Solving the Birkhoff Interpolation Problem via the Critical Point Method: an Experimental Study

F. Rouillier[1], M. Safey El Din[2], É. Schost[3]

[1] LORIA, INRIA-Lorraine, Nancy, France
Fabrice.Rouillier@loria.fr
[2] CALFOR team, LIP6, Université de Paris VI, Paris, France
Mohab.Safey@lip6.fr
[3] Laboratoire GAGE, École polytechnique, 91128 Palaiseau Cedex, France
schost@gage.polytechnique.fr

Abstract. Following the work of Gonzalez-Vega, this paper is devoted to show how to use recent algorithmic tools of computational real algebraic geometry to solve the *Birkhoff Interpolation Problem*. We recall and partly improve two algorithms to find at least one point in each connected component of a real algebraic set defined by a single equation or a polynomial system of equations, both based on the computation of the critical points of a distance function.
These algorithms are used to solve the Birkhoff Interpolation Problem in a case which was known to be an open problem. The solution is available at the U.R.L.:
http://www-calfor.lip6.fr/~safey/applications.html.

1 Introduction

The problem of interpolating a function $f : \mathbb{R} \longrightarrow \mathbb{R}$ by a univariate polynomial from the values of f and some of its derivatives on a set of sample points is one of the main questions in Numerical Analysis and Approximation Theory.

Let $\chi = \{x_1, \ldots, x_n\}$ be a set of real numbers such that $x_1 < \ldots < x_n$, r an integer, and let $\mathcal{I} \subset \{1, \ldots, n\} \times \{0, \ldots, r\}$ be the set of pairs (i, j) such that the value $f_{i,j} = f^{(j)}(x_i)$ is known. The problem of determining the existence and uniqueness of a polynomial Q in $\mathbb{R}[X]$ of degree bounded by r such that:

$$\forall (i, j) \in \mathcal{I}, \quad Q^{(j)}(x_i) = f_{i,j}$$

is called the *Birkhoff Interpolation Problem.*

In [17], Gonzalez-Vega focuses on determining, for fixed integers n and r, the family of $\mathcal{I}$'s for which this question is solvable for any choice of χ and the values $f_{i,j}$. To this end, he shows that the problem can be reduced to decide if some hypersurfaces contain real points with non zero coordinates. In [17] the cases $n = 2$, $r \in \{1, 2, 3, 4, 5, 6\}$, $n = 3$, $r \in \{1, 2, 3\}$ and $n = 4$, $r \in \{1, 2, 3, 4\}$ are solved, using techniques adapted from the Cylindrical Algebraic Decomposition. In 1998, the case $n = 5$ and $r = 4$ is presented as an open problem in [18]. The aim of the paper is to show how we solved this case.

The most popular algorithm deciding the emptiness of semi-algebraic sets – as a particular case of deciding the truth of a first order formula – is Collins' Cylindrical Algebraic Decomposition (CAD) [12, 11] whose complexity is doubly exponential in the number of variables in terms of basic arithmetic operations and size of output.

From Grigoriev and Vorobjov's paper [16], new algorithms appeared, based on the critical point method. These algorithms have a single exponential complexity in the number of variables, in terms of basic arithmetic operations and size of output. Still, in [20], Hong shows that the algorithms proposed in the papers [24] and [16] are not usable in practice. According to the experiments in [31], the same conclusions apply for more recent methods like in [9, 30, 19]. These algorithms adopt strategies of the following kind:

– In the first place, solve the problem in favorable cases, such as a compact and smooth variety.

- Get back from general situations to the favorable cases using various tricks, such as infinitesimal deformations or sums of squares.

The papers of the TERA group [7, 8] treat the case of a smooth complete intersection variety, and propose an algorithm based on evaluation techniques, whose complexity is polynomial in terms of intrinsic real degrees. Still, these favorable situations, in particular compactness, are not easily detectable, and systematically applying the tricks above makes the computations difficult in practice.

In the papers [28, 6], two algorithms inspired by the ideas in [16, 19, 24, 9, 30] are proposed. Both are based on the computation of the critical points of a distance function (thus avoiding the hypothesis of compactness) and improve the aforementioned algorithms. The first algorithm [28] computes at least one point in each connected component of a real algebraic set defined by a single equation. The second [6] applies to a real algebraic set defined by a polynomial system, in the spirit of [10, 13]. The experiments in [28, 6] show that these strategies are competitive with the CAD on small examples and allow to deal with more significant ones, unreachable with the CAD.

In this paper, we pursue the investigations of [20] by analyzing the practical behavior of these two recent algorithms on the Birkhoff Interpolation Problem. In sections 3, 4 and 5, we recall the algorithm given in [28], our contribution being a new way to solve a system with infinitesimal parameters arising in its course, then the algorithm given in [6]. We conclude this paper with our experimental results, which solve the case $n = 5$, $r = 4$ of Birkhoff's problem. This gives us the opportunity to compare the size of the outputs and computational times of both algorithms.

Throughout this paper, the base field is the rational field $\mathbb{Q}$. The algorithms presented here generalize to the case of an ordered field K, replacing the field $\mathbb{R}$ by the real closure of K and the complex field $\mathbb{C}$ by its algebraic closure.

2 The Birkhoff Interpolation Problem

2.1 Formulation

We want to determine the sets $\mathcal{I}$ for which the Birkhoff Problem Interpolation admits a unique solution for all choices of χ and of the $f_{i,j}$. To this end, we follow closely [17], and adopt his convenient matricial formulation.

