabla_{\mathcal{L}_1}\pi \subseteq \pi$ are true by construction. We prove that $\pi^2 \subseteq \nabla_{\mathcal{L}_2}\pi$. Let x^α be a monomial in the support of π and $i \in \mathcal{L}_2$. If the monomial $x_i x^\alpha$ were in π^2 we could write it as the product of two monomials of π: $x_i x^\alpha = x^\beta x^\gamma$. One of them would be a multiple of x_i, say x^β, then x^α would be a multiple of x^γ. This is a contradiction. □

The last technical result about gradients of ideals we need is:

Proposition 5 *For any positive integer $\lambda \geq 1$ the following inclusion holds:*

$$(x_1^\lambda, x_2, \ldots, x_n)\nabla_{\{1,\ldots,n\}}(x_1^\lambda, x_2, \ldots, x_n) \subseteq \nabla_{\{1,\ldots,n\}}(x_1^{2\lambda}, x_2, \ldots, x_n).$$

Proof. Let ζ_λ denote $(x_1^\lambda, x_2, \ldots, x_n)$. By construction we have:

$$\mathrm{supp}(\nabla_{\{1,\ldots,n\}}\zeta_\lambda) = \{x_1^j,\ 0 \leq j \leq \lambda - 1\} \cup \{x_1^j x_i,\ 1 \leq i \leq n,\ 0 \leq j \leq \lambda - 1\}.$$

We want to prove that any monomial of the support of $\nabla_{\{1,\ldots,n\}}\zeta_{2\lambda}$ is not in $\mathfrak{I} = \zeta_\lambda \nabla_{\{1,\ldots,n\}}\zeta_\lambda$; let x^α be one of these monomials. First, if x^α is x_1^j, with $j \leq 2\lambda$, then it can not be in $\mathfrak{I}$, since the smallest power of x_1 in $\nabla_{\{1,\ldots,n\}}\zeta_\lambda$ is $\lambda + 1$ and λ in ζ_λ. If $x^\alpha = x_1^j x_i$, with $0 \leq j \leq 2\lambda - 1$ and $i \neq 1$, were in $\mathfrak{I}$ then it could only be the product of x_1^l and $x_1^m x_i$, with $m \geq \lambda$ and $l \geq \lambda$, which is not possible. □

2.4 Deflation Lemma

The deflation lemma is the key for bounding the complexity of the deflation algorithm of §4: it shows that our deflation process has a good complexity behavior.

Lemma 4 *Let k be a field and Ψ be an ideal of $S := k[[x_1, \ldots, x_n]]$ of valuation m such that*

- *S/Ψ is a finite dimensional k-vector space of dimension $M \geq 1$;*
- *x_n is in Weierstraß position with respect to Ψ;*
- *Either k has characteristic zero or $m+1 \leq \mathrm{char}(k)$.*

*We define $\widetilde{\Psi}$, the **deflated ideal** of Ψ by*

$$\widetilde{\Psi} := \frac{\partial^{m-1}\Psi}{\partial {x_n}^{m-1}};$$

this is an ideal containing Ψ of valuation 1 and the dimension $\widetilde{M}$ of the k-vector space $S/\widetilde{\Psi}$ satisfies the following inequality:

$$1 \leq \widetilde{M} \leq M/m.$$

Proof. Since $\widetilde{\Psi}$ contains Ψ, the quotient $S/\widetilde{\Psi}$ is a finite dimensional k-vector space. Now we order the monomials according to the *anti-graded lexicographic order* defined by $x_1^{\alpha_1}\cdots x_n^{\alpha_n} > x_1^{\beta_1}\cdots x_n^{\beta_n}$ if $\alpha_1+\cdots+\alpha_n < \beta_1+\cdots+\beta_n$ or if $\alpha_1+\cdots+\alpha_n = \beta_1+\cdots+\beta_n$ and $(\alpha_1,\ldots,\alpha_n)$ is greater than $(\beta_1,\ldots,\beta_n)$ for the pure lexicographic order:

$$\begin{array}{ccccccccccc} 1 & > & x_n & > & x_{n-1} & > & \cdots & > & x_1 & & \\ & > & x_n^2 & > & x_nx_{n-1} & > & x_nx_{n-2} & > & \cdots & > & x_nx_1 \\ & & & > & x_{n-1}^2 & > & \cdots & & & & \end{array}$$

This order is compatible with the differentiation with respect to x_n: if both the derivatives of two monomials are not zero, they are in the same order as the monomials.
We denote by $\mathrm{lm}(\Psi)$ (resp. $\mathrm{lm}(\widetilde{\Psi})$) the monomial ideal constituted by the leading monomials of Ψ (resp. $\widetilde{\Psi}$) according to the above order. Note that the cardinal of the complement of $\mathrm{lm}(\Psi)$ (resp. $\mathrm{lm}(\widetilde{\Psi})$) is M (resp. $\widetilde{M}$). Let T be the subset of the complement of $\mathrm{lm}(\Psi)$ composed of the monomials having the power $m-1$ with respect to x_n:

$$T := \{x_1^{\alpha_1}\cdots x_n^{\alpha_n} \notin \mathrm{lm}(\Psi),\ \alpha_n = m-1\}.$$

Since M is at least $m|T|$ (where $|T|$ denotes the cardinal of T), it suffices to prove that $\widetilde{M}$ is at most $|T|$. Let A be a leading monomial $x_1^{\alpha_1}x_2^{\alpha_2}\cdots x_n^{\alpha_n}$ of Ψ such that $\alpha_n = m-1$ then $\frac{\partial^{m-1}A}{\partial {x_n}^{m-1}}$ is not zero (for we have $\mathrm{char}(k) \geq m+1$) and is a leading monomial of $\widetilde{\Psi}$ (Note that $x_1^{\alpha_1}x_2^{\alpha_2}\cdots x_n^{\alpha_n}$ cannot be x_n^{m-1}).
Now using the monomial $A = x_1^{\alpha_1}\cdots x_{n-1}^{\alpha_{n-1}}x_n^m$ which belongs to $\mathrm{lm}(\Psi)$ for any tuple $(\alpha_1,\ldots,\alpha_{n-1})$ by the Weierstraß position property and from the fact that $\mathrm{char}(k) \geq$

$m+1$, we deduce that the monomial $x_1^{\alpha_1}\cdots x_{n-1}^{\alpha_{n-1}}x_n$ is a leading monomial of $\widetilde{\Psi}$. We deduce that the complement of $\mathrm{lm}(\widetilde{\Psi})$ is included in the set

$$\{x_1^{\alpha_1}\cdots x_{n-1}^{\alpha_{n-1}} \quad | \quad x_1^{\alpha_1}\cdots x_{n-1}^{\alpha_{n-1}}x_n^{m-1} \in T\}.$$

We are done. □

We exhibit an example where the Weierstraß condition is a necessary hypothesis: let $n := 2$ and $\Psi := (x_1x_2, x_1^a, x_2^b)$, with $a \geq 3$ and $b \geq 3$. With $m = 2$ and $M = a+b-1$, the ideal Ψ satisfies the hypotheses of the deflation lemma save the Weierstraß position. Its corresponding deflated ideal $\widetilde{\Psi} = \frac{\partial\Psi}{\partial x_2}$ is (x_1, x_2^{b-1}) and has degree $\widetilde{M} := b-1$ so that the conclusion of the lemma holds if and only if $b \leq a+1$.

3 Deflation Sequence

We recall that $F := \{f_1, \ldots, f_s\}$ is a finite subset of $\mathfrak{o}[x_1, \ldots, x_n]$ and x^* is an isolated root of $f_1 = \cdots = f_s = 0$ with multiplicity M. First we construct a sequence of ideals of $\overline{k}[[x_1 - x_1^*, \ldots, x_n - x_n^*]]$ starting from F with decreasing multiplicity. Then we examine the branchings and deduce conditions under which the computations remain valid when replacing $\overline{k}$ by $A = \hat{\mathfrak{o}}[T]/(q(T))$ and x^* by an approximate root. We assume that either $\mathrm{char}(k) = 0$ or $\mathrm{char}(k) \geq M+1$.

3.1 Exact Construction

In this subsection we assume that we are given a root $x^* \in \overline{k}^n$ of $f_1, \ldots, f_s$ isolated for the Zariski topology, and of multiplicity M. We denote by Φ the set F viewed as a subset of $S := \overline{k}[[x_1 - x_1^*, \ldots, x_n - x_n^*]]$. The ideal generated by the elements of Φ in S is not trivial and the quotient ring S/Φ is a finite dimensional $\overline{k}$-vector space of dimension M ($M \geq 1$). We construct a sequence of *deflated ideals*. This sequence starts from Φ, each step is achieved by combining the action of a well chosen differentiation and the elimination of well chosen variables.

By induction we define the sequences of integers R_i and the deflated subsets Φ_i of S, for $i \geq 1$ as follows:

- $R_1 := 1$;
- $\Phi_1 := \Phi$.

At step $i \geq 1$ we know R_i and Φ_i, we now describe how we compute R_{i+1} and Φ_{i+1}. We use the following notations:

- $S_i := \overline{k}[[x_{R_i} - x_{R_i}^*, \ldots, x_n - x_n^*]]$;
- $M_i := \dim_{\overline{k}}(S_i/\Phi_i)$;
- $m_i := \mathrm{val}(\Phi_i)$.

We assume that

(W_i) x_{R_i} is in Weierstraß position with respect to Φ_i.

The ideal Φ_i satisfies the conditions of the deflation lemma (Lemma 4), we deflate it once. We write $\frac{\partial \widetilde{\Phi}_i}{\partial\{x_{R_i},\ldots,x_n\}}$ for the Jacobian matrix of the elements of $\widetilde{\Phi}_i$ with respect to the variables $x_{R_i},\ldots,x_n$.

- $\widetilde{\Phi}_i := \frac{\partial^{m_i-1}}{\partial {x_{R_i}}^{m_i-1}}\Phi_i$;
- $\widetilde{M_i} := \dim(S_i/\widetilde{\Phi}_i)$;
- $r_i := \operatorname{rank}\Big(\frac{\partial \widetilde{\Phi}_i}{\partial\{x_{R_i},\ldots,x_n\}}(x^*_{R_i},\ldots,x^*_n)\Big)$, this is the rank of the set of the gradient vectors at x^* of the elements of $\widetilde{\Phi}_i$.

According to Lemma 4 the following properties hold:

- $1 \leq r_i \leq n - R_i + 1$;
- $1 \leq \widetilde{M_i} \leq M_i/m_i$.

We set $R_{i+1} := R_i + r_i$ and extract a subset Ω_i of cardinal r_i from $\widetilde{\Phi}_i$ such that: the gradient of Ω_i at $(x^*_{R_i},\ldots,x^*_n)$ has rank r_i and, up to a permutation of the variables, there exist power series $y_{R_i},\ldots,y_{R_{i+1}-1}$ in S_{i+1} satisfying $x_j = y_j$ in S_i/Ω_i, for $R_i \leq j \leq R_{i+1}-1$ (thanks to the implicit function theorem). We define the elimination map G_i as follows:

$$\begin{array}{rccl} G_i: & S_i & \to & S_{i+1} \\ & \phi & \mapsto & \phi(y_{R_i},\ldots,y_{R_{i+1}-1},x_{R_{i+1}},\ldots,x_n). \end{array}$$

From a practical point of view the rank computation and the extraction of such a subset Ω_i can be achieved using classical Gaussian elimination. The lucky choice of the maximal ideal $\mathfrak{m}$, involved with this part of the computation, relates to all the equality tests and inversions. We discuss these aspects more in Proposition 14 of §4.3.

Lemma 5 *The following property holds:*

$$\operatorname{val}(G_i(\widetilde{\Phi}_i)) \geq 2.$$

Proof. From Lemma 4, $\widetilde{\Phi}_i$ has valuation 1. Let ϕ be an element of $\widetilde{\Phi}_i$ of valuation 1; since the gradient of ϕ is a linear combination of the gradients of the elements of Ω_i, the valuation of $G_i(\phi)$ is at least 2. □

Corollary 5 $m_i \geq 2$, *for* $i \geq 2$.

Last, we define

$$\Phi_{i+1} := G_i(\widetilde{\Phi}_i).$$

The above construction stops once we have exhausted all the variables, that is when $R_{i+1} = n+1$. We let ν be such that $R_{\nu+1} = n+1$ and call it the **depth** of the deflation. The main output of this process is the sequence $(\Omega_i)_{i=1,\ldots,\nu}$ such that

$$\begin{cases} \Omega_1(x^*_{R_1}, \ldots, x^*_n) = 0, \\ \Omega_2(x^*_{R_2}, \ldots, x^*_n) = 0, \\ \cdots \\ \Omega_\nu(x^*_{R_\nu}, \ldots, x^*_n) = 0, \end{cases}$$

and the Jacobian matrix of the union of the Ω_i is invertible at x^*. Our algorithm presented in §4 consists essentially in applying the classical Newton iterator on the above system. The difficulty is to find an efficient way for evaluating the Ω_is and their gradient vectors in a neighborhood of x^*.

The sequence $m_1, \ldots, m_\nu$ is called the **multiplicity sequence** associated to the deflation. The crucial quantity appearing in the complexity estimate of our algorithm is $m_1 \cdots m_\nu$. Noting that M_{i+1} equals $\widetilde{M_i}$, we deduce the following proposition:

Proposition 6 *For $1 \leq i \leq \nu$ we have the following inequalities:*

$$m_1 \cdots m_{i-1} M_i \leq M, \qquad m_1 \cdots m_\nu \leq M.$$

Concerning the Weierstraß conditions (W_i), we successively apply Lemma 1 to deduce:

Proposition 7 *The linear changes of variables for which not all the (W_i) hold are enclosed in an algebraic hypersurface of $GL_n(k)$.*

From a computational point of view we first perform a generic linear change of coordinates and then apply a deterministic deflation process (performing no random choice). With respect to this generic linear change of coordinates we can speak about a generic multiplicity sequence as the multiplicity sequence found on a Zariski open subset of $GL_n(k)$. We detail this point of view later in §3.5 more precisely.

Block Notations It is natural to introduce the following block notation, for each i, $1 \leq i \leq \nu$:

- $X_i := x_{R_i}, \ldots, x_{R_{i+1}-1}$;
- $Y_i := y_{R_i}, \ldots, y_{R_{i+1}-1}$.

By convention we consider that Y_0 is the empty sequence.

