A multivariate Weierstrass iterative rootfinder

Olivier Ruatta
INRIA, GALAAD
BP 93,06902 Sophia-Antipolis
France
oruatta@sophia.inria.fr

ABSTRACT

We propose an algorithm to compute simultaneously all the solutions of an algebraic system (of n equations in n variables) that define a zero-dimentional variety. This new approach generalises the univariate Weierstrass's method. We study the arithmetic complexity of this method that has a quadratic convergence in a neighbourhood of the solutions. Hereafter, we describe a method based on the iteration function of the multivariate Weierstrass's method and on the continuation method for computing the roots of polynomial systems. Finally we describe some numerical experiments of those methods.

1. INTRODUCTION

There are several methods for computing simultaneously all the roots of univariate polynomials in $\mathbb{R}$ or $\mathbb{C}$ such as the Aberth's method [1], [5], Durand-Kerner's method [14] and the Weierstrass's method [23], [24], [3]. In the multivariate algebraic case, the situation is more complex. Generally, for multivariate polynomials, solvers proceed in two distinct steps:

1. Extract informations about the structure of the quotient algebra associated with the algebraic system. For instance, the structure of the vector space and some multiplication operators.

2. Solve a simpler problem. For instance an eigenvalue problem obtained thanks to the knowledge of the quotient's vector space structure and multiplication operator.

There are several efficient algorithms to perform step 1, for instance [11] and [19]. Moreover, if we are not interested by the whole structure of the quotient algebra, computations using modular arithmetic for step 1 may be accelerated in order to get a basis. But the computations for step 2 remain heavy. In this paper we propose a new algorithm for solving multivariate algebraic systems. We generalize the Weierstrass's method in the framework of polynomial systems which define zero-dimensional varieties. We follow an inverse approach in the sense that we deduce first the structure of a quotient algebra from the solutions of an equivalent system. Next we deform it and follow the solutions in order to recover the required quotient algebra. As we will see, each correction step of this algorithm consists in applying the Newton's iteration method to a set of relations. These relations link the solutions and the coefficients of the polynomials. Thanks to this iteration function which generalizes the Weierstrass's method, we use tools from homotopy theory for polynomial systems to develop a continuation method to compute simultaneously all the solutions. This method is connected with the continuation methods exposed in [8], [20], [13], [16] and [22]. For these method, one path gives one solution. With our algorithm, one path gives all the solutions.

This approach differs from usual methods used nowadays to solve polynomial systems in the following way : We suppose to know a basis of the quotient algebra's vector space structure. Then we construct a system with the same vector space structure and with known solutions (for that, we use interpolation) and we deform the multiplicative structure of the well known system to the unknown one, following the solutions of the systems. So at the end of the process, we get the wanted solutions. There are several advantages to this new method. Only few informations on the vector space structure (a monomial basis) are needed. We do not need to compute normal forms or multiplication operators here. So we can compute a basis using modular arithmetic for instance. Consequently, the step of preprossessing is simplified. It is possible to generalize the algorithm to the non complete intersection case. In the given algorithm (for complete intersection), we inverse explicitly some matrices. But in the non complete intersection, we will have rectangular linear systems that we will solve by a least square method. The other step of this algorithm remains unchanged. For a class of system those all correspond to the same vector space structure, we can obtain a dedicated algorithm. The preprocessing step of the Weierstrass method is done only one time and the solving of a system is reduced the coefficents specialisations and the numerical step.

The paper is organized in the following way. In section 2, we recall briefly some results on the classical univariate Weierstrass's method for the simultaneous computation of all roots of a polynomial. Section 3 is a theoretical description of our method and of the framework we need. In section 4, we describe an algorithm for an iteration of this method

ISSAC 2001, UWO, Canada

and we study the arithmetic complexity of this iteration. Section 5 describes a general solver that use the preceding iteration algorithm and continuation method. In section 6 , we give some numerical experiences of those two methods. And in section 7, we explain some possible improvements and open new perspectives for this kind of methods.

In what follows, $\mathbb{K}$ denotes $\mathbb{R}$ or $\mathbb{C}$, but we can imagine an adaptation for other fields.

2. THE CLASSICAL WEIERSTRASS'S METHOD

Let $P(x) = \sum_{i=0}^{d} a_{d-i}x^d = \prod_{i=1}^{d}(x-\zeta_i) \in \mathbb{K}[x]$ be a monic polynomial with all roots distinct in $\mathbb{K}$: $\zeta_1,\ldots,\zeta_d$. We denote $\zeta = (\zeta_1,\ldots,\zeta_d)$ the vector of the roots of P. For any $\mathbf{z} = (z_1,\ldots,z_d) \in \mathbb{K}^d$, let $P_{\mathbf{z}}$ be the polynomial $P_{\mathbf{z}}(x) = \prod_{i=1}^{d}(x-z_i)$ and $F(\mathbf{z},x) = P_{\mathbf{z}}(x) - P(x)$. We remark that $F(\mathbf{z},x) = 0$ if and only if there exists a permutation τ of $\{1,\ldots,d\}$ such that

$$(z_1,\ldots,z_d) = \left(\zeta_{\tau(1)},\ldots,\zeta_{\tau(d)}\right).$$

DEFINITION 2.1. *Let $\mathbf{z} \in \mathbb{K}^d$ and $i \in \{1,\ldots,d\}$, we define* $\sigma_i(\mathbf{z}) = (-1)^i \sum_{1\le j_1<\ldots<j_i\le d} z_{j_1}\ldots z_{j_i}$.

We refer to [2] for the following proposition and theorem :

PROPOSITION 2.2. *Let $\mathbf{z} \in \mathbb{K}^d$, $\mathbf{z}$ is a solution of the following system :*

$$(\Sigma)\left\{\begin{array}{l} a_i - \sigma_i(\mathbf{z}) = 0 \\ i \in \{1,\ldots,d\} \end{array}\right.$$

if and only if there exists a permutation τ of $\{1,\ldots,d\}$ such that $(z_1,\ldots,z_d) = \left(\zeta_{\tau(1)},\ldots,\zeta_{\tau(d)}\right)$.

The idea of the Weierstrass's method is to take a point $\mathbf{z}^{(0)}$ in a neighbourhood of ζ (up to a permutation) and to apply the Newton's method to the system (Σ) at $\mathbf{z}^{(0)}$. The key point is that there is an explicit inversion of the jacobian matrix of the system (Σ). So we obtain an iteration function $\mathcal{I}^W$ that defines a sequence $\mathbf{z}^{(n+1)} = I^W\left(\mathbf{z}^{(n)}\right)$ of points. They are the successive iteration points of the Newton's method. Finally we have :

THEOREM 2.3. *Let $z^{(0)} \in \mathbb{K}^n$ be in a neighbourhood of ζ up to a permutation. Then the following sequence :*

$$\mathbf{z}^{(n+1)} = I\left(\mathbf{z}^{(n)}\right) = \left(I_1\left(\mathbf{z}^{(n)}\right),\ldots,I_d\left(\mathbf{z}^{(n)}\right)\right)$$

where $I_i\left(\mathbf{z}^{(n)}\right) = z_i^{(n)} + \frac{P\left(z_i^{(n)}\right)}{\prod_{j=1,j\neq i} z_i^{(n)} - z_j^{(n)}}$ *converges quadratically to a vector formed by the roots of P.*

An old conjecture says that the Weierstrass's method (more precisely Aberth's or Durand-Kerner's methods) converges from all points excepted for a null measurement set of points. We will suggest a natural generalisation hereafter, and give a theorical and practical study of this new method.