Consider the matrix $\mathcal{E} = (e_{i,j})$ with n lines and $r+1$ columns [17], filled with 0's and 1's, such that $e_{i,j} = 1$ if and only if $(i, j) \in \mathcal{I}$. The problem admits a solution only if $\mathcal{E}$ has as many 1's as columns. This amounts to saying that the coefficients of the interpolating polynomial Q are solution of a linear *square* system, with associated matrix $\mathcal{M}_{\mathcal{E}}$. This matrix is parametrized by χ and its shape depends on $\mathcal{E}$. We are interested in determining the matrices $\mathcal{E}$ for which the determinant of $\mathcal{M}_{\mathcal{E}}$ is non-zero for all χ, in which case the matrix $\mathcal{E}$ is said to be *poised.*

Example 1. Let $n = 4$ and $r = 3$ and consider the matrix:

$$\mathcal{E} = \begin{pmatrix} 1\,1\,0\,0 \\ 0\,0\,1\,0 \\ 0\,1\,0\,0 \end{pmatrix}$$

Let $Q(x) = a_0 + a_1x + a_2x^2 + a_3x^3$ be the generic polynomial of degree 3. Writing $Q^{(j)}(x_i) = f_{i,j}$ if and only if $e_{i,j} = 1$, we obtain the following linear system:

$$\begin{cases} a_0 + a_1x_1 + a_2x_1^2 + a_3x_1^3 = f_{1,1} \\ \quad a_1 + 2a_2x_1 + 3a_3x_1^2 = f_{1,2} \\ \quad a_1 + 2a_2x_3 + 3a_3x_3^2 = f_{3,2} \\ \qquad 2a_2 + 6a_3x_2 = f_{3,2} \end{cases}$$

whose matrix is:

$$\begin{pmatrix} 1 & x_1 & x_1^2 & x_1^3 \\ 0 & 1 & 2x_1 & 3x_1^2 \\ 0 & 1 & 2x_3 & 3x_3^2 \\ 0 & 0 & 2 & 6x_2 \end{pmatrix}$$

The interpolation problem is solvable if and only if

$$12x_3x_2 + 6x_1^2 - 12x_2x_1 - 6x_3^2$$

does not vanish for all values x_1, x_2, x_3 satisfying $x_1 < x_2 < x_3$.

In [17], Gonzalez-Vega shows that the question can be reduced to test if a particular factor of the determinant of the matrix $\mathcal{M}_{\mathcal{E}}$ has real roots with non zero coordinates. Replacing $(x_1, \ldots, x_n)$ by $(x_1, x_1 + t_1^2, \ldots, x_1 + t_1^2 + \ldots + t_{n-1}^2)$ yields a homogeneous polynomial in $(t_1, \ldots, t_{n-1})$. Letting $t_1 = 1$, we are brought to test if a hypersurface defined by a polynomial $P \in \mathbb{R}[t_2, \ldots, t_{n-1}]$ has real roots with non zero coordinates.

2.2 Sketch of the Resolution

In order to determine all the poised matrices in the case $n = 5$ and $r = 4$, we have to study quasi-algebraic sets defined by a unique equation in $\mathbb{R}[t_2, t_3, t_4]$ and several inequations. While algorithms deciding the emptiness of real algebraic sets have known recent significant progress [13, 28, 6], more algorithmic work is necessary in the semi-algebraic case [9].

We thus treat this question using and adapting the algorithms for the algebraic case described in [28] and [6]. The algorithm described in [28] will be named **Algorithm 1**. It takes as input a single polynomial equation and returns at least one point in each connected component of the real algebraic variety defined by the equation. The algorithm described in [6] will be named **Algorithm 2**. It takes as input a polynomial system of equations and returns at least one point in each connected component of the real algebraic variety defined by the system.

To solve our problem, given a polynomial P in $\mathbb{Q}[t_2, t_3, t_4]$, we adopt the following scheme.

First Step. We study the hypersurface defined by $P = 0$, using either **Algorithm 1** or **Algorithm 2**. If this subroutine returns no real point, or a real point with non zero coordinates, we can give a positive (resp. negative) answer to the Interpolation Problem.

Second Step. We are in the case when all the real points we obtained have at least one coordinate equal to zero. Using **Algorithm 1**, we study the hypersurface defined by $P^2 + (Tt_1t_2t_3 - 1)^2 = 0$; when **Algorithm 2** we study the polynomial system $P = 0$ and $Tt_1t_2t_3 - 1 = 0$.

Note that in some situations, we can avoid such extra computations, using for example the following result:

Lemma 1. *Let $P \in \mathbb{R}[t_1, \ldots, t_n]$ and $\mathcal{V}(P)$ the hypersurface defined by $P = 0$. Let $M = (\mu_1, \ldots, \mu_n) \in \mathcal{V}(P) \cap \mathbb{R}^n$ such that $\mathbf{grad}_{\mathbf{M}}(P) \neq \mathbf{0}$, and $I \subset \{1, \ldots, n\}$ the set of indexes for which μ_i is zero. If $\mathbf{grad}_{\mathbf{M}}(P)$ is collinear to none of the axes $(t_i)_{i \in I}$, then there exists a point M' in $\mathcal{V}(P) \cap \mathbb{R}^n$ with non zero coordinates.*

3 Preliminary Results

This section describes the basics of the critical point method, valid for both **Algorithm 1** and **Algorithm 2**.

