CONTEMPORARY MATHEMATICS

604

Recent Advances in Real Complexity and Computation

UIMP-RSME Lluís A. Santaló Summer School
Recent Advances in Real Complexity and Computation
July 16–20, 2012
Universidad Internacional Menéndez Pelayo,
Santander, Spain

José Luis Montaña
Luis M. Pardo
Editors

American Mathematical Society
Real Sociedad Matemática Española

American Mathematical Society

Recent Advances in Real Complexity and Computation

CONTEMPORARY MATHEMATICS

604

Recent Advances in Real Complexity and Computation

UIMP-RSME Lluís A. Santaló Summer School
Recent Advances in Real Complexity and Computation
July 16–20, 2012
Universidad Internacional Menéndez Pelayo,
Santander, Spain

José Luis Montaña
Luis M. Pardo
Editors

American Mathematical Society
Real Sociedad Matemática Española

American Mathematical Society
Providence, Rhode Island

2010 *Mathematics Subject Classification*. Primary 03D15, 14Qxx, 14Q20, 65-xx, 65H20.

Library of Congress Cataloging-in-Publication Data

UIMP-RSME Lluis Santaló Summer School (2012 : Santander, Spain) Recent advances in real complexity and computation : UIMP-RSME Lluis Santaló Summer School 2012, recent advances in real complexity and computation, July 16–20, 2012, UIMP Palacio de la Magdalena, Santander (Cantabria), Spain / Jose Luis Montana, Luis M. Pardo, editors.
pages cm – (Contemporary Mathematics ; volume 604)
Includes bibliographical references.
ISBN 978-0-8218-9150-6 (alk. paper)
1. Computational complexity–Congresses. I. Pardo, L. M. (Luis M.), editor of compilation. II. Montana, Jose Luis, 1961–editor of compilation. III. Title.

QA267.7.U36 2012
511.3′52–dc23 2013022512

Contemporary Mathematics ISSN: 0271-4132 (print); ISSN: 1098-3627 (online)

DOI: http://dx.doi.org/10.1090/conm/604

Dedicated to our beloved friend Jean-Pierre Dedieu.

Contents

Editors' preface

This volume is composed of six contributions derived from the lectures given during the UIMP–RSME Lluís Santaló Summer School on "Recent Advances in Real Complexity and Computation". The goal of this Summer School was to present some of the recent advances on *Smale's 17th Problem*. This Problem was stated by Steve Smale as follows:

PROBLEM 1 (Smale's 17th Problem). *Can a zero of n complex polynomial equations in n unknowns be found approximately, on the average, in polynomial time with a uniform algorithm?*

These contributions cover several aspects around this problem: from numerical to symbolic methods in polynomial equation solving, computational complexity aspects (both worse and average cases, both upper and lower complexity bounds) and even aspects of the underlying geometry of the problem. Some of the contributions also deal with either real or multiple solutions solving.

The School was oriented to graduate mathematicians, as to Master or Ph. D. students in Mathematics and to senior researchers interested on this topic.

The School was promoted and supported by the Spanish Royal Mathematical Society (RSME) and hosted by the Universidad Internacional Menéndez Pelayo (UIMP), from July 16th to July 20th of 2012, in El Palacio de la Magdalena, Santander. Partial financial support was also granted by the University of Cantabria and the Spanish Ministry of Science Grant MTM2010-16051. We thank these institutions and grants for their financial support.

The speakers (in alphabetical order) and their courses in this Summer School were the following ones:

- Carlos Beltrán, *"Stability, precision and complexity in some numerical problems"*.
- Marc Giusti, *"Polar, co–polar and bipolar varieties: real solving of algebraic varieties with intrinsic complexity"*.
- Joos Heintz, *"On the intrinsic complexity of elimination problems in effective algebraic geometry"*.
- Gregorio Malajovich, *"From the quadratic convergence of Newton's method to problems of counting of the number of solutions"*.
- Klaus Meer, *"Real Number Complexity Theory and Probabilistically Checkable Proofs (PCPs)"*.
- Michael Shub, *"The geometry of condition and the analysis of algorithms"*.
- Jean-Claude Yakoubsohn, *" Tracking multiplicities"*.

The present volume extends the Summer School by expository articles presenting the state of art of each of the topics. The reader will find the following contributions in forthcoming pages:

(1) Martijn Baartse and Klaus Meer, *"Topics in real and complex number complexity theory"*
The contribution intends to introduce into topics relevant in real and complex number complexity theory. This is done in a survey style. Taking as starting point the computational model introduced by Blum, Shub, and Smale the following issues are addressed: Basic results concerning decidability and NP-completeness, transfer results of open questions between different models of computation, structural complexity inside $\mathrm{NP}_{\mathbb{R}}$, computational universality, and probabilistically checkable proofs over the real and complex numbers.

(2) Bernd Bank, Marc Giusti and Joos Heintz, *"Polar, bipolar and copolar varieties: Real solving of algebraic varieties with intrinsic complexity"*.
This survey covers a decade and a half of joint work with L. Lehmann, G. M. Mbakop, and L. M. Pardo. The authors address the problem of finding a smooth algebraic sample point for each connected component of a real algebraic variety, being only interested in components which are generically smooth locally complete intersections. The complexity of their algorithms is essentially polynomial in the degree of suitably defined generalized polar varieties and is therefore intrinsic to the problem under consideration.

(3) Carlos Beltrán and Michael Shub, *"The complexity and geometry of numerical solving polynomial equations"*.
This contribution contains a short overview on the state of the art of efficient numerical analysis methods that solve systems of multivariate polynomial equations. The authors focus on the work of Steve Smale who initiated this research framework, and on the collaboration between Stephen Smale and Michael Shub, which set the foundations of this approach to polynomial system–solving, culminating in the more recent advances of Carlos Beltrán, Luis Miguel Pardo, Peter Bürgisser and Felipe Cucker.

(4) Marc Giusti and Jean–Claude Yakoubsohn, *"Multiplicity hunting and approximating multiple roots of polynomials systems"*.
The computation of the multiplicity and the approximation of isolated multiple roots of polynomial systems is a difficult problem. In recent years, there has been an increase of activity in this area. Our goal is to translate the theoretical background developed in the last century on the theory of singularities in terms of computation and complexity. This paper presents several different views that are relevant to address the following issues : predict the multiplicity of a root and/or determine the number of roots in a ball, approximate fast a multiple root and give complexity results for such problems. Finally, we propose a new method to determine a regular system, called equivalent but deflated, i.e., admitting the same root as the initial singular one.

(5) Joos Heintz, Bart Kuijpers and Andrés Rojas Paredes, *"On the intrinsic complexity of elimination problems in effective algebraic geometry"*.
The representation of polynomials by arithmetic circuits evaluating them is an alternative data structure which allowed considerable progress in polynomial equation solving in the last fifteen years. The authors present in this contribution a circuit based computation model which captures the core of all known symbolic elimination algorithms that avoid unnecessary branchings in effective algebraic geometry and show the intrinsically exponential complexity character of elimination in this complexity model.

(6) Gregorio Malajovich, *"Newton iteration, conditioning and zero counting"*.
This contribution deals with the problem of counting the number of real solutions of a system of multivariate polynomial equations with real coefficients. You can also find in this contribution a crash-course in Newton iteration. We will state and analyze a Newton iteration based 'inclusion-exclusion' algorithm to count (and find) roots of real polynomials.

In recent months, two members of our scientific community left us: our colleague *Mario Wschebor* and our beloved friend *Jean-Pierre Dedieu*. Jean-Pierre was invited to the Summer School and his talk was scheduled as the closing talk of the School. Unfortunately, a long illness prevented him from being with us at the School and, sadly, he left us on 15 June 2012. Let this volume serve as a remembrance of both of them.

The editors wish to thank the RSME for giving us the opportunity to organize this event. It is also a pleasure to thank the patronage of the UIMP. Their help in the organization and the experience in Las Caballerizas del Palacio de la Magdalena are not to be easily forgotten. Our deepest gratitude goes to the speakers, who did an excellent job, and also to the students, whose interest and dedication created a great atmosphere. We finally wish to thank the authors for their excellent contributions to this volume.

José Luis Montaña & Luis M. Pardo

Contemporary Mathematics
Volume **604**, 2013
http://dx.doi.org/10.1090/conm/604/12067

Topics in real and complex number complexity theory

Martijn Baartse and Klaus Meer

Abstract. The paper intends to introduce into topics relevant in real and complex number complexity theory. This is done in a survey style. Taking as starting point the computational model introduced by Blum, Shub, and Smale the following issues are addressed: Basic results concerning decidability and NP-completeness, transfer results of open questions between different models of computation, structural complexity inside $\mathrm{NP}_{\mathbb{R}}$, computational universality, and probabilistically checkable proofs over the real and complex numbers.

1. Introduction

Complexity theory as a mathematical discipline is a relatively young subject. In a systematic way it was basically developed since the 1970's in Theoretical Computer Science based on the Turing machine as underlying model of computation. This led to a theory nowadays basically devoted to study complexity issues of discrete problems over finite structures. Problem instances are coded by sequences of bits and the complexity of algorithms is measured by counting the number of elementary bit operations necessary. It seems that Turing himself was as well interested in complexity and accuracy issues of numerical algorithms. He also addressed an idealized model in which floating-point numbers are used as kind of entities and was working on notions like the conditioning of a problem [**104**].

In contrast to the observation that complexity theory often is considered as a discipline in computer science mathematicians have designed and analysed algorithms already since centuries. Some of the most important and prominent ones were developed long before computers existed. Their inventors certainly had as well an intuition about complexity issues, though often under other perspectives. Think about algorithms like Gaussian elimination, Newton's method and notions like the order of convergence in numerical analysis, or algorithms for deciding the existence of complex solutions of a polynomial system related to Hilbert's Nullstellensatz.

Algorithms located in more classical areas of mathematics usually work with objects from uncountable continuous domains like the real or complex numbers, respectively. Often the number of basic arithmetic operations and test operations

2010 *Mathematics Subject Classification.* Primary 68Q05, 68Q15; Secondary 68Q17, 03D15, 03D35.

The authors gratefully acknowledge support of both authors by project ME 1424/7-1 of the Deutsche Forschungsgemeinschaft DFG. The second author wants to cordially thank L.M. Pardo and J. L. Montaña for the hospitality during the Santaló summer school in Santander, on which occasion this paper was written.

reflecting the underlying structure are of major interest. One then disregards the influence of round-off errors and uses an idealized model that computes with real or complex numbers as entities. This allows to focus on algebraic properties of problems and algorithms solving them. One of the first formalizations of such a viewpoint in complexity theory has been worked with in the area of Algebraic Complexity Theory [**25**] and can be traced back at least to the 1950's. Models of computation used there are algebraic circuits and straight line programs. In 1989, Blum, Shub and Smale introduced a model of computation now called the BSS model, see [**18**, **19**]. It gives a general approach to computability over rings and fields with a particular emphasis on $\mathbb{R}$ and $\mathbb{C}$. When considered over the finite field $\mathbb{Z}_2$ it results in the classical Turing machine model, whereas over fields like the real or complex numbers it gives a uniform model generalizing the ones previously used in algebraic complexity theory.

Let us mention that different models of computation have become more and more interesting in recent years both in computer science and mathematics. Think about such diverse models as Neural Networks [**48**], Quantum Computers [**81**], Analog Computers [**101**], Biological Computers [**43**] to mention a few. Beside in algebraic complexity the BSS model is also used as underlying computational model in the area of Information Based Complexity in which algorithms for numerical problems without complete information are studied [**82**, **108**]. Computational models dealing with real numbers are also studied in Recursive Analysis. Here, objects like real numbers or real functions are coded in a certain way by Cauchy sequences leading to notions like that of a computable real (already introduced by Turing) and computable real number functions. The theory arising from this approach is focussing more on stability of real number algorithms and thus different from the setting of this paper. For introduction and some additional controversial discussions on the question which model to use in what situation we refer the reader to the following literature: [**16**, **22**, **53**, **96**, **107**, **108**].

In this paper the Blum-Shub-Smale model builds the main topic of interest. The intention is to give an introduction into problems and methods relevant in real number complexity theory. The paper is organized as follows. Section 2 starts with a motivating example from kinematics that leads to several interesting questions in complexity, both with respect to the classical Turing and the real/complex number model. These problems are outlined and lead to a formal introduction of the real number model in the following section. There, basic complexity classes as well as the concept of $\mathrm{NP}_{\mathbb{R}}$-completeness are introduced and some first results are presented. We then focus on structural complexity theory for the real and complex numbers by discussing three different topics: Transfer results between different computational models, analysis of the structure inside $\mathrm{NP}_{\mathbb{R}}$ and $\mathrm{NP}_{\mathbb{C}}$ along the lines of a classical result by Ladner in the Turing model, and recursion theory over the reals. The rest of the paper then focusses on Probabilistically Checkable Proofs PCPs. The PCP theorem by Arora et al. [**2**, **3**] was a cornerstone in Theoretical Computer Science giving a new surprising characterization of complexity class NP and having tremendous applications in the area of approximation algorithms. After introducing the main ideas behind probabilistically checkable proofs we give a detailed proof of the existence of long transparent proofs for $\mathrm{NP}_{\mathbb{R}}$ and $\mathrm{NP}_{\mathbb{C}}$. Then, we outline how one can obtain a real analogue of the PCP theorem along the lines of a more recent proof of the classical PCP theorem by Dinur [**38**].