3.2 Example 3 continued

We are coming back to the example of the beginning: $n = 3$, $\mathfrak{o} = \mathbb{Z}$, $k = \mathbb{Q}$, and $F = f_1, f_2, f_3$, where

$$\begin{aligned} f_1 &:= 2\,x_1 + 2\,x_1^2 + 2\,x_2 + 2\,x_2^2 + x_3^2 - 1, \\ f_2 &:= (x_1 + x_2 - x_3 - 1)^3 - x_1^3, \\ f_3 &:= (2\,x_1^3 + 5\,x_2^2 + 10\,x_3 + 5\,x_3^2 + 5)^3 - 1000\,x_1^5. \end{aligned}$$

The root $x^* = (0,0,-1)$ lies in k^3 and has multiplicity 18. We illustrate the construction of the deflation sequence. At x^* we examine the rank of the Jacobian matrix of f_1, f_2, f_3:

$$\frac{df_1}{dx}(x^*) = (2,2,2), \frac{df_2}{dx}(x^*) = (0,0,0), \frac{df_3}{dx}(x^*) = (0,0,0).$$

We find that $m_1 = 1$, $\widetilde{\Phi}_1 = \Phi_1$, x_1 is in Weierstraß position, $r_1 = 1$ and $\Omega_1 = \{f_1\}$. We compute y_1 as the power series solution of $f_1(y_1, x_2, x_3) = 0$ in $k[[x_2, x_3+1]]$ such that $y_1 = 0 + \mathcal{O}(x_2, x_3+1)$:

$$\begin{aligned}
y_1 = {} & (x_3+1) - \frac{3}{2}(x_3+1)^2 + 3\,(x_3+1)^3 - \frac{33}{4}(x_3+1)^4 \\
& + x_2\Big(-1 + 2\,(x_3+1) - 7\,(x_3+1)^2 + 26\,(x_3+1)^3 - \frac{203}{2}(x_3+1)^4\Big) \\
& + x_2^2\Big(-2 + 8\,(x_3+1) - 40\,(x_3+1)^2 + 196\,(x_3+1)^3 - 949\,(x_3+1)^4\Big) \\
& + x_2^3\Big(-4 + 32\,(x_3+1) - 216\,(x_3+1)^2 + 1320\,(x_3+1)^3 - 7610\,(x_3+1)^4\Big) \\
& + x_2^4\Big(-12 + 136\,(x_3+1) - 1148\,(x_3+1)^2 + 8360\,(x_3+1)^3 - 55710\,(x_3+1)^4\Big) \\
& + x_2^5\Big(-40 + 592\,(x_3+1) - 6008\,(x_3+1)^2 + 50736\,(x_3+1)^3 - 383124\,(x_3+1)^4\Big) \\
& + x_2^6\Big(-144 + 2624\,(x_3+1) - 31104\,(x_3+1)^2 + 298592\,(x_3+1)^3 \\
& -2517368\,(x_3+1)^4\Big) + \mathcal{O}(x_2^7, (x_3+1)^5).
\end{aligned}$$

Substituting x_1 by y_1 in F we deduce $\Phi_2 = \{\phi_1, \phi_2, \phi_3\}$ where:

$$\begin{aligned}
\phi_1 = {} & 0 + \mathcal{O}(x_2^7, (x_3+1)^5), \\
\phi_2 = {} & -(x_3+1)^3 + \frac{9}{2}(x_3+1)^4 \\
& + x_2\Big(3\,(x_3+1)^2 - 15\,(x_3+1)^3 + \frac{255}{4}(x_3+1)^4\Big) \\
& + x_2^2\Big(-3\,(x_3+1) + \frac{45}{2}(x_3+1)^2 - 123\,(x_3+1)^3 + \frac{2367}{4}(x_3+1)^4\Big) \\
& + x_2^3\Big(1 - 18\,(x_3+1) + 135\,(x_3+1)^2 - 822\,(x_3+1)^3 + \frac{9357}{2}(x_3+1)^4\Big) \\
& + x_2^4\Big(6 - 84\,(x_3+1) + 708\,(x_3+1)^2 - 5112\,(x_3+1)^3 + 33699\,(x_3+1)^4\Big) \\
& + x_2^5\Big(24 - 360\,(x_3+1) + 3636\,(x_3+1)^2 - 30480\,(x_3+1)^3 + 228324\,(x_3+1)^4\Big) \\
& + x_2^6\Big(84 - 1560\,(x_3+1) + 18468\,(x_3+1)^2 - 176520\,(x_3+1)^3 \\
& +1480290\,(x_3+1)^4\Big) + \mathcal{O}(x_2^7, (x_3+1)^5), \\
\phi_3 = {} & 5000\,x_2(x_3+1)^4 \\
& + x_2^3\Big(-10000\,(x_3+1)^3 + 95375\,(x_3+1)^4\Big) \\
& + x_2^4\Big(10000\,(x_3+1)^2 - 130000\,(x_3+1)^3 + 1031450\,(x_3+1)^4\Big) \\
& + x_2^4\Big(-5000\,(x_3+1) + 107875\,(x_3+1)^2 - 1113950\,(x_3+1)^3 + 8959725\,(x_3+1)^4\Big) \\
& + x_2^5\Big(1000 - 50000\,(x_3+1) + 774250\,(x_3+1)^2 - 8043910\,(x_3+1)^3 + \frac{137720739}{2}(x_3+1)^4\Big) \\
& + x_2^6\Big(10125 - 319550\,(x_3+1) + 4815785\,(x_3+1)^2 - 53037098\,(x_3+1)^3 \\
& + \frac{978677779}{2}(x_3+1)^4\Big) + \mathcal{O}(x_2^7, (x_3+1)^5).
\end{aligned}$$

We see that $m_2 = 3$ and that x_2 is in Weierstraß position. We deduce $\widetilde{\Phi}_2$ at precision $\mathcal{O}(x_2^5, (x_3+1)^5)$, it contains 9 elements:

$$\widetilde{\Phi}_2 = \left\{ \phi_1, \phi_2, \phi_3, \frac{\partial \phi_1}{\partial x_2}, \frac{\partial \phi_2}{\partial x_2}, \frac{\partial \phi_3}{\partial x_2}, \frac{\partial^2 \phi_1}{\partial {x_2}^2}, \frac{\partial^2 \phi_2}{\partial {x_2}^2}, \frac{\partial^2 \phi_3}{\partial {x_2}^2} \right\}.$$

The gradients at point $(0, -1)$ of these elements are respectively

$$(0,0),\ (0,0),\ (0,0),\ (0,0),\ (0,0),\ (0,0),\ (0,0),\ (6,-6),\ (0,0).$$

Therefore we get $r_2 = 1$, $\Omega_2 = \{\frac{\partial^2 \phi_2}{\partial {x_2}^2}\}$ and compute y_2 as the power series solution of $\frac{\partial^2 \phi_2}{\partial {x_2}^2}(y_2, x_3) = 0$ in $k[[x_3+1]]$ such that $y_2 = 0 + \mathcal{O}(x_3+1)$:

$$y_2 = (x_3+1) - \frac{3}{2}(x_3+1)^2 + 3\,(x_3+1)^3 + \frac{9}{4}(x_3+1)^4 + \mathcal{O}((x_3+1)^5).$$

Substituting x_2 by y_2 in $\widetilde{\Phi}_2$ we find $\Phi_3 = \{\varphi_1, \ldots, \varphi_9\}$ where

$$\begin{array}{rcl} \varphi_l & = & \phi_l(y_2, x_3), \text{ for } l = 1,2,3, \\ \varphi_l & = & \frac{\partial \phi_l}{\partial dx_2}(y_2, x_3), \text{ for } l = 4,5,6, \\ \varphi_l & = & \frac{\partial^2 \phi_l}{\partial {dx_2}^2}(y_2, x_3), \text{ for } l = 7,8,9. \end{array}$$

We compute $\varphi_l = 0 + \mathcal{O}((x_3+1)^5)$ for $l = 1, \ldots, 8$ and $\varphi_9 = 9000(x_3+1)^4 + \mathcal{O}((x_3+1)^5)$. We deduce that $m_3 = 4$ and $\widetilde{\Phi}_3$:

$$\widetilde{\Phi}_3 = \left\{ \frac{\partial^j \varphi_l}{\partial {dx_3}^j},\ l = 1, \ldots, 9,\ j = 0, \ldots, 3 \right\}.$$

All the elements of $\widetilde{\Phi}_3$ equal $0 + \mathcal{O}((x_3+1)^2)$ except $\frac{\partial^3 \varphi_9}{\partial {dx_3}^3} = 216000(x_3+1) + \mathcal{O}((x_3+1)^2)$. It is obvious that x_3 is in Weierstraß position. We deduce $r_3 = 1$ and finally the depth $\nu = 3$. As expected, the product $m_1 m_2 m_3 = 12$ is bounded by the multiplicity $M = 18$.

3.3 Smoothness Hypotheses

We revisit the construction of the deflation sequence presented above. We gather conditions on $\mathfrak{m}$ under which the Y_i are well-defined over A. Theses conditions are weak in the sense that they do not allow the computations of the ranks and the selections of Ω_i over A instead of $k(u)$. Stronger conditions are established in Proposition 14 of §4.3 during the presentation of the algorithm.

Let $A_i := A[[x_{R_i} - x^*_{R_i}, \ldots, x_n - x^*_n]]$, for $i = 1, \ldots, \nu$. We assume by induction on i that Y_i is well-defined in A_{i+1}. Since Y_0 is the empty sequence, this is trivially true for $i = 0$. Let us assume that this holds for $i \geq 1$. To go from step $i-1$ to i we assume that the Jacobian matrix of Ω_i with respect to the variables in X_i evaluated at x^* is invertible in $A/\mathfrak{m}A$:

(H_{Ω_i}) The determinant of $\frac{\partial \Omega_i}{\partial X_i}(x^*_{R_i}, \ldots, x^*_n)$ is a unit of $A/\mathfrak{m}A$.

In particular, from the implicit function theorem this condition implies that Y_i belongs to A_{i+1}. We denote by (H_Ω) the set of hypotheses (H_{Ω_i}) for $i = 1, \ldots, \nu$. We recall from §1.1 that p is the canonical projection from $k[T]$ to $k[u]$ and p^{-1} the linear map such that $p \circ p^{-1} = \mathrm{Id}$ and $p^{-1}(z)$ has degree strictly less than $\deg(Q)$. We define $a_i \in k$ as the resultant of $p^{-1}(\det(\frac{\partial \Omega_i}{\partial X_i}(x^*_{R_i}, \ldots, x^*_n)))$ with Q. We can write $a_i = \rho_{\Omega_i}/\overline{\rho}_{\Omega_i}$ with $\rho_{\Omega_i} \in o$, and $\overline{\rho}_{\Omega_i} \notin \mathfrak{m}$.

Proposition 8 *Let $\rho_\Omega = \rho_{\Omega_1} \cdots \rho_{\Omega_\nu}$. For any maximal ideal $\mathfrak{m}$ of $\mathfrak{o}$ that does not contain ρ_Ω, hypothesis (H_Ω) is satisfied.*

Indeed (H_Ω) generalizes (H_J) of §1.1 in the following sense:

Proposition 9 *Under hypothesis (H_Ω). Let z be an approximation of x^* (that is $z - x^* \in (\mathfrak{m}A)^n$) such that*

$$\begin{cases} \Omega_1(z_{R_1}, \ldots, z_n) = 0, \\ \Omega_2(z_{R_2}, \ldots, z_n) = 0, \\ \ldots \\ \Omega_\nu(z_{R_\nu}, \ldots, z_n) = 0, \end{cases}$$

then $z = x^*$.

Proof. The proof is immediate since the Jacobian matrix of the above system is invertible at x^* modulo $\mathfrak{m}A$. □

In the example of §3.2 the computation of the deflation sequence can be performed modulo any prime number p different from 2, 3 and 5: we need to invert 2, 6 and $216000 = 2^6 3^3 5^3$.

3.4 Nested Coordinates

The idea behind our lifting algorithm is to use the classical Newton iterator on the system of equations of Proposition 9. Hence, we only need to compute values and gradients of the Ω_i at various points in a neighborhood of x^* efficiently. Computing these values straightforwardly as in §3.2 is not the best way: the requested precision with respect to the variable x_2 is $m_2 + m_3 = 7$ whereas the method we are to present requires precision $\nabla_{\{1,2,3\}}(x_1^{m_1}, x_2^{m_2}, x_3^{m_3})$ only. Our method relies on what we call the *nested coordinates* associated to the deflation sequence.

We introduce a new set of variables $dx_1, \ldots, dx_n$ with its associated blocks dX_i (as in §3.1). For $i = 1, \ldots, \nu$:

- $dX_i := dx_{R_i}, \ldots, dx_{R_{i+1}-1}$.
- The associated power series rings over $\overline{k}$: $dS_i := \overline{k}[[dx_{R_i}, \ldots, dx_n]]$.
- The associated power series rings over A: $dA_i := A[[dx_{R_i}, \ldots, dx_n]]$.

We also introduce the function $\mathbf{coeff_i}(f, dx^\alpha)$ returning the coefficient of f with respect to the monomial dx^α, seen as a power series in the variables $dX_1, \ldots, dX_i$ only.
The definition of Φ_{i+1} of §3.1 yields the following recursive formula:

$$\begin{aligned}
\Phi_{i+1} &= G_i(\widetilde{\Phi}_i) \\
&= \frac{\partial^{m_i-1}\Phi_i}{\partial {x_{R_i}}^{m_i-1}}(Y_i, x_{R_{i+1}}, \ldots, x_n) \\
&= \{\alpha_i!\,\mathrm{coeff}_i(\phi(Y_i + dX_i, x_{R_{i+1}}, \ldots, x_n), dx_{R_i}^{\alpha_i}), \\
&\quad \phi \in \Phi_i,\ 0 \le \alpha_i \le m_i - 1\}.
\end{aligned}$$

Solving this recurrence leads to a fast evaluation scheme for the Φ_is. For this purpose and each $i = 1, \ldots, \nu$, we introduce the ith **nested coordinates** Y^i as the sequence $(Y_1^i, \ldots, Y_i^i)$ with entries in

$$\overline{k}[[dX_1, \ldots, dX_i]][[x_{R_{i+1}} - x^*_{R_{i+1}}, \ldots, x_n - x^*_n]].$$

The entries of Y^i are recursively defined from the last one to the first one:

$$\begin{aligned}
Y_i^i &:= Y_i(x_{R_{i+1}}, \ldots, x_n), \\
Y_{i-1}^i &:= Y_{i-1}(Y_i^i + dX_i, x_{R_{i+1}}, \ldots, x_n), \\
&\cdots \\
Y_1^i &:= Y_1(Y_2^i + dX_2, \ldots, Y_i^i + dX_i, x_{R_{i+1}}, \ldots, x_n).
\end{aligned}$$

By convention we set $Y^0(x_1, \ldots, x_n)$ to be the empty sequence. Under hypothesis (H_Ω), Y^i is well-defined over A instead of $\overline{k}$. Next proposition describes how nested coordinates are related. Note that we use the comma operator for the concatenation of sequences.