3. A DESCRIPTION OF A MULTIVARIATE WEIERSTRASS'S METHOD

We consider a set of polynomials $\{f_1,\ldots,f_n\}$ of $\mathbb{K}[x_1,\ldots,x_n]$, and we want to compute the points of the variety defined by those polynomials. We will assume that this variety is a finite set of points. This assumption gives very strong algebraic structures that will be described first. After that we will show how this allows us to generalise the Weierstrass's method and we will describe this method.

3.1 Hypothesis, notations and properties

We denote by $(f_1,\ldots,f_n)$ the ideal that they define, by $\mathcal{A} = \mathbb{K}[x_1,\ldots,x_n]/(f_1,\ldots,f_n)$ the quotient algebra by this ideal and by $Z(f_1,\ldots,f_n) = \{\mathbf{z} \in \mathbb{K}^n \,|\, f_1(\mathbf{z}) = \cdots = f_n(\mathbf{z}) = 0\} \subset \mathbb{K}^n$ the variety defined by those polynomials. We define now the framework that we will use for our generalisation.

HYPOTHESIS 3.1. *We assume that $Z(F)$ is a finite set $\{\zeta_1, \ldots,\zeta_D\}$ of simple roots (under this assumption, $\mathcal{A}$ is a finite $\mathbb{K}$-vector space of dimension D, see [10] and [21]).*

PROPOSITION 3.2 ([21] AND [10]). *Under the hypothesis (3.1), there is a family of polynomials $(\mathbf{e}_{\zeta_i})_{i=1,\ldots,D}$ such that for i and $j \in \{1,\ldots,D\}$:*

i) $\mathbf{e}_{\zeta_i}(\zeta_j) = \delta_{i,j}$,

ii) $x_j\mathbf{e}_{\zeta_i} = \zeta_{i,j}\mathbf{e}_{\zeta_i}$, *where* $\zeta_i = (\zeta_{i,1},\ldots,\zeta_{i,n})$,

iii) $\mathbf{e}_{\zeta_i}\mathbf{e}_{\zeta_j} = \delta_{i,j}\mathbf{e}_{\zeta_i}$ *if* $i \neq j$,

iv) $\mathbf{e}_{\zeta_1} + \ldots + \mathbf{e}_{\zeta_D} = 1$.

The polynomial $\mathbf{e}_{\zeta_i}$ is called the idempotent associated with ζ_i and this set of polynomials forms a basis of $\mathcal{A}$, i.e.:

$$\mathcal{A} = \mathbb{K}\mathbf{e}_{\zeta_1} \oplus \cdots \oplus \mathbb{K}\mathbf{e}_{\zeta_D}.$$

We have the following projection formula:

$$\forall P \in \mathcal{A}, P(\mathbf{x}) = \sum_{i=1}^{D} P(\zeta_i)\,\mathbf{e}_{\zeta_i}(\mathbf{x}).$$

This theoretical framework is clearly similar to the univariate case.

We denote by $\widehat{\mathcal{A}}$ the dual space of $\mathcal{A}$ as $\mathbb{K}$ vector space, that is to say $\widehat{\mathcal{A}} = \mathrm{Hom}_{\mathbb{K}}(\mathcal{A},\mathbb{K})$. Let $\zeta \in Z(f_1,\cdots,f_n)$. The dual basis of the idempotent $\mathbf{e}_\zeta$ is the evaluation at ζ denoted by 1_ζ and such that $\forall P \in \mathcal{A}$, $1_\zeta(P) = P(\zeta)$. We know that $\widehat{\mathcal{A}}$ is a $\mathbb{K}$ vector space of dimension D. We have the following proposition (see [17]):

PROPOSITION 3.3. *We have* $\widehat{\mathcal{A}} = < 1_{\zeta_1},\cdots,1_{\zeta_D} >$.

3.2 The representation of generic algebra

HYPOTHESIS 3.4. *We assume that we know a set $E = \{\alpha_1, \ldots,\alpha_D\} \subset \mathbb{N}^n$ such that $(x^\alpha)_{\alpha\in E}$ is a basis of $\mathcal{A}$ a $\mathbb{K}$-vector space. This basis can be computed from a Grobner basis using modular arithmetic for instance.*

REMARK 3.5. *Furthermore, we assume that we know the normal forms of a set of polynomials $\{Q_1,\ldots,Q_m\}$, with $m \ge n$ and such that Q_i contains a monomial that is not in*

$\{\mathbf{x}^{\alpha}\}_{\alpha\in E}$ *for all* $i \in \{1,\dots,n\}$*, those polynomials are supposed to be linearly independent. So we know the representant of the class of* Q_i *in* $\mathcal{A}$ *expressed in the basis* $(x^{\alpha})_{\alpha\in E}$ *for all* $i\in\{1,\dots,n\}$*. If we denote* $N_{Q_i}(\mathbf{x}) = \sum_{j=1}^{D} a_{i,j}\mathbf{x}^{\alpha_j}$ *the representant of* Q_i*, then* $Q_i - N_{Q_i}(\mathbf{x}) \in (f_1,\cdots,f_n)$*. For instance, we have* $N_{f_i}=0$*, for all* $i\in\{1,\dots,n\}$*.*

We denote by $\mathbb{K}[x_1,\dots,x_n]_E$ the set of the polynomials $P\in\mathbb{K}[x_1,\dots,x_n]$ which have as monomial support $\subset (\mathbf{x}^{\alpha})_{\alpha\in E}$, i.e. $P(\mathbf{x}) = \sum_{i=0}^{D} p_i x^{\alpha_i}$. We will give some representations and properties of the vector space $\mathbb{K}[x_1,\dots,x_n]_E$.

Let $\mathbf{z}=\{\mathbf{z}_1,\dots,\mathbf{z}_D\}$ be distinct points in $\mathbb{K}^D$. Let us denote by $\mathcal{A}_{\mathbf{z}}$ the coordinates algebra of $\mathbf{z}$ and let us assume that $\mathcal{A}_{\mathbf{z}}$ admits $(\mathbf{x}^{\alpha})_{\alpha\in E}$ as a $\mathbb{K}$-vector space basis. We denote by $(\mathbf{e}_i(\mathbf{z},\mathbf{x}))_{1\in\{1,\dots,D\}}$, the basis of idempotents of $\mathcal{A}_{\mathbf{z}}$.