Let P be a square-free polynomial in $\mathbb{Q}[x_1, \ldots, x_n]$, and $\mathcal{V}(P) \subset \mathbb{C}^n$ the complex variety defined by $P = 0$. Our strategy to compute at least one point in each connected component of $\mathcal{V}(P) \cap \mathbb{R}^n$ relies on the computation of the critical points on $\mathcal{V}(P) \cap \mathbb{R}^n$ of a "distance function". Given a point $\mathbf{A} = (a_1, \ldots, a_n)$ in $\mathbb{R}^n$, the function $d_{\mathbf{A}} : \mathbf{M} \mapsto ||\mathbf{AM}||^2$ admits a minimum on each connected component of $\mathcal{V}(P) \cap \mathbb{R}^n$. These minima are solutions of the system

$$S(P, \mathbf{A}) = \{P(\mathbf{M}) = 0,\ \mathbf{grad}_{\mathbf{M}}(P)//\mathbf{AM}\},$$

where the condition $\mathbf{grad}_{\mathbf{M}}(P)//\mathbf{AM}$ is expressed by setting the determinants to zero. The set of all complex roots of this system is denoted by $\mathcal{C}(P, A)$.

Two cases will be distinguished, according to the dimension of the singular locus of $\mathcal{V}(P)$. The following result from [28] deals with the first case, where there is a finite number of singularities:

Theorem 1. *Let P be a square-free polynomial in $\mathbb{Q}[x_1, \ldots, x_n]$.*

If $\mathcal{V}(P)$ contains a finite number of singular points, there exists a Zariski-dense set $\mathcal{F}' \subset \mathbb{C}^n$ such that for $\mathbf{A}$ in $\mathcal{F}'$, the system $S(P, \mathbf{A})$ is zero-dimensional.

Moreover, if $\mathcal{V}(P)$ is a smooth hypersurface, there exists a Zariski-dense set $\mathcal{F} \subset \mathbb{C}^n$ such that for $\mathbf{A}$ in $\mathcal{F}$, the system $S(P, \mathbf{A})$ is zero-dimensional and radical.

Hence, if $\mathcal{V}(P)$ contains a finite number of singular points, one can choose a point $\mathbf{A}$ such that $S(P, \mathbf{A})$ is zero-dimensional. Since $S(P, \mathbf{A})$ intersects each semi-algebraically connected component of $\mathcal{V}(P) \bigcap \mathbb{R}^n$, the problem is reduced to isolate the real roots of $S(P, \mathbf{A})$.

Throughout this paper, we will represent the solutions of a zero-dimensional system with coefficients in a field k using primitive element techniques, in a way that can be traced back to Kronecker [22]. Such a representation consists in:

- a linear form $u = \sum u_i x_i$ which separates the zeros of the system[1];
- its minimal polynomial q in $k[t]$ and the parameterizations $(v_1, \ldots, v_n)$ in $k[t]^n$, where $\deg v_i < \deg q$, such that the zeros of the system are described by

$$q(t) = 0, \quad \begin{cases} \widetilde{q}(t)x_1 = v_1(t), \\ \quad\vdots \\ \widetilde{q}(t)x_n = v_n(t), \end{cases}$$

  where $\widetilde{q}(t)$ is the derivative of the square-free part of $q(t)$. We will denote this kind of representation $(u, \mathcal{R})$, where $\mathcal{R}$ is the vector $[q, v_1, \ldots, v_n]$.

Modern presentations and algorithms include the Rational Univariate Representation [26, 25] or the Geometric Resolution [14, 15], which coincide in the case of radical ideals of dimension zero. Such representations are useful, in that they allow to count and isolate the real roots of the system using for instance Sturm-Habicht sequences [30].

When the singular locus of $\mathcal{V}(P)$ is of positive dimension, difficulties arise, as the system $S(P, \mathbf{A})$ is also of positive dimension for any choice of the point $\mathbf{A}$. In the following sections, we focus on this case, and present in turn the strategies used in algorithms **1** and **2**.

- The algorithm described in [28] performs an infinitesimal deformation on the hypersurface $\mathcal{V}(P)$ to get back to a smooth situation; the required information is extracted from the solution of the deformed problem.
- The approach from [6] is based on the iterated study of singular loci, as varieties of lower dimension.

4 Algorithm 1: Using Infinitesimal Deformations

In the first subsection, we recall the main steps of the algorithm in [28], to which we refer for further details. We then present our solution to the specific subproblem of computing a univariate representation depending on the deformation parameter.

[1] the image of two distinct zeros by u are distinct

4.1 Overview of the Algorithm

Let ε be an infinitesimal; we denote by $\mathbb{C}\langle\varepsilon\rangle$ the algebraically closed field of algebraic Puiseux series with coefficients in $\mathbb{C}$. The sub-ring of elements of non-negative valuation (called "bounded elements") is naturally equipped with the operation denoted $\lim_{\varepsilon\to 0}$. If $\mathcal{C}$ is a set of elements of $\mathbb{C}\langle\varepsilon\rangle$, $\lim_{\varepsilon\to 0}\mathcal{C}$ denotes the set of the limits of the bounded of elements of $\mathcal{C}$. Finally, if x is $\sum_{i\geq i_0} a_i\varepsilon^{i/q} \in \mathbb{C}\langle\varepsilon\rangle$, where $i_0 \in \mathbb{Z}, q \in \mathbb{Q}, a_i \in \mathbb{C}$, we denote by $o(x)$ the rational number i_0/q.

The following result shows that the study of $S(P-\varepsilon, \mathbf{A})$ enables to solve the original problem.