Proposition 10 *Let V denote $x_{R_{i+2}}, \ldots, x_n$. The nested coordinates are related to each other by the following formula:*

$$Y^{i+1} = Y^i(Y_{i+1}(V) + dX_{i+1}, V), Y_{i+1}(V),$$

for $i = 0, \ldots, \nu - 1$.

The deflated sets Φ_i can be computed by evaluating the input set F at the nested coordinates.

Proposition 11 *For $i = 0, \ldots, \nu - 1$, the $(i+1)$st deflated set Φ_{i+1} is given by:*

$$\begin{aligned}
\Phi_{i+1} = \Big\{ &\alpha_1! \cdots \alpha_i!\,\mathrm{coeff}_i\Big(\phi(Y_1^i + dX_1, \ldots, Y_i^i + dX_i, x_{R_{i+1}}, \ldots, x_n), \\
&dx_{R_1}^{\alpha_1} \cdots dx_{R_i}^{\alpha_i}\Big),\quad \phi \in F,\ 0 \le \alpha_j \le m_j - 1,\ 1 \le j \le i\Big\}.
\end{aligned}$$

Proof. The proof is done by induction on i. If $i = 0$, we have $\Phi_1 = F$ and Y^0 is the empty sequence, the formula is true. Assume that the formula holds for $i \geq 0$. First we have:

$$\begin{aligned}\Phi_{i+2} &= \frac{\partial^{m_{i+1}-1}\Phi_{i+1}}{\partial {x_{R_{i+1}}}^{m_{i+1}-1}}(Y_{i+1}, x_{R_{i+2}}, \ldots, x_n) \\ &= \Big\{\alpha_{i+1}!\,\mathrm{coeff}_{i+1}(\phi(Y_{i+1} + dX_{i+1}, x_{R_{i+2}}, \ldots, x_n), dx_{R_{i+1}}^{\alpha_{i+1}}), \\ &\quad \phi \in \Phi_{i+1},\ 0 \leq \alpha_{i+1} \leq m_{i+1} - 1\Big\}.\end{aligned}$$

Using the induction hypothesis giving Φ_{i+1} and letting

$$V_i := Y_{i+1} + dX_{i+1}, x_{R_{i+2}}, \ldots, x_n,$$

we obtain:

$$\begin{aligned}\Phi_{i+2} &= \Big\{\alpha_1! \cdots \alpha_{i+1}!\,\mathrm{coeff}_{i+1}\Big(\phi(Y_1^i(V_i) + dX_1, \ldots, Y_i^i(V_i) + dX_i, V_i), \\ &\quad dx_{R_1}^{\alpha_1} \cdots dx_{R_{i+1}}^{\alpha_{i+1}}\Big),\ \phi \in F,\ 0 \leq \alpha_j \leq m_j - 1,\ 1 \leq j \leq i+1\Big\}.\end{aligned}$$

Using Proposition 10 we recognize the desired formula for $i + 1$. □

At last, we show in §4 how to compute Y^ν efficiently:

$$\begin{aligned}Y_\nu^\nu &= Y_\nu(), \\ Y_{\nu-1}^\nu &= Y_{\nu-1}(Y_\nu^\nu + dX_\nu), \\ &\cdots \\ Y_1^\nu &= Y_1(Y_2^\nu + dX_2, \ldots, Y_\nu^\nu + dX_\nu).\end{aligned}$$

But before this, it remains to make the deflation process deterministic with respect to the choices of the sets Ω_i and to present the complexity model we consider.

Example 3 (continued from §3.2) Here are the nested coordinates Y^3:

$$\begin{aligned}Y_3^3 &= -1, \\ Y_2^3 &= y_2(Y_3^3 + dx_3) \\ &= dx_3 - \frac{3}{2}\,dx_3^2 + 3\,dx_3^3 + \frac{9}{4}\,dx_3^4 + \mathcal{O}(dx_3^5), \\ Y_1^3 &= y_1(Y_2^3 + dx_2, Y_3^3 + dx_3) \\ &= -\frac{21}{2}\,{dx_3}^4 + dx_2\Big(-1 - 2\,dx_3 + 3\,dx_3^2 - 6\,dx_3^3 - \frac{51}{2}\,dx_3^4\Big) \\ &\quad + dx_2^2\Big(-2 - 4\,dx_3 + 2\,dx_3^2 - 126\,dx_3^4\Big) \\ &\quad + dx_2^3\Big(-4 - 16\,dx_3 + 8\,dx_3^3 + 18542\,dx_3^4\Big) \\ &\quad + dx_2^4\Big(-12 - 64\,dx_3 - 48\,dx_3^2 + 19120\,dx_3^3 - 351000\,dx_3^4\Big) \\ &\quad + dx_2^5\Big(-40 - 272\,dx_3 + 11032\,dx_3^2 - 162096\,dx_3^3 + 1733652\,dx_3^4\Big) \\ &\quad + dx_2^6\Big(-144 + 2624\,dx_3 - 31104\,dx_3^2 + 298592\,dx_3^3 - 2517368\,dx_3^4\Big) \\ &\quad + \mathcal{O}(dx_2^7, dx_3^5).\end{aligned}$$

3.5 Tracing the Deflation

During the deflation process presented in §3 the choice of the Ω_is are not really specified: different choices may be possible for the Weierstraß positions and the variables with respect to which the implicit function theorem is applied. We adopt the following point of view: the deflation process is seen as a deterministic procedure. In particular the variables on which the implicit function theorem is applied are not chosen at random. But the deflation process is parametrized by the change of the coordinates. From this point of view, Proposition 7 tells us that the deflation process works well for almost all choices of this parameter.

In order to explain how we track the deflation process in practice, we introduce notations. For $i = 1, \ldots, \nu$, we let $\boldsymbol{\pi_i}$ denote the monomial ideal

$$\pi_i := (dx_{R_i}^{m_i}, dx_{R_i+1}, \ldots, dx_{R_{i+1}-1})$$

and $\boldsymbol{\pi_{i,j}}$ denote the monomial ideal $\pi_i + \cdots + \pi_j$, for $i \leq j$. We use these ideals over various base rings. We shall specify the considered base ring in each case.

From Proposition 11 it follows that each element of Φ_{i+1} is determined by an element of F and a monomial dx^α in the support of $\pi_{1,i}$, where $\pi_{1,i}$ is seen as an ideal in the variables $dX_1, \ldots, dX_i$. In the same way, to each element of $\widetilde{\Phi}_{i+1}$ corresponds a triple $\tau := (l, dx^\alpha, \mu)$ in $\{1, \ldots, s\} \times \mathrm{supp}(\pi_{1,i}) \times \{0, \ldots, m_{i+1} - 1\}$: this element of $\widetilde{\Phi}_{i+1}$ is the μth derivative with respect to $dx_{R_{i+1}}$ of the element of Φ_{i+1} corresponding to f_l and dx^α.

The **trace** of the deflation is a sequence $(\mathcal{T}_i)_{1 \leq i \leq \nu}$; each $\mathcal{T}_i$ is itself a sequence $\tau_1, \ldots, \tau_{r_i}$ of r_i elements of $\{1, \ldots, s\} \times \mathrm{supp}(\pi_{1,i-1}) \times \{0, \ldots, m_i - 1\}$. Each $\tau_j = (l, dx^\alpha, \mu)$ is bijectively related to an element of Ω_i. That is: $\Omega_i = \{\omega_{\tau_j},\ \tau_j \in \mathcal{T}_i\}$, where

$$\begin{aligned} \gamma_\tau &= f_l(Y_1^{i-1} + dX_1, \ldots, Y_{i-1}^{i-1} + dX_{i-1}, x_{R_i}, \ldots, x_n) \text{ and} \\ \omega_{\tau_j} &= \alpha_1! \cdots \alpha_{i-1}! \, \frac{\partial^\mu}{\partial {dx_{R_i}}^\mu} \mathrm{coeff}_{i-1}(\gamma_\tau, dx^\alpha). \end{aligned} \tag{4}$$

The smoothness hypothesis (H_Ω) is equivalent to the fact that the trace of the deflation sequence over $\mathfrak{o}/\mathfrak{m}$ is the same as over $\overline{k}$.

It is important to notice that there exists a Zariski open subset of $GL_n(k)$ for the change of coordinates on which the trace is constant: we call this constant the **generic trace**. From now on we say that the change of coordinates taken as parameter of the deflation process is **generic enough** if the Weierstraß positions are satisfied and if the trace of the deflation process equals the generic trace.

The generic trace depends on the deflation algorithm used: two different deflation programs may lead to distinct generic traces according to their choices of the subsets Ω_i. Therefore our lifting process has two stages: first we must compute a generic trace, then we can apply the classical Newton iterator on the corresponding system of Ω_i. The smoothness conditions (H_Ω) restricting the choices of $\mathfrak{m}$ ensure that the second stage works fine but not necessarily the first one. We come back to this point in §4.3.

Example 3 (continued from §3.2) The trace is $\mathcal{T}_1 = (1,1,0)$, $\mathcal{T}_2 = (2, dx_1^0, 2)$, $\mathcal{T}_3 = (3, dx_1^0 dx_2^2, 3)$. For instance the element of Ω_3 is obtained from the evaluation of f_3 this way:

$$\begin{aligned}
\Omega_3(Y_3^3 + dX_3) &= \left\{ 2\frac{\partial^3}{\partial {dx_3}^3}\mathrm{coeff}_2(f_3(Y_1^3 + dX_1, Y_2^3 + dX_2, Y_3^3 + dX_3), dx_1^0 dx_2^2) \right\} \\
&= \left\{ 2\frac{\partial^3}{\partial {dx_3}^3}\mathrm{coeff}_2(4500\, dx_2^2 dx_3^4 + \mathcal{O}(dx_2^3, dx_3^5), dx_1^0 dx_2^2) \right\} \\
&= \left\{ 2\frac{\partial^3}{\partial {dx_3}^3} 4500\, dx_3^4 + \mathcal{O}(dx_3^5) \right\} \\
&= \left\{ 216000\, dx_3 + \mathcal{O}(dx_3^2) \right\}.
\end{aligned}$$

4 Algorithm

We are now ready to present our algorithm. It is based on the following idea: we compute the nested coordinates Y^ν associated to the input system F at the root x^* in A modulo $\mathfrak{m}A$ first and then lift these coordinates modulo $(\mathfrak{m}A)^\kappa$ for arbitrary integer κ. During the first stage we compute a generic trace as defined in §3.5, $\mathfrak{m}$ must be lucky and the coordinates generic to ensure the correctness of the answer. Once a generic trace is found together with the nested coordinates modulo $\mathfrak{m}A$ then we are sure that the lifting process works fine. We first detail the complexity model we use and the costs of each elementary operation.

4.1 Complexity Model

Two data structures are used by the algorithm. First the input polynomials $f_1, \ldots, f_s \in \mathfrak{o}[x_1, \ldots, x_n]$ are given by a **straight-line program** [Str72, Gat86, Sto89, Hei89] of length L containing neither test nor branching: for any ring R, for any partially defined ring morphism c from $\mathfrak{o}$ to R and any point $a := (a_1, \ldots, a_n)$ in R^n we can compute the values $f_1(a), \ldots, f_s(a)$ performing L binary operations in R (additions, subtractions and multiplications) and at most L calls to c. Therefore each evaluation costs at most L times the maximum of the cost of an elementary binary arithmetic operation in R or call to c.

The second data structure we use is for multivariate power series. For any ring R and any zero-dimensional monomial ideal π of $S := R[[x_1, \ldots, x_n]]$, we need to compute in S/π. We count the number of arithmetic operations performed in R ($+$, $\times$, $=$) but also the arithmetic operations with the exponents of the monomials.

Proposition 12 *For a given π, we can construct S/π in*

$$\mathcal{O}\Big(n^2 \deg(\pi)^2 \log(\deg(\pi))\Big).$$

Then we have the following complexity estimates.

- *The construction of an element given by a list of l terms (couple of coefficient and exponent) costs* $\mathcal{O}\Big(nl\log(\deg(\pi))\Big)$.
- *The extraction of a coefficient costs* $\mathcal{O}\Big(n\log(\deg(\pi))\Big)$.
- *The cost of each elementary binary arithmetic operations* ($+$, $-$, $\times$, $=$) *in* S/π *is in* $\mathcal{O}\Big(\deg(\pi)^2\Big)$.
- *The inverse costs* $\mathcal{O}\Big(\deg(\pi)^2\log(\deg(\pi))\Big)$.
- *For any* i, $1 \leq i \leq n$, *and any* $\psi \in S/\pi$, $\frac{\partial\psi}{\partial x_i}$ *costs* $\mathcal{O}\Big(\deg(\pi)\Big)$.
- *For any* $x^\alpha \notin \pi$ *and any* $\psi \in S/\pi$, $\frac{\partial^{|\alpha|}\psi}{\partial x^\alpha}$ *costs* $\mathcal{O}\Big(\deg(\pi)^2\Big)$.
- *For any* $x^\alpha \in R[[x_1,\ldots,x_i]]$ *with* $x^\alpha \notin \pi$ *and any* $\psi \in S/\pi$, *the extraction of the coefficient of the monomial* x^α *of* ψ *seen as an element of* $R[[x_{i+1},\ldots,x_n]][[x_1,\ldots,x_i]]$ *costs* $\mathcal{O}\Big(n\deg(\pi)\log(\deg(\pi))\Big)$.