DEFINITION 3.6. *We define the pseudo Vandermonde's determinant associated to* $\mathbf{z}$ *as*

$$V_E(\mathbf{z}) = \begin{vmatrix} \mathbf{z}_1^{\alpha_1} & \dots & \mathbf{z}_1^{\alpha_D} \\ \vdots & \ddots & \vdots \\ \mathbf{z}_D^{\alpha_1} & \dots & \mathbf{z}_D^{\alpha_D} \end{vmatrix}$$

PROPOSITION 3.7. *Let* $E\subset\mathbb{N}^n$*, then* $(\mathbf{x}^{\alpha})_{\alpha\in E}$ *is a basis of* $\mathcal{A}_{\mathbf{z}}$ *if and only if* $V_E(\mathbf{z})\neq 0$*.*

PROOF. We saw in proposition 3.3 that $(1_{\zeta_i})_{i\in\{1,\cdots,D\}}$ is a basis of $\widehat{\mathcal{A}}$. So the set $(\mathbf{x}^{\alpha_j})_{j\in\{1,\cdots,D\}}$ is a basis of $\mathcal{A}$ if and only if the matrix $(1_{\zeta_i}(\mathbf{x}^{\alpha_j}))_{i,j\in\{1,\cdots,D\}}$ has an inverse matrix, and the proposition is proved. $\square$

We consider $\mathbb{K}^D$ with its canonical basis $v_i = (\delta_{i,1},\dots,\delta_{i,D})^t$, for $i\in\{1,\dots,D\}$. We detail now the isomorphism:

$$\Phi : \left\{ \begin{array}{ccc} \mathbb{K}[x_1,\dots,x_n]_E & \longrightarrow & \mathbb{K}^D \\ b(\mathbf{x}) = \sum_{i=0}^{D} b_i\mathbf{x}^{\alpha_i} & \longmapsto & \vec{b} = \sum_{i=0}^{D} b_i v_i \end{array}\right.$$

REMARKS 3.8.

- *In this case, the family* $(\mathbf{w}_i)_{i=1,\dots,D}$ *forms a basis of* $\mathbb{K}^D$*, where* $\mathbf{w}_i = \Phi(\mathbf{e}_i(\mathbf{z},\mathbf{x}))$*.*
- *Let* $P(\mathbf{x}) = \sum_{i=1}^{D} P(\mathbf{z}_i)\,\mathbf{e}_i(\mathbf{z},\mathbf{x})$*, then* $\Phi(P) = \sum_{i=1}^{D} P(\mathbf{z}_i)\,\mathbf{w}_i$*.*

DEFINITION 3.9. *For each* $i\in\{1,\dots,D\}$ *we define the pseudo Lagrange's polynomial associated with* $\mathbf{z}_i$ *(this is exactly the idempotent associated with* $\mathbf{z}_i$*) by:*

$$\mathbf{e}_i(\mathbf{z},\mathbf{x}) = -\begin{vmatrix} \mathbf{x}^{\alpha_1} & \dots & \mathbf{x}^{\alpha_D} \\ \mathbf{z}_1^{\alpha_1} & \dots & \mathbf{z}_1^{\alpha_D} \\ \vdots & & \vdots \\ \mathbf{z}_{i-1}^{\alpha_1} & \dots & \mathbf{z}_{i-1}^{\alpha_D} \\ \mathbf{z}_{i+1}^{\alpha_1} & \dots & \mathbf{z}_{i+1}^{\alpha_D} \\ \vdots & & \vdots \\ \mathbf{z}_D^{\alpha_1} & \dots & \mathbf{z}_D^{\alpha_D} \end{vmatrix} / V_E(\mathbf{z}).$$

It is easy to check that the pseudo Lagrange polynomials are the idempotent by checking that they have the following properties which characterize the idempotents:

PROPERTIES 3.10.

- $\mathbf{e}_i(\mathbf{z},\mathbf{z}_j) = 0$ *if* $i\neq j$ *for* i *and* $j\in\{1,\dots,D\}$*.*
- $\mathbf{e}_i(\mathbf{z},\mathbf{z}_i) = 1$ *for* $i\in\{1,\dots,D\}$*.*
- $\mathbf{e}_i(\mathbf{z},\mathbf{x}) \in \mathbb{K}[x_1,\dots,x_n]_E$*.*

We have a $\mathbb{K}$-vector space isomorphism between $\mathcal{A}_{\mathbf{z}}$ and $\mathbb{K}[x_1,\dots,x_n]_E$ which associates to each class in $\mathcal{A}_{\mathbf{z}}$ the representant of the class which has $\{\mathbf{x}^{\alpha}\}_{\alpha\in E}$ as monomial support.

3.3 Relations

We will introduce some new polynomials, that we call relations. They are given by determinantal expressions. Those polynomials are useful tools to obtain polynomials in the ideal associated with the algebraic system.

Let Q be a polynomial that contains a monomial that is not in the given basis, we denote by $N_Q(\mathbf{x}) = \sum_{\alpha\in E}\lambda_{Q,\alpha}x^{\alpha}$ the normal form of the polynomial Q in the basis $(\mathbf{x}^{\alpha})_{\alpha\in E}$ of $\mathcal{A}$. We consider the polynomial $P_Q(\mathbf{x}) = Q(\mathbf{x}) - N_Q(\mathbf{x})$. Then, the polynomial $P_Q\in(f_1,\cdots,f_n)$.

DEFINITION 3.11. *Let* $\mathbf{z}\in(\mathbb{K}^n)^D$*, we define the polynomial* $R_Q(\mathbf{z},\mathbf{x})$ *as the following determinant:*

$$R_Q(\mathbf{z},\mathbf{x}) = (-1)^D \begin{vmatrix} \mathbf{x}^{\alpha_1} & \dots & \mathbf{x}^{\alpha_D} & Q(\mathbf{x}) \\ \mathbf{z}_1^{\alpha_1} & \dots & \mathbf{z}_1^{\alpha_D} & Q(\mathbf{z}_1) \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{z}_D^{\alpha_1} & \dots & \mathbf{z}_D^{\alpha_D} & Q(\mathbf{z}_D) \end{vmatrix} \tag{1}$$

It is easy to check the following properties:

PROPERTIES 3.12.

- $R_Q(\mathbf{z},\mathbf{z}_i) = 0$,
- $R_Q(\zeta,\mathbf{x}) = V_E(\zeta)\,P_Q(\mathbf{x})$.

Now, we will define a function associated to a relation.

DEFINITION 3.13. *We define the following polynomial:*

$$F_Q(\mathbf{z},\mathbf{x}) = \frac{R_Q(\mathbf{z},\mathbf{x})}{V_E(\mathbf{z})} - P_Q(\mathbf{x}) \tag{2}$$

This function has an interesting property:

PROPOSITION 3.14. *Let* $Q_1,\dots,Q_m$ *be polynomials such that* $(P_{Q_1},\dots,P_{Q_m}) = (f_1,\dots,f_n)$*, then* $F_{Q_i}(\mathbf{z},\mathbf{x}) = 0$ *for* $i\in\{1,\dots,m\}$ *if and only if* $\{\mathbf{z}_1,\dots,\mathbf{z}_D\} = \{\zeta_1,\dots,\zeta_D\}$*.*