Proposition 1. *[28] The set* $(\lim_{\varepsilon\to 0}\mathcal{C}(P-\varepsilon,\mathbf{A}))\cap\mathbb{R}^n$ *intersects each connected component of* $\mathcal{V}(P)\cap\mathbb{R}^n$.

Since $\mathcal{V}(P-\varepsilon)\subset\mathbb{C}\langle\varepsilon\rangle^n$ is smooth, Theorem 1 implies that for a generic choice of the point $\mathbf{A}$, the system $S(P-\varepsilon,\mathbf{A})$ is radical of dimension zero. In this case, its solutions can be described by a univariate representation $(u,\mathcal{R})$, with coefficients in $\mathbb{Q}(\varepsilon)$. The paper [28] then gives an algorithm to compute the limits of the bounded solutions it describes, when u is a *well separating element*. This is the case when:

- for all $\alpha\in\mathcal{C}(P-\varepsilon,\mathbf{A})$, $o(u(\alpha)) = \min(o(X_i(\alpha)), i=1,\ldots,n)$,
- for all $(\alpha,\beta)\in\mathcal{C}(P-\varepsilon,\mathbf{A})^2$, $u(\alpha)$ and $u(\beta)$ are infinitesimally close if and only if α and β are infinitesimally close.

Indeed, from a univariate representation associated to a well separating element

$$q(\varepsilon,t)=0, \begin{cases} \widetilde{q}(\varepsilon,t)x_1 = v_1(\varepsilon,t), \\ \quad\vdots \\ \widetilde{q}(\varepsilon,t)x_n = v_n(\varepsilon,t), \end{cases}$$

if $q_1q_2^2\ldots q_m^m$ is the square-free decomposition of $q(0,t)$, then, $\forall j\in\{1,\ldots,m\}$, one can compute polynomials $\widetilde{q}^{(j)}$ and $v_i^{(j)}$ such that the limits of the bounded solutions are represented by

$$\left(q_j(t)=0, \begin{cases} \widetilde{q}^{(j)}(0,t)x_1 = v_1^{(j)}(0,t), \\ \quad\vdots \\ \widetilde{q}^{(j)}(0,t)x_n = v_n^{(j)}(0,t). \end{cases} \right)_{j\in\{1,\ldots,m\}}$$

We can now give the first algorithm, which is the synthesis of all the previous points:

Algorithm 1

Input: A squarefree polynomial P
Output: At least one point on each connected component of $\mathcal{V}(P)$

1. Find by trial and error a point $\mathbf{A}$ such that $S(P-\varepsilon,\mathbf{A})$ is zero-dimensional and radical.
2. Compute a parametric resolution $(u,\mathcal{R})$ of its roots, with coefficients in $\mathbb{Q}(\varepsilon)$.
3. If u is a well-separating element for $S(P-\varepsilon,\mathbf{A})$, compute the limits of the bounded solutions described by $\mathcal{R}$ as $\varepsilon\to 0$.
4. else change u, check if it is a separating element for $S(P-\varepsilon,\mathbf{A})$ and return to step 3.

We briefly detail a solution to point 1. above. In [28] the authors prove the following results:

Lemma 2. *Let G be a Gröbner basis of the system $S(P-\varepsilon,\mathbf{A})$ in $\mathbb{Q}[\varepsilon,x_1,\ldots,x_n]$, for a block-ordering such that $[\varepsilon]<[x_1,\ldots,x_n]$. Then G is a non-reduced Gröbner basis of $S(P-\varepsilon,\mathbf{A})$ in $\mathbb{Q}\langle\varepsilon\rangle[x_1,\ldots,x_n]$.*

If there exists a value ε_0 in $\mathbb{Z}$ which doesn't cancel any of the leading coefficients of the polynomials in G, and such that the system $S(P-\varepsilon_0,\mathbf{A})$ is radical of dimension zero, then $S(P-\varepsilon,\mathbf{A})$ is radical of dimension zero. Moreover, for such ε_0 and $\mathbf{A}$, if u is a separating element for $S(P-\varepsilon_0,\mathbf{A})$ then u is a separating element for $S(P-\varepsilon,\mathbf{A})$.

Given any point A that fits the hypothesis of Theorem 1, one can show that only a finite number of ε_0 will not fit the conditions of Lemma 2. Hence, a simultaneous search by trial and error of $\mathbf{A} \in \mathbb{Z}^n$ and $\varepsilon_0 \in \mathbb{Z}$ can be performed to obtain a point $\mathbf{A}$ such that $S(P-\varepsilon, \mathbf{A})$ is radical. For a given $\mathbf{A}$, this requires to compute the basis G; testing that $S(P-\varepsilon_0, \mathbf{A})$ is radical can be done using Hermite's quadratic form (see [26] for example).

4.2 Computing a Parametric Resolution

We now turn to the second of the tasks mentioned above, namely computing a resolution parametrized by ε. In [33], Schost proposes a probabilistic algorithm to do so, based on the work of the TERA group [14, 3, 15]. The algorithm relies on a formal Newton approximation process, which is an analog of numerical root-finding techniques, and reminiscent of Hensel lifting methods. In the sequel, we first recall the main steps of this algorithm, then provide solutions to certify its output in our case (radical and zero-dimensional ideal).

Overview of the Algorithm. For a random choice of ε_0 in $\mathbb{Q}$, given any resolution $(u, \mathcal{R}_0)$ of the system $S(P-\varepsilon_0, \mathbf{A})$, there exists a resolution $(u, \mathcal{R})$ of the system $S(P-\varepsilon, \mathbf{A})$ with coefficients in $\mathbb{Q}(\varepsilon)$ whose specialization at ε_0 is $(u, \mathcal{R}_0)$. The strategy presented in [33] consists in approximating the solution $(u, \mathcal{R})$ starting from $(u, \mathcal{R}_0)$.