The paper is written in the style of an exposition. We do not cover all the interesting work that has been done since introduction of the Blum-Shub-Smale model about 20 years ago. Instead, the focus will be on topics the authors have also worked on themselves. With one exception dealing with long transparent proofs we most of the time do not present full proofs of the results treated. Instead it is tried to give the reader a feeling of the ideas behind such proofs, some more detailed and some not. More interested readers will easily find all details in the cited literature. Finally, we expect the reader to have a basic knowledge of classical complexity theory and the theory of NP-completeness [**44**], [**1**]. This is not crucial for understanding the flow of ideas, but we frequently refer to the Turing model in order to pinpoint similarities and differences between real number and classical complexity theory.

2. A motivating example

A typical problem in kinematics asks for finding suitable mechanisms that fulfil given motion tasks. Having chosen a mechanism which in principle can solve the problem the dimensions of the mechanism's components have to be determined. In its mathematical formulation this often leads to solving polynomial systems. As example of such a motion synthesis task consider the following one. Construct a mechanism which is able to generate a rigid body motion such that some constraints are satisfied. Constraints, for example, could be certain positions that have to be reachable by the mechanism. Figure 1[1] shows as typical example the motion of a plane in relation to a fixed base. Here, a $\xi - \eta$-system with its origin P is attached to the moving plane and a $x - y$-system is attached to the base. The rigid body motion now can be defined by certain poses of the $\xi - \eta$-system with respect to the $x - y$-system.

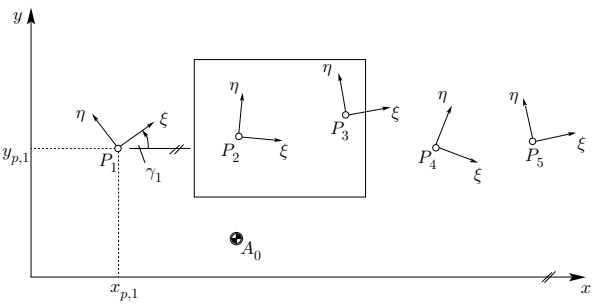


FIGURE 1. Synthesis-task "Motion generation for five precision poses"

The engineer's task is to choose a suitable mechanism being able to solve the task. Then, its precise dimensions have to be determined. Here it is often desirable to perform a *complete synthesis*, i.e., to find all possible realizations of a synthesis task. This gives the engineer the possibility to choose particular mechanisms optimized with respect to additional criteria not reflected in the mathematical description, and to fine-tune. A class of mechanisms suitable for the above task are so-called Stephenson mechanisms, one example of which is shown in Figure 2.

[1]the figures are taken from [**92**]

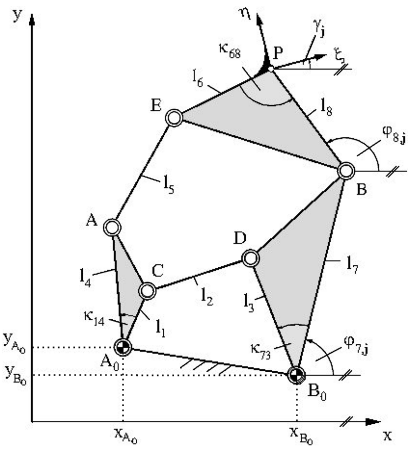


FIGURE 2. Six-bar Stephenson-1I mechanism; kinematic parameters such as lengths of linkages, sizes of angles etc. have to be determined in the *dimensional synthesis* step.

Having chosen the kind of mechanism that is suitable to solve the problem (structural synthesis), in the dimensional synthesis step the unknown kinematic dimensions of the chosen mechanism have to be calculated. Mathematically, the problem leads to a polynomial system that has to be solved either over the real or the complex numbers depending on the formalization. Though both the number of variables and the degrees of the involved equations remain moderate, computing a complete catalogue of solutions in many cases already is demanding. Note that of course not all solutions of the resulting polynomial system are meaningful from an engineering point of view. A first complete dimensional synthesis for Stephenson mechanisms has been performed in [**92**], for a general introduction to solution algorithms for such kinematic problems see [**100**].

An important numerical technique to practically solve polynomial systems are homotopy methods. Here, the basic idea for solving $F(x) = 0$ is to start with another polynomial system G that in a certain sense has a similar structure as F. The idea then is to build a homotopy between G and F and follow the zeros of G numerically to those of F. A typical homotopy used is the linear one $H(x,t) := (1-t)\cdot G(x) + t\cdot F(x), 0 \le t \le 1$. In order to follow this approach the zeros of the starting system should be easily computable.

Homotopy methods for solving polynomial systems are a rich source for many interesting and demanding questions in quite different areas. Their analysis has seen tremendous progress in the last 20 years and is outside the scope of this survey. There will be contributions in this volume by leading experts (which the authors of the present paper are not!) in the area, see the articles by C. Beltrán, G. Malajovich, and M. Shub. We just point to some of the deep results obtained and recommend both the other contributions in this volume and the cited literature as starting point for getting deeper into homotopy methods. Recent complexity analysis for homotopy methods was strongly influenced by a series of five papers in the 1990'ies starting with [**94**] and authored by M. Shub and S. Smale. A question that remained open at the end of this series and now commonly is addressed as Smale's 17th

problem, see [**98**], was the following: 'Can a zero of n complex polynomial equations in n unknowns be found approximately, on the average, in polynomial time with a uniform algorithm?' After a lot of work on the problem involving different authors a major breakthrough was obtained by Beltrán and Pardo, see [**11**, **13**]. They showed how to solve polynomial systems by a uniform randomized homotopy algorithm that runs in polynomial time on the average. Another important progress based on that work was made by Bürgisser and Cucker in [**28**], where a deterministic algorithm for Smale's problem running in pseudo-polynomial time was designed. For a more accurate account on the history of the problem we refer to the survey [**12**].

For the purposes of the present paper we are just interested in some particular aspects arising from the above discussions. They lead into different directions of complexity theory, both with respect to the classical Turing model and real number complexity theory. In the rest of this section we discuss a problem resulting from the above approach that leads to a hard combinatorial optimization problem in classical complexity theory. The following sections then deal with the problem to decide solvability of such polynomial systems; as we shall see this is a task at the heart of real and complex number complexity theory.

For the moment let us restrict ourselves to considering polynomial systems of the form $F : \mathbb{C}^n \mapsto \mathbb{C}^n, F := (f_1, \ldots, f_n)$ over the complex numbers. Here, each component polynomial f_i is supposed to have a degree $d_i \in \mathbb{N}$. Since the system has as many equations as variables it is canonically solvable. For a successful application of homotopy methods the choice of the starting system G is of huge importance. One aspect is that if the zero structure of G significantly differs (for example, with respect to its cardinality) from that of the target system F, then many zeros from G are followed in vain, thus wasting computation time. A common idea for choosing G therefore is to get as far as possible the same number of zeros as F. There are different ways to estimate the number of complex zeros that a canonical system $F : \mathbb{C}^n \to \mathbb{C}^n$ has. A first classical result is Bézout's theorem which upper bounds the number of isolated zeros by $d := \prod_{i=1}^{n} d_i$. Though this number can be easily calculated and a system G with d many isolated zeros is easily found, the disadvantage is that it often drastically overestimates the number of zeros of F. A prominent example is the computation of eigenvalues and -vectors of an (n, n)-matrix M, formulated via the polynomial system $Mx - \lambda x = 0, \|x\|^2 - 1 = 0$ in variables $(x, \lambda) \in \mathbb{C}^{n+1}$. The Bézout number is exponential in n, whereas clearly only n solutions exist.

To repair this disadvantage one might try to use better bounds for the number of solutions. A famous theorem by Bernstein [**15**] determines for *generic* systems the exact number of zeros in $(\mathbb{C}^*)^n$. Though giving the exact number applying this theorem algorithmically suffers from another aspect. In order to compute this bound one has to calculate so-called mixed volumes. The latter is a computational problem that is expected to be even much harder than solving problems in NP because in suitable formulations it leads to $\#P$-hard problems.[2] Thus at least in general one has to be careful whether to compute this bound for the target system F in order to construct G.

A third approach has been used as well, relying on so-called multi-homogeneous Bézout numbers, see [**61**, **80**] for more. Here, the idea is to obtain better estimates

[2]A bit more information about the class $\#P$ can be found at the end of section 4.

by first partitioning the problem's variables into groups and then applying Bézout's theorem to each group. In many cases like the eigenvalue problem mentioned above the resulting bound is much closer to the true number of zeros than it is the case for the Bézout number. However, the question then again is how difficult it is to find an optimal grouping of the variables such that the resulting upper bound is minimal. Though we deal with solving numerically systems of polynomials over the complex numbers, the above question leads to a typical problem about a *combinatorial* optimization problem and thus into the framework of classical complexity theory. This is due to the structure of multi-homogeneous Bézout numbers. More precisely, the optimal grouping mentioned above only depends on the support of the given system, i.e., the structure of monomials with non-zero coefficients. It is not important how these coefficients look like. As consequence, the problem changes to a purely combinatorial one. The question of how difficult it is to compute the optimal variable partitioning has been answered in [**66**] which gives a hardness result for the problem. It is therefore sufficient to focus on particular polynomial systems, namely systems $F := (f_1, \ldots, f_n) = 0$ in which all f_i have the same support. More precisely, consider $n \in \mathbb{N}$, a finite $A \subset \mathbb{N}^n$ and a polynomial system

$$f_1(z) \;=\; \textstyle\sum_{\alpha\in A} f_{1\alpha} z_1^{\alpha_1} z_2^{\alpha_2} \cdots z_n^{\alpha_n} \quad , \;\ldots\; , \quad f_n(z) \;=\; \textstyle\sum_{\alpha\in A} f_{n\alpha} z_1^{\alpha_1} z_2^{\alpha_2} \cdots z_n^{\alpha_n},$$

where the $f_{i\alpha}$ are non-zero complex coefficients. Thus, all f_i have the same support A. A multi-homogeneous structure is a partition of $\{1, \ldots, n\}$ into k subsets $(I_1, \ldots, I_k)$, $I_j \subseteq \{1, \ldots, n\}$. For each such partition we define the block of variables related to I_j as $Z_j = \{z_i | i \in I_j\}$; the corresponding degree of f_i with respect to Z_j is $d_j := \max\limits_{\alpha\in A} \sum_{l\in I_j} \alpha_l$. It is the same for all polynomials f_i because all have the same support.

Definition 2.1. a) The multi-homogeneous Bézout number with respect to support A and partition $(I_1, \ldots, I_k)$ is the coefficient of $\prod_{j=1}^k \zeta_j^{|I_k|}$ in the formal polynomial $(d_1\zeta_1 + \cdots + d_k\zeta_k)^n$, which is

$$\text{Béz}(A, I_1, \ldots, I_k) = \left(\begin{array}{c} n \\ |I_1| \; |I_2| \; \cdots \; |I_k| \end{array} \right) \prod_{j=1}^{k} d_j^{|I_j|}.$$

Here, we assume the f_i to be not yet homogeneous with respect to variable group Z_j; otherwise, replace d_j's exponent by $|I_j| - 1$.

b) The minimal multi-homogeneous Bézout number for a system having support A is

$$\min_{I \text{ partition}} \text{Béz}(A, I).$$

It is known that this minimal number bounds the number of isolated solutions in a suitable product of projective spaces and trivially is never worse than the Bézout number, see [**64**, **100**] for a proof. Unfortunately, as it is the case with Bernstein's bound computing such an optimal partition is a hard task. Even if one would be satisfied with only approximating the minimal multi-homogeneous Bézout number using an efficient Turing algorithm this is not likely possible. More precisely, the following holds:

Theorem 2.2 ([**66**]). *a) Given a polynomial system $F : \mathbb{C}^n \to \mathbb{C}^n$ with support A there is no polynomial time Turing-algorithm that computes the minimal multi-homogeneous Bézout number for A unless $P = NP$.*

b) The same holds with respect to the task of efficiently approximating the minimal multi-homogeneous Bézout number within an arbitrary constant factor of the minimum.

PROOF. As mentioned already above the task of computing the best variable partition is a purely discrete one because its definition only depends on the discrete structure of the support of the given system. The proof thus shows that an efficient algorithm for any of the two mentioned tasks would result in an efficient algorithm for the 3-colouring problem in graph theory. This problem is well known to be NP-complete in discrete complexity theory. Relating graph colouring with the problem at hand is done by assigning to a given graph G over vertex set V monomials that have the vertices of G as its variables and reflect the presence of edges and triangles in G. Doing this appropriately will result in a polynomial system whose minimal multi-homogeneous Bézout number equals $C := \frac{(3n)!}{n!n!n!}$ in case the graph has a 3-colouring and otherwise is at least $\frac{4}{3}C$. This gives claim a). For the non-approximability result one performs a similar construction which allows to blow up the factor $\frac{4}{3}$ to an arbitrary constant. For this construction, a multiplicative structure of the multi-homogeneous Bézout numbers is exploited. □

In practice this means that one has to decide whether one would prefer a longer pre-computation for getting a better starting system either by using mixed volumes or by determining a suitable multi-homogeneous structure or abstains from such a pre-computation. Choosing a random starting system also in theory is an important alternative here.

A more recent application of multi-homogeneous Bézout numbers can be found in [**7**]. Finally note that they also play some role outside the realm of polynomial equation solving. An example is given in [**37**], where the number of roots is used to bound geometrical quantities such as volume and curvature which is applied to the theory of Linear Programming.