PROOF. We have $F_{Q_i}(\zeta,\mathbf{x})\in\mathbb{K}[x_1,\dots,x_n]_E$, the $F_{Q_i}(\zeta,\mathbf{x}) = \sum_{j=1}^{D} F_{Q_i}(\zeta,\zeta_j)\,\mathbf{e}_{\zeta_j}(\mathbf{x})$. But $F_{Q_i}(\zeta,\zeta_j) = \frac{R_{Q_i}(\zeta,\zeta_j)}{V_E(\mathbf{z})} - P_{Q_i}(\zeta_j)$ with $R_{Q_i}(\zeta,\zeta_j) = 0$ and $P_{Q_i}(\zeta_j) = 0$, then $F_{Q_i}(\zeta,\mathbf{x}) = 0$. Reciprocally, if $F_{Q_i}(\mathbf{z},\mathbf{x}) = 0$ for all $i\in\{1,\dots,D\}$, it implies that $\sum_{j=1}^{D} F_{Q_i}(\mathbf{z},\mathbf{z}_j)\,\mathbf{e}_j(\mathbf{z},\mathbf{x}) = 0$ and so $\frac{R_{Q_i}(\mathbf{z},\mathbf{z}_j)}{V_E(\mathbf{z})} - P_{Q_i}(\mathbf{z}_j) = 0$ for all $i\in\{1,\cdots,n\}$ and $j\in\{1,\cdots,D\}$. But $R_{Q_i}(\mathbf{z},\mathbf{z}_j) = 0$ for all i and j, then $P_{Q_i}(\mathbf{z}_j) = 0$ and because $(P_{Q_1},\cdots,P_{Q_n}) = (f_1,\dots,f_n)$, we have $(\mathbf{z}_1,\cdots,\mathbf{z}_D) = (\zeta_1,\cdots,\zeta_D)$. $\square$

We will describe the links between the different objects introduced above.

3.4 Links between relations and idempotents

We denote $\mu_i = (\delta_{i,j})_{j=1,\ldots,n} \in \mathbb{N}^n$. Let $Q \notin \mathbb{K}[x_1,\ldots,x_n]_E$. The partial derivative function of F_Q relatively to the coordinate of $\mathbf{z}_i$, $i=1,\ldots,D$, have a strong property described in the following proposition.

Proposition 3.15. *We have:*

$$\frac{\partial}{\partial z_{i,j}} F_Q(\mathbf{z},\mathbf{x}) = -\frac{\frac{\partial R_Q}{\partial \mathbf{x}_j}(\mathbf{z},\mathbf{z}_i)}{V_E(\mathbf{z})}\mathbf{e}_i(\mathbf{z},\mathbf{x}) \qquad (3)$$

Proof. We consider the determinantal expression of this polynomial:

$$\frac{\partial}{\partial z_{i,j}} R_Q(\mathbf{z},\mathbf{x}) = (-1)^D \frac{\partial}{\partial z_{i,j}} \begin{vmatrix} \mathbf{x}^{\alpha_1} & \ldots & \mathbf{x}^{\alpha_D} & Q(\mathbf{x}) \\ \mathbf{z}_1^{\alpha_1} & \ldots & \mathbf{z}_1^{\alpha_D} & Q(\mathbf{z}_1) \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{z}_D^{\alpha_1} & \ldots & \mathbf{z}_D^{\alpha_D} & Q(\mathbf{z}_D) \end{vmatrix}.$$

So we have:

$$\frac{\partial}{\partial z_{i,j}} R_Q(\mathbf{z},\mathbf{x}) = \begin{vmatrix} \mathbf{x}^{\alpha_1} & \ldots & \mathbf{x}^{\alpha_D} & Q(\mathbf{x}) \\ \vdots & & \vdots & \vdots \\ \mathbf{z}_{i-1}^{\alpha_1} & \ldots & \mathbf{z}_{i-1}^{\alpha_D} & Q(\mathbf{z}_{i-1}) \\ \alpha_{1,j}\mathbf{z}_i^{\alpha_1-\mu_j} & \ldots & \alpha_{D,j}\mathbf{z}_i^{\alpha_D-\mu_j} & \frac{\partial Q}{\partial x_j}(\mathbf{z}_i) \\ \mathbf{z}_{i+1}^{\alpha_1} & \ldots & \mathbf{z}_{i+1}^{\alpha_D} & Q(\mathbf{z}_{i+1}) \\ \mathbf{z}_D^{\alpha_1} & \ldots & \mathbf{z}_D^{\alpha_D} & Q(\mathbf{z}_D) \end{vmatrix}.$$

Then, with this expression we remark that $\frac{\partial}{\partial z_{i,j}} R_Q(\mathbf{z},\mathbf{z}_j) = 0$ for all $j \neq i$, and that $\frac{\partial}{\partial z_{i,j}} R_Q(\mathbf{z},\mathbf{z}_i) \neq 0$. So, $\frac{\partial}{\partial z_{i,j}} R_Q(\mathbf{z},\mathbf{x})$ is a multiple of the idempotent $\mathbf{e}_{\zeta_i}$. If we permute the line indexed by i with the first line we obtain :

$$\frac{\partial R_Q}{\partial z_{i,j}}(\mathbf{z},\mathbf{z}_i) = -\begin{vmatrix} \alpha_{1,j}\mathbf{z}_i^{\alpha_1-\mu_j} & \ldots & \alpha_{D,j}\mathbf{z}_i^{\alpha_D-\mu_j} & \frac{\partial Q}{\partial x_j}(\mathbf{z}_i) \\ \vdots & & \vdots & \vdots \\ \mathbf{z}_{i-1}^{\alpha_1} & \ldots & \mathbf{z}_{i-1}^{\alpha_D} & Q(\mathbf{z}_{i-1}) \\ \mathbf{z}_i^{\alpha_1} & \ldots & \mathbf{z}_i^{\alpha_D} & Q(\mathbf{z}_i) \\ \mathbf{z}_{i+1}^{\alpha_1} & \ldots & \mathbf{z}_{i+1}^{\alpha_D} & Q(\mathbf{z}_{i+1}) \\ \vdots & & \vdots & \vdots \\ \mathbf{z}_D^{\alpha_1} & \ldots & \mathbf{z}_D^{\alpha_D} & Q(\mathbf{z}_D) \end{vmatrix}.$$

The proposition can be deduced directly now with the projection formula applied to $\mathcal{F}_Q$. □

3.5 Construction of a new system

Let $Q_1,\ldots,Q_n \in \mathbb{K}[x_1,\ldots,x_n] - \mathbb{K}[x_1,\ldots,x_n]_E$. We define the function

$$\mathcal{F} : \begin{cases} (\mathbb{K}^n)^D & \longrightarrow \quad (\mathbb{K}[x_1,\ldots,x_n]_E)^n \\ \mathbf{z} & \longmapsto \quad \mathcal{F}(\mathbf{z},\mathbf{x}) = \begin{pmatrix} F_{Q_1}(\mathbf{z},\mathbf{x}) \\ \vdots \\ F_{Q_n}(\mathbf{z},\mathbf{x}) \end{pmatrix} \end{cases} \qquad (4)$$

We can associate to the equations $\mathcal{F}(\mathbf{z},\mathbf{x}) = 0$ a system that we will describe now. If we look at the polynomials $F_{Q_i}(\mathbf{z},\mathbf{x}) = \frac{R_{Q_i}(\mathbf{z},\mathbf{x})}{V(\mathbf{z})} - P_{Q_i}(\mathbf{x})$ we can remark that this polynomial has $(\mathbf{x}^\alpha)_{\alpha\in E}$ as monomial support, for all $i \in \{1,\ldots,n\}$. We have $F_{Q_i}(\mathbf{z},\mathbf{x}) = \sum_{\alpha\in E}(\psi_{i,\alpha}(\mathbf{z}) - \lambda_{i,\alpha})\mathbf{x}^\alpha$, for all $i \in \{1,\ldots,n\}$. Then we obtain the following system:

$$(\Psi) : \begin{cases} \psi_{i,\alpha}(\mathbf{z}) - \lambda_{i,\alpha} = 0 \\ \forall\alpha \in E, \forall i \in \{1,\ldots,n\} \end{cases} \qquad (5)$$

Then, this is a $nD \times nD$ square system. By construction we have $\psi_{i,\alpha}(\zeta) - \lambda_{i,\alpha} = 0$, $\forall\alpha \in E$, $\forall i \in \{1,\ldots,n\}$. The idea is again to apply the Newton method to (Ψ). As in the univariate case, we will exploit the structure of the problem in order to invert explicitly the jacobian and give a simple iteration function.