Proof. We use a dense representation and naive algorithms for the arithmetic operations. The dense representation is implemented as follows: first we choose a total order on the monomials such that each comparison between two monomials of $\mathrm{supp}(\pi)$ costs $\mathcal{O}(n)$ (for instance we can take the lexicographical order). Then we sort the support of π according to this order, we obtain an array E representing a map from the integer range $N := [1,\ldots,\deg(\pi)]$ to $\mathrm{supp}(\pi)$. Each element of S/π is stored in an array of size $\deg(\pi)$: the ith entry is its coefficient with respect to the monomial $E(i)$. During the initialization of S/π we also precompute the multiplication table of the monomials of the support of π: this is a two dimensional array $F : N \times N \to N$, such that $F(i,j)$ is zero if $x^\beta := E(i)E(j)$ is in π and $E^{-1}(x^\beta)$ otherwise. Let us detail the costs of these precomputations:

- The construction of E costs $\mathcal{O}(n\deg(\pi)\log(\deg(\pi)))$, using a classical fast sorting algorithm.
- Each call to E has cost $\mathcal{O}(1)$.
- Each call to E^{-1} has cost $\mathcal{O}(n\log(\deg(\pi)))$, by dichotomic search.
- F can be built with cost $\mathcal{O}(n\deg(\pi)^2\log(\deg(\pi)))$, performing one call to E^{-1} for each possible product of the monomials of the support.
- Each call to F costs $\mathcal{O}(1)$.

Finally this part of initialization of S/π requires

$$\mathcal{O}\Big(n\deg(\pi)^2\log(\deg(\pi))\Big)$$

operations. Let us now specify the costs of the arithmetic operations in S/π:

- The construction of an element given by a list of l terms requires l calls to E^{-1}.
- The function coeff just performs one call to E^{-1}.
- Binary additions (resp. subtractions) are done in $\deg(\pi)$ binary additions (resp. subtractions) in R.
- A binary multiplication requires at most $\deg(\pi)^2$ binary additions and multiplications in R and also $\deg(\pi)^2$ calls to the multiplication table F.

If $\psi \in S/\pi$ is a unit, we compute its inverse thanks to the classical iterator $N(z) = z + z(1 - z\psi)$, starting from the inverse of the constant coefficient of ψ. The number of iterations is in $\mathcal{O}(\log(\deg(\pi)))$.

In order to speed up the derivations we compute look-up tables $D_1, \ldots, D_n$. For $i = 1, \ldots, n$:

$$\begin{array}{rccl} D_i: & N & \to & N \times \mathbb{N} \\ & l & \mapsto & \left(E^{-1}(\frac{\partial x^\alpha}{\partial x_i}), \alpha_i\right), \text{ where } x^\alpha = E(l). \end{array}$$

The construction of each D_i requires $\mathcal{O}(n\deg(\pi)\log(\deg(\pi)))$. The cost of the construction of all the D_i dominates the cost of the initialization of S/π.

Let $\psi \in S/\pi$. In order to compute $\frac{\partial \psi}{\partial x_i}$ we differentiate ψ term by term: for each $l \in N$ we look up $(m, \lambda) = D_i(l)$ and set the mth entry of $\frac{\partial \psi}{\partial x_i}$ to λ times the lth entry of ψ. The total cost is in $\mathcal{O}(\deg(\pi))$.

We deduce that for any $x^\alpha \notin \pi$ the cost of $\frac{\partial^{|\alpha|}\psi}{\partial x^\alpha}$ is in $\mathcal{O}(\deg(\pi)^2)$, for we have $|\alpha| \leq \deg(\pi)$.

As for the last statement of the proposition it suffices to select in ψ the monomials matching x^α and construct the answer in S/π. This is done within $\mathcal{O}(n\deg(\pi)\log(\deg(\pi)))$.

□

For the sake of clarity, we denote by $\mathcal{C}_S(\pi)$ the following expression

$$\mathcal{C}_S(\pi) := \deg(\pi)^2 \log(\deg(\pi)).$$

It denotes the maximum of the numbers of binary additions, subtractions, multiplications and inversion in R needed to compute a binary addition, subtraction, multiplication or inverse in S/π. Unfortunately, we do not know better complexity result in general. We only know an optimal algorithm (up to logarithmic factors and when R is a field of characteristic zero) when π is a power of $(x_1, \ldots, x_n)$ [LS01].

Concerning the complexity of linear algebra, Ω is a constant such that matrix determinant and adjoint computation require $\mathcal{O}(n^\Omega)$ arithmetic operations in the base ring. Therefore Ω is less than 4 [Abd97, Ber84, Csa76, Lev40]. In order to simplify the presentation we take $\Omega \geq 3$. Using a better Ω would not improve our complexity estimates.

4.2 Incremental Lifting Step

The method we propose to compute the nested coordinates follows an approximation and correction scheme. The function `IncrementalLift` presented below is the core of our algorithm: for a given i it lifts an approximate value of Y^i. The input and output precisions are governed by integers $\kappa \geq 1$ and $\lambda \geq 1$. We define the ideals ζ_1 and ζ_2:

$$\begin{aligned} \zeta_1 &:= (dx_{R_{i+1}}^{\lambda}, dx_{R_{i+1}+1}, \ldots, dx_n) + \mathfrak{m}^{\kappa}, \\ \zeta_2 &:= (dx_{R_{i+1}}^{2\lambda}, dx_{R_{i+1}+1}, \ldots, dx_n) + \mathfrak{m}^{2\kappa}. \end{aligned}$$

We lighten the notations: $\nabla \pi_{l_1,l_2}$ stands for $\nabla_{\{R_{l_1},\ldots,R_{l_2+1}-1\}}\pi_{l_1,l_2}$ and we write $\nabla\zeta_i$ instead of $\nabla_{\{R_{i+1},\ldots,n\}}\zeta_i$, for $i = 1, 2$. Quantities appearing in the core of the function that are not input nor local variables refer to global variables. As discussed in §3.5 we assume that the coordinates are generic enough.

`IncrementalLift`

Input:

- $W_{i+1} := z_{R_{i+1}}, \ldots, z_n$ in a neighborhood of $x^*_{R_{i+1}}, \ldots, x^*_n$, let $dW_{i+1} := z_{R_{i+1}} + dx_{R_{i+1}}, \ldots, z_n + dx_n$.
- Two integers $\kappa \geq 1$ and $\lambda \geq 1$.
- $\mathcal{T}_1, \ldots, \mathcal{T}_i$, the first i elements of the generic trace.
- $Z_1 := Y_1^i(dW_{i+1}), \ldots, Z_i := Y_i^i(dW_{i+1})$ at precision $\pi_{1,i} + \nabla\zeta_1$.

Output:

- $Y^i(dW_{i+1})$ at precision $\pi_{1,i} + \nabla\zeta_2 + \mathfrak{m}^{\kappa}\zeta_1$.

This subroutine will be used in two situations: either we want to improve the precision λ with respect to the variables $dx_{R_{i+1}}$ and $\kappa = 1$ holds (§4.3) or we want to improve the precision κ with respect to $\mathfrak{m}$ and $i = \nu$ holds (§4.6). Our presentation combines both these situations.

Algorithm:

We improve the precision of the nested coordinates in sequence. At step l we know an improved value of Y^l. More precisely:

(L_l) At step l we know $Y^l(Z_{l+1}+dX_{l+1}, \ldots, Z_i+dX_i, dW_{i+1})$ at precision $\pi_{1,l}+\sigma_{l+1}$, where

$$\sigma_{l+1} := \nabla\pi_{l+1,i} + \pi_{l+1,i}\zeta_1 + \nabla\zeta_2 + \mathfrak{m}^{\kappa}\zeta_1.$$

The computation goes from (L_0) to (L_i). Along the presentation of the algorithm we introduce the quantities θ, θ^{Ω}, θ_{Ω} and θ^{∇}. They are detailed in Lemma 6 below.

The initialization is trivial since Y^0 is the empty sequence. At the end (when $l = i$) we get $Y^i(dW_{i+1})$ at precision $\pi_{1,i} + \sigma_{i+1} = \pi_{1,i} + \nabla\zeta_2 + \mathfrak{m}^{\kappa}\zeta_1$.

Going from (L_l) to (L_{l+1}) is performed by the following computations:

1. First we let $dW_{l+2} := Z_{l+2} + dX_{l+2}, \ldots, Z_i + dX_i, dW_{i+1}$ and we compute for each $j = 1, \ldots, s$:

$$\gamma_j := f_j\Big(Y_1^l(Z_{l+1} + dX_{l+1}, dW_{l+2}) + dX_1, \ldots,$$
$$Y_l^l(Z_{l+1} + dX_{l+1}, dW_{l+2}) + dX_l, Z_{l+1} + dX_{l+1}, dW_{l+2}\Big).$$

2. Since $\mathrm{coeff}_l(\gamma_j, dx^\alpha)$ is known at precision σ_{l+1}, then for each $\tau := (j, dx^\alpha, \mu) \in \mathcal{T}_{l+1}$ and according to formula (4), we deduce the value

$$\omega_\tau(Z_{l+1} + dX_{l+1}, dW_{l+2}) := \alpha_1! \cdots \alpha_l! \, \frac{\partial^\mu \mathrm{coeff}_l(\gamma_j, dx^\alpha)}{\partial dx_{R_{l+1}}{}^\mu} \tag{5}$$

at precision at least

$$\theta := \frac{\partial^{m_{l+1}-1}\sigma_{l+1}}{\partial dx_{R_{l+1}}{}^{m_{l+1}-1}}.$$

3. We deduce $\Omega_{l+1}(Z_{l+1}, dW_{l+2}) = (\omega_\tau(Z_{l+1}, dW_{l+2}),\ \tau \in \mathcal{T}_{l+1})$ at precision at least $\theta^\Omega := \theta \cap dA_{l+2}$ and valuation at least $\theta_\Omega := \pi_{l+2,i} + \nabla\zeta_1$ (by hypothesis).

4. We also deduce

$$\frac{\partial \omega_\tau}{\partial dX_{l+1}}(Z_{l+1}, dW_{l+2}) := \left(\frac{\partial \omega_\tau}{\partial dx_j}(Z_{l+1}, dW_{l+2}),\ j = R_{l+1}, \ldots, R_{l+2} - 1\right)$$

at precision θ^∇.

5. We are looking for a value dZ_{l+1} of dX_{l+1} such that $Z_{l+1} + dZ_{l+1}$ is $Y_{l+1}(dW_{l+2})$ at precision σ_{l+2}. This value satisfies the following equation obtained from the first order Taylor expansion of Ω_{l+1} at (Z_{l+1}, dW_{l+2}):

$$\Omega_{l+1}(Z_{l+1} + dZ_{l+1}, dW_{l+2}) \in \sigma_{l+2}$$
$$\Longleftrightarrow$$
$$\Omega_{l+1}(Z_{l+1}, dW_{l+2}) + \frac{\partial \Omega_{l+1}}{\partial dX_{l+1}}(Z_{l+1}, dW_{l+2}) dZ_{l+1} \in \sigma_{l+2}.$$

We prove in Lemma 6 below that $\theta_\Omega^2 + \theta^\nabla \theta_\Omega + \theta^\Omega \subseteq \sigma_{l+2}$. Therefore dZ_{l+1} exists, is unique at precision σ_{l+2} and is given by:

$$dZ_{l+1} := -\left(\frac{\partial \Omega_{l+1}}{\partial dX_{l+1}}(Z_{l+1}, dW_{l+2})\right)^{-1} \Omega_{l+1}(Z_{l+1}, dW_{l+2}).$$

6. From the value $Y_{l+1}(dW_{l+2}) = Z_{l+1} + dZ_{l+1}$ at precision σ_{l+2} and using Proposition 10, it remains to compute $Y^l(Y_{l+1}(dW_{l+2}) + dX_{l+1}, dW_{l+2})$ at precision

$\pi_{1,l+1} + \sigma_{l+2}$ in order to reach (L_{l+1}). For this purpose we use the following first order Taylor expansion formula:

$$\begin{aligned} Y^l(Z_{l+1} + dX_{l+1} + dZ_{l+1}, dW_{l+2}) &= Y^l(Z_{l+1} + dX_{l+1}, dW_{l+2}) \\ &+ \frac{\partial Y^l(Z_{l+1} + dX_{l+1}, dW_{l+2})}{\partial dX_{l+1}} dZ_{l+1} + \mathcal{O}((dZ_{l+1})^2) \end{aligned} \tag{6}$$

We know dZ_{l+1} at precision σ_{l+2} and each of its entries is in θ_Ω. The derivative $\frac{\partial Y^l(Z_{l+1}+dX_{l+1},dW_{l+2})}{\partial dx_j}$ for $dx_j \in dX_{l+1}$ is known at precision $\theta' := \frac{\partial(\pi_{1,l}+\sigma_{l+1})}{\partial dx_j}$. From Corollary 3 and Proposition 4 we deduce that

$$\begin{aligned} \theta' &\subseteq \pi_{1,l} + \tfrac{\partial \nabla \pi_{l+1,i}}{\partial dx_j} + \zeta_1 \\ &\subseteq \pi_{1,l} + \nabla_{\{R_{l+1},\ldots,R_{i+1}-1\}} \tfrac{\partial \pi_{l+1,i}}{\partial dx_j} + \zeta_1 \\ &\subseteq \pi_{1,l} + \nabla_{\{j\}} \tfrac{\partial \pi_{l+1,i}}{\partial dx_j} + \zeta_1 \\ &\subseteq \pi_{1,l} + \pi_{l+1,i} + \zeta_1. \end{aligned}$$

Therefore the precision of Y^l we reached is in

$$\pi_{1,l+1} + (\pi_{l+2,i} + \zeta_1)(\pi_{l+2,i} + \nabla\zeta_1),$$

which is itself included in $\pi_{1,l+1} + \sigma_{l+2}$, using again Lemma 6 below.

Lemma 6 *The following inclusions hold* $\theta^\Omega \subseteq \sigma_{l+2}$, $\theta^\nabla \theta_\Omega \subseteq \sigma_{l+2}$ *and* $\theta_\Omega^2 \subseteq \sigma_{l+2}$.