The discussion in this section intended to show the wide range of interesting questions arising from different areas related to polynomial system solving. In engineering, many tasks can be formalized using such systems. Solving them then leads to demanding problems in many different disciplines, ranging from algebraic geometry over numerical analysis to algorithm design and complexity theory. Being the focus of the present paper we concentrate on complexity theory. Above we have seen a question arising from polynomial system solving and being located in the framework of combinatorial opimization. This is a typical area of interest in classical discrete complexity theory, where also (non-)approximability results like the one given in Theorem 2.2 are studied, see [**4**, **51**, **52**, **59**].

However, taking into account domains like $\mathbb{R}$ and $\mathbb{C}$ over which the systems are to be solved nearby other questions arise: Can we design deterministic algorithms that decide whether a general such system has a solution at all in the respective domain? General here in particular means that we do not longer relate the number of variables and polynomials. If such decision algorithms exist what is the intrinsic complexity of this problem, i.e., can we give good lower and upper bounds on the running time of such algorithms? Is there a way to compare related problems with respect to their complexity? These are also typical questions in classical complexity theory when dealing with a class of problems. Since we are interested in real and/or complex number instances the Turing model seems at least not appropriate to deal

with all above questions. Also the homotopy methods mentioned above usually are formulated in a framework in which real numbers are considered as entities and complexity is measured, for example, in terms of Newton steps that are applied to follow the homotopy.

This led Blum, Shub, and Smale [**19**] to introduce a computational model formalizing algorithms over quite general domains together with a related complexity theory. This model will be the central one considered in this paper. In the next section we give a short summary of its definition, focussing on the real and complex numbers as underlying domains.

3. The real number model by Blum, Shub, and Smale

As already mentioned when dealing with algorithms over uncountable structures like $\mathbb{R}$ and $\mathbb{C}$ as they often occur in many areas of mathematics it is quite natural to formulate such algorithms in a computational model which does not take care about a concrete representation of objects in modern computers. Then the real or complex numbers to compute with are considered as entities and each elementary operation on such numbers is supposed to take unit time. Of course, this does not mean that issues related to such a number representation are not important in algorithm design and analysis. But if one focusses on certain aspects of a computational problem, for example, on the number of basic arithmetic operations intrinsically necessary to solve it, this abstraction makes sense. One important new aspect for the algorithmic treatment of algebraic problems is to place them into the framework of a uniform P versus NP question. This has also inspired a lot of further interesting new questions in the area of algebraic complexity, see [**24**, **25**].

In 1989 Blum, Shub, and Smale [**19**] introduced a formal framework that allows to carry over important concepts from classical complexity theory in the Turing machine model to computational models over a large variety of structures. For computations over the real and complex numbers they obtained an analogue of the currently most important open question of classical complexity theory, namely the P versus NP problem. We remark that the Blum-Shub-Smale model was introduced over general ring structures; in case the underlying ring is the finite field $\mathbb{Z}_2$ it gives back the classical Turing model.

We now give a brief introduction into the model, its main complexity classes and then turn to the above mentioned version of a P versus NP question. Full details can be found in [**18**]. We give the basic definitions for real number computations; they easily extend to other structures.

Definition 3.1. A (real) Blum-Shub-Smale (shortly: BSS) machine is a Random Access Machine over $\mathbb{R}$. Such a machine has a countable number of registers each storing a real number. It is able to perform the basic arithmetic operations $\{+, -, *, :\}$ together with branch instructions of the form: is a real number $x \geq 0$? These operations are performed on the input components and the intermediate results; moreover, there is a finite number of constants from the underlying domain used by the algorithm. They are called *machine constants*. In addition, there are instructions for direct and indirect addressing of registers. A BSS machine M now can be defined as a directed graph. Each node of the graph corresponds to an instruction. An outgoing edge points to the next instruction to be performed; a branch node has two outgoing edges related to the two possible answers of the

test. Such a machine handles finite sequences of real numbers as inputs, i.e., elements from the set $\mathbb{R}^\infty := \bigcup_{k\in\mathbb{N}} \mathbb{R}^k$. Similarly, after termination of a computation it outputs an element from $\mathbb{R}^\infty$ as result.

A machine does not necessarily terminate for each of the suitable inputs. For computations over other structures one has to adjust the set of operations that can be performed accordingly. For example, when computing with complex numbers there is no ordering available, therefore tests are of the form: is a complex number $z = 0$? In a more formal treatment of the model one additionally has to specify how inputs are presented to the machine in form of a start configuration and how a terminal configuration leads to the output. This can easily be done by specifying a set of registers in which an input is placed and others where the result of a computation has to be delivered. However, being almost straightforward we skip to go through the related formalism and refer instead once more to [**18**].

The problems we are mainly interested in are decision problems.

Definition 3.2. A set $A \subseteq \mathbb{R}^\infty$ is called *real decision problem*. It is called *decidable* if there is a real BSS algorithm that decides it, i.e., given an input $x \in \mathbb{R}^\infty$ the algorithm terminates with result 1 in case $x \in A$ and result 0 otherwise.

The problem is *semi-decidable* if the algorithm stops for all $x \in A$ with result 1 but computes forever for inputs $x \notin A$. Similarly for complex decision problems.

Before turning to complexity issues one natural question in computability theory is whether there exist decision problems that cannot be decided at all by an algorithm in the respective model.

Definition 3.3 (Real Halting Problem). The *real Halting Problem* $\mathbb{H}_\mathbb{R}$ is the following decision problem: Given a code $c_M \in \mathbb{R}^\infty$ of a real BSS machine M and an $x \in \mathbb{R}^\infty$, does machine M stop its computation on input x?

The Halting Problem was one of the first that has been shown to be undecidable in the real number model in [**19**]. There are further problems shown to be undecidable by simple topological arguments. Recall that the Mandelbrot set $\mathcal{M}$ is defined as the set of those $c \in \mathbb{C}$ whose iterates under the map $z \mapsto z^2 + c$ remain bounded when starting the iteration in $z = 0$.

Theorem 3.4 ([**19**]). *The following problems are undecidable in the real number model: The real Halting problem $\mathbb{H}_\mathbb{R}$, the problem $\mathbb{Q}$ to decide whether a given real number is rational, the Mandelbrot set $\mathcal{M}$ seen as subset of $\mathbb{R}^2$. Moreover, $\mathbb{H}_\mathbb{R}, \mathbb{Q}$ and the complement of $\mathcal{M}$ in $\mathbb{R}^2$ are semi-decidable.*

In the complex BSS model, the corresponding complex version of the Halting problem is undecidable. The same holds for deciding the integers $\mathbb{Z}$. Both problems are semi-decidable.

Proof. For proving undecidability of $\mathbb{H}_\mathbb{R}$ in a first step one constructs a universal BSS machine, i.e, a machine U that takes as its input pairs (c_M, x), where $c_M \in \mathbb{R}^\infty$ codes a BSS machine M as element in $\mathbb{R}^\infty$ and $x \in \mathbb{R}^\infty$ is an input for this machine M. Machine U on such an input simulates the computation of M on x. The computational model is strong enough to guarantee U's existence, though the precise construction is tedious. Now, undecidability is obtained by a typical diagonalization argument in which x is taken to be c_M. Semi-decidability easily follows from performing U's computation. If the universal machine halts the input belongs to $\mathbb{H}_\mathbb{R}$ by definition, otherwise not. The argument over $\mathbb{C}$ is the same.

For the other two real number problems undecidability follows from the topological structure of the respective problems. First, every semi-decidable set in $\mathbb{R}^\infty$ is an at most countable union of semi-algebraic sets.[3] This follows from the algebraic structure of the basic operations allowed in algorithms. Both the complement of $\mathbb{Q}$ in $\mathbb{R}$ and the Mandelbrot set are known not to be such a countable union, so it follows these that sets cannot be semi-decidable. But since decidability of a problem A is equivalent to semi-decidability of both A and its complement both problems can neither be decidable. Semi-decidability of $\mathbb{Q}$ is straightforward by enumerating $\mathbb{Q}$, that of $\mathbb{R}^2 \setminus \mathcal{M}$ follows immediately from $\mathcal{M}$'s definition: As soon as an iterate of an input $c \in \mathbb{R}^2$ in absolute value becomes larger than 2 this c belongs to $\mathcal{M}$'s complement. This condition as well characterizes the complement.

As to undecidability of the integers over $\mathbb{C}$ another frequently used topological argument is helpful. Consider a potential machine deciding the problem. Then any input x^* that is algebraically independent of the extension field obtained when joining the complex machine constants to $\mathbb{Q}$ must be branched along the not-equal-alternative of each test node. This computation path must be finite. But the set of inputs that are branched at least once along an equal-alternative is finite by the fundamental theorem of algebra. Thus there must exist integers for which the machine uses the same computation path and gives the same result as for x^*. On such integers the computed result is false and thus the machine has to fail. □

The above statements for $\mathbb{Q}$ and $\mathbb{Z}$ are closely related to so called definability issues in real and algebraically closed fields, see [**20**]. We shall exploit similar arguments again below when analysing computationally universal problems in section 4.3.

Next, algorithms should be equipped with a time measure for their execution. As usual, in order to then classify problems with respect to the running time needed to solve them one also has to define the size of an instance. The time consumption is considered as function in the input size. The intuitive approach for measuring the algebraic complexity of a problem described at the beginning of this section is now made more precise as follows.

Definition 3.5. Let M be a real BSS machine. The *size* of an element $x \in \mathbb{R}^k$ is $size_{\mathbb{R}}(x) := k$. The *cost* of each basic operation is 1. The cost of an entire computation is the number of operations performed until the machine halts. The (partial) function from $\mathbb{R}^\infty$ to $\mathbb{R}^\infty$ computed by M is denoted by Φ_M. The cost of M's computation on input $x \in \mathbb{R}^\infty$ is also called its *running time* and denoted by $T_M(x)$. If $\Phi_M(x)$ is not defined, i.e., M does not terminate on x we assign the running time $T_M(x) := \infty$.

Most of the well known Boolean time-complexity classes can now be defined analogously over the reals. We give a precise definition of the two main such classes.

Definition 3.6 (Complexity classes, completeness).

a) A problem $A \subseteq \mathbb{R}^\infty$ is in class $\mathrm{P}_{\mathbb{R}}$ (*decidable in polynomial time over* $\mathbb{R}$) iff there exist a polynomial p and a real BSS machine M deciding A such that $T_M(x) \leq p(size_{\mathbb{R}}(x))\ \forall x \in \mathbb{R}^\infty$.

[3] A semi-algebraic set in $\mathbb{R}^n$ is a finite Boolean combination of sets defined as solution of finitely many polynomial equalities and inequalities.

b) A is in $\mathrm{NP}_{\mathbb{R}}$ (*verifiable in non-deterministic polynomial time over* $\mathbb{R}$) iff there exist a polynomial p and a real BSS machine M working on input space $\mathbb{R}^\infty \times \mathbb{R}^\infty$ such that
   (i) $\Phi_M(x,y) \in \{0,1\}\ \forall x \in \mathbb{R}^\infty, y \in \mathbb{R}^\infty$
   (ii) $\Phi_M(x,y) = 1 \implies x \in A$
   (iii) $\forall x \in A\ \exists y \in \mathbb{R}^\infty\ \Phi_M(x,y) = 1$ and $T_M(x,y) \leq p(size_{\mathbb{R}}(x))$

c) A problem A in $\mathrm{NP}_{\mathbb{R}}$ is $\mathrm{NP}_{\mathbb{R}}$-*complete* iff every other problem in $\mathrm{NP}_{\mathbb{R}}$ can be reduced to it in polynomial time. *Polynomial time reducibility* from problem B to problem A means: There is a polynomial time computable function $f : \mathbb{R}^\infty \to \mathbb{R}^\infty$ which satisfies: $\forall x \in \mathbb{R}^\infty : x \in B \Leftrightarrow f(x) \in A$. This type of reduction is also called polynomial time many one reduction.

d) The corresponding definitions over $\mathbb{C}$ lead to classes $\mathrm{P}_{\mathbb{C}}, \mathrm{NP}_{\mathbb{C}}$, and $\mathrm{NP}_{\mathbb{C}}$-completeness.

When talking about a problem $A \in \mathrm{NP}_{\mathbb{R}}$, for an input $x \in A$ the y whose existence is required in part b,ii) above can be seen as a proof of x's membership in A. The definition then requires that correctness of this proof can be checked efficiently in the size of x. Below we often use the phrase that on input x machine M guesses a proof y for establishing $x \in A$.

The definition directly implies that $\mathrm{P}_{\mathbb{K}}$ is included in $\mathrm{NP}_{\mathbb{K}}$ for $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$. The currently most important open question in real and complex number complexity theory is whether these inclusions are strict. This is easily seen to be equivalent to the existence of already one single $\mathrm{NP}_{\mathbb{K}}$-complete problem which does not belong to the corresponding class $\mathrm{P}_{\mathbb{K}}$.

The following closely related two problems turn out to be extremely important for the entire theory and will occur in one or the other form throughout the rest of this paper.

Definition 3.7. Let $\mathbb{K}$ be a field of characteristic 0.

a) The Hilbert-Nullstellensatz problem is the problem to decide whether a given system of polynomial equations

$$p_1(x_1, \ldots, x_n) = 0\ , \ldots,\ p_m(x_1, \ldots, x_n) = 0,$$

where all p_i are polynomials in $\mathbb{K}[x_1, \ldots, x_n]$ has a common solution in $\mathbb{K}^n$.