3.6 Computation of an iteration function

We consider the map $\Lambda : \mathbb{K}[x_1,\ldots,x_n]_E^n \longrightarrow (\mathbb{K}^D)^n$ defined by $(P_1,\ldots,P_n) \longmapsto (\Phi(P_1),\ldots,\Phi(P_n))$. We define $\overrightarrow{\mathcal{F}}$:

$$\begin{cases} (\mathbb{K}^n)^D & \longrightarrow \quad (\mathbb{K}^D)^n \\ \mathbf{z} & \longmapsto \quad \overrightarrow{\mathcal{F}}(\mathbf{z}) = (\Phi(F_{Q_1}(\mathbf{z},\mathbf{x})),\ldots,\Phi(F_{Q_n}(\mathbf{z},\mathbf{x}))) \end{cases}$$

By proposition 3.14, $\overrightarrow{\mathcal{F}}(\mathbf{z}) = 0$ if and only if $\mathbf{z} = \zeta$.

So we apply the Newton's method to $\overrightarrow{\mathcal{F}}$ and to do that we need to compute the sequence of Newton iteration points:

$$\mathbf{z}^{(k+1)} = \mathbf{z}^{(k)} - J_{\overrightarrow{\mathcal{F}}}\left(\mathbf{z}^{(k)}\right)^{-1}\overrightarrow{\mathcal{F}}\left(\mathbf{z}^{(k)}\right).$$

Theorem 3.16. *In the above situation, to apply a Newton's iteration at $\overrightarrow{\mathcal{F}}$, for $i \in \{1,\ldots,D\}$, we only need to compute*

$$\mathbf{z}_i^{(k+1)} = \mathbf{z}_i^{(k)} - \begin{pmatrix} \frac{\frac{\partial R_{Q_1}}{\partial x_1}\left(\mathbf{z}^{(k)},\mathbf{z}_i^{(k)}\right)}{V\left(\mathbf{z}^{(k)}\right)} & \ldots & \frac{\frac{\partial R_{Q_1}}{\partial x_n}\left(\mathbf{z}^{(k)},\mathbf{z}_i^{(k)}\right)}{V\left(\mathbf{z}^{(k)}\right)} \\ \vdots & \ddots & \vdots \\ \frac{\frac{\partial R_{Q_n}}{\partial x_1}\left(\mathbf{z}^{(k)},\mathbf{z}_i^{(k)}\right)}{V\left(\mathbf{z}^{(k)}\right)} & \ldots & \frac{\frac{\partial R_{Q_n}}{\partial x_n}\left(\mathbf{z}^{(k)},\mathbf{z}_i^{(k)}\right)}{V\left(\mathbf{z}^{(k)}\right)} \end{pmatrix}^{-1} \begin{pmatrix} P_{Q_1}\left(\mathbf{z}_i^{(k)}\right) \\ \vdots \\ P_{Q_n}\left(\mathbf{z}_i^{(k)}\right) \end{pmatrix} \qquad (6)$$

And the sequence $\left(\mathbf{z}^{(k)}\right)_{k\in\mathbb{N}}$ converge quadraticaly in a neighbourhood of the solutions of the system.

Remark 3.17. *Because of the linearity of Φ we have $\frac{\partial}{\partial z_{i,j}}\Phi(F_{Q_k}(\mathbf{z})) = \Phi\left(\frac{\partial}{\partial z_{i,j}}F_{Q_k}(\mathbf{z})\right)$ for j and $k \in \{1,\ldots,n\}$ and $i \in \{1,\ldots,D\}$.*

Proof. Let us solve $J_{\overrightarrow{\mathcal{F}}}(\mathbf{z})\mathbf{u} = \overrightarrow{\mathcal{F}}(\mathbf{z})$. Thanks to proposition (3.15), for all $k \in \{1,\ldots,n\}$ we have

$$\Phi\left(\sum_{i=1}^{D}\sum_{j=1}^{n} -\frac{\frac{\partial R_{Q_k}}{\partial x_j}(\mathbf{z},\mathbf{z}_i)}{V_E(\mathbf{z})}u_{i,j}\mathbf{e}_i(\mathbf{z},\mathbf{x})\right) = \Phi\left(\sum_{i=1}^{D} F_{Q_k}(\mathbf{z},\mathbf{z}_i)\mathbf{e}_i(\mathbf{z},\mathbf{x})\right).$$

Then we have

$$\sum_{i=1}^{D}\sum_{j=1}^{n} -\frac{\frac{\partial R_{Q_k}}{\partial z_{i,j}}(\mathbf{z},\mathbf{z}_i)}{V_E(\mathbf{z})}u_{i,j}\mathbf{w}_i = \sum_{i=1}^{D} F_{Q_k}(\mathbf{z},\mathbf{z}_i)\mathbf{w}_i.$$

So for each $i \in \{1, \ldots, D\}$ we have the following system

$$\left\{ \begin{array}{l} \displaystyle\sum_{j=1}^{n} - \frac{\frac{\partial R_{Q_k}}{\partial x_j}(\mathbf{z}, \mathbf{z}_i)}{V_E(\mathbf{z})} u_{i,j} = F_{Q_k}(\mathbf{z}, \mathbf{z}_i) \\ \quad \forall k \in \{1, \ldots, n\} \end{array} \right. \tag{7}$$

Then for each $i \in \{1, \ldots, D\}$ we define

$$\Delta_{\mathbf{z}_i} \overrightarrow{\mathcal{F}}(\mathbf{z}) = \begin{pmatrix} -\frac{\frac{\partial R_{Q_1}}{\partial x_1}(\mathbf{z},\mathbf{z}_i)}{V_E(\mathbf{z})} & \cdots & -\frac{\frac{\partial R_{Q_1}}{\partial x_n}(\mathbf{z},\mathbf{z}_i)}{V_E(\mathbf{z})} \\ \vdots & \ddots & \vdots \\ -\frac{\frac{\partial R_{Q_n}}{\partial x_1}(\mathbf{z},\mathbf{z}_i)}{V_E(\mathbf{z})} & \cdots & -\frac{\frac{\partial R_{Q_n}}{\partial x_n}(\mathbf{z},\mathbf{z}_i)}{V_E(\mathbf{z})} \end{pmatrix}$$

Following these definitions and since $F_{Q_k}(\mathbf{z}, \mathbf{z}_i) = -P_{Q_k}(\mathbf{z}_i)$, the system (7) can be written

$$\begin{pmatrix} u_{i,1} \\ \vdots \\ u_{i,n} \end{pmatrix} = \Delta_{\mathbf{z}_i} \overrightarrow{\mathcal{F}}(\mathbf{z})^{-1} \begin{pmatrix} P_{Q_1}(\mathbf{z}, \mathbf{z}_i) \\ \vdots \\ P_{Q_n}(\mathbf{z}, \mathbf{z}_i) \end{pmatrix}$$

The formula of the theorem is proved. The covergence of the method come from the facts that this method is equivalent to a Newton method and that all roots assumed to be simples. □

REMARK 3.18. *If there is only one root to the ssytem, then the formula 6 is the Newton's iteration formula, and if there is only one variable, the formula 6 is the formula of the classical Weierstrass's method.*

4. ALGORITHM AND ARITHMETIC COMPLEXITY

Recall that we assume to know a monomial basis of the algebra $\mathcal{A}$ as $\mathbb{K}$-vector space and the normal forms in this basis of n polynomials that are not in this basis. We use the notations of the preceding sections.