Proof. First we prove that $\theta^\Omega \subseteq \sigma_{l+2}$. Combining Corollary 3 and Lemma 2 we get that:

$$\frac{\partial^{m_{l+1}-1} \nabla \pi_{l+1,i}}{\partial dx_{R_{l+1}}{}^{m_{l+1}-1}} = \nabla_{\{R_{l+1},\ldots,R_{i+1}-1\}}(dX_{l+1} + \pi_{l+2,i}).$$

We deduce that

$$\begin{aligned} \theta \subseteq \frac{\partial^{m_{l+1}-1} \sigma_{l+1}}{\partial dx_{R_{l+1}}{}^{m_{l+1}-1}} &= \nabla_{\{R_{l+1},\ldots,R_{i+1}-1\}}(dX_{l+1} + \pi_{l+2,i}) \\ &\quad + (dX_{l+1} + \pi_{l+2,i})\zeta_1 + \nabla\zeta_2 + \mathfrak{m}^\kappa \zeta_1. \end{aligned} \tag{7}$$

As for the projection $\theta^\Omega = \theta \cap dS_{l+2}$ we use Corollary 4:

$$\nabla_{\{R_{l+1},\ldots,R_{i+1}-1\}}(dX_{l+1} + \pi_{l+2,i}) \cap dS_{l+2} = \nabla\pi_{l+2,i}.$$

This leads to

$$\theta^\Omega = \nabla\pi_{l+2,i} + \pi_{l+2,i}\zeta_1 + \nabla\zeta_2 + \mathfrak{m}^\kappa\zeta_1 = \sigma_{l+2}.$$

From (7) we deduce roughly that

$$\theta^\nabla \subseteq \pi_{l+2,i} + \zeta_1.$$

Finally

$$\theta^\nabla \theta_\Omega \subseteq (\pi_{l+2,i} + \zeta_1)(\pi_{l+2,i} + \nabla\zeta_1) = \pi_{l+2,i}^2 + \pi_{l+2,i}\zeta_1 + \zeta_1\nabla\zeta_1,$$

and using Propositions 4 and 5 yields

$$\theta^{\nabla}\theta_{\Omega} + \mathfrak{m}^{\kappa}\zeta_1 \subseteq \nabla\pi_{l+2,i} + \pi_{l+2,i}\zeta_1 + \nabla\zeta_2 = \sigma_{l+2}.$$

□

For complexity estimate we recall that the constant Ω is less than 4 and at least 3 (see definition in §4.1). We observe that all the computations can be done modulo the ideal $\nabla_{\{1,\ldots,n\}}(\pi_{1,i} + \zeta_2) + \mathfrak{m}^{2\kappa}$:

Lemma 7 *For any $i = 0, \ldots, \nu$ and any $l = 1, \ldots, i$, the following inclusion holds:*

$$\nabla_{\{1,\ldots,n\}}(\pi_{1,i} + \zeta_2) \subseteq \pi_{1,l} + \sigma_{l+1}.$$

Proof. It suffices to prove that

$$\nabla_{\{1,\ldots,n\}}(\pi_{1,i} + \zeta_2) \subseteq \pi_{1,l} + \nabla\pi_{l+1,i} + \pi_{l+1,i}\zeta_1 + \nabla\zeta_2.$$

We prove the reverse inclusion for the support of the ideals. We write $dx^{\alpha}dx^{\beta}dx^{\gamma}$ a monomial of the support of $\pi_{1,l} + \nabla\pi_{l+1,i} + \pi_{l+1,i}\zeta_1 + \nabla\zeta_2$, where dx^{α} is in the variables $dX_1, \ldots, dX_l$, dx^{β} in the variables $dX_{l+1}, \ldots, dX_i$ and dx^{γ} in the variables $dx_{R_{i+1}}, \ldots, dx_n$. Necessarily we have: $dx^{\alpha} \in \mathrm{supp}(\pi_{1,l})$, $dx^{\beta} \in \mathrm{supp}(\nabla\pi_{l+1,i})$, $dx^{\gamma} \in \mathrm{supp}(\nabla\zeta_2)$. We examine the two possible cases. In the first case $dx^{\beta} \in \mathrm{supp}(\pi_{l+1,i})$, we conclude thanks to Corollary 4. The second situation is $dx^{\beta} \in \pi_{l+1,i}$: necessarily we have $dx^{\gamma} \in \mathrm{supp}(\zeta_1) \subseteq \mathrm{supp}(\zeta_2)$, hence $dx^{\alpha}dx^{\beta}dx^{\gamma} \in \mathrm{supp}(\nabla_{\{1,\ldots,n\}}(\pi_{1,i} + \zeta_2))$. □

Proposition 13 *Let c denote $nm_1 \cdots m_i\lambda$. In terms of arithmetic operations in $A/(\mathfrak{m}^{2\kappa}A)$, the complexity of the function* `IncrementalLift` *is in*

$$\mathcal{O}\Big((nL + n^{\Omega})c^2\log(c)\Big).$$

Proof. Let $S := A/(\mathfrak{m}A)^{2\kappa}[[x_1 - x_1^*, \ldots, x_n - x_n^*]]$ and $\pi = \nabla_{\{1,\ldots,n\}}(\pi_{1,i} + \zeta_2)$. Using Lemma 7, we perform the computations is S/π. Let $\mathcal{C} := \mathcal{C}_S(\pi)$. From Proposition 3 we know that $\deg(\pi) \in \mathcal{O}(c)$. Using Proposition 12 we deduce that $\mathcal{C} \in \mathcal{O}(c^2\log(c))$ and that the initialization of S/π is done within $\mathcal{O}(n^2\mathcal{C})$.
We analyze the complexity for going from (L_l) to (L_{l+1}) in terms of the number of operations in $A/(\mathfrak{m}^{2\kappa}A)$.

Step 1 costs $\mathcal{O}(L\mathcal{C})$. Since $n \leq c$, coeff_l requires $\mathcal{O}(\mathcal{C})$. Therefore Step 2 costs $\mathcal{O}(r_{l+1}\mathcal{C})$. Step 3 can be done within $\mathcal{O}(r_{l+1}\mathcal{C})$. Step 4 performs r_{l+1}^2 calls to coeff, this needs $\mathcal{O}(r_{l+1}^2\mathcal{C})$. In Step 5 one has to inverse an $r_{l+1} \times r_{l+1}$ invertible matrix over a ring it can be done in $\mathcal{O}(r_{l+1}^{\Omega}\mathcal{C})$. Last, Step 6 performs at most nr_{l+1} differentiations and n matrix vector products in dimension r_{l+1}. This yields $\mathcal{O}(nr_{l+1}^2\mathcal{C})$. Summing all these costs yields $\mathcal{O}((L + r_{l+1}^{\Omega} + nr_{l+1}^2)\mathcal{C})$. Now summing for l from 0 to i and the assumption $\Omega \geq 3$ yield the claimed bound $\mathcal{O}((nL + n^{\Omega})\mathcal{C})$. □

4.3 Computation of the Nested Coordinates

Now we explain how to compute the nested coordinates over the residue field $\mathfrak{o}/\mathfrak{m}$ together with a generic trace. We call this part of the algorithm `NestedCoordinates`. This function is always called before entering a lifting process.

`NestedCoordinates`

Input:

- $f_1, \ldots, f_s$ polynomials in $\mathfrak{o}[x_1, \ldots, x_n]$.
- A monic polynomial q in $\mathfrak{o}/\mathfrak{m}[T]$. We let $A := \hat{\mathfrak{o}}[T]/(q)$.
- An approximation x^* in $(A/\mathfrak{m}A)^n$ of an isolated root with multiplicity M of $f_1 = \cdots = f_s = 0$.

Output:

- If the coordinates are generic enough and if $\mathfrak{m}$ is lucky then the function returns the nested coordinates $Y^\nu()$ in $\mathfrak{o}/\mathfrak{m}[[dx_1, \ldots, dx_n]]$ at precision $\pi_{1,\nu}$. The function also returns a generic trace for the deflation.

Algorithm:

In order to shorten the notations we let

$$dW_{i+1} := x^*_{R_{i+1}} + dx_{R_{i+1}}, \ldots, x^*_n + dx_n,$$

$$\zeta_\lambda := (dx^\lambda_{R_{i+1}}, dx_{R_{i+1}+1}, \ldots, dx_n) \text{ and } \nabla\zeta_\lambda := \nabla_{\{R_{i+1},\ldots,n\}}\zeta_\lambda.$$

The computation of the nested coordinates is incremental: we get the ith nested coordinates $Y^i(dW_{i+1})$ in sequence for $i = 0, \ldots, \nu$. The initialization (for $i = 0$) is trivial since Y^0 is the empty sequence.

We enter the ith step of the computation with

($\mathtt{C}_i$) We know

- $r_1, \ldots, r_i$;
- $m_1, \ldots, m_i$;
- $Y^i(dW_{i+1})$ at precision $\pi_{1,i} + \nabla\zeta_1$.

We explain how to go from ($\mathtt{C}_i$) to ($\mathtt{C}_{i+1}$). All the calls to `IncrementalLift` are done at precision $\kappa = 1$.

In a first stage we search the value of m_{i+1}. These computations are organized around one main loop. We initialize $\lambda = 1$ and enter the following loop:

1. repeat:

   - Compute $\Phi_{i+1}(dW_{i+1})$ at precision $\nabla\zeta_\lambda$ using the formula of Proposition 11.
   - Find the smallest $l \leq \lambda$ such that the monomial $dx^l_{R_{i+1}}$ is in the support of the elements of $\Phi_{i+1}(dW_{i+1})$. If such an l exists then it equals m_{i+1}. In this case break the loop.
   - Call `IncrementalLift` in order to reach the precision $\nabla\zeta_{2\lambda}$ and the replace λ by 2λ.

   The value of m_{i+1} is known. It remains to compute r_{i+1} and $\mathcal{T}_{i+1}$.

2. Compute
$$\widetilde{\Phi}_{i+1}(dW_{i+1}) = \frac{\partial^{m_{i+1}-1}}{\partial {dx_{R_{i+1}}}^{m_{i+1}-1}}\Phi_{i+1}(dW_{i+1})$$
at precision $\nabla\zeta_1$.

3. Build the $(sm_1 \cdots m_{i+1}) \times (n - R_{i+1} + 1)$ matrix J:
$$J := \frac{\partial\widetilde{\Phi}_{i+1}(dW_{i+1})}{\partial\{dx_{R_{i+1}}, \ldots, dx_n\}}(0, \ldots, 0).$$
This is the row matrix of the gradients of the elements of $\widetilde{\Phi}_{i+1}$ at x^*.

4. Let
$$dW_{i+2} := x^*_{R_{i+2}} + dx_{R_{i+2}}, \ldots, x^*_n + dx_n.$$
Compute the rank r_{i+1} of the matrix J and the expression of $Y_{i+1}(dW_{i+2})$ at precision $\nabla_{\{R_{i+2},\ldots,n\}}(dx_{R_{i+2}}, \ldots, dx_n)$. Deduce the trace $\mathcal{T}_{i+1}$. We let
$$dZ_{i+1} := Y_{i+1}(dW_{i+2}) - (x^*_{R_{i+1}}, \ldots, x^*_{R_{i+2}-1}).$$

5. We deduce the value
$$Y^{i+1}(dW_{i+2}) = (Y^i(Y_{i+1}(dW_{i+2}) + dX_{i+1}, dW_{i+2}), Y_{i+1}(dW_{i+2}))$$
at precision $\nabla_{\{R_{i+2},\ldots,n\}}(dx_{R_{i+2}}, \ldots, dx_n)$ using the following first order Taylor expansion:
$$Y^i(Y_{i+1}(dW_{i+2}) + dX_{i+1}, dW_{i+2}) = Y^i(dW_{i+1}) + \frac{\partial Y^i(dW_{i+1})}{\partial dX_{i+1}}dZ_{i+1} + \mathcal{O}((dZ_{i+1})^2).$$

The computation of the rank and the determination of $\mathcal{T}_{i+1}$ can be done using the classical Gaussian elimination process for instance. Indeed we can use any linear algebra algorithm. But it is important to notice that, according to this choice, the luckiness of $\mathfrak{m}$ differs. We can only say that:

Proposition 14 *There exists an element $a \neq 0$ in $\mathfrak{o}$ such that any $\mathfrak{m}$ that does not contain a is lucky for* `NestedCoordinates`.

Proof. Formally, one can think about the execution of `NestedCoordinates` over $k(u)$ instead of A. The function performs a finite number of equality tests and divisions. The maximal ideal $\mathfrak{m}$ is lucky if this mental computation can be specialized modulo $\mathfrak{m}$. Therefore the element a is a multiple of all the denominators of all the elements of k occurring in this computation. □

Proposition 15 *If the coordinates are generic enough and if $\mathfrak{m}$ is lucky then the function* `NestedCoordinates` *requires*

$$\mathcal{O}\Big(n^3(nL+n^\Omega)M^2\log(nM)\Big)$$

arithmetic operations in $A/\mathfrak{m}A$.

Proof. Let $c_\lambda := nm_1\cdots m_i\lambda$ and $\mathcal{C}_\lambda = c_\lambda^2\log(c_\lambda)$. From Proposition 13 the call of `IncrementalLift` with input precision $\nabla\zeta_\lambda$ and output precision $\nabla\zeta_{2\lambda}$ costs $\mathcal{O}((nL+n^\Omega)\mathcal{C}_{2\lambda})$ operations in $A/\mathfrak{m}A$. The computation of $\Phi_{i+1}(dW_{i+1})$ requires $\mathcal{O}(L\mathcal{C}_{2\lambda} + sm_1\cdots m_i nc_{2\lambda}\log(c_{2\lambda})) \subseteq \mathcal{O}((L+s)\mathcal{C}_{2\lambda}) \subseteq \mathcal{O}(L\mathcal{C}_{2\lambda})$.
We deduce that the total cost of the loop is in $\mathcal{O}((nL+n^\Omega)\mathcal{C}_{m_{i+1}})$.