We denote the problem by $\mathrm{QPS}_{\mathbb{K}}$ for Quadratic Polynomial Systems if, in addition, all p_i have a total degree bounded by 2.

b) If $\mathbb{K} = \mathbb{R}$ the feasibility problem 4-$FEAS_{\mathbb{R}}$ is the task to decide whether a polynomial $f \in \mathbb{R}[x_1, \ldots, x_n]$ of total degree at most 4 has a zero in $\mathbb{R}^n$.

We shall see that the above problems in the BSS models over $\mathbb{R}$ and $\mathbb{C}$, respectively, take over the role of the famous 3-SAT problem in the Turing model.

Example 3.8. The following problems are easily seen to belong to the respective class NP over $\mathbb{R}$ or $\mathbb{C}$.

a) $\mathrm{QPS}_{\mathbb{K}}$ belongs to $\mathrm{NP}_{\mathbb{K}}$ for $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$, 4-$FEAS_{\mathbb{R}}$ belongs to $\mathrm{NP}_{\mathbb{R}}$. In all cases the verification procedure guesses a common zero of the given system or the given polynomial, respectively. The polynomials then are evaluated in this point and it is finally checked whether the guess actually was a zero. The (algebraic) size of the guess equals the number of variables the polynomials depend on. The evaluation procedure obviously only needs a number of arithmetic steps and tests that is polynomially bounded in the input size of the system. For the latter we

take a dense representation, i.e., also zero-coefficients for monomials not present contribute to the size by 1. This in principle could be done differently, but here we want to avoid discussions about sparse polynomials.

As an easy example of a polynomial time reduction note that $\mathrm{QPS}_{\mathbb{R}}$ straightforwardly is reducible to 4-$FEAS_{\mathbb{R}}$ by defining an instance of the latter as the sum of the squared degree-2-polynomials of the given $\mathrm{QPS}_{\mathbb{R}}$ instance. Obviously, over $\mathbb{C}$ this reduction does not work correctly.

b) Another example of a problem in $\mathrm{NP}_{\mathbb{R}}$ is the Linear Programming problem $\mathrm{LP}_{\mathbb{R}}$. Here, the input is a real (m,n)-matrix A together with a vector $b \in \mathbb{R}^m$. The question to decide is whether there is a real solution $x \in \mathbb{R}^n$ satisfying $A \cdot x \leq b$. The input size is $O(mn+m)$, a verification proof once again guesses a potential solution x and then verifies whether it solves the system. The problem, when restricted to rational input data and considered in the Turing model, is known to be decidable in polynomial time. This is the well known result implied by the ellipsoid and the interior-point methods. The running time of those algorithms, however, are polynomial in the input size only because the discrete input size is larger than the algebraic one, taking into account the bit-length necessary to represent the rational data. It is a major open question in the theory of Linear Programming whether a polynomial time algorithm also exists in the real number model [**98**].

By introducing slack variables and reducing the number of variables per equation to at most 3 by using additional variables, the real Linear Programming problem is polynomial time reducible to $\mathrm{QPS}_{\mathbb{R}}$. Even though it is currently open whether $\mathrm{LP}_{\mathbb{R}} \in \mathrm{P}_{\mathbb{R}}$ it is not expected that the problem becomes much harder in terms of complexity classes it belongs to, for example, becoming $\mathrm{NP}_{\mathbb{R}}$-complete. This would have strange consequences [**69**]. So $\mathrm{LP}_{\mathbb{R}}$ might well be a kind of intermediate problem between $\mathrm{P}_{\mathbb{R}}$ and $\mathrm{NP}_{\mathbb{R}}$-complete ones. However, even the theoretical existence of such 'intermediate' problems is currently open. We comment on this point once again below after Theorem 4.12.

In order to justify the importance of the $\mathrm{NP}_{\mathbb{R}}$-completeness notion first it has to be shown that such problems exist.

THEOREM 3.9 ([**19**]). *For $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$ the Hilbert-Nullstellensatz problem $\mathrm{QPS}_{\mathbb{K}}$ is $\mathrm{NP}_{\mathbb{K}}$-complete. Over the reals the same holds for 4-$FEAS$.*

PROOF. The proof in principle follows the structure of the one for Cook's theorem, i.e., the proof of NP-completeness of the 3-SAT problem in the Turing model. However, certain adjustments to the framework are needed. Given a problem $A \in \mathrm{NP}_{\mathbb{R}}$, a BSS machine M witnessing this membership, and an input $x \in \mathbb{R}^\infty$ a quadratic polynomial system has to be constructed that is solvable if and only if $x \in A$. The system is obtained by representing M's possible computation on x and a suitable guess y by a rectangular matrix. Rows represent the time steps of M during computations, columns represent the registers in which input, guess and intermediate results are stored. Given the polynomial running time of M this matrix has polynomial size only. Now for each assignment to the matrix with real numbers one tries to express by polynomial equations that the first row's assignment corresponds to the input configuration for M on x, each row's assignment implies that of the next row by applying a computational step of M, and the last row-assignment indicates that M's computation is accepting. Then $x \in A$ if and

only if a suitable assignment of the matrix exists if and only if the generated $\mathrm{QPS}_{\mathbb{R}}$ instance is solvable. □

The list of complete problems still is relatively small compared to the thousands of discrete problems that since Cook's theorem have been established to be NP-complete. For some further completeness results related to $\mathrm{NP}_{\mathbb{R}}$ and $\mathrm{NP}_{\mathbb{C}}$, respectively, we refer (non-exhaustively) to [**27**, **36**, **57**, **58**, **90**].

An important difference between $\mathrm{NP}_{\mathbb{R}}$ and its discrete counterpart NP is the fact that the guess y in the above definition of $\mathrm{NP}_{\mathbb{R}}$ is taken from an uncountable space. This is very much different to the classical setting where the search space for a correct membership proof is an element in $\{0,1\}^*$, i.e., a finite bit-string. Since the length of the guess is polynomially bounded in the length of the input, over finite alphabets the search space is finite, implying that each problem in NP is decidable in single exponential time. Over the reals and the complex numbers this turns out to be true as well relying on much deeper results.

THEOREM 3.10. *Let $d \in \mathbb{N}$ and let A be a (basic) semi-algebraic set that is given as*

$$A := \{x \in \mathbb{R}^n | p_i(x) \Delta_i 0, 1 \leq i \leq s\} ,$$

where each p_i is a polynomial of degree at most d with real coefficients and $\Delta_i \in \{=, \neq, \geq, >\}$.

Then emptiness of A can be decided by an algorithm in the BSS model that runs in $O\left((s \cdot d)^{O(n)}\right)$ arithmetic steps, i.e., in single exponential time.

A similar statement is true for the complex numbers. It follows that both all problems in $\mathrm{NP}_{\mathbb{R}}$ and in $NP_{\mathbb{C}}$ can be decided by a BSS algorithm of the respective model in single exponential time.

The proof of this theorem is out of the scope of this paper. It is a special case of Quantifier Elimination (the existential quantifiers are removed by the decision procedure), a question having a long tradition for real and algebraically closed fields. The first procedure for general quantifier elimination in these structures was given by Tarski [**102**]. Then, starting in the 1970'ies research has focussed on getting better complexity estimates, until finally for existentially quantified problems single exponential complexity bounds could be established. Significant contributions to the complexity of such methods have been made, for example, in [**9**, **47**, **49**, **88**]. The above result in this particular form if taken from [**88**]. Once again, with M. Giusti and J. Heintz two experts in complexity aspects of elimination theory will contribute with related issues to this volume, so the interested reader should consult their article [**6**].

Thus we arrived at some first cornerstones in BSS complexity theory. There are problems that are algorithmically undecidable in the model. And there is a reasonable theory of efficient algorithms and hard problems in form of the $\mathrm{P}_{\mathbb{R}}$ versus $\mathrm{NP}_{\mathbb{R}}$ question. Its importance is justified by the existence of natural $\mathrm{NP}_{\mathbb{R}}$-complete problems, and all such problems can be decided within the algorithmic framework in single exponential time. It is currently not known whether more efficient algorithms exist. Similar statements are true in the complex number model. In the following sections we shall discuss several structural questions taking their starting points in the above results.

4. Structural complexity

In this section we exemplify typical methods and questions analysed in structural complexity theory for the BSS model over $\mathbb{R}$ and $\mathbb{C}$ on the basis of three thematic areas. These topics include transfer theorems, the structure inside $\mathrm{NP}_{\mathbb{R}}$, and recursion theory on $\mathbb{R}$.

4.1. Transfer principles for P versus NP. One of the research lines from the beginning of real and complex number complexity theory was to study similarities and differences between classical complexity theory for the Turing model dealing with finite alphabets and complexity theory in alternative than discrete structures. Defining literally the main problem whether the classes P and NP coincide is an easy task once the underlying model has been specified, but this of course does not automatically make the question interesting. Nevertheless, it fortunately turned out to be the case that several non-trivial problems arise in such alternative models. The most prominent one certainly again is the P versus NP question, this time over $\mathbb{R}$ and $\mathbb{C}$.

It is natural to wonder whether the answer(s) to this question are related for different structures. More generally, it seems interesting to combine major open complexity theoretic question in one computational model with related questions in another. Ideally, this enlarges the tools of methods that could be used to attack such open problems. Results following this guideline are usually called transfer theorems. This subsection intends to highlight some such transfer results relating different versions of the BSS model with classical complexity theory.

Of course in general it is not clear what difficulties one meets when studying a problem over continuous domains which literally is similar to the respective question in the Turing model. Sometimes, an easy solution over the reals is possible because new arguments can be used that are not applicable in the Turing model. Sometimes, a deep analysis combines different models, and sometimes, new interesting questions arise which do not have a significant counterpart in the discrete world. All of that has been observed, and the present section tries to outline some results into these directions.

Note that we saw already one good example where a seemingly similar question needs much more efforts to be solved in the real number model. The decidability of all problems in class $\mathrm{NP}_{\mathbb{R}}$ mentioned in the previous section relies on the non-trivial task to perform quantifier elimination in real closed fields. And obtaining a comparable time bound to the discrete world, where NP trivially can be decided in simple exponential time, becomes even more difficult.

An opposite example where a question being extremely difficult in the Turing setting turned out to be much easier in the BSS framework is the following early result by Cucker on non-parallelizability of the class $\mathrm{P}_{\mathbb{R}}$. The real number class $\mathrm{NC}_{\mathbb{R}}$ used in the statement intuitively is defined as those decision problems in $\mathrm{P}_{\mathbb{R}}$ that can be parallelized. This means they can be solved using a polynomial number of BSS machines working in parallel which but all running in poly-logarithmic time only. The theorem states that not all problems in $\mathrm{P}_{\mathbb{R}}$ can be parallelized that way.

THEOREM 4.1 ([**33**]). *$NC_{\mathbb{R}} \subsetneq P_{\mathbb{R}}$*

PROOF. The proof relies on an irreducibility argument by defining a decision problem whose solution requires to compute an irreducible polynomial of exponential degree. It is then shown that this computation basically cannot be split into different parallel branches. □

The above argument relies on the structure of irreducible varieties over $\mathbb{R}$. Thus, it cannot be applied to the analogue question for the Turing model. It is still is one of the major open problems in classical complexity theory.

Comparing different models is most interesting when dealing with problems that somehow can be considered in both models. This often can be achieved by restricting the set of input instances. An important example is the Hilbert Nullstellensatz problem. Given a system of polynomial equations as input on the one hand side can be considered as problem for both $\mathrm{NP}_{\mathbb{R}}$ and $\mathrm{NP}_{\mathbb{C}}$, depending on the set of coefficients. Solvability accordingly can be required either over the reals or the complex numbers. If we restrict coefficients to be rationals or integers the question as well makes sense in the Turing model, even when asking for real solutions. Moreover, if one is interested in $\{0,1\}$-solutions this can be forced by binding each single variable x through an additional equation $x\cdot(x-1)=0$. Note that also in the Turing model it is an NP-complete task to decide whether a system of quadratic equations with integer coefficients has a real or complex solution, respectively.

Thus, for a comparison of these models an important question to solve is: Suppose, the real (or complex) QPS problem could be decided by an efficient algorithm. Could this algorithm somehow be turned into a Turing algorithm such that the changed algorithm turns out to be efficient as well for the discrete variant of QPS? Clearly, a main aspect for attacking this question is whether the potentially real or complex machine constants which the given algorithm uses could be replaced somehow in order to obtain a Turing algorithm. This topic of replacement of machine constants has turned out to be extremely interesting and demanding. In the results below such a replacement in one or the other way always is crucial. The resulting effects can be quite different. For some tasks constants can be replaced without much harm to complexity aspects, for some others such a replacement introduces new aspects like non-uniformity of algorithms, and there are situations where it remains an open question whether a replacement is possible.

A first deep result dealing with such issues was given in [**17**]. Here, the QPS problem is studied over arbitrary algebraically closed fields of characteristic 0. The proof of Theorem 3.9 shows that QPS defined accordingly is complete for the corresponding class NP in all such fields. Thus it is natural to ask whether an answer to any of the related P versus NP problems would have implications for the other fields as well. In fact, this is true.

THEOREM 4.2 ([**17**]). *For all algebraically closed fields* $\mathbb{K}$ *of characteristic* 0 *the* P *versus* NP *question has the same answer in the BSS model over* $\mathbb{K}$*, i.e., either in all these fields* $\mathrm{NP}_{\mathbb{K}}=\mathrm{P}_{\mathbb{K}}$ *or in all such fields* $\mathrm{P}_{\mathbb{K}}$ *is strictly contained in* $\mathrm{NP}_{\mathbb{K}}$.