4.1 Algorithm

The starting points are given thanks to a matrix $\mathbf{z}$ that has its column equal to the vector of the coordinates of the starting points.

ALGORITHM 4.1 (WEIERSTRASS'S ITERATION).

- INPUT *The matrix $\mathbf{z}$ formed with the coordinates of the initial points.*
- STEP 1 *We compute the matrices (specialized at $\mathbf{z}$)*

$$MV_E(\mathbf{z}) = \begin{pmatrix} \mathbf{z}_1^{\alpha_1} & \cdots & \mathbf{z}_1^{\alpha_D} \\ \vdots & \ddots & \vdots \\ \mathbf{z}_D^{\alpha_1} & \cdots & \mathbf{z}_D^{\alpha_D} \end{pmatrix}$$

$$MR_{Q_k}(\mathbf{z}, \mathbf{x}) = \begin{pmatrix} \mathbf{z}_1^{\alpha_1} & \cdots & \mathbf{z}_1^{\alpha_D} & Q_1(\mathbf{z}_1) \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{z}_D^{\alpha_1} & \cdots & \mathbf{z}_D^{\alpha_D} & Q_n(\mathbf{z}_D) \end{pmatrix}$$

  for $k = 1, \cdots, n$. After we compute the rows:

$$dv_{Q_k}^{(i)}(\mathbf{x}) = \left(\frac{\partial}{\partial x_i} x^{\alpha_1}, \ldots, \frac{\partial}{\partial x_i} x^{\alpha_D}, \frac{\partial}{\partial x_i} Q_k(\mathbf{x}) \right)$$

  for $i \in \{1, \cdots, n\}$ and $k \in \{1, \cdots, n\}$.

- STEP 2 *Make the LU-decomposition of $MV_E(\mathbf{z})$. Use it to compute the kernels of the matrices MR_{Q_k}. This gives us the vectors $r_{Q_k} \in \mathbb{K}^{D+1}$, for each $i \in \{1, \cdots, n\}$. If the dimension of the kernel of $MV_E(\mathbf{z})$ does not vanish, stop the algorithm with an error setting.*
- STEP 3 *For each $i \in \{1, \cdots, D\}$, we construct the matrices $\Delta_{\mathbf{z}_i} \overrightarrow{\mathcal{F}}(\mathbf{z})$ by computing their coefficients as follows: we specialize $dv_{Q_k}^{(i)}$ at $\mathbf{z}_i$, compute its product with $r_{\beta_k}(\mathbf{z})$ and normalize it by its last coefficient (that is the value of the pseudo Vandermonde determinant), for each $k \in \{1, \cdots, n\}$. Then we obtain: $\Delta_{\mathbf{z}_i} \overrightarrow{\mathcal{F}}(\mathbf{z})$.*
- STEP 4 *For each $i \in \{1, \cdots, D\}$ we solve the system*

$$(P_{Q_1}(\mathbf{z}_i), \cdots, P_{Q_n}(\mathbf{z}_i))^t = \Delta_{\mathbf{z}_i} \overrightarrow{\mathcal{F}}(\mathbf{z}) \mathbf{z}'_i,$$

  where $\mathbf{z}'_i$ is the column indexed by i of the output matrix.

- OUTPUT *The matrix $\mathbf{z}'$.*

4.2 Arithmetic complexity for an iteration

PROPOSITION 4.2. *The arithmetic complexity of an iteration of the algorithm (4.1) is bounded by $\mathcal{O}(D^3 + n^2 D^2 + Dn^3)$ arithmetic operations.*

PROOF. We detail here the cost of the different steps:

- In step 1: We begin by making n specialization of $D \times (D+1)$, this gives us $\mathcal{O}(nD^2)$.
- In step 2 and 3, we compute the LU-decomposition the matrix $MV_E(\mathbf{z})$, this gives a complexity $\mathcal{O}(D^3)$. Finally we use this decomposition to compute the kernels of the n matrices MR_{Q_k}, this cost $\mathcal{O}(n * D^2)$ arithmetic operations. We specialize nD vectors of length $D+1$, this gives a $\mathcal{O}(nD^2)$ complexity. After we compute $n^2 D$ inner products of vectors of length $D+1$, we obtain a complexity term $\mathcal{O}(n^2 D^2)$.
- In step 4: We specialize D vectors of length n, this gives us a complexity $\mathcal{O}(nD)$. We solve D linear systems of size $n \times n$, this costs $\mathcal{O}(Dn^3)$ arithmetic operations. Finally we compute n products of $n \times n$ matrices by vectors and we have a complexity $\mathcal{O}(n^2)$. If we denote by L the maximal cost of the evaluation of a P_{Q_k} there is also a contribution in $\mathcal{O}(nL)$ arithmetic operations, but this contribution is marginal.

With all these elements we obtain a $\mathcal{O}(D^3 + n^2 D^2 + Dn^3)$ arithmetic complexity. □

REMARK 4.3. *In step 2, we do not use the pseudo Vandermonde's structure of the matrices $MV_E(\mathbf{z})$. However the use of this structure could give a better complexity. We refer to [18] for the use of the matrix structure for solving systems.*

5. A GENERAL SOLVER BASED ON MULTIVARIATE WEIERSTRASS'S ITERATION

We propose now a new approach by combining the multivariate Weierstrass's method and a continuation method as described in [20] or [8].

Let $\mathbf{z}^{(0)} \in (\mathbb{K}^n)^D$ such as the quotient algebra is defined by n relations $R_{Q_1}\left(\mathbf{z}^{(0)}, \mathbf{x}\right), \cdots, R_{Q_n}\left(\mathbf{z}^{(0)}, \mathbf{x}\right)$, i.e., if we denote $\mathcal{R}$ the ideal generated by those relations, and by $\mathcal{A}_{\mathbf{z}^{(0)}}$ the coordinates algebra $\mathbb{K}[x_1, \ldots, x_n]/\mathcal{R}$. We assume that $\mathcal{A}_{\mathbf{z}^{(0)}}$ is isomorphic to $\mathbb{K}[x_1, \ldots, x_n]_E$ as a $\mathbb{K}$-vector space. The aim is to deform the system defined by those relations into the initial system.