The cardinal of Φ_{i+1} equals $sm_1\cdots m_i$. The computation of the value $\widetilde{\Phi}_{i+1}(dW_{i+1})$ requires $\mathcal{O}(sm_1\cdots m_{i+1}c_{m_{i+1}}) \subseteq \mathcal{O}(s\mathcal{C}_{m_{i+1}})$. The construction of J requires

$$\mathcal{O}(sm_1\cdots m_{i+1}nc_{m_{i+1}}\log(c_{m_{i+1}})) \subseteq \mathcal{O}(s\mathcal{C}_{m_{i+1}}).$$

The Gaussian elimination can be done within

$$\mathcal{O}(sm_1\cdots m_{i+1}n^2) \subseteq \mathcal{O}(s\mathcal{C}_{m_{i+1}}).$$

Last, updating the value of Y^i requires

$$\mathcal{O}(nr_{i+1}c_{m_{i+1}} + nr_{i+1}\mathcal{C}_{m_{i+1}}) \subseteq \mathcal{O}(nr_{i+1}\mathcal{C}_{m_{i+1}}).$$

The total of these last operations is in $\mathcal{O}((s+nr_{i+1})\mathcal{C}_{m_{i+1}})$.
We conclude that going from Step $(\mathtt{C}_i)$ to $(\mathtt{C}_{i+1})$ requires $\mathcal{O}((nL+n^\Omega)\mathcal{C}_{m_{i+1}})$. It remains to sum these costs from $(\mathtt{C}_0)$ to $(\mathtt{C}_\nu)$ to get the claimed bound. □

4.4 An *a posteriori* probabilistic luckiness test

We give a variant of the function `NestedCoordinates` taking the generic trace as argument and returning the nested coordinates. Computations are essentially the same as in `NestedCoordinates` but no rank determination is necessary.

`NestedCoordinatesWithTrace`

Input:

- $f_1, \ldots, f_s$ polynomials in $\mathfrak{o}[x_1, \ldots, x_n]$.
- a monic polynomial q in $\mathfrak{o}/\mathfrak{m}[T]$. We let $A := \hat{\mathfrak{o}}[T]/(q)$.
- A approximation x^* in $(A/\mathfrak{m}A)^n$ of an isolated root of $f_1 = \cdots = f_s = 0$.
- The generic trace $\mathcal{T}$.

Output:

- If the coordinates are generic enough and if $\mathfrak{m}$ satisfies the smoothness hypotheses of §3.3 then the function returns the nested coordinates $Y^\nu()$ in $\mathfrak{o}/\mathfrak{m}[[dx_1, \ldots, dx_n]]$ at precision $\pi_{1,\nu}$.

Proposition 16 *If the coordinates are generic enough and if* $\mathfrak{m}$ *is lucky then* `NestedCoordinatesWithTrace` *requires*

$$\mathcal{O}\Big(n^3(nL + n^\Omega)M^2 \log(nM)\Big)$$

arithmetic operations in $A/\mathfrak{m}A$.

Once we know the generic trace it is sometimes useful to have a probabilistic test of the smoothness hypotheses for another maximal ideal $\mathfrak{m}$. From a practical point of view, `NestedCoordinatesWithTrace` raises a "division by zero" error message if the inversion of an element which is not zero is not possible. This message occurs when $\mathfrak{m}$ does not satisfy the smoothness hypotheses.

Proposition 17 *Let* $\mathcal{T}$ *be the generic trace of* x^* *if the coordinates are generic enough then if* `NestedCoordinates` *does not raise "division by zero" then the smoothness hypotheses* (H_Ω) *are satisfied.*

Proof. The only inversions occurring in `NestedCoordinates` are the ones of the Jacobian matrices of the Ω_i. These inversions involve determinants only. This corresponds to the smoothness hypotheses exactly. □

4.5 Example 3 continued from §3.2

We illustrate `NestedCoordinates` on Example 3 of §3.2. In order to make the reading of the computation easier we work over k instead of $\mathfrak{o}/\mathfrak{m}$, for some lucky maximal ideal $\mathfrak{m}$. Moreover we do not change the coordinates to generic ones since we know from 3.2 that all the Weierstraß positions we need are satisfied.

We enter `NestedCoordinates` at step $(\mathtt{C}_0)$. It is easy to check that $m_1 = 1$, $r_1 = 1$ and compute

$$Y_1^1(dx_2, -1 + dx_3) = -dx_2 + dx_3 + \mathcal{O}((dx_1) + \nabla_{\{2,3\}}(dx_2, dx_3)).$$

We arrive at step ($\mathtt{C}_1$).

We compute $Y_1^1(dx_2, -1+dx_3) = Y_1(dx_2, -1+dx_3)$ as the solution of $f_1(Y_1^1(dx_2, -1+dx_3), dx_2, -1+dx_3)$ satisfying the condition $Y_1^1(dx_2, -1+dx_3) = 0$ at precision $\mathcal{O}((dx_1)+\nabla_{\{2,3\}}(dx_2^3, dx_3))$. After 2 calls to `IncrementalLift` we obtain:

$$\begin{aligned} Y_1^1(dx_2, -1+dx_3) &= dx_3 + dx_2(-1+2\,dx_3) + dx_2^2(-2+8\,dx_3) - 4\,dx_2^3 \\ &\quad +\mathcal{O}((dx_1) + \nabla_{\{2,3\}}(dx_2^3, dx_3)). \end{aligned}$$

Substituting x_1 by $Y_1^1(dx_2, -1+dx_3)$ in F, we get $\gamma_1, \gamma_2, \gamma_3$:

$$\begin{aligned} \gamma_1 &= f_1(Y_1^1(dx_2, -1+dx_3) + dx_1, dx_2, -1+dx_3) \\ &= 0 + \mathcal{O}((dx_1) + \nabla_{\{2,3\}}(dx_2^3, dx_3)), \\ \gamma_2 &= f_2(Y_1^1(dx_2, -1+dx_3) + dx_1, dx_2, -1+dx_3) \\ &= -3\,dx_2^2 dx_3 + dx_2^3 + \mathcal{O}((dx_1) + \nabla_{\{2,3\}}(dx_2^3, dx_3)), \\ \gamma_2 &= f_3(Y_1^1(dx_2, -1+dx_3) + dx_1, dx_2, -1+dx_3) \\ &= 0 + \mathcal{O}((dx_1) + \nabla_{\{2,3\}}(dx_2^3, dx_3)). \end{aligned}$$

Then we deduce $\Phi_2 = \{\phi_1, \phi_2, \phi_3\}$ where

$$\begin{aligned} \phi_1(dx_2, -1+dx_3) &= \mathrm{coeff}_1(\gamma_1, dx_1^0) \\ &= 0 + \mathcal{O}(\nabla_{\{2,3\}}(dx_2^3, dx_3)), \\ \phi_2(dx_2, -1+dx_3) &= \mathrm{coeff}_1(\gamma_2, dx_1^0) \\ &= -3\,dx_2^2 dx_3 + dx_2^3 + \mathcal{O}(\nabla_{\{2,3\}}(dx_2^3, dx_3)), \\ \phi_3(dx_2, -1+dx_3) &= \mathrm{coeff}_1(\gamma_3, dx_1^0) \\ &= 0 + \mathcal{O}(\nabla_{\{2,3\}}(dx_2^3, dx_3)). \end{aligned}$$

We find that $m_2 = 3$ and we have to study the rank r_2 of the gradients of the elements of $\widetilde{\Phi}_2$:

$$\widetilde{\Phi}_2(dx_2, -1+dx_3) = \left\{\phi_1, \phi_2, \phi_3, \frac{\partial\phi_1}{\partial x_2}, \frac{\partial\phi_2}{\partial x_2}, \frac{\partial\phi_3}{\partial x_2}, \frac{\partial^2\phi_1}{\partial {x_2}^2}, \frac{\partial^2\phi_2}{\partial {x_2}^2}, \frac{\partial^2\phi_3}{\partial {x_2}^2}\right\}.$$

All the elements of $\widetilde{\Phi}_2$ equal $0+\mathcal{O}(\nabla_{\{2,3\}}(dx_2, dx_3))$ except $\frac{\partial^2\Phi_2}{\partial {dx_2}^2}$: we deduce that $r_2 = 1$ and examine

$$\frac{\partial^2\phi_2}{\partial {dx_2}^2}(dx_2, -1+dx_3) = -6\,dx_3 + 6\,dx_2 + \mathcal{O}(\nabla_{\{2,3\}}(dx_2, dx_3)).$$

This yields the first order approximation of $Y_2^2(-1+dx_3)$:

$$Y_2^2(-1+dx_3) = Y_2(-1+dx_3) = dx_3 + \mathcal{O}((dx_1, dx_2^3) + \nabla_{\{3\}}(dx_3)).$$

We can now update $Y_1^2(1+dx_3)$:

$$\begin{aligned} Y_1^2(-1+dx_3) &= Y_1^1(Y_2^2(-1+dx_3)+dx_2,-1+dx_3) \\ &= Y_1^1(dx_2,-1+dx_3)+dx_3\frac{\partial Y_1^1(dx_2,1+dx_3)}{\partial dx_2} \\ &= Y_1^1(dx_2,-1+dx_3)+dx_3\Big(-1-4\,dx_2-12\,dx_2^2 \\ &\quad +\mathcal{O}((dx_1,dx_2^3)+(dx_3))\Big) \\ &= dx_2(-1-2\,dx_3)+dx_2^2(-2-4\,dx_3) \\ &\quad +\mathcal{O}((dx_1,dx_2^3)+\nabla_{\{3\}}(dx_3)). \end{aligned}$$

Step ($\mathtt{C}_2$) of `NestedCoordinates` is reached.

We check that the precision is not enough and enter `IncrementalLift` again. Let z_1 and z_2 denote the former approximations of $Y_1^2(-1+dx_3)$ and $Y_2^2(-1+dx_3)$ respectively. The evaluation of f_1 yields:

$$\begin{aligned} f_1(z_1+dx_1,z_2+dx_2,-1+dx_3) &= 3\,dx_3^2+8\,dx_2^2dx_3^2+8\,dx_2^3 \\ &\quad +dx_1(2-4\,dx_2-8\,dx_2^2) \\ &\quad +\mathcal{O}(\nabla_{\{1,2\}}(dx_1,dx_2^3)+(dx_1,dx_2^3)(dx_3)+\nabla_{\{3\}}(dx_3^2)). \end{aligned}$$

We deduce

$$\begin{aligned} dz_1 &= -\frac{3\,dx_3^2+8\,dx_2^2dx_3^2+8\,dx_2^3}{2-4\,dx_2-8\,dx_2^2}+\mathcal{O}(\nabla_{\{2\}}(dx_2^3)+(dx_2^3)(dx_3)+\nabla_{\{3\}}(dx_3^2)) \\ &= -\frac{3}{2}dx_3^2-3\,dx_2dx_3^2-16\,dx_2^2dx_3^2-4\,dx_2^3 \\ &\quad +\mathcal{O}(\nabla_{\{2\}}(dx_2^3)+(dx_2^3)(dx_3)+\nabla_{\{3\}}(dx_3^2)). \end{aligned}$$

Therefore we arrive at Step ($\mathtt{L}_1$) with

$$\begin{aligned} Y_1^1(z_2+dx_2,-1+dx_3) &= z_1+dz_1 \\ &= -\frac{3}{2}dx_3^2+dx_2(-1-2\,dx_3-3\,dx_3^2)+dx_2^2(-2-4\,dx_3-16\,dx_3^2) \\ &\quad -4\,dx_2^3+\mathcal{O}((dx_1)+\nabla_{\{2\}}(dx_2^3)+(dx_2^3)(dx_3)+\nabla_{\{3\}}(dx_3^2)). \end{aligned}$$

We deduce the new value of Ω_2:

$$\begin{aligned} &\Omega_2(z_2+dx_2,-1+dx_3) = \\ &\quad \Big\{\tfrac{\partial^2}{\partial {dx_2}^2}\mathrm{coeff}_1(f_2(Y_1^1(z_2+dx_2,-1+dx_3)+dx_1,z_2+dx_2,-1+dx_3),1)\Big\} \\ &\qquad = \Big\{9\,dx_3^2+6\,dx_2+\mathcal{O}(\nabla_{\{2\}}(dx_2)+(dx_2)(dx_3)+\nabla_{\{3\}}(dx_3^2))\Big\}. \end{aligned}$$

We get $dz_2=-\frac{3}{2}dx_3^2+\mathcal{O}(\nabla_3(dx_3^2))$ and improve the precision of $Y_2^2(-1+dx_3)$:

$$\begin{aligned} Y_2^2(-1+dx_3) &= z_2+dz_2 \\ &= dx_3-\frac{3}{2}dx_3^2+\mathcal{O}((dx_1,dx_2^3)+\nabla_3(dx_3^2)). \end{aligned}$$

The new value of Y_1^2 becomes:

$$\begin{aligned}
Y_1^2(-1+dx_3) &= Y_1^1(z_2+dz_2+dx_2,-1+dx_3)\\
&= Y_1^1(z_2+dx_2,-1+dx_3)+dz_2\frac{\partial Y_1^1(z_2+dx_2,1+dx_3)}{\partial dx_2}\\
&= Y_1^1(z_2+dx_2,-1+dx_3)-\frac{3}{2}dx_3^2\Big(-1-4\,dx_2\\
&\quad -12\,dx_2^2+\mathcal{O}((dx_1)+\nabla_{\{2\}}(dx_2^2)+(dx_3))\Big)\\
&= Y_1^1(z_2+dx_2,-1+dx_3)+\frac{3}{2}dx_3^2+6\,dx_2dx_3^2\\
&\quad +18\,dx_2^2dx_3^2+\mathcal{O}((dx_1,dx_2^3)+\nabla_{\{3\}}(dx_3^2))\\
&= dx_2(-1-2\,dx_3+3\,dx_3^2)+dx_2^2(-2-4\,dx_3+2\,dx_3^2)\\
&\quad +\mathcal{O}((dx_1,dx_2^3)+\nabla_{\{3\}}(dx_3^2)).
\end{aligned}$$

Step ($\mathtt{L}_2$) is reached but the precision is not enough to find m_3 and r_3. We enter `IncrementalLift` again. Let z_1 and z_2 denote the last approximations of $Y_1^2(-1+dx_3)$ and $Y_2^2(-1+dx_3)$ respectively. The evaluation of f_1 yields:

$$\begin{aligned}
&f_1(z_1+dx_1,z_2+dx_2,-1+dx_3)=\\
&\quad -6\,dx_3^3+\frac{9}{2}dx_3^4+dx_2^2(-24\,dx_3^3+18\,dx_3^4)+dx_2^3(8+32\,dx_3)\\
&\quad +dx_1(2+dx_2(-4-8\,dx_3)+dx_2^2(-8-16\,dx_3))\\
&\quad +\mathcal{O}(\nabla_{\{1,2\}}(dx_1,dx_2^3)+(dx_1,dx_2^3)(dx_3^2)+\nabla_{\{3\}}(dx_3^4)).
\end{aligned}$$