PROOF. The theorem's first proof in [**17**] performs the elimination of constants using number theoretic arguments. We outline an alternative proof given by Koiran [**56**] which is based on results on quantifier elimination. A first observation using the introductory remarks before the statement of the theorem shows that a central problem to consider is QPS over the algebraic closure $\bar{\mathbb{Q}}$ of the rational number field. Since each field $\mathbb{K}$ under consideration has to contain $\bar{\mathbb{Q}}$ it suffices to analyse

the QPS problem over $\mathbb{K}$ and over $\bar{\mathbb{Q}}$. There are then two directions to prove; the first asserts that the existence of an efficient algorithm for a hard problem over $\mathbb{K}$ implies the same for a hard problem over $\bar{\mathbb{Q}}$. This is the more difficult statement. The converse direction states that an efficient algorithm for the Hilbert Nullstellensatz problem in $\bar{\mathbb{Q}}$ can be lifted to one for the same problem over $\mathbb{K}$. It is true by well known results from model theory, basically applying the so called strong transfer principle for the theory of algebraically closed fields. This was first done by Michaux [**79**]. Since its proof does not rely on techniques for eliminating machine constants we do not go into more details here.

Let us thus focus on the other direction. Note that a QPS instance with coefficients from $\bar{\mathbb{Q}}$ has a solution over $\mathbb{K}$ if and only if it has a solution over $\bar{\mathbb{Q}}$. This follows from Hilbert's Nullstellensatz. Below we choose $\mathbb{K} := \mathbb{C}$, but the arguments remain basically the same for any other $\mathbb{K}$.

Suppose then there were an efficient algorithm solving QPS over $\mathbb{C}$, i.e., proving $\mathrm{P}_{\mathbb{C}} = \mathrm{NP}_{\mathbb{C}}$. This algorithm also solves QPS over $\bar{\mathbb{Q}}$ efficiently, but in order to conclude $\mathrm{P}_{\bar{\mathbb{Q}}} = \mathrm{NP}_{\bar{\mathbb{Q}}}$ the algorithm is not allowed to use constants from $\mathbb{K} \setminus \bar{\mathbb{Q}}$. Suppose the potential decision algorithm uses transcendental constants; with a moderate technical effort one can additionally assume without loss of generality that all these machine constants are algebraically independent. For the computation on a fixed input from $\bar{\mathbb{Q}}$ one can view each equality test performed by the algorithm as a polynomial with coefficients in $\bar{\mathbb{Q}}$ that is evaluated in the set of transcendental machine constants. Thus, no such algebraic equality test is answered positively in a reasonably small neighbourhood of the set of machine constants. Consequently, for all such points the machine computes the same yes-no answers. The task is then to find a rational point in such a neighbourhood. A clever application of the complexity statements behind Theorem 3.10 for $\mathbb{C}$ guarantees such points to exist and being not too large. 'Not too large' here means that they can be computed fast enough starting from the constant 1 by a machine working over $\bar{\mathbb{Q}}$. Thus, replacement of transcendental constants by efficiently computable algebraic ones can be accomplished and an efficient algorithm for the $\mathrm{NP}_{\bar{\mathbb{Q}}}$-complete problem QPS is found. □

The theorem unfortunately does not solve the P versus NP problem in any of those structures but just guarantees the currently unknown answer to be the same for all related fields. The next transfer theorem discussed relates the P versus NP question in the complex BSS model with randomized complexity classes in classical complexity. Here, the well known class BPP denotes decision problems L that can be solved by randomized polynomial time algorithms allowing a small two-sided error, i.e., the procedure might fail both on elements in L and its complement with a small constant probability. BPP thus stands for bounded error probability polynomial time. The precise placement of class BPP in discrete complexity theory with respect to its relations to P and NP is a major open problem. Though recent results have led to the reasonable possibility that BPP equals P, it is not even known whether BPP is a proper subset of the set of problems that can be solved in non-deterministic exponential time, see [**1**]. The next result shows an interesting connection between the complex BSS model and BPP. The main ingredients for its proof were independently given by Koiran [**54**] and Smale [**97**], though in the cited sources the theorem seems not outspoken explicitly.

THEOREM 4.3 ([**54**, **97**]). *Suppose* $P_{\mathbb{C}} = NP_{\mathbb{C}}$ *in the complex number BSS model, then* NP $\subseteq$ BPP *in the Turing model.*

PROOF. Once again, the main idea is to extract from an efficient complex algorithm for $\mathrm{QPS}_{\mathbb{C}}$ a randomized Turing algorithm for a suitable NP-complete variant of QPS. In order to replace non-rational constants used by the given algorithm randomization enters at two places. First, relying once more on arguments like those used in the previous theorem one tries to find small rational constants that could be used instead of the original ones. These constants are chosen by random from a large enough set and their appropriateness with high probability is established by using the famous Schwartz-Zippel Lemma [**111**]. However, even if the new rational coefficients work fine, it might be the case that intermediate results produced in the initial $\mathrm{P}_{\mathbb{C}}$ algorithm get too large when counting bit-operations in the Turing model. This is solved by doing all computations modulo randomly chosen integers located in a suitable set. For most of them the computation then still works correctly, but now running in polynomial time as well in the Turing model. □

Even though the relation between BPP and NP is currently unknown nobody expects NP $\subseteq$ BPP to be true. The inclusion would have dramatical consequences concerning complexity classes above NP, and here foremost the collapse of the so called polynomial hierarchy. So if one could prove that this hierarchy does not collapse in classical complexity theory it would follow $\mathrm{P}_{\mathbb{C}} \neq \mathrm{NP}_{\mathbb{C}}$ in the complex number model.

The two previous results are not known to hold for the real numbers as well. The attempt to obtain transfer results here seems to meet more obstacles. We shall encounter this phenomenon once again in the next subsection. It is then natural to first consider restrictions of the real number model in order to figure out whether more could be said for such restricted models. In addition, this might shed more light on where the difficulties lie.

The first such restriction considered here is called additive BSS model. The difference with the full real model is that only additions and subtractions are allowed as arithmetic operations. For the following discussion we also restrict ourselves to so called *constant-free* additive algorithms, i.e., there are no other machine constants used than 0 and 1. Nevertheless note that for an $\mathrm{NP}_{\mathbb{R}}^{add}$ verification algorithm it is still allowed to work with real guesses. Similar results as those described below can be obtained as well if arbitrary constants are allowed. We comment on that at the end of this subsection.

In the additive model classes $\mathrm{P}_{\mathbb{R}}^{add}$ and $\mathrm{NP}_{\mathbb{R}}^{add}$ are defined analogously to the full model.[4] Algorithms in the additive model still can work with inputs being vectors of real numbers. However, when inputs are restricted to stem from $\{0,1\}^*$, each additive computation can be simulated with only polynomial slow down by a Turing machine. This is true because the sizes of intermediate results in such a computation cannot grow too much, in contrast to the case in the full model when repeated squaring of a number is performed. The following theorem by Fournier and Koiran shows that proving lower bounds in the additive model is of the same difficulty as in classical complexity theory.

[4]In literature the constant-free classes usually are denoted with an additional superscript 0. We skip that here in order to minimize the notational overhead.

THEOREM 4.4 ([**41**]). *It is* P = NP *in the Turing model if and only if it holds* $P_{\mathbb{R}}^{add} = NP_{\mathbb{R}}^{add}$ *in the additive (constant-free) model over* $\mathbb{R}$.

PROOF. In a first step one analyses the power of additive machines on discrete languages. For $L \subseteq \mathbb{R}^*$ one denotes it Boolean (i.e., discrete) part as $BP(L) := L \cap \{0,1\}^*$, and similarly for entire complexity classes. The above argument on a moderate growing of intermediate results implies the equality $BP(\mathrm{P}_{\mathbb{R}}^{add}) = \mathrm{P}$. The analogue equality $BP(\mathrm{NP}_{\mathbb{R}}^{add}) = \mathrm{NP}$ is true as well, though for proving it one first has to show that guessing real components in a verification proof can be replaced by guessing small rational components. This is only known to be true in the additive model, for the full BSS model it is an open question and conjectured to be false. These observations suffice to show the easier direction, namely that $\mathrm{P}_{\mathbb{R}}^{add} = \mathrm{NP}_{\mathbb{R}}^{add}$ implies P = NP. The difficult one is the converse, and as usual we only outline the main proof ingredients. Suppose that P = NP. The idea is to show that any problem in $\mathrm{NP}_{\mathbb{R}}^{add}$ can be efficiently decided by an additive machine which has access to a discrete oracle for NP. The latter means that the algorithm is allowed to generate questions to a classical NP-complete problem and gets a correct answer at unit cost.[5] Since we assume P = NP such an oracle device can be replaced by an efficient algorithm in the Turing model, which of course is efficient as well in the additive model. This would yield the assertion. The design of this oracle algorithm is the heart of the proof. It relies on a deep result by Meyer auf der Heide [**77**, **78**] on point location for arrangements of hyperplanes. This result establishes how to construct non-uniformly a so called linear decision tree for solving the point location problem. The proof shows how this algorithm can be made uniform if an NP oracle is available. We only outline it very roughly here. Given a problem $L \in \mathrm{NP}_{\mathbb{R}}^{add}$ the non-deterministic additive algorithm generates an exponential family of hyperplanes describing membership in L. These hyperplanes arise from accepting computations, and since it suffices to guess small rational numbers only in non-deterministic algorithms the coefficients of those hyperplanes remain small rational numbers. Moreover, the set of hyperplanes decomposes the respective part of the input space $\mathbb{R}^n$ into regions each of which either belongs to L or its complement. The main part of the proof now shows that an additive machine which is allowed to use a classical NP oracle can solve the following task: For an input $x \in \mathbb{R}^n$ it computes a set S described by few affine inequalities with small coefficients such that $x \in S$ and S is either completely contained in L or in its complement. The construction of S needs Meyer auf der Heide's results in a clever way, using at several stages the oracle. The final decision whether $S \subseteq L$ or not again is decided by means of the NP oracle. □

The theorem shows that there are deep relations between major open questions in classical complexity theory and real number models. If additive machines are allowed to use real constants similar results have been proved in the same paper [**41**] relying on results from [**34**, **55**]. Basically the use of such constants introduces non-uniformity for discrete problems, that is Boolean parts of the class $\mathrm{P}_{\mathbb{R}}^{add}$ when constants can be used turn out to equal the class P/*poly* in the Turing model; the latter defines problems that can be decided efficiently by additional use of a moderate non-uniformity, see also below. The same is true for $\mathrm{NP}_{\mathbb{R}}^{add}$ and leads to a corresponding version of the previous theorem. Note that further restricting the

[5] Oracle algorithms will be considered once more in Section 4.3.

model, for example by only allowing equality branches and therefore considering $\mathbb{R}$ as unordered vector space does not lead to a similar transfer result. In such a model the corresponding class P provably is a proper subclass of NP, see [**68**].

Of course, it is challenging to extend connections to discrete complexity like the ones shown above to the full real number model as well.

4.2. Inside $\mathrm{NP}_{\mathbb{R}}$ and $\mathrm{NP}_{\mathbb{C}}$. The problems to be considered in this subsection as well require to deal with the machine constants of algorithms and how to replace them by more suitable ones. This time, however, the goal will not be to replace arbitrary constants by rational ones. Instead, a family of constants used non-uniformly should be replaced by a single fixed set of machine constants. Before understanding the task and how the replacement in some situations can be achieved we introduce the problem to be studied now.

Starting point of the investigations is the following classical result by Ladner [**60**], which in the Turing model analyses the internal structure of complexity class NP in case $\mathrm{P} \neq \mathrm{NP}$ is supposed to be true:

THEOREM 4.5 ([**60**]). *Suppose* $\mathrm{NP} \neq \mathrm{P}$. *Then there are problems in* $\mathrm{NP} \setminus \mathrm{P}$ *which are not* NP*-complete under polynomial time many-one reductions.*

PROOF. The proof relies intrinsically on the countability of both the family $\{P_1, P_2, \ldots\}$ of polynomial time Turing machines and the family $\{R_1, R_2, \ldots\}$ of polynomial time reduction machines in the Turing model. A diagonalization argument is performed to fool one after the other each machine in the two sets. This is briefly done as follows. Given an NP-complete problem L one constructs a problem $\tilde{L} \in \mathrm{NP}$ such that all machines R_i fail to reduce L to $\tilde{L}$ on some input and all machines P_i fail to decide $\tilde{L}$ correctly on some input. Towards this aim the definition of $\tilde{L}$ proceeds dimension-wise while intending to fool step by step $P_1, R_1, P_2, R_2, \ldots$. In order to fool an P_i the language $\tilde{L}$ is taken to look like L for inputs of sufficiently large size. Conversely, in order to fool reduction algorithm R_i for sufficiently many of the following input-sizes $\tilde{L}$ is defined to look like an easy problem. Both steps together imply that none of the machines P_i, R_i works correctly for the new language $\tilde{L}$. Finally, a typical padding argument guarantees $\tilde{L} \in NP$. □

Extensions of Ladner's result can be found, for example, in [**91**].

Considering computational models over uncountable structures like $\mathbb{R}$ and $\mathbb{C}$ the above diagonalization argument - at least at a first sight - fails since the corresponding algorithm classes become uncountable. So it is not obvious whether similar statements hold for $\mathrm{NP}_{\mathbb{R}}$ and/or $\mathrm{NP}_{\mathbb{C}}$. We shall now see that studying this question in the extended framework leads to interesting insights and new open problems.