The output $\mathcal{R}$ can be rewritten in terms of $\varepsilon' = \varepsilon - \varepsilon_0$, as a vector $\mathcal{R}'$ of polynomials in $\mathbb{Q}(\varepsilon')[t]$, where none of the denominators of the expressions in ε' vanishes at zero. Denote by $\mathcal{R}'_i$ the vector of polynomials of $\mathbb{Q}[[\varepsilon']][t]$, where all coefficients are expanded at precision 2^i. The initial value is obtained by solving the system $S(P-\varepsilon_0, \mathbf{A})$; the formal Newton operator, denoted $\texttt{Lift}(\mathcal{R}'_i)$ computes $\mathcal{R}'_{i+1}$ from the argument $\mathcal{R}'_i$.

The whole algorithm is organized around a while loop. Each pass begins by computing the resolution $\mathcal{R}'_{i+1}$, of precision 2^{i+1}. The subroutine `RationalReconstruction` then computes Padé approximants [35] of all the coefficients, as rational functions in ε', with numerators and denominators of degree at most 2^i; a boolean value b indicates success. If the reconstruction is possible, the subroutine `StopCriterion`, detailed below, tests if the resolution, rewritten in terms of ε, is correct. If this is not the case, we go trough another loop.

A Certified Result. The probabilistic aspect of the algorithm in [33] is twofold: it lies in the choice of the specialization value ε_0, and in the test `StopCriterion`. We provide here certified versions of these subroutines.

The value ε_0 must satisfy some genericity conditions: the system $S(P-\varepsilon_0, \mathbf{A})$ must be zero-dimensional, its roots must be simple and in maximal number. The algorithm in [33] is based on the fact that all choices of ε_0 but a finite number fulfill these conditions, and that a bound on the number of bad values is readily available. Instead, we use the following result: if ε_0 cancels none of the leading coefficients of the polynomials in the basis G computed above, and if the system $S(P-\varepsilon_0, \mathbf{A})$ is radical, then ε_0 satisfies the genericity conditions.

Finally, to check that a solution $(u = \sum u_i x_i, \mathcal{R})$ is the correct solution, it is enough to check that the minimal polynomial and the parameterizations in $\mathcal{R}$, with t evaluated at $\sum u_i x_i$, reduce to zero modulo the basis G. This implies that the resolution $\mathcal{R}$ describes a set of points containing $\mathcal{C}(P-\varepsilon, \mathbf{A})$. As these two sets have the same cardinality, we are done.

Computing the parametric resolution

Input : a point $\mathbf{A}$ such that the roots of the system $S(P-\varepsilon, \mathbf{A})$ are simple.
a Gröbner basis G of $S(P-\varepsilon, \mathbf{A})$ in $\mathbb{Q}\langle\varepsilon\rangle[x_1, \dots, x_n]$.
Output : a parametric resolution $(u, \mathcal{R})$ of $S(P-\varepsilon, \mathbf{A})$

1. Find by trial and error a random rational number ε_0 such that the system $S(P-\varepsilon_0, \mathbf{A})$ is zero-dimensional and has simple roots, in maximal number.
2. Compute a univariate representation $\mathcal{R}'_0$ of the roots of $S(P-\varepsilon_0, \mathbf{A})$
3. Use the Newton-Hensel lifting process to compute the successive approximations $\mathcal{R}'_i$:

   $i \leftarrow 0$; finished $\leftarrow$ `false`
   while not finished do
       $\mathcal{R}'_{i+1} \leftarrow$ `Lift`$(\mathcal{R}'_i)$
       $b, \mathcal{R}' \leftarrow$ `RationalReconstruction`$(\mathcal{R}'_{i+1})$
       if b then
           $\mathcal{R} \leftarrow$ Substitute ε' by $\varepsilon + \varepsilon_0$ in $\mathcal{R}'$
           finished $\leftarrow$ `StopCriterion`$(\mathcal{R})$
       end if
       $i \leftarrow i + 1$
   end while

4. return $(u, \mathcal{R})$

The output is a list of polynomials in $\mathbb{Q}(\varepsilon)[t]$. The degree in t, the degrees in ε of the numerators and the denominators of the coefficients are bounded by the Bézout number d^n, where d is the degree of the polynomial P [33]. The "size" of the output is thus $O(n(d+1)^{2n})$ elements of the base-field $\mathbb{Q}$.

The details of the computations done in Newton's operator are given in [15, 33]. Following the usual numeric Newton operator, they rely on the evaluation of the vector $\mathbf{Jac}(S(P-\varepsilon, \mathbf{A}))^{-1} S(P-\varepsilon, \mathbf{A})$ in suitable quotient rings, where $\mathbf{Jac}(S)$ denotes the jacobian matrix of the system S.

As in [7, 15, 8], we use the *Straight-Line Program* model to encode the input polynomial P. In this model, the complexity of the whole lifting process is polynomial in the size of the output, in the number of variables n, and the number of operations L necessary to evaluate the polynomial P [33]. Still, the whole algorithm requires the precomputation of the Gröbner basis G, and the subroutine `StopCriterion` relies on normal form computations. This part dominates the whole cost of the algorithm.

5 Algorithm 2: Iterated Study of Singular Loci

The second approach generalizes the critical point methods used here to the case of polynomial systems. In the presence of infinitely many singular points, the infinitesimal deformation is avoided by studying the singular locus as a variety of lower dimension.