We find

$$\begin{aligned}
dz_1 &= -\frac{-6\,dx_3^3+\frac{9}{2}dx_3^4+dx_2^2(-24\,dx_3^3+18\,dx_3^4)+dx_2^3(8+32\,dx_3)}{2+dx_2(-4-8\,dx_3)+dx_2^2(-8-16\,dx_3)}\\
&\quad +\mathcal{O}(\nabla_{\{2\}}(dx_2^3)+(dx_2^3)(dx_3^2)+\nabla_{\{3\}}(dx_3^4))\\
&= 3\,dx_3^3-\frac{9}{4}dx_3^4+dx_2(6\,dx_3^3+\frac{15}{2}dx_3^4)+dx_2^2(36\,dx_3^3+45\,dx_3^4)\\
&\quad +dx_2^3(-4-16\,dx_3)+\mathcal{O}(\nabla_{\{2\}}(dx_2^3)+(dx_2^3)(dx_3^2)+\nabla_{\{3\}}(dx_3^4)).
\end{aligned}$$

Therefore we arrive at Step ($\mathtt{L}_1$) with

$$\begin{aligned}
&Y_1^1(z_2+dx_2,-1+dx_3)=z_1+dz_1=\\
&\quad 3\,dx_3^3-\frac{9}{4}dx_3^4+dx_2(-1-2\,dx_3+3\,dx_3^2+6\,dx_3^3+\frac{15}{2}dx_3^4)\\
&\quad +dx_2^2(-2-4\,dx_3+2\,dx_3^2+36\,dx_3^3+45\,dx_3^4)\\
&\quad +dx_2^3(-4-16\,dx_3)+\mathcal{O}((dx_1)+\nabla_{\{2\}}(dx_2^3)+(dx_2^3)(dx_3^2)+\nabla_{\{3\}}(dx_3^4)).
\end{aligned}$$

We deduce the new value of Ω_2:

$$\Omega_2(z_2+dx_2,-1+dx_3)=$$

$$\left\{\frac{\partial^2}{\partial {dx_2}^2}\mathrm{coeff}_1(f_2(Y_1^1(z_2+dx_2,-1+dx_3)+dx_1,z_2+dx_2,-1+dx_3),1)\right\}$$
$$=\Big\{-18\,dx_3^3-\tfrac{243}{2}dx_3^4+dx_2(6+36\,dx_3)$$
$$+\mathcal{O}(\nabla_{\{2\}}(dx_2^3)+(dx_2^3)(dx_3^2)+\nabla_{\{3\}}(dx_3^4))\Big\}.$$

We get $dz_2 = 3dx_3^3 + \frac{9}{4}dx_3^4 + \mathcal{O}(\nabla_3(dx_3^4))$ and improve the precision of $Y_2^2(-1+dx_3)$:

$$Y_2^2(-1+dx_3) = z_2 + dz_2 =$$
$$dx_3 - \frac{3}{2}dx_3^2 + 3\,dx_3^3 + \frac{9}{4}dx_3^4 + \mathcal{O}((dx_1, dx_2^3) + \nabla_3(dx_3^4)).$$

The new value of Y_1^2 becomes:

$$\begin{aligned} Y_1^2(-1+dx_3) &= Y_1^1(z_2+dz_2+dx_2,-1+dx_3)\\ &= Y_1^1(z_2+dx_2,-1+dx_3)+dz_2\frac{\partial Y_1^1(z_2+dx_2,-1+dx_3)}{\partial dx_2}\\ &= -\frac{21}{2}dx_3^4+dx_2(-1-2\,dx_3+3\,dx_3^2-6\,dx_3^3-\frac{51}{2}dx_3^4)\\ &\quad +dx_2^2(-2-4\,dx_3+2\,dx_3^2-126\,dx_3^4)\\ &\quad +\mathcal{O}((dx_1,dx_2^3)+\nabla_{\{3\}}(dx_3^4)). \end{aligned}$$

Step ($\mathtt{L}_2$) is reached. We deduce $\Phi_3 = \{\varphi_1, \ldots, \varphi_9\}$, where $\varphi_l = 0 + \mathcal{O}(\nabla_{\{3\}}(dx_3^4))$ for $l = 1, \ldots, 8$ and $\varphi_9 = 9000\,dx_3^4 + \mathcal{O}(\nabla_{\{3\}}(dx_3^4))$. Therefore, $m_3 = 4$, $r_3 = 1$ and $\widetilde{\Phi}_3$. All the elements of $\widetilde{\Phi}_3$ equal $0 + \mathcal{O}(\nabla_{\{3\}}(dx_3))$ except $\frac{\partial^3\varphi_9}{\partial {dx_3}^3} = 216000\,dx_3 + \mathcal{O}(\nabla_{\{3\}}(dx_3))$.

4.6 Lifting the Nested Coordinates

Once we know the nested coordinates Y^ν over $A/(\mathfrak{m}^\kappa A)$ we can lift them in $A/(\mathfrak{m}^{2\kappa}A)$, for any $\kappa \geq 1$.

`LiftNestedCoordinates`

Input:

- $f_1, \ldots, f_s$ polynomials in $\mathfrak{o}[x_1, \ldots, x_n]$.
- a monic irreducible polynomial q in $\mathfrak{o}/\mathfrak{m}[T]$. We let $A := \hat{\mathfrak{o}}/\mathfrak{m}[T]/(q)$.
- Y^ν at precision $\mathfrak{m}^\kappa A + \pi_{1,\nu}$, the nested coordinates with respect to an isolated root x^* of $f_1 = \cdots = f_s = 0$.
- The generic trace $\mathcal{T}$ of the deflation computed by `NestedCoordinates`.

Output:

- Y^ν at precision $\mathfrak{m}^{2\kappa}A + \pi_{1,\nu}$.

Algorithm: Call straightforwardly the procedure `IncrementalLift`. Recall that the coordinates must be generic enough.

Proposition 18 *If the coordinates are generic enough and if* $\mathfrak{m}$ *is lucky then the function* `LiftNestedCoordinates` *requires*

$$\mathcal{O}(n^2(nL+n^\Omega)M^2\log(nM))$$

arithmetic operations in $A/(\mathfrak{m}^{2\kappa}A)$.

4.7 Example 3 continued from §4.5

In §4.5 we computed over $k=\mathbb{Q}$ the nested coordinates Y^3. As discussed in §3.3, $\mathfrak{m}=(7)$ is a lucky maximal ideal in $\mathfrak{o}=\mathbb{Z}$. We could have computed the nested coordinates in $\mathbb{Z}/p\mathbb{Z}$, they coincide with the value modulo 7 of the ones computed over $\mathbb{Q}$:

$$\begin{aligned}
Y_1^3 &= dx_2(6+5\,dx_3+3\,dx_3^2+dx_3^3+6\,dx_3^4)+dx_2^2(5+3\,dx_3+2\,dx_3^2)\\
&\quad +\mathcal{O}(dx_1,dx_2^3,dx_3^4),\\
Y_2^3 &= dx_3+2\,dx_3^2+3\,dx_3^3+4\,dx_3^4+\mathcal{O}(dx_1,dx_2^3,dx_3^4),\\
Y_3^3 &= 6+\mathcal{O}(dx_1,dx_2^3,dx_3^4).
\end{aligned}$$

Modulo 7 our approximated root z is $(0,0,6)$. We call the function `LiftNested-Coordinates` and get modulo 7^2 the new nested coordinates Y^3:

$$\begin{aligned}
Y_1^3 &= 21\,dx_3^4+dx_2(48+47\,dx_3+3\,dx_3^2+43\,dx_3^3+27\,dx_3^4)\\
&\quad +dx_2^2(47+45\,dx_3+2\,dx_3^2)+\mathcal{O}(dx_1,dx_2^3,dx_3^4),\\
Y_2^3 &= dx_3+23\,dx_3^2+3\,dx_3^3+4\,dx_3^4+\mathcal{O}(dx_1,dx_2^3,dx_3^4),\\
Y_3^3 &= 48+\mathcal{O}(dx_1,dx_2^3,dx_3^4).
\end{aligned}$$

The new approximation of $x^*=(0,0,-1)$ we get modulo 49 is $(0,0,48)$, which is correct.

4.8 Summary of the Algorithm

We gather the above algorithms in order to get a proof of Theorem 1. The existence of a in the theorem comes from Proposition 8 and Proposition 14. The restriction on the characteristic of k comes from the deflation lemma 4.

In §3.5 it is shown that the change of coordinates that are not generic enough are contained in an algebraic hypersurface of $GL_n(k)$. From Proposition 15 the computation of the nested coordinates is done within

$$\mathcal{O}\Big(n^3(nL+n^\Omega)M^2\log(nM)\Big)$$

arithmetic operations in $A/\mathfrak{m}A$.

Once the nested coordinates are computed modulo $\mathfrak{m}$ we can lift them using the procedure `LiftNestedCoordinates`. Reaching precision $\mathfrak{m}^{2\kappa}$ from precision $\mathfrak{m}^\kappa$ costs

$$\mathcal{O}\Big(n^2(nL+n^\Omega)M^2\log(nM)\Big)$$

arithmetic operations in $A/(\mathfrak{m}^{2\kappa}A)$.

5 Splittings

From the beginning we have assumed that the root x^* is given by means of an irreducible minimal polynomial Q. The question arising naturally is to know what happens if Q is not irreducible any more: in this case x^* represents several irreducible sets of roots. It is a well-known fact that such a systematic extension of our method for a reducible square free polynomial Q is possible via *dynamic evaluation*. This general framework has been developed both from theoretical and practical point of views in the Axiom [Sut92] computer algebra system in a series of papers by Duval and her collaborators: Della Dora, Dellière, Dicrescenzo, Gómez Díaz, Reynaud [DDD85, Duv87, DD89, Duv89, Duv94, Día94, Duv94, DR94a, DR94b, BGW95, Duv95, Del99]. Following the dynamic evaluation scheme, we develop in this section a special function for handling a non irreducible situation and estimate its complexity.

From now on we assume that Q is not necessarily irreducible anymore: Q is square free in $k[T]$, we let $k[u] := k[T]/(Q(T))$ and x^* is given by a vector of n polynomials of $k[T]$. If $Q = Q_1 \cdots Q_r$ is the irreducible decomposition of Q in $k[T]$ then $k[u]$ is isomorphic to the Cartesian product of the field extensions $k[T]/(Q_1(T)) \times \cdots \times k[T]/(Q_r(T))$. We still assume that hypotheses (H_Q) and (H_{x^*}) hold. In consequence, (H_{Q_i}) holds and (H_{x^*}) holds in each $k[T]/(Q_i(T))$, for $i = 1, \ldots, r$.

We denote by A_i the quotient $\hat{\mathfrak{o}}[T]/(Q_i(T))$ and still write x^* for the image of x^* in A_i. Note that A is isomorphic to the product $A_1 \times \cdots \times A_r$. We shall write $\mathcal{T}^i$ for the generic trace of the deflation process executed in A_i, as defined in §3.5. Having the same trace induces a partition $p_1, \ldots, p_t$ of the set $\{1, \ldots, r\}$. Let Q_{p_i} be the product of the Q_j for all $j \in p_i$. The aim of the following algorithm is to compute the Q_{p_i} and their associated generic traces $\mathcal{T}^{p_i}$. We write M_i for the multiplicity of x^* as a root of the system $f_1 = \cdots = f_s = 0$ in $k[T]/(Q_i(T))$, for $i = 1, \ldots, r$. We say that x^* is **DA-irreducible** if $t = 1$.

If we execute the function `NestedCoordinates` in A directly to compute the nested coordinates modulo $\mathfrak{m}$ from the knowledge of x^* modulo $\mathfrak{m}$ we probably run into an error due to the fact that $A/\mathfrak{m}A$ is not a field. Such an error occurs when the computation requires a division by an element that is not zero and not invertible. In order to build up the splitting method we assume that the function `NestedCoordinates` raises the message "division by zero" and returns the element causing the trouble.

We introduce the function `DaSplit` which takes as input the polynomial Q known modulo $\mathfrak{m}$ and the value of x^* modulo $\mathfrak{m}$ in the algebra A. It returns the sequences $(Q_{p_i})_{i=1,\ldots,t}$ and $(\mathcal{T}^{p_i})_{i=1,\ldots,t}$.

`DaSplit`

Input:

- a square free monic polynomial Q in $\mathfrak{o}[T]$. We let $A = \hat{\mathfrak{o}}/\mathfrak{m}[T]/(Q)$.
- a vector $(V_1, \ldots, V_n)$ of polynomials in $\mathfrak{o}[T]$ of degree less than $\deg(Q)$ such that $x^* = (V_1, \ldots, V_n)$ in $A/\mathfrak{m}A$.

- a sequence $f_1, \ldots, f_s$ of polynomials such that x^* represents an isolated set of roots of $f_1 = \cdots = f_s = 0$.

Output:

- If the coordinates are generic enough and if $\mathfrak{m}$ is lucky then it returns the sequences $Q_{p_i},\ i = 1, \ldots, t$ and $\mathcal{T}^{p_i}, i = 1, \ldots, t$.

Algorithm:

1. Call `NestedCoordinates` with Q. If the error "division by zero" is raised then the function returns an element P of $\hat{\mathfrak{o}}/\mathfrak{m}[T]$ that is not divisible by Q. Let Q_e be the greatest common divisor of P and Q and $Q_n = Q/Q_e$. We have $\deg(Q_e) \geq 1$ and $\deg(Q_n) \geq 1$, this induces a splitting. If no error occurs then return Q and $\mathcal{T}$ found by `NestedCoordinates`.

2. Call recursively the function with Q_e, then with Q_n, merge the returned sequences and return.

As for complexity estimate we need to count the costs of the operations in A. We introduce the function $\mathcal{U}$ such that all the elementary arithmetic operations (multiplication, division, greatest common divisor) can be performed within $\mathcal{O}(\mathcal{U}(\deg(Q)))$ operations in $\hat{\mathfrak{o}}/\mathfrak{m}$.

Proposition 19 *In case of success the complexity of* `DaSplit` *is in*

$$\mathcal{O}\Big(\log(d)n^4(nL + n^\Omega)D^2\mathcal{U}(\deg(Q))\Big),$$

in terms of arithmetic operations in the residue field $\mathfrak{o}/\mathfrak{m}$*, where*

$$D := \sum_{i=1}^{r} M_i \deg(Q_i).$$

Proof. There are $2t - 1$ calls to `NestedCoordinates`: t of them leading to the Q_{p_j} and $t-1$ raising the "division by zero" error. We perform the complexity analysis in the worst case as if all the computations were performed in A. First we examine the cost of the failing calls. It suffices to observe that the failing computations can be embedded in the successful ones: to each failing call we associate a successful one performing the same operations in A until the error. Therefore, in terms of arithmetic operations in A, the sum of the costs of the calls to `NestedCoordinates` raising the error is bounded by the sum of the costs of the finishing ones.