Definition 5.1. *We define the following application :*

$$\mathcal{H} : \begin{cases} [0,1] \times (\mathbb{K}^n)^D & \rightarrow \mathbb{K}[x_1, \ldots, x_n]^n \\ (t, \mathbf{z}) & \mapsto \mathcal{H}(t, \mathbf{z}) = \begin{pmatrix} tf_1(\mathbf{x}) + (1-t) R_{Q_1}(\mathbf{z}, \mathbf{x}) \\ \vdots \\ tf_n(\mathbf{x}) + (1-t) R_{Q_n}(\mathbf{z}, \mathbf{x}) \end{pmatrix} \end{cases}$$

Remark 5.2. *For all $i \in \{1, \cdots, D\}$, $\mathbf{z}_i^{(0)}$ is a solution of $\mathcal{H}\left(0, \mathbf{z}^{(0)}\right) = 0$. Then for Δt small enough, each point $\mathbf{z}_i$ ($i \in \{1, \cdots, D\}$) must be an approximation of a solution of the system $\mathcal{H}\left(\Delta t, \mathbf{z}^{(0)}\right) = 0$.*

The idea of this algorithm is to use this remark in order to apply the multivariate Weierstrass's iteration function to $\mathcal{H}\left(\Delta t, \mathbf{z}^{(0)}\right) = 0$. Then obtain an approximation of the solution of this system, denoted $\mathbf{z}^{(1)}$. So $\mathbf{z}^{(1)}$ must be an approximation of the solution of $\mathcal{H}\left(2\Delta t, \mathbf{z}^{(0)}\right) = 0$, and so we can apply the multivariate Weierstrass's iteration again, and so on. Finally we will get approximation of the solutions of $\mathcal{H}\left(1, \mathbf{z}^{(0)}\right) = 0$, that is to say an approximation of the solutions we sought. In that method, Δt must be small enough. We suppose we have such a value for Δt with $\frac{1}{\Delta t} \in \mathbb{N}$.

Remark 5.3. *We denote by $\mathcal{R}^{(i)}$ the ideal generated by $j\Delta t f_j(x) + (1 - j\Delta t) R_{Q_j}\left(\mathbf{z}^{(0)}, \mathbf{x}\right)$, for $j \in \{1, \cdots, n\}$. We assume that at each step $\mathbb{K}[x_1, \ldots, x_n]/\mathcal{R}^{(i)}$ is isomorphic to $\mathbb{K}[x_1, \ldots, x_n]_E$ as a $\mathbb{K}$-vector space. Then $\mathcal{H}$ is an application from $[0,1] \times (\mathbb{K}^n)^D$ to $\mathbb{K}[x_1, \ldots, x_n]_E^n$.*

Under this assumption we have the following algorithm :

Algorithm 5.4.

- Input : Δt, $\mathcal{H}\left(t, \mathbf{z}^{(0)}\right)$.
- *For i from 1 to $\frac{1}{\Delta t}$ do*

$$\mathbf{z}^{(i)} = Weierstrass\left(\mathcal{H}\left(i\Delta t, \mathbf{z}^{(0)}\right), \mathbf{z}^{(i-1)}\right)$$

- *Return* $\mathbf{z}^{\left(\frac{1}{\Delta t}\right)}$.

Where the function $Weierstrass$ is the iteration function of the multivariate Weierstrass's method, just as described in section 3.

Remark 5.5. *If we have only one root, this algorithm is the Newton's method with continuation, just as defined in [20].*

6. SOME NUMERICAL EXPERIENCES

We present here some numerical results used to experiment the two methods. The implementation has been made in Maple. We experiment the quadratic convergence of the Weierstrass's method and the global behaviour of the modified method.

6.1 Local convergence of the multivariate Weierstrass's method

Here follow an experimentation about the quadratic convergence of the algorithm in a neighbourhood of the solutions. We start close from the solutions. The norm of the difference between the matrix formed with the coordinates of the solutions and the matrix formed with the coortinates of the starting points is denoted by γ. The cosidered system has 24 solutions : $f1 := -852423.2347701474 * x + 169004.4536412536 * y - 19587.68428441587 * x^2 -.2719592445098297 * x^3 * y^2 - .2378686977250152e-2 * x^2 * y^4 - .8344236516137206e-1 * x * y^4 - 1.268543666683052 * x^3 * y + .2617294749371950e-2 * x^3 * y^3 + 4.819813945556113 * x * y^3 - 4795.362970530327 * x * y + .2644928939975801e-4 * x^3 * y^4 - .2107340021576993e-6 * x^3 * y^5 - .2947515061103828 * x^2 * y^3 + 154.2508123690707 * x^2 * y + 50.84034152607777 * y^3 + 223.7907787208135 * x^3 - .9697537223091198e-2 * y^5 + 3.526073665425521 * y^4 - .7361951940053530e-3 * x * y^5 + .3720958545997405e-4 * x^2 * y^5 + 18.33954272510618 * x^2 * y^2 - 28421.31022609652 * y^2 + 734.3372661438380 * x * y^2 + x^4 + 33433273.36167341$,
$f2 := -144829332.0786109 * x + 40709354.58370014 * y + 3824871.458683409 * x^2 - 23.18560279310779 * x^3 * y^2 +.6693010338539543 * x^2 * y^4 - 6.082009869353424 * x * y^4 - 304.8916006475366 * x^3 * y - .4322974862463313e-1 * x^3 * y^3 - 2323.970839246147 * x * y^3 + 3799581.127924402 * x * y + .5466585020145390e-2 * x^3 * y^4 - .2980440728102852e-4 * x^3 * y^5 + 48.77890395419961 * x^2 * y^3 - 24217.46315683881 * x^2 * y - 76253.38337080877 * y^3 + 22127.66616730034 * x^3 + 17.15474628585860 * y^5 - 11134.75238509252 * y^4 +.3546068648236057 * x * y^5 - .8332410245728632e-2 * x^2 * y^5 - 4082.673924664958 * x^2 * y^2 + 32505846.59397477 * y^2 + 132026.4287217977 * x * y^2 + y^6 - 26006535160.89200$

- $\gamma = 0.1$.

Number of iterations	distance between iteration points
1	.9991002315300692e-1
2	.6104845338391919e-2
3	.4134651430850353e-5
4	.5327469723622636e-11

- $\gamma = 0.5$.

Number of iterations	distance between iteration points
1	1.550672273638374
2	1.482054913234396
3	.6859425831384290e-1
4	.3418779114072579e-3
5	.8479437935480125e-8
6	.6620809049017181e-11

- $\gamma = 1$.

Number of iterations	distance between iteration points
1	6.184060161791787
2	5.517965427969140
3	.4599625264660909
4	.1555314356637456e-1
5	.1527148419754666e-4
6	.1345221563471728e-10

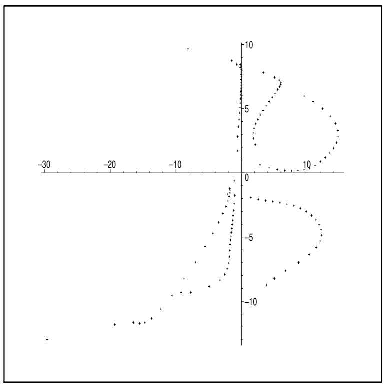
By using parallelisation of this method we can treat problem with 30 solutions with a number of variables between 2 and 5.