As it was the case in the previous subsection also for Ladner's problem the complex BSS model is easier to handle than the real model. The first Ladner like result in the BSS framework in [**65**] was shown for the complex classes $\mathrm{P}_{\mathbb{C}}$ and $\mathrm{NP}_{\mathbb{C}}$:

THEOREM 4.6 ([**65**]). *Suppose* $NP_{\mathbb{C}} \neq P_{\mathbb{C}}$. *Then there are problems in* $NP_{\mathbb{C}} \setminus P_{\mathbb{C}}$ *which are not* $NP_{\mathbb{C}}$*-complete under polynomial time many-one reductions in the complex number BSS model.*

PROOF. The proof relies on Theorem 4.2 from the previous subsection. It will be crucial to transfer the question from the uncountable structure $\mathbb{C}$ of complex numbers to the countable one $\bar{\mathbb{Q}}$, the algebraic closure of $\mathbb{Q}$ in $\mathbb{C}$.

In a first step we answer Ladner's problem positively in the BSS model over $\bar{\mathbb{Q}}$. This can be done along the lines of the classical proof sketched above since both families of algorithms mentioned therein are countable. Let $\tilde{L}$ be the diagonal problem constructed.

In order to apply Theorem 4.2 some observations are necessary. They all are immediate consequences of Theorem 3.9 and Tarski's Quantifier Elimination for algebraically closed fields of characteristic 0. First, the Hilbert Nullstellensatz decision problem is $\mathrm{NP}_{\mathbb{K}}$-complete in the BSS model over $\mathbb{K}$ for $\mathbb{K} \in \{\bar{\mathbb{Q}}, \mathbb{C}\}$. The strong transfer principle mentioned already in the proof of Theorem 4.2 implies that an instance over $\bar{\mathbb{Q}}$ is solvable over $\mathbb{C}$ if and only if it is as well solvable already over $\bar{\mathbb{Q}}$. Since the Hilbert Nullstellensatz problem can be defined without additional complex constants Theorem 4.2 can be applied. This allows to lift the diagonal problem $\tilde{L}$ from $\bar{\mathbb{Q}}$ to $\mathbb{C}$ such that the lifted problem has the same properties there. Thus, Ladner's theorem holds as well over the complex numbers. □

Since Theorem 4.2 is not known to be true for the real number model the above proof cannot be applied to show Ladner's result for $\mathrm{NP}_{\mathbb{R}}$. Thus a new idea is necessary. If we could group an uncountable set of real algorithms into a countable partition, then may be one could at least construct diagonal problems for such a partition. But how should a reasonable partition look like?

This idea was first considered by Michaux who introduced the notion of *basic machines* in [**79**].

Definition 4.7. A *basic machine over* $\mathbb{R}$ in the BSS-setting is a BSS-machine M with rational constants and with two blocks of parameters. One block x stands for a concrete input instance and takes values in $\mathbb{R}^\infty$, the other block c represents real constants used by the machine and has values in some $\mathbb{R}^k$ ($k \in \mathbb{N}$ fixed for M).

Basic machines for variants of the BSS model are defined similarly.

Basic machines split the discrete skeleton of an original BSS machine from its real machine constants. That is done by regarding those constants as a second block of parameters. Fixing c we get back a usual BSS machine $M(\bullet, c)$ that uses the same c as its constants for all input instances x. Below, when we speak about the machine's constants we refer to the potentially real ones only.

Basic machines give rise to define a non-uniform complexity class P/const for the different model variants we consider. The non-uniformity is literally weaker than the well-known *P/poly* class from classical complexity theory since the non-uniform advice has fixed dimension for all inputs. In *P/poly* it can grow polynomially with the input size.

Definition 4.8 ([**79**]). A problem L is in class $\mathrm{P}_{\mathbb{R}}$/const if and only if there exists a polynomial time basic BSS machine M and for every $n \in \mathbb{N}$ a tuple $c^{(n)} \in [-1,1]^k \subset \mathbb{R}^k$ of real constants for M such that $M(\bullet, c^{(n)})$ decides L for inputs up to size n.

Similarly for other models.

Note that in the definition $c^{(n)}$ works for all dimensions $\leq n$. The reason for this becomes obvious below. Note as well that assuming all machine constants to be bounded in absolute value is no severe restriction; if a larger constant should be used it can be split into the sum of its integer part and its non-integral part. The integer part then is taken as rational machine constant, thus belonging to the discrete skeleton.

The class P/const turned out to be important in unifying Ladner like results in different models and to get as well a (weaker) real version. The class of basic machines clearly is countable as long as the particular choice of machine constants is not fixed. Thus, in principle we can diagonalize over P/const decision and reduction machines in the different models.

THEOREM 4.9 ([**14**]). *Suppose* $\mathrm{NP}_{\mathbb{R}} \not\subseteq \mathrm{P}_{\mathbb{R}}/\mathrm{const}$. *Then there exist problems in* $\mathrm{NP}_{\mathbb{R}} \setminus \mathrm{P}_{\mathbb{R}}/\mathrm{const}$ *not being* $\mathrm{NP}_{\mathbb{R}}$*-complete under* $\mathrm{P}_{\mathbb{R}}/\mathrm{const}$ *reductions.*

Similarly for the other model variants.

PROOF. The proof again uses the usual padding argument along the classical line. The main new aspect, however, is the necessity to establish that for each basic machine M which is supposed to decide the intended diagonal problem $\tilde{L}$ an input-dimension where M's result disagrees with $\tilde{L}$'s definition can be computed effectively. The condition that M disagrees with $\tilde{L}$ for all possible choices of machine constants can be expressed via a quantified first-order formula. Deciding the latter then is possible due to the existence of quantifier elimination algorithms in the respective structures. □

Since the assumption of Theorem 4.9 deals with $\mathrm{P}_{\mathbb{R}}/\mathrm{const}$ instead of $\mathrm{P}_{\mathbb{R}}$ it gives a non-uniform version of Ladner's result. Note that because of $\mathrm{P}_{\mathbb{R}} \subseteq \mathrm{P}_{\mathbb{R}}/\mathrm{const}$ the theorem's implication also holds for uniform reductions. In order to achieve stronger versions one next has to study the relation between the classes P and P/const. If both are equal, then a uniform version of the theorem follows.

At this point some model theory enters. Very roughly, a structure is called recursively saturated if for each recursive family of first-oder formulas $\{\varphi_n(c) | n \in \mathbb{N}\}$ with free variables c the following holds: if each finite subset of formulas can be commonly satisfied by a suitable choice for c, then the entire family is satisfiable.[6]

THEOREM 4.10 ([**79**],[**14**]). *If a structure is recursively saturated, then it holds* $\mathrm{P} = \mathrm{P}/\mathrm{const}$.

PROOF. Let L be a language in P/const and M the respective basic machine. The proof basically is a combination of the definition of saturation with a reasonable description of M's behaviour on instances up to a given dimension. This description, being folklore in BSS theory, gives the recursive family $\{\varphi_n(c)\}_n$ of formulas required, where n stands for the input dimension and the free variables c for the machine constants taken for the basic machine M. Saturation then implies that a single choice for the machine constants can be made which works for all dimensions. This choice turns M into a uniform polynomial time algorithm for L. □

As a consequence, Ladner's results holds uniformly over structures like $\{0, 1\}$ and $\mathbb{C}$ which are well known to be recursively saturated. Thus, Ladner's original result as well as Theorem 4.6 are reproved.

However, since $\mathbb{R}$ is not recursively saturated – take as family $\varphi_n(c) \equiv c > n$ for $c \in \mathbb{R}$ – the theorem's consequence does not apply to $\mathbb{R}$. So once again the

[6]Recursiveness here is understood in the Turing sense and just requires that one should be able to enumerate the formulas without using additional machine constants. In the present applications the formulas of the family always represent computations of certain basic machines up to a certain dimension. By 'hiding' constants from the underlying computational structure as variables it follows that such a family satisfies the recursiveness assumption. For more details see [**79**]

above technique does not give a uniform analogue of Ladner's result over the reals and additional ideas seem necessary. Due to its importance for the above questions Chapuis and Koiran in [**30**] have undertaken a deep model-theoretic analysis of P/const and related classes. They argue that for the full real model already the equality $\mathrm{P}_{\mathbb{R}} = \mathrm{P}_{\mathbb{R}}/1$ is highly unlikely unless some major complexity theoretic conjecture is violated. Here, $\mathrm{P}_{\mathbb{R}}/1$ is defined by means of basic machines which use a finite number of uniform and a single non-uniform machine constant only. Nevertheless, for the reals with addition and order (additive model) they were able to show once again $\mathrm{P}^{add}_{\mathbb{R}} = \mathrm{P}^{add}_{\mathbb{R}}/\mathrm{const}$ and thus

THEOREM 4.11 ([**30**]). *Suppose $NP^{add}_{\mathbb{R}} \neq P^{add}_{\mathbb{R}}$. Then there are problems in $NP^{add}_{\mathbb{R}} \setminus P^{add}_{\mathbb{R}}$ which are not $NP^{add}_{\mathbb{R}}$-complete.*

Their proof for showing the inclusion $\mathrm{P}^{add}_{\mathbb{R}}/\mathrm{const} \subseteq \mathrm{P}^{\mathrm{add}}_{\mathbb{R}}$ once more makes use of the moderate growth of intermediate results in an additive computation. This allows to bound the size of and compute efficiently and uniformly for each input dimension n a set of rational machine constants $c^{(n)}$ such that the given $\mathrm{P}^{add}_{\mathbb{R}}/\mathrm{const}$-machine works correctly on $\mathbb{R}^{\leq n}$ if $c^{(n)}$ is taken as vector of constants.

This idea is one of the starting points to extend the result to yet another variant of the full real number model named restricted model in [**73**]. In this model, the use of machine constants is restricted in that all intermediate results computed by a restricted algorithm should only depend linearly on the machine constants. In contrast to additive machines input variables can be used without limitation, i.e., they can be multiplied with each other. The motivation of considering this model is that it is closer to the original full real BSS model than the additive one. As one indication for this fact note that the $\mathrm{NP}_{\mathbb{R}}$-complete feasibility problem QPS over $\mathbb{R}$ is $\mathrm{NP}^{\mathrm{rc}}_{\mathbb{R}}$-complete as well in the restricted model, where the superscript rc is used to denote respective complexity classes in the restricted model. Since Theorem 4.9 holds as well here the main task once more is to analyse the relation between $\mathrm{P}^{\mathrm{rc}}_{\mathbb{R}}$ and $\mathrm{P}^{\mathrm{rc}}_{\mathbb{R}}/\mathrm{const}$.

THEOREM 4.12 ([**73**]). *It is $P^{\mathrm{rc}}_{\mathbb{R}} = P^{\mathrm{rc}}_{\mathbb{R}}/\mathrm{const}$. As a consequence, supposing* QPS $\notin NP^{\mathrm{rc}}_{\mathbb{R}}$ *there exist non-complete problems in $NP^{\mathrm{rc}}_{\mathbb{R}} \setminus P^{\mathrm{rc}}_{\mathbb{R}}$.*

PROOF. Crucial for showing $\mathrm{P}^{\mathrm{rc}}_{\mathbb{R}} = \mathrm{P}^{\mathrm{rc}}_{\mathbb{R}}/\mathrm{const}$ is a certain convex structure underlying the set of suitable machine constants. Given a problem $L \in \mathrm{P}^{\mathrm{rc}}_{\mathbb{R}}/\mathrm{const}$ and a corresponding basic machine M using k constants define $E_n \subset \mathbb{R}^k$ as set of constants that can be used by M in order to decide $L \cap \mathbb{R}^{\leq n}$ correctly. It can be shown that without loss of generality the $\{E_n\}_n$ build a nested sequence of bounded convex sets. If the intersection of all E_n is non-empty any point in it can be taken as uniform set of machine constants and we are done. Thus suppose the intersection to be empty. The main point now is to establish by a limit argument in affine geometry the following: There exist three vectors $c^*, d^*, e^* \in \mathbb{R}^k$ such that for all $n \in \mathbb{N}$ and small enough $\mu_1 > 0, \mu_2 > 0$ (μ_2 depending on μ_1 and both depending on n) machine M correctly decides $L \cap \mathbb{R}^{\leq n}$ when using $c^* + \mu_1 \cdot d^* + \mu_2 \cdot e^*$ as its constants. This is sufficient to change M into a polynomial time restricted machine that decides L and uses c^*, d^*, e^* as its *uniform* machine constants. □

Let us summarize the methods described so far in view of the main open problem in this context, namely Ladner's result for $\mathrm{NP}_{\mathbb{R}}$. The diagonalization technique used above allows some degree of freedom as to how to define $\mathrm{P}_{\mathbb{R}}/\mathrm{const}$. This means

that we can put some additional conditions onto the set of constants that we allow for a fixed dimension to work. To make the diagonalization work there are basically two aspects that have to be taken into account. First, the resulting class has to contain $\mathrm{P}_{\mathbb{R}}$. Secondly, the conditions we pose on the constants have to be semi-algebraically definable without additional real constants. Playing around with suitable definitions might be a way to attack Ladner's problem as well in the full real number model. However, for a problem L in $\mathrm{P}_{\mathbb{R}}/\mathrm{const}$ the topological structure of the set of suitable constants is more complicated since now each branch results in a (potentially infinite) intersection of semi-algebraic conditions. Then one has to study how the topology of the sets $\bigcap_{i=1}^{N} E_i$ evolves for increasing N. For example, could one guarantee the existence of say a semi-algebraic limit curve along which one could move from a point c^* into an E_n? In that case, a point on the curve might only be given by a semi-algebraic condition. As consequence, though one would likely not be able to show $\mathrm{P}_{\mathbb{R}}/\mathrm{const} \subseteq \mathrm{P}_{\mathbb{R}}$ may be at least a weaker uniform version of Ladner's result could be settled.