Let $\mathcal{V} \subset \mathbb{C}^n$ be an equidimensional algebraic variety of dimension d and $P = \{P_1, \dots, P_s\}$ polynomials in $\mathbb{Q}[x_1, \dots, x_n]$ such that $\mathcal{I}(\mathcal{V}) = (P_1, \dots, P_s)$. Following the notation of the two previous sections, given any point $\mathbf{A}$ in $\mathbb{C}^n$, we define the polynomial system:

$$S(P, \mathbf{A}) = \{P_1(\mathbf{M}) = \cdots = P_s(\mathbf{M}) = 0, \mathrm{rank}(\mathbf{grad}_{\mathbf{M}}(P_1), \dots, \mathbf{grad}_{\mathbf{M}}(P_s), \mathbf{AM}) \leq n - d\},$$

where the rank condition is expressed by setting to zero the $(n-d+1) \times (n-d+1)$ minors of the matrix $[\mathbf{Jac}(P), \mathbf{AM}]$. The roots of this system form a set denoted $\mathcal{C}(\mathcal{V}, \mathbf{A})$.

The algorithm proposed in [6] is based on the following theorem:

Theorem 2. *[6] The set $\mathcal{C}(\mathcal{V}, \mathbf{A})$ meets every connected component of $\mathcal{V} \cap \mathbb{R}^n$. Moreover, there exists a Zariski-dense subset $\mathcal{F}$ of $\mathbb{C}^n$ such that, for all $\mathbf{A}$ in $\mathcal{F}$, $\mathcal{C}(\mathcal{V}, \mathbf{A})$ can be written* $\mathrm{Sing}(\mathcal{V}) \cup \mathcal{V}_0$*, where*

– $\mathcal{V}_0$ *is a finite set of points in* $\mathbb{C}^n$,
– $\mathrm{Sing}(\mathcal{V}) = \{\mathbf{M} \in \mathcal{V} \mid \mathrm{rank}(\mathbf{grad}_{\mathbf{M}}(P_1), \ldots, \mathbf{grad}_{\mathbf{M}}(P_s)) < n - d\}$.

In particular, in this case, $\dim(\mathcal{C}(\mathcal{V}, \mathbf{A})) < \dim(\mathcal{V})$.

Suppose that $\mathcal{V} \subset \mathbb{C}^n$ and $\mathbf{A} \in \mathbb{R}^n$ satisfy the conditions of Theorem 2, and that $\mathcal{V}_1$ is a finite set of points that meets each connected component of $\mathrm{Sing}(\mathcal{V}) \cap \mathbb{R}^n$. Theorem 2 implies that $\mathcal{V}_1 \cup \mathcal{V}_0$ meets each connected component of $\mathcal{V} \cap \mathbb{R}^n$. The set $\mathcal{V}_1$ can in turn be obtained by applying Theorem 2 to each equidimensional component of $\mathrm{Sing}(\mathcal{V})$. The algorithm in [6] consists in applying inductively the above process, performing at each step equidimensional decompositions of intermediate varieties $\mathcal{C}(\mathcal{V}_i, \mathbf{A}_i)$. In the end, we obtain a family of zero-dimensional sets which meets each connected component of $\mathcal{V} \cap \mathbb{R}^n$.

At each step, we need to apply a subroutine taking as input a polynomial system S and returning a set of generators of radical equidimensional ideals whose intersection is $\sqrt{S}$. This can be done using the algorithms mentioned in [4, 31] or by performing a decomposition into regular and separable triangular sets [5, 23, 21, 34] and computing a Gröbner basis of the saturated ideal of each triangular set. We denote by `EquiDimDecomposition` such a radical equidimensional decomposition algorithm.

Algorithm 2

Input: A polynomial system P
Output: At least one point on each connected component of $\mathcal{V}(P)$

1. list $\leftarrow$ `EquiDimDecomposition`(P), result $\leftarrow$ []
2. while list $\neq$ [] do
   – $\widetilde{P} \leftarrow$ first(list) and remove $\widetilde{P}$ from list,
   – if $\dim(\widetilde{P}) = 0$ then result $\leftarrow$ result $\cup\, \widetilde{P}$
   – else find by trial and error a point $\mathbf{A}$ such that $\dim(\mathcal{C}(\mathcal{V}(\widetilde{P}), \mathbf{A})) < \dim(\widetilde{P})$ and list $\leftarrow$ list $\cup$ `EquiDimDecomposition`$(S(\widetilde{P}, \mathbf{A}))$
3. count and isolate the real roots of all the polynomial systems in result.

6 Experimental Results

6.1 Methodology and Basic Algorithms

Both algorithms presented above have been implemented, using the following softwares:

– Gb/AGb: implemented in C++ by J.-C. Faugère [4] and devoted to Gröbner basis computations;
– RS: implemented in C by F. Rouillier, devoted to computing Rational Univariate Representations, and to counting and isolating real roots of univariate polynomials;
– Kronecker: implemented in Magma by G. Lecerf [15], devoted to compute Geometric Resolutions, from which we borrowed the formal Newton iterator.

The subroutine `EquiDimDecomposition` was implemented using Maple and a file connection with Gb, following the algorithm described in [31] and based on the results in [5, 23].

All the computations were done on the computers of the UMS MEDICIS [2], on a PC Pentium II 400 MHz with 512 MB of RAM.

6.2 Solution of the Problem

The case $n = 5$, $r = 4$ of Birkhoff's problem generates 53130 matrices, which produces as many hypersurfaces of $\mathbb{C}^3$ to study.