Let $M_{p_j} = \min_{i \in p_j} M_j$, for $j = 1, \ldots, t$. In terms of arithmetic operations in $\mathfrak{o}/\mathfrak{m}$, the sum of the costs of the `NestedCoordinates` finishing without any error is in

$$\mathcal{O}\Big(\sum_{j=1}^{t} n^3(nL + n^\Omega)M_{p_j}^2 \log(nM_{p_j})\mathcal{U}(\deg(Q))\Big).$$

Noticing that $M_i \leq d^n$ for each $i = 1, \ldots, r$, this yields the bound $\log(nM_i) \in \mathcal{O}(\log(nd^n)) \subseteq \mathcal{O}(n\log(d))$. We conclude the proof this way:

$$\sum_{j=1}^{t} M_{p_j}^2 \leq \sum_{j=1}^{t} M_{p_j}^2 \deg(Q_{p_j})^2 \leq \left(\sum_{j=1}^{t} M_{p_j} \deg(Q_{p_j}) \right)^2 \leq D^2.$$

□

Concerning the probabilistic aspects it is easy to deduce that

Proposition 20 *There exists an element $a \neq 0$ in $\mathfrak{o}$ such that for any $\mathfrak{m}$ which does not contain a* `DaSplit` *returns a correct answer if the linear change of coordinates is chosen outside an algebraic hypersurface of $GL_n(k)$.*

Conclusion

A numerical translation of the algorithm presented here seems possible if we take care of the convergence of the power series. Experiments on small examples indicates that everything works fine. The main problem is the numerical computation of the generic trace.

During our deflation process we do not compute the multiplicity of the root, we only get a lower bound. The natural question we would like to answer is: how to compute efficiently the multiplicity? Another direction we are planning to to explore is the extension of the Jacobian criterion: can we prove that a given root is isolated within a small complexity?

Acknowledgments

I greatly thank M. Giusti, B. Salvy, M. Stillman and the anonymous referees for their useful comments.

References

[Abd97] J. Abdeljaoued. *Algorithmes rapides pour le calcul du polynôme caractéristique.* PhD thesis, Université de Franche-Comté, Besançon, France, 1997.

[Ass94] A. Assi. On flatness of generic projections. *Journal of Symbolic Computation*, 18(5):447–462, 1994.

[BC89] G. Butler and J. Cannon. Cayley version 4: The user language. In *Proceedings of ISSAC'88*, volume 358 of *Lecture Notes in Computer Science*, pages 456–466. New York: Springer, July 1989.

[BC90] G. Butler and J. Cannon. The design of Cayley, a language for modern algebra. In A. Miola, editor, *Design and Implementation of Symbolic Computation Systems*, volume 429 of *Lecture Notes in Computer Science*, pages 10–19. New York: Springer, July 1990.

[BC95] W. Bosma and J. Cannon. Handbook of Magma functions. Sydney: School of Mathematics and Statistics, University of Sydney, 1995.

[BCM94] W. Bosma, J. Cannon, and J. Matthews. Programming with algebraic structures: design of the Magma language. In M. Giesbrecht, editor, *Proceedings of ISSAC'94*. ACM, 1994.

[BCP97] W. Bosma, J. Cannon, and C. Playoust. The Magma algebra system I: The user language. *Journal of Symbolic Computation*, 24, 1997.

[Ber84] S. J. Berkowitz. On computing the determinant in small parallel time using a small number of processors. *Information Processing Letters*, 18:147–150, 1984.

[BGW95] P. A. Broadbery, T. Gómez Díaz, and S. M. Watt. On the implementation of dynamic evaluation. In *ISSAC'95*, pages 77–84, 1995.

[CP96] J. Cannon and C. Playoust. Magma: A new computer algebra system. *Euromath Bulletin*, 2(1):113–144, 1996.

[Csa76] L. Csanky. Fast parallel matrix inversion algorithms. *SIAM Journal of Computing*, 5(4):618–623, 1976.

[DD89] C. Dicrescenzo and D. Duval. Symbolic and algebraic computation. *Lecture Note in Computer science, Springer Verlag*, 358:440–446, 1989.

[DDD85] J. Della Dora, C. Dicrescenzo, and D. Duval. About a new method for computing in algebraic number fields. In *EuroCal'85*, volume 204 of *Lecture Notes in Computer Science*, pages 289–290, 1985.

[Del99] S. Dellière. *Triangularisation de systèmes constructibles — Application à l'évaluation dynamique.* PhD thesis, Université de Limoges, 1999.

[Día94] T. Gómez Díaz. *Quelques applications de l'évaluation dynamique.* PhD thesis, Université de Limoges, 1994.
ftp://medicis.polytechnique.fr/pub/src/dynamic_evaluation.

[Dix82] J. Dixon. Exact solution of linear equations using p-adic expansions. *Numerische Mathematik*, 40:137–141, 1982.

[DR94a] D. Duval and J.-C. Reynaud. Sketches and computation - I : Basic definition and static evaluation. In *Mathematical Structures in Computer Science*, volume 4. Cambridge University Press, 1994.

[DR94b] D. Duval and J.-C. Reynaud. Sketches and computation - II : Dynamic evaluation and applications. In *Mathematical Structures in Computer Science*, volume 4. Cambridge University Press, 1994.

[Duv87] D. Duval. *Diverses questions relatives au calcul formel avec des nombres algebriques*. PhD thesis, Université de Grenoble 1, 1987.

[Duv89] D. Duval. Simultaneous computation in fields of arbitrary characteristic. In *Computer and mathematics 89*, pages 321–326. Springer-Verlag, 1989.

[Duv94] D. Duval. Algebraic numbers: an example of dynamic evaluation. *Journal of Symbolic Computation*, 18:429–445, 1994.

[Duv95] D. Duval. Évaluation dynamique et clôture algébrique en Axiom. *Journal of pure and Applied Algebra*, 99:267–295, 1995.

[Gat86] J. von zur Gathen. Parallel arithmetic computations: a survey. In B. Rovan J. Gruska and J. Wiedermann, editors, *Proceedings of the 12th Symposium on Mathematical Foundations of Computer Science*, volume 233 of *Lecture Notes in Computer Science*, pages 93–112, Bratislava, Czechoslovakia, August 1986. Springer.

[GFT00] P. Gianni, E. Fortuna, and B. Trager. Degree reduction under specialization. Exposé à MEGA 2000, 2000.

[GHH+97] M. Giusti, K. Hägele, J. Heintz, J. E. Morais, J. L. Montaña, and L. M. Pardo. Lower bounds for Diophantine approximation. In *Proceedings of MEGA'96*, volume 117,118, pages 277–317. Journal of Pure and Applied Algebra, 1997.

[GHL+00] M. Giusti, K. Hägele, G. Lecerf, J. Marchand, and B. Salvy. Computing the dimension of a projective variety: the projective Noether Maple package. *Journal of Symbolic Computation*, 30(3):291–307, September 2000.

[GHM+98] M. Giusti, J. Heintz, J. E. Morais, J. Morgenstern, and L. M. Pardo. Straight-line programs in geometric elimination theory. *Journal of Pure and Applied Algebra*, 124:101–146, 1998.

[GHMP97] M. Giusti, J. Heintz, J. E. Morais, and L. M. Pardo. Le rôle des structures de données dans les problèmes d'élimination. *Comptes Rendus de l'Académie des Sciences de Paris*, 325:1223–1228, 1997.

[Gia87] P. Gianni. Properties of Gröbner bases under specializations. In *European Conference on Computer Algebra*, number 378 in Lecture Notes in Computer Science, pages 293–297, 1987.

[GLS01] M. Giusti, G. Lecerf, and B. Salvy. A Gröbner free alternative for polynomial system solving. *Journal of Complexity*, 17(1):154–211, 2001.

[GM89] P. Gianni and T. Mora. Algebraic solution of systems of polynomial equations using Gröbner bases. In *Applied Algebra, Algebraic Algorithms and Error Correcting Codes, Proceedings of AAECC-5*, volume 356 of *Lecture Notes in Computer Science*, pages 247–257. Springer, 1989.

[GPS01] G.-M. Greuel, G. Pfister, and H. Schönemann. Singular 2.0. A Computer Algebra System for Polynomial Computations, Centre for Computer Algebra, University of Kaiserslautern, 2001.
http://www.singular.uni-kl.de.

[Grä93] H.-G. Gräbe. On lucky primes. *Journal of Symbolic Computation*, 15:199–209, 1993.

[Häg98] K. Hägele. *Intrinsic height estimates for the Nullstellensatz.* PhD thesis, Universidad de Cantabria, Santander, 1998.

[Hei89] J. Heintz. On the computational complexity of polynomials and bilinear mappings. A survey. In *Applied Algebra, Algebraic Algorithms and Error Correcting Codes, Proceedings of AAECC-5*, volume 356 of *Lecture Notes in Computer Science*, pages 269–300. Springer, 1989.

[HKP+00] J. Heintz, T. Krick, S. Puddu, J. Sabia, and A. Waissbein. Deformation techniques for efficient polynomial equation solving. *Journal of Complexity*, 16(1), 2000.

[HMW01] J. Heintz, G. Matera, and A. Waissbein. On the time-space complexity of geometric elimination procedures. *Applicable Algebra in Engineering, Communication and Computing*, 11(4):239–296, 2001.

[JS00] G. Jeronimo and J. Sabia. Probabilistic equidimensional decomposition. *Comptes rendus de l'Académie des sciences de Paris*, 331(1), 2000.

[Kal87] M. Kalkbrener. Solving systems of algebraic equations by using Gröbner bases. In *European Conference on Computer Algebra*, number 378 in Lecture Notes in Computer Science, pages 282–292, 1987.

[Kal97] M. Kalkbrener. On stability of Gröbner bases under specializations. *Journal of Symbolic Computation*, 24(1):51–58, 1997.

[KP96] T. Krick and L. M. Pardo. A computational method for Diophantine approximation. In L. González-Vega and T. Recio, editors, *Algorithms in Algebraic Geometry and Applications. Proceedings of MEGA'94*, volume 143 of *Progress in Mathematics*, pages 193–254. Birkhäuser Verlag, 1996.

[Lan93] S. Lang. *Algebra.* Addison Wesley, 1993.

[Lec99] G. Lecerf. Kronecker, a Magma package for polynomial system solving, from 1999.
http://kronecker.medicis.polytechnique.fr.

[Lec00] G. Lecerf. Computing an equidimensional decomposition of an algebraic variety by means of geometric resolutions. In *Proceedings of ISSAC'2000*, pages 209–216. ACM, 2000.

[Lec01] G. Lecerf. *Une alternative aux méthodes de réécriture pour la résolution des systèmes algébriques*. PhD thesis, École polytechnique, 2001.

[Lev40] U. J. J. Leverrier. Sur les variations séculaires des éléments elliptiques des sept planètes principales : Mercure, Vénus, la terre, Mars, Jupiter, Saturne et Uranus. *Journal de Mathématiques Pures et Appliquées*, 4:220–254, 1840.

[LS01] G. Lecerf and É. Schost. Fast multivariate power series multiplication in characteristic zero. Manuscrit, Laboratoire GAGE, École polytechnique, France, April 2001.

[Mat86] H. Matsumura. *Commutative Ring Theory*. Cambridge University Press, 1986.

[Mor97] J. E. Morais. *Resolución eficaz de sistemas de ecuaciones polinomiales*. PhD thesis, Universidad de Cantabria, Santander, Spain, 1997.

[MS95] H. M. Möller and H. J. Stetter. Multivariate polynomial equations with multiple zeros solved by matrix eigenproblems. *Numerische Mathematik*, 70:311–329, 1995.

[Nau98] R. Nauheim. Systems of algebraic equations with bad reduction. *Journal of Symbolic Computation*, 25(5):619–641, 1998.

[Oji82] T. Ojika. Deflation algorithm for the multiple roots of simultaneous nonlinear equations. *Memoirs of Osaka Kyoiku University. III. Natural Science and Applied Science*, 30:197–209, 1982.

[Oji87] T. Ojika. Modified deflation algorithm for the the solution of singular problems. I. a system of nonlinear algebraic equations. *Journal of Mathematical Analysis and Applications*, 123:199–221, 1987.

[OWM83] T. Ojika, S. Watanabe, and T. Mitsui. Deflation algorithm for the multiple roots of a system of nonlinear equations. *Journal of Mathematical Analysis and Applications*, 96:463–479, 1983.

[Pau92] F. Pauer. On lucky ideals for Gröbner basis computation. *Journal of Symbolic Computation*, 14(5):471–482, 1992.

[Sch00] É. Schost. *Sur la résolution des systèmes polynomiaux à paramètres*. PhD thesis, École polytechnique, 2000.
http://www.gage.polytechnique.fr/schost.html.

[Ste96] H. J. Stetter. Analysis of zero clusters in multivariate polynomial systems. In *International Symposium on Symbolic and Algebraic Computation*, pages 127–136, 1996.

[Sto89] H.-J. Stoß. On the representation of rational functions of bounded complexity. *Theoretical Computer Science*, 64:1–13, 1989.

[Str72] V. Strassen. Berechnung und Programm. I, II. *Acta Informatica*, 1(4):320–355; *ibid.* 2(1), 64–79 (1973), 1972.

[Sut92] J. Sutor. *Axiom The Scientific Computation System.* Springer-Verlag, 1992.

[Tri85] W. Trinks. On improving approximate results of Buchberger's algorithm by Newton's method. In B. Caviness, editor, *Proceedings of EUROCAL'85*, number 204 in Lecture Notes in Computer Science, pages 608–611. Springer-Verlag, 1985.

[Win88] F. Winkler. A p-adic approach to the computation of Gröbner bases. *Journal of Symbolic Computation*, 6:287–304, 1988.

[Yak00] J.-C. Yakoubsohn. Finding a cluster of zeros of univariate polynomials. *Journal of Complexity*, 16, 2000.

[Zas69] H. Zassenhaus. Hensel factorization I. *Journal of Number Theory*, 1:291–311, 1969.