To finish this subsection let us refer the interested reader to [**23**], where similar questions concerning Ladner like results are studied in Valiant's model of computation.

4.3. Recursion theory. Whereas so far the focus was on decidable problems, in this subsection we consider problems of increased computational difficulty, i.e., undecidable ones in the BSS model. Recursion theory which deals with degrees of undecidability certainly was one of the main topics that at the beginning stimulated research in classical computability theory, see [**84**]. For alternative models it is in particular interesting with respect to the so called area of *hypercomputation*, i.e., whether there are (natural) computational devices that are more powerful than Turing machines and thus violate the famous Church-Turing hypothesis. For an introduction to hypercomputation and an extended list of references see [**101**], and [**110**] for a particular focus on real hypercomputation.

The real Halting Problem $\mathbb{H}_{\mathbb{R}}$ was already mentioned earlier. We consider it here in the following version: Given a code $c_M \in \mathbb{R}^\infty$ of a real BSS machine M, does this machine stop its computation on input 0? The problem was the first that has been shown to be undecidable in the real number model in [**19**]. We now deal with the following question: Are there problems which in a reasonable sense are strictly easier than $\mathbb{H}_{\mathbb{R}}$ yet undecidable? In the Turing model this was a famous question asked by Post in 1944 and solved about 15 years later independently by Friedberg and Muchnik, see [**99**]. Nevertheless, until today there is no natural problem with this properties known in the Turing model. We shall see that the question turns out to be much easier (though not trivial) in our framework. A second question to be discussed then is that of finding as well more natural problems that are equivalent to $\mathbb{H}_{\mathbb{R}}$, i.e., have the same degree of undecidability. Finally, aspects of bounded query computation are treated briefly.

Before explaining some of the results obtained we have to be more specific on what should be understood under terms like *easier* and *equivalent* if we deal with computability issues. As above with $\mathrm{NP}_{\mathbb{R}}$-completeness this again is formalized using special reductions, this time focussing on computability only instead of complexity.

DEFINITION 4.13. A real decision problem $A \subseteq \mathbb{R}^\infty$ is *Turing reducible* to another problem $B \subseteq \mathbb{R}^\infty$ iff there exists an oracle BSS machine M working as follows. For inputs $x \in \mathbb{R}^\infty$ M works like a normal BSS machine except that it additionally has repeatedly access to an oracle for B. In such an oracle state the machine queries the oracle whether a previously computed $y \in \mathbb{R}^\infty$ belongs to B and gets the correct answer in one step. After finitely many steps (normal and oracle) M stops and gives the correct answer whether x belongs to A or not.

Turing reducibility gives a straightforward way to compare undecidable problems. If A can be decided by an oracle machine using B as oracle but not vice versa, then A is strictly easier than B. If both are Turing reducible to each other they are said to be equivalent. Note that all problems below (i.e., easier than) or equivalent to $\mathbb{H}_\mathbb{R}$ at least are semi-decidable: there is an algorithm which halts exactly for inputs from the problem under consideration. This follows from the existence of a Turing reduction and semi-decidability of $\mathbb{H}_\mathbb{R}$. Problems equivalent to $\mathbb{H}_\mathbb{R}$ are also called *computationally complete* for the real BSS model.

Now our first question, the real version of Post's problem reads: Is there a semi-decidable problem A which is neither decidable nor reducible from $\mathbb{H}_\mathbb{R}$?

THEOREM 4.14 ([**75**]). *The rational numbers $\mathbb{Q}$ represent an undecidable decision problem which is strictly easier than $\mathbb{H}_\mathbb{R}$. Thus, there is no real BSS oracle machine that decides $\mathbb{H}_\mathbb{R}$ by means of accessing $\mathbb{Q}$ as oracle.*

PROOF. Undecidability of $\mathbb{Q}$ was already shown in Theorem 3.4. The same arguments give undecidability of the real algebraic numbers $\mathbb{A}$. The main step now is to show that $\mathbb{Q}$ is strictly easier than $\mathbb{A}$. Note that this implies the result because $\mathbb{A}$ easily can be decided by a machine accessing $\mathbb{H}_\mathbb{R}$ as oracle. So if the statement was false, i.e., $\mathbb{H}_\mathbb{R}$ would be Turing reducible to $\mathbb{Q}$, transitivity of the reduction notion implies that $\mathbb{A}$ should also be decidable using $\mathbb{Q}$ as oracle.

Assume to the contrary that M is an oracle algorithm deciding $\mathbb{A}$ by means of accessing $\mathbb{Q}$. The arguments used to get a contradiction combine some elementary topology and number theory similar to those used in the proof of Theorem 3.4. Topology enters for dealing with inequality branches of M, whereas number theory is used for branches caused by queries to the $\mathbb{Q}$-oracle. Since all intermediate results computed by M are rational functions in the input it turns out to be crucial for analysing the outcome of oracle queries to see how a rational function maps algebraic numbers to rationals. The main observation is the following: Suppose f is a rational function computed by M as oracle query for an input $x \in \mathbb{R}$, i.e., M asks whether $f(x) \in \mathbb{Q}$. If f maps a large enough yet finite set of algebraic numbers to $\mathbb{Q}$, then f will map all algebraic numbers of a large enough degree to a non-rational real. Consequently, such an oracle query is not able to distinguish any algebraic number of large enough degree from a transcendental number. Using this fact together with basic continuity and counting arguments one can conclude that an oracle machine for $\mathbb{A}$ accessing $\mathbb{Q}$ will always fail on certain algebraic numbers. Thus, M cannot work correctly. □

The above proof actually can be extended to get an infinite family of problems that are all strictly easier than $\mathbb{H}_\mathbb{R}$ but pairwise incomparable with respect to Turing reductions. Thus there is a rich structure between decidable problems and computationally complete ones in the real BSS model. Similar results have been obtained in [**45**] for the additive BSS model, whereas [**32**] studies related questions

for higher levels of undecidability above $\mathbb{H}_\mathbb{R}$. Degrees of undecidability in the BSS model are as well studied in [**109**] and [**29**].

Having solved Post's problem we turn to the question whether there are other problems beside $\mathbb{H}_\mathbb{R}$ being computationally universal in the BSS model. In the Turing model several very different problems turned out to be such examples. To mention some of the most prominent ones there is Post's Correspondence Problem [**86**] which asks for matching a finite set of strings according to some rules, Hilbert's 10th problem [**67**] which asks for solvability of diophantine equations, and the word problem in finitely presented groups [**21**, **83**]. The problems considered so far in BSS theory naturally have a very strong connection to semi-algebraic geometry because of the underlying set of operations implying that all intermediate results in an algorithm are related to rational functions. So it is demanding to find other significant problems in the theory which basically are not problems in semi-algebraic geometry. We shall now discuss that a suitable variant of the discrete word problem is such an example. Note that the first two of the above mentioned problems do not provide such examples. The Post Correspondence Problem by nature has strong discrete aspects as a kind of matching problem, whereas a real analogue of Hilbert's 10th problem, i.e., deciding real solvability of a real polynomial system is decidable by quantifier elimination.

To understand the word problem let us start with an easy discrete example. Suppose we are given a formal string bab^2ab^2aba in a free group $\langle\{a,b\}\rangle$ generated by the two generators a, b. Here, x^i denotes the i-fold repetition of element x, and concatenation represents the group operation. Now we add some relations between certain elements of the freely generated group. That way a quotient group of the original free group is obtained. For example, assume the equation $ab = 1$ to hold. It is then easy to see that the given element in the resulting quotient group represents b^2. But it cannot be reduced to 1 in this group. However, if as well the relation $a^4 = a^2$ holds, then the given word in the resulting new quotient group does represent the neutral element 1.

This leads to the definition of the word problem.

DEFINITION 4.15. a) Let X denote a set. The *free group* generated by X, denoted by $(\langle X\rangle, \circ)$, is the set $(X \cup X^{-1})^*$ of all finite sequences $\bar{w} = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$ with $n \in \mathbb{N}$, $x_i \in X$, $\alpha_i \in \{-1, +1\}$, equipped with concatenation $\circ$ as group operation subject to the rules

$$x \circ x^{-1} \quad = \quad 1 \quad = \quad x^{-1} \circ x, \qquad x \in X\ , \tag{4.1}$$

where $x^1 := x$ and 1 denotes the empty word, that is, the unit element.

b) A group $(G, \bullet)$ is called *finitely presented* if $G \cong \langle X\rangle/\langle R\rangle_{\langle X\rangle} =: \langle X|R\rangle$ is (isomorphic to) the quotient of a free group $\langle X\rangle$ with finite set of generators X and the normal subgroup $\langle R\rangle_{\langle X\rangle}$ of $\langle X\rangle$ generated by the finite set $R \subseteq \langle X\rangle$.

c) The *word problem for* $\langle X|R\rangle$ is the task of deciding, given $\bar{w} \in \langle X\rangle$, whether $\bar{w} = 1$ holds in $\langle X|R\rangle$.

Intuitively, R describes finitely many rules "$\bar{r} = 1$", $\bar{r} \in R$ additional to those necessarily satisfied in a group. The famous work of Novikov and, independently, Boone establishes the existence of a finitely presented group $\langle X|R\rangle$ whose associated word problem is many-one reducible by a Turing Machine from the discrete Halting

Problem H and thus computationally complete in the Turing model. The result is interesting in linking a purely algebraic problem with recursion theory. Since algebra of course is not restricted to discrete groups it is natural to ask whether similar relations can be established between other groups and BSS recursion theory. In the following we shall outline that this indeed is possible.

A natural generalization of Definition 4.15 to the real number setting is obtained by allowing the sets X and R to become uncountable. Formally, this is expressed by considering sets $X := \{x_r\}_r$ of abstract generators indexed with real vectors r ranging over some subset of $\mathbb{R}^\infty$, and similarly for the relations R. Then interesting word problems arise by putting restrictions on these sets in $\mathbb{R}^\infty$. For sake of notational simplicity we identify X with the sets in $\mathbb{R}^\infty$ the corresponding r's belong to, and similarly for R and the rules.

DEFINITION 4.16. Let $X \subseteq \mathbb{R}^\infty$ and $R \subseteq \langle X\rangle \subseteq \mathbb{R}^\infty$. The elements in $\langle X\rangle$ are coded as elements in $\mathbb{R}^\infty$. The tuple (X, R) is called a *presentation* of the *real group* $G = \langle X|R\rangle$. This presentation is *algebraically generated* if X is BSS-decidable and $X \subseteq \mathbb{R}^N$ for some $N \in \mathbb{N}$. G is termed *algebraically enumerated* if R in addition is BSS semi-decidable; and if R is BSS-decidable we call G *algebraically presented.* The *word problem* for the presented real group $G = \langle X|R\rangle$ is the task of BSS-deciding, given $\bar{w} \in \langle X\rangle$, whether $\bar{w} = 1$ holds in G.

EXAMPLE 4.17 ([**76**]). The following three examples should clarify the above notions. The first two give different presentations $\langle X|R\rangle$ of the additive group $(\mathbb{Q}, +)$ of rational numbers with decidable word problem, whereas the third has an undecidable word problem due to its connection to deciding $\mathbb{Q}$ in $\mathbb{R}$.

i) $X = \{x_r : r \in \mathbb{Q}\}$, $R = \{x_r x_s = x_{r+s} : r, s \in \mathbb{Q}\}$;

ii) $X = \{x_{p,q} : p, q \in \mathbb{Z}, q \neq 0\}$,
$R = \{x_{p,q}x_{a,b} = x_{(pb+aq,qb)} : p, q, a, b \in \mathbb{Z}\} \cup \{x_{p,q} = x_{(np,nq)} : p, q, n \in \mathbb{Z}, n \neq 0\}$;

iii) $X = \{x_r : r \in \mathbb{R}\}$, $R = \{x_{nr} = x_r, x_{r+k} = x_r : r \in \mathbb{R}, n \in \mathbb{N}, k \in \mathbb{Z}\}$.

Case ii) yields an algebraic presentation, i) is not even algebraically generated, but iii) is algebraically presented. The word problem is trivially decidable for i) because after embedding the task into $(\mathbb{R}, +)$ one can simply compute on the indexes and check whether the result is 0. Also for ii) it is decidable by a similar argument. For iii) the word problem is undecidable because it holds $x_r = x_0 \Leftrightarrow r \in \mathbb{Q}$. Note, however, that case iii) by means of Theorem 4.14 does not provide a group for which the word problem is computationally universal.