6.2 Behaviour of the modified method

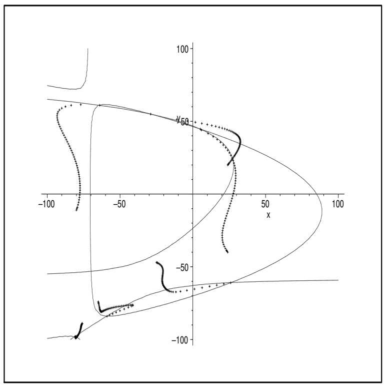
Here follow some experimentations of the method using continuation. Because the codimension of the singular configuration is two with $\mathbb{K} = \mathbb{C}$ we use complex numbers. In the following graphics, only real part of the iteration points are represented by solutions of the considered systems are real.

At each step we use several times the Weierstrass iteration in order to get the solutions of the intermediate system with a given precision. Initial configaration of points are choosen randomly.

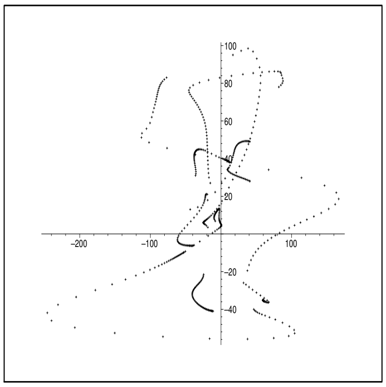
- A system with 4 solutions solved in 20 iterations :



- A system with 6 solutions solved in 50 iterations :



- A system with 10 solutions solved in 50 iterations :



With a parallelised implemantation of this algorithm, we solved a system with 24 solutions and 2 variables.

7. POSSIBLE IMPROVEMENTS AND PERSPECTIVES

In algoritm 4.1, we take as many relations as variables. But we can take more relations to obtain an overconstrain system. So we can expect a better numerical behaviour of the multivariate Weierstrass's method with using more informations. We have mentioned that we can use the pseudo Vandermonde structure of the matrices to improve complexity. A great perspective comes from the possibility to have a complexity theory as in [20] or [8]. We expect to find explicit conditions on the convergence of the last algorithm (for instance geometrical conditions as the distance between solutions). For the univariate polynomials there are faster algorithms than the Weierstrass method. We expect a generalisation for the situation described here in order to obtain more consistant numerical methods using a generalisation of Newton's basis as in the univariate case (see [5]).

Notice that it is possible to generalize this method if there are solutions with multiplicities. Here we used the fact that if all solutions are simple, the evaluations at those solutions form a basis of $\widehat{\mathcal{A}}$. If we have solutions with multiplicity, we can give a basis of $\widehat{\mathcal{A}}$ formed by evaluation at those solutions and evaluation of derivative. The dual set of this basis forms a basis of $\mathcal{A}$ that has all the properties needed to compute an iteration function. We refer to [17] for a more complete description of this formalism.

Implemantations of algorithms are made in Maple and are available at the following url : `http:/www-sop.inria.fr/galaad/personnel/ruatta/Weier.html`. A new implemantation in C++ is in progress, and we expect to treat problems more efficiently and faster. This algorithm is easily parallelisable. Experiments on a cluster where done and revealed a good behaviour of the method.

8. ACKNOWLEDGMENTS

I thank Bernard Mourrain for his kindness and for the time he spends with me to discuss of the problems I meet. I am deeply thankfull for the ideas that professor Jean-Claude Yakoubsohn gave me and for his contribution to this work. I thanks Alexis Bonnecase for his careful reading.

9. REFERENCES

[1] Aberth, O. Iteration methods for finding all zeros of a polynomial simultaneously. *Mathematics of Computation 27* (1973).

[2] Bellido, A.-M. Construction de fonction d'itération pour le calcul simultané des racines d'une équation $f(s) = 0$. *C.R. Acad. Sci. Paris* (1992).

[3] Bellido, A.-M. *Construction de fonctions d'itération pour le calcul simultané des solutions d'équations et de systèmes d'équations algébriques.* PhD thesis, Université Paul Sabatier de Toulouse, 1992.

[4] Bellido, A.-M. Construction of iteration functions for the simultaneous computation of the solutions of equations and algebraic systems. *Numerical Algorithms 6* (1994), 313–351.

[5] Bini, D. Numerical computation of polynomial zeros by means of aberth's method. *Numerical Algorithms 13* (1996), 179–200.

[6] Blum, L., Shub, M., and Smale, S. On a theory of computation and complexity over the real numbers: Np-compleness, recursive functions and universal machines. *Bulletin of the American Mathematical Society 21*, 1 (July 1989), 1–46.

[7] Cox, D., Little, J., and O'Shea, D. *Ideals, Variety and Algorithms.* Springer-Verlag, New York, 1996.

[8] Dedieu, J.-P., and Smale, S. Some lower bounds for the complexity of continuation methods. *Journal of Complexity 14* (1998), 454–465.

[9] Durand, E. *Solution numérique des équations algébriques*, vol. 1. 1968.

[10] Elkadi, M., and Mourrain, B. Géométrie algébrique effective : de la théorie aux applications. Notes de cours, prépublication.

[11] Faugére, J.-C. A new efficient algorithm for computing gröbner bases (f_4). *Journal of Pure and Applied Algebra 139* (1999), 61–88.

[12] Fromer, A. A unified approach to methods for the simultaneous computation of all zeros of generalized polynomials. *Numer. Math. 54* (1988), 105–116.

[13] Huber, B., and Sturmfelds, B. A polyhedral method for solving sparce polynomial systems. *Math. Comp. 64*, 212 (1995), 1542–1555.

[14] Kerner, I. Ein Gesamtschrittverfahrenzur Berechnung der Nullstellen von Polynomen. *Numer. Math. 8* (1966), 290–294.

[15] Lascoux, A. Note on interpolation in one and several variables. http://schubert.univ-mlv.fr/ al/.

[16] Li, T. Numerical solution of multivariate polynomial systems by homotopy continuation methods. *Acta Numerica* (1997), 399–436.

[17] Mourrain, B. Isolated points,duality and residues. *Journal of Pure and Applied Algebra 117 & 118* (1996), 469–493. Special issue for the Proc. of the 4th Int. Symp. on Effective Methods in Algebraic Geometry (MEGA).

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

[19] Mourrain, B., and Trebuchet, P. Solving projective complete intersection faster. In *Proc. ISSAC* (2000), pp. 231–238.

[20] Smale, S. Complexity theory and numerical analysis. *Acta Numerica* (1997), 523–551.

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

[22] Verschelde, J., Verlinden, K., and Cools, R. Homotopies exploiting newton polytopes for solving sparce polynomial systems. *SIAM J. Numerical Analysis 31*, 3 (1994), 915–930.

[23] Weierstrass, K. Neuer Beweis des Satzes, dass Jede Ganze Rationale Function einer Veränderlichen Dargestellt werden kann als ein Product aus Linearen Functionen derselben Veränderlichen. *Mathematische Werke* (1903). Tome 3.

[24] Yakoubsohn, J.-C. Simultaneous computation of all the zero-clusters of a univariate polynomial. Submitted, 2000.