– 42925 of these hypersurfaces are defined by constant polynomials.

- For the non-constant polynomials, to avoid unnecessary computations, we specialized all variables but one at random non-zero values and applied Uspensky's algorithm on the univariate polynomials we obtained, looking for non-zero real roots. At the end of this preprocessing, about one thousand hypersurfaces remained to study.
- About 900 of these hypersurfaces had zero or a finite number of singularities. In all these cases, the first step given in section 2.2 was sufficient to conclude: the situation where $\mathcal{C}(P, \mathbf{A})$ had exclusively real roots with a coordinate equal to zero was never encountered.

On a PC bi-Pentium 400 MHz with 512 MB of RAM, 4 hours are necessary to perform these 3 steps for all hypersurfaces.

- There remained 102 hypersurfaces containing an infinity of singularities. We will see below that our implementation of **Algorithm 1** can not solve all of them, whereas **Algorithm 2** succeeded in all cases. In 60 out of these 102 cases, we had to go through the second step given in section 2.2.
  On the same machine, 2 additional hours are necessary to perform this final step with **Algorithm 2**.

As a conclusion, 19092 out of the 53130 matrices are poised. Their complete list can be found in Maple format at the web page [1].

6.3 Comparing Algorithm 1 and Algorithm 2

We give a more detailed account of the behavior of algorithms **1** and **2** on a sample of the family of hypersurfaces with infinitely many singularities. These hypersurfaces are denoted Birk.3-1,...,Birk3-15. All of them have degree less than 8; the whole list can be found at the web page [1].

Table 1 summarizes our results on this family. The sign ∞ indicates that the computations were stopped after 24 hours.

- **Algorithm 1:** The first column gives the number of bounded roots we obtain, which is a measure of the size of the output. The second and third columns give the degrees in t and ε of the generic resolution, which is a measure of the size of intermediate data. The last column gives the time necessary to perform all computations, in seconds.
- **Algorithm 2:** The first column gives the sum of the degrees of the zero-dimensional systems produced in the course of the computations; the second indicates the total time of computations, given in seconds.

Hypersurface	Algorithm 1				Algorithm 2	
Birk.3-1	12	16	3	5.6	12	0,08
Birk.3-2	7	16	3	5.2	7	0,13
Birk.3-3	25	34	5	32	25	0,37
Birk.3-4	16	36	5	46	16	0,18
Birk.3-5	31	40	5	116	31	0,46
Birk.3-6	37	52	7	149	37	0,86
Birk.3-7	38	52	7	115	38	0,72
Birk.3-8	45	130	19	3927	45	7,11
Birk.3-9	47	132	19	2945	47	7,88
Birk.3-10	48	136	31	18843	48	8,04
Birk.3-11	50	138	31	26536	50	8,88
Birk.3-12	50	138	31	17508	50	10,01
Birk.3-13	32	252	29	∞	32	9,26
Birk.3-14	60	264	31	∞	60	67
Birk.3-15	60	272	31	∞	60	83

Table 1. Algorithms **1**/**2**: Size of the output and computation times

For the last examples, the time spent in **Algorithm 1** in the checking phase becomes largely predominant. Other strategies to certify this output, based on a sufficient number of sample checks, could lower this time. Even *without* this certification phase, the computation is longer than with **Algorithm 2**. Still, with regard to the degrees in t and ε of the parametric resolution, we consider that our implementation of **Algorithm 1** shows good performance.

A relevant criterion to analyze the algorithms based on the critical point method is the degrees of the zero-dimensional systems they produce. For **Algorithm 1**, this is the cardinality of the set of bounded roots $\lim_{\varepsilon \to 0} \mathcal{C}(P - \varepsilon, \mathbf{A})$. To this regard, the outputs of **Algorithm 1** and **Algorithm 2** are of similar size. The size of the intermediate data in **Algorithm 1**, such as the degree of the parametric resolution, is bigger, as several points of $\mathcal{C}(P - \varepsilon, \mathbf{A})$ collapse on a same point when $\varepsilon \to 0$.

Nevertheless, the degrees of the output and of the intermediate bivariate polynomials in **Algorithm 1** are bounded by d^n, while we have no similar bound for **Algorithm 2**. An open problem is to precise such a bound for **Algorithm 2**. In all these examples, the dimension of the singular locus was 1, so that there was at most one recursive call in **Algorithm 2**. Experiments with more intricate singular loci should tell us more about this question.

7 Conclusions

The case $n = 5$, $r = 4$ of the Birkhoff Interpolation Problem is now automatically solved. The case $n = 6$, $r = 5$, requires to study 1947792 hypersurfaces in $\mathbb{C}^4$; this combinatorial number is now the limiting factor. More research on qualitative nature should be devised to have a better control on this number; in this sense, the conclusions and suggestions in [17] are still a topical question.

This problem gave us the opportunity to compare two recent algorithms of computational real algebraic geometry and illustrate their practical use. It appears that the algorithms based on the critical point method can now solve application problems.

In particular, we have implemented computations with an infinitesimal, considering it as a parameter. Another approach consists in implementing an infinitesimal arithmetic; we refer to [27] for such a realization in Axiom. Nevertheless, obtaining good performance in practice using this type of arithmetic is still a computer science challenge.

Besides, the use of infinitesimals in computational real algebraic geometry is not exclusive to the desingularization of hypersurfaces: they are required in several algorithms to decide the emptiness of semi-algebraic sets, such as [19, 9, 30].

References

1. `http://www-calfor.lip6.fr/~safey/applications.html`
2. `http://www.medicis.polytechnique.fr`
3. `http://www.tera.medicis.polytechnique.fr`
4. `http://www-calfor.lip6.fr/~jcf`
5. P. Aubry, *Ensembles triangulaires de polynômes et résolution de systèmes algébriques, Implantations en Axiom*, PhD thesis, Université de Paris VI, 1999.
6. P. Aubry, F. Rouillier, M. Safey El Din, *Real Solving for positive dimensional systems,* Rapport de Recherche du Laboratoire d'Informatique de Paris VI, Mars 2000.
7. B. Bank, M. Giusti, J. Heintz, and M. Mbakop *Polar Varieties and Efficient Real Equation Solving*, Journal of Complexity, vol.13:5–27, 1997, Best paper award 1997.
8. B. Bank, M. Giusti, J. Heintz, and M. Mbakop *Polar Varieties and Efficient Real Elimination*, to appear in Mathematische Zeitschrift (2000).
9. S. Basu, R. Pollack, M.-F. Roy, *On the combinatorial and algebraic complexity of Quantifier elimination.* J. Assoc. Comput. Machin., 43, 1002–1045, 1996.
10. E. Becker, R. Neuhaus, *Computation of real radicals for polynomial ideals*, in Computational Algebraic geometry, Progress in Math., vol.109, 1–20, Birkhaüser, 1993.

11. G. E. Collins, H. Hong, *Partial Cylindrical Algebraic Decomposition,* Journal of Symbolic Computation, vol.12, No.3: 299–328, 1991.
12. G. E. Collins, *Quantifier elimination for real closed field by cylindrical algebraic decomposition,* Lectures Notes in Computer Science, 33, 515–532, 1975.
13. P. Conti, C. Traverso, *Algorithms for the real radical,* unpublished manuscript
14. M. Giusti, J. Heintz, *La détermination des points isolés et de la dimension d'une variété algébrique réelle peut se faire en temps polynomial*, Computational Algebraic Geometry and Commutative Algebra, Eds D. Eisenbud and L. Robbiano, 1993.
15. M. Giusti, G. Lecerf, B. Salvy, *A Gröbner free alternative for solving polynomial systems*, Journal of Complexity, vol.17, No.1, 2001
16. D. Grigor'ev, N. Vorobjov , *Solving Systems of Polynomial Inequalities in Subexponential Time,* Journal of Symbolic Computation, vol.5, No.1-2: 37–64, 1988.
17. L. Gonzalez-Vega, *Applying quantifier elimination to the Birkhoff Interpolation Problem,* Journal of Symbolic Computation vol.22, No.1, 1996.
18. M.-J. Gonzalez-Lopez and L. Gonzalez-Vega, *Project 2 : The Birkhoff Interpolation Problem,* In: Some tapas of computer algebra, A. Cohen ed. Springer, 297–310, 1999.
19. J. Heintz, M.-F. Roy, P. Solerno, *On the theoretical and practical complexity of the existential theory of the reals,* Comput. J. vol.36, No.5: 427–431, 1993.
20. H. Hong, *Comparison of Several Decision Algorithms for the Existential Theory of the Reals*, Research report, RISC, 1991.
21. M. Kalkbrener, *Three contributions to elimination theory*, PhD thesis, Johannes Kepler University, RISC, 1991.
22. L. Kronecker, *Grundzüge einer arithmetischen Theorie de algebraischen Grössen*, J. reine angew. Math. 1882.
23. M. Moreno Maza, *Calculs de Pgcd au-dessus des Tours d'Extensions Simples et Résolution des Systèmes d'Equations Algébriques,* PhD thesis, Université de Paris VI, 1997.
24. J. Renegar *On the computational complexity and geometry of the first order theory of the reals,* Journal of Symbolic Computation vol.13, No.3: 255–352, 1992.
25. F. Rouillier, *Algorithmes efficaces pour l'étude des zéros réels des systèmes polynomiaux,* PhD thesis, Université de Rennes I, 1996.
26. F. Rouillier, *Solving Zero-Dimensional Systems through the Rational Univariate Representation,* Applicable Algebra in Engineering Communications and Computing vol.9, No.5: 433–461, 1999.
27. R. Rioboo, *Computing with infinitesimals*, manuscript.
28. F. Rouillier, M.-F. Roy, M. Safey El Din, *Finding at least one point in each connected component of a real algebraic set defined by a single equation,* Journal of Complexity, vol.16, No.4, 2000.
29. F. Rouillier, P. Zimmermann, *Uspensky's algorithm : improvements and applications,* in preparation (2000).
30. M.-F. Roy, *Basic algorithms in real algebraic geometry: from Sturm theorem to the existential theory of reals,* Lectures on Real Geometry in memoriam of Mario Raimondo, Expositions in Mathematics 23, 1–67. Berlin, New York: de Gruyter 1996.
31. M. Safey El Din, *Résolution réelle des systèmes polynomiaux en dimension positive*, PhD thesis, Université Paris VI, 2001.
32. É. Schost, *Computing parametric geometric resolutions*, preprint École polytechnique, 2000.
33. É. Schost, *Sur la résolution des systèmes polynomiaux à paramètres*, PhD thesis, École polytechnique, 2000.
34. D. Wang, *Computing Triangular Systems and Regular Systems,* Journal of Symbolic Computation, vol.30, No.2: 221–236, 2000.
35. J. Von Zur Gathen, J. Gerhardt, *Modern Computer Algebra,* Cambridge University Press, 1999.