It is not hard to establish that for all algebraically enumerated groups the corresponding word problem is semi-decidable in the BSS model. This just requires a folklore argument based on quantifier elimination. The more interesting result is

THEOREM 4.18 ([**76**]). *There exists an algebraically presented real group $\mathcal{H} = \langle X|R\rangle$ such that the real Halting problem $\mathbb{H}_\mathbb{R}$ is reducible to the word problem in $\mathcal{H}$. This word problem thus is computationally universal for the real BSS model.*

The proof in a first step embeds the membership problem for any set in $\mathbb{R}^\infty$ to the word problem in a suitable group. Then, it proceeds showing that for $\mathbb{H}_\mathbb{R}$ this embedding can be arranged such that the resulting group is algebraically presented. We skip further details because they rely on a lot of classical techniques

in combinatorial group theory such as HNN extensions and Britton's lemma. For more on that see [**63**] and the full proof in [**76**].

The theorem is interesting in that it gives a problem computationally significant in the BSS model over $\mathbb{R}$ yet only indirectly related to semi-algebraic features. The list of such problems at the moment is much smaller than in classical recursion theory and it seems an interesting topic for future research to find more such problems. Open questions related immediately to the above theorem are the following. Can the corresponding universality result be established for algebraically presented groups for which as well the set R of rules comes from a finite dimensional space $\mathbb{R}^k$? In the construction of the proof it turns out to be crucial that R lives in $\mathbb{R}^\infty$, i.e., there have to be included rules for vectors of arbitrarily large dimension. Recall that in the original result by Boone and Novikov both X and R are finite, and it seems that a suitable analogue of finiteness in the discrete setting is finite dimensionality of these sets in the real number framework. Another interesting question is that of finding particularly structured groups whose respective word problems are universal for complexity classes. One such task thus would be to find particular algebraically generated groups for which the word problem is $\mathrm{NP}_{\mathbb{R}}$-complete.

To close this section we briefly mention yet another area of recursion theory which has intensively been studied in the Turing model and only seen some initial considerations in our framework, namely bounded query computations. Here, the interest is shifted from the direct consideration of decision problems, i.e., computing the characteristic function χ_A of an $A \subseteq \mathbb{R}^\infty$ to the following type of questions: Given an $n \in \mathbb{N}$ how many oracle queries to a set $B \subseteq \mathbb{R}^\infty$ are needed in order to compute the n-fold characteristic function χ_A^n of A on n many inputs $x_i \in \mathbb{R}^\infty, 1 \leq i \leq n$. More precisely, this function is defined as $\chi_A^n(x_1, \ldots, x_n) := (\chi_A(x_1), \ldots, \chi_A(x_n))$. Different choices of A and B, where also $A = B$ is possible, give quite different results. An easy example shows that by using binary search for each semi-decidable set A the n-fold characteristic function can be computed by $\lceil \log_2 n + 1 \rceil$ many calls to an oracle for $\mathbb{H}_{\mathbb{R}}$. The following result is much less obvious

THEOREM 4.19 ([**74**]). *Let $n \in \mathbb{N}$ and consider the n-fold characteristic function $\chi_{\mathbb{Q}}^n$ on $\mathbb{Q}$. Let $B \subseteq \mathbb{R}$ be an arbitrary subset of the reals. Then no BSS oracle machine having access to B as oracle can compute $\chi_{\mathbb{Q}}^n$ with less many than n queries.*

PROOF. Suppose such an oracle machine exists it must in a certain way reduce n questions about $\mathbb{Q}$ to at most $n-1$ many questions about the arbitrary real set B. Now the main idea is to arrange the situation for an application of the implicit function theorem for functions from $\mathbb{R}^n \mapsto \mathbb{R}^{n-1}$. Along the one-dimensional solution curve which the theorem guarantees to exist the oracle machine then can be shown to necessarily err. □

The application of classical tools from analysis like the implicit function theorem shows a significant difference to proofs in the Turing framework. So we expect a lot of interesting problems to exist in this area which need other methods not available in discrete recursion theory.

This section aimed to present some challenging questions and techniques in structural complexity theory for real or complex number computations. The problems treated just reflect a small fraction of topics studied in the last two decades in this area. We close by pointing to some more literature.

A prominent class of problems that have been studied intensively in classical complexity theory are counting problems. This has lead to the definition of a counting analogue of NP denoted by #P and the search for complete problems in that class. Roughly speaking, #P captures functions that count the number of accepting computations of an NP-algorithm. Assuming $\mathrm{P} \neq \mathrm{NP}$ this counting class contains much harder problems than those in NP. This is justified by Toda's famous result [**103**] which says that using an oracle from #P in deterministic polynomial time computations captures all problems in the so called polynomial hierarchy, a set conjectured to be much larger than NP. A prominent result by Valiant [**105**] shows that the computation of the permanent for a matrix with $\{0, 1\}$-entries reflects the difficulty of this class, i.e., is a #P-complete problem.

In the real number framework counting problems have been extensively studied in several papers by Bürgisser, Cucker and co-authors. Many of the relevant problems have a strong algebraic flavour, for example tasks like computing Betti numbers of algebraic varieties. As a starting point for readers being interested in such questions we just refer to [**26**, **27**]. Analogues of Toda's theorem both in real and complex number complexity theory were recently obtained in [**8**, **10**].

Another branch of complexity theory that was studied in the real number framework is descriptive complexity. Here the goal is to describe complexity classes independently of the underlying computational model. Instead, the logical shape in which a problem can be expressed reflects the algorithmic complexity sufficient to solve it. The first result into this direction can be found in [**46**], where both for $\mathrm{P}_{\mathbb{R}}$ and $\mathrm{NP}_{\mathbb{R}}$ such logical characterizations are given. [**35**] contains further such results, [**70**] deals with counting problems from a logical point of view.

Transfer results, one of the main topics in this section, have as well been analysed with respect to other algebraic approaches to complexity, and here foremost Valiant's complexity classes VP and VNP, see [**106**]. This approach focusses on families of polynomials over a field that have a polynomially bounded degree in the number of their variables and can be computed by a non-uniform family of small circuits. This constitutes class VP, whereas VNP essentially is the family of polynomials whose coefficients are functions in VP, though there might be exponentially many monomials. A notion of reduction then is introduced by using a projection operator and once again the permanent polynomials turn out to be a complete family for VNP. This gives another algebraic variant of a P versus NP problem, this time for VP versus VNP and it is nearby to ask whether this question as well is related to some of the other problems of that style mentioned before. Readers interested in learning more about progress being made into this direction are refered to [**24**] as starting point.

5. Probabilistically checkable proofs over $\mathbb{R}$

For the rest of this paper we shall now turn to the area of probabilistically checkable proofs, for short PCPs. The PCP theorem first shown by Arora et al. [**2**, **3**] certainly is one of the landmark results in Theoretical Computer Science in the last two decades. It gives a new characterization of class NP in the Turing model and had tremendous impact on obtaining non-approximability results in combinatorial optimization. More recently, an alternative proof of the theorem was given by Dinur [**38**].

The first subsection below briefly surveys the classical PCP theorem and its currently existing proofs. The main part of this section is then devoted to studying PCPs in the BSS model. We shall give a complete proof of the existence of so-called long transparent proofs for both $\mathrm{NP}_{\mathbb{R}}$ and $\mathrm{NP}_{\mathbb{C}}$, see [**71**]. Then, we outline how the full PCP theorem can be shown to hold as well in these two models.

5.1. The classical PCP theorem: A short outline. The PCP theorem gives a surprising alternative characterization of the class NP. It is based on a new point of view concerning the verification procedure necessary to establish membership of a problem L in class NP. Recall that according to the definition of NP verifying that an input x belongs to L can be done by guessing a suitable proof y and then verifying by a deterministic polynomial time algorithm in the size of x that the pair (x, y) satisfies the property defining L. For example, verifying satisfiability of a given Boolean formula $x := \phi$ in conjunctive normal form can be done by guessing a satisfying assignment y and then evaluating $\phi(y)$. Clearly such a verification algorithm in general must read all components of y in order to work correctly. Note that for $x \in L$ at least one such y has to exist, whereas for $x \notin L$ all potential proofs y have to be rejected.

In the PCP theorem the requirements for the verification procedure are changed. Here is a brief outline of these new aspects, precise definitions are given in the next subsection. Suppose membership of x in L should be verified using proof y. The verifier is randomized in that it first generates a random string. This string and input x are then used to determine a number of proof components in y it wants to read. This number is intended to be dramatically smaller than the size of y, actually only constant in the PCP theorem. Finally, using the input, the random string and those particular proof components the verifier makes its decision whether to accept or rejct the input. This way there will be a possibility that the verifier comes to a wrong conclusion, but as long as this probability is not too big this is allowed. $\mathrm{PCP}(r(n), q(n))$ denotes the class of those languages that have a verifier using $r(n)$ random bits and inspecting $q(n)$ components of the given proof y for inputs x of size n. The PCP theorem states that $\mathrm{PCP}(O(\log(n)), O(1)) = \mathrm{NP}$. It thus shows that there exists a format of verification proofs for languages in NP which is stable in the following sense: If $x \in L$, then there is a proof y that is always accepted (just as in the original definition of NP); and if $x \notin L$ for each proof y the verifier detects a fault in that proof with high probability by reading a constant number of components only. The number of components of y to be seen in particular is independent of the length of the input!

The PCP theorem implies lots of inapproximability results. One of the first such result is this one. Assume that $\mathrm{P} \neq \mathrm{NP}$. Given a propositional Boolean formula ϕ in conjunctive normal form having m clauses, there is no algorithm running in polynomial time in the size of ϕ which for an arbitrary given $\epsilon > 0$ does the following. It computes a value k such that for the maximum number $\max(\phi)$ of clauses of ϕ that are satisfiable in common the inequality $\max(\phi)/k \leq 1 + \epsilon$ holds. Thus the results tells us that unless $\mathrm{P} = \mathrm{NP}$ this maximal number of clauses satisfiable in common cannot be approximated efficiently with arbitrary relative accuracy. Recall that we saw a similar negative result in Theorem 2.2, part b). However, that result was much easier to obtain than the one above which could only be shown as an application after the PCP theorem was proven. The close

relation between PCPs and approximability is the starting point of Dinur's proof and will also be important for studying such questions in the real number setting.

Let us shortly outline the two existing proofs of the PCP theorem. The original one by Arora et al. is very algebraic in nature. Here, different verifiers are constructed which are then combined to a single new verifier with the desired properties. The particular way how verifiers are combined requires a new technique called verifier-composition that was developed in [**3**]. One of the verifiers used for the composition needs a large amount of randomness but inspects constantly many proof components only. It is based on coding a satisfying assignment of a Boolean formula via certain linear functions. The second verifier uses logarithmic randomness but needs to read more components. Here, the used coding of an assignment is done via multivariate polynomials of not too high degree. Both verifiers are then cleverly combined by the above mentioned technique of verifier-composition. This yields a third verifier with the required resources.

The second proof of the PCP theorem given by Dinur [**38**] in 2005 is more combinatorial in structure. The basic idea of this proof is to exploit the strong relation between PCPs and (non-)approximability results. More precisely, Dinur's proof uses an NP-complete problem called CSP which stands for constraint satisfiability problem; such problems are extensions of the Boolean satisfiability problem. An instance of the CSP problem consists of a number of constraints in a finite number of variables taking values in a finite alphabet. The question is again whether there exists an assignment of the variables that satisfies all constraints. Instead of directly constructing a verifier for this problem one considers the following approximation problem: Is there an efficient algorithm which for any given $\epsilon > 0$ approximates the maximal number of constraints that are commonly satisfiable within a factor at most $1 + \epsilon$. Clearly, since the decision problem is NP-complete computing the maximal number exactly is an NP-hard problem as well. But it is not clear whether the above optimization task can be accomplished more easily. This question is intimately related to the PCP theorem as follows. Suppose there exists a polynomial time reduction from CSP instances to CSP instances such that a satisfiable CSP instance is mapped to a satisfiable CSP instance and a non-satisfiable CSP instance is mapped to a CSP instance for which no assignment satisfies more than a certain fixed fraction of the constraints. Then the PCP theorem would follow from the existence of that reduction. A verifier for CSP first performs the reduction on an input instance. It then expects the proof to give an assignment to the variables of the instance resulting from the reduction. Now if this resulting instance is not satisfiable, then the assignment the proof encodes violates at least a fixed fraction of the constraints. So the verifier can check the proof by selecting a constant number of constraints, reading the constantly many values that the proof assigns to the variables occurring in these constraints, and checking if one of these constraints is violated by the assignment. Due to the existence of the fixed fraction repeating this test constantly many times will guarantee that the verifier respects the necessary error bounds.

Dinur's proof constructs such a polynomial time reduction between CSP instances. There are two major steps involved in the construction. Given an unsatisfiable set of constraints at the beginning we only know that at least one among the constraints is not satisfiable together with the remaining ones. Thus at the beginning we have no constant fraction of unsatisfied constraints. The first step is

an amplification step that increases this fraction by a constant factor. Repeating it logarithmically many times would yield a constant fraction. However, the amplification also increases the size of the finite alphabet used. To control this a second step called alphabet reduction is necessary. This second step as well heavily relies on the existence of long transparent proofs, i.e., verifiers that accept CSP using a large (super-logarithmic) amount of randomness and inspecting constantly many proof components. Note that for using the corresponding verifier, in both proofs its structure is much more important than the values of the parameters r and q. This is due to the fact that the long-transparent-proof verifier is applied to instances of constant size only. This as well will be important below in the real number setting.

This short outline of the classical proof structures should be sufficient here. Similar ideas will be described much more explicitly in the next subsections in relation to PCPs for the real and complex BSS model. Complete descriptions of the two classical proofs can be found in the already cited original papers as well as in [**1**, **50**, **87**].

5.2. Verifiers in BSS setting; long transparent proofs. For $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$ consider once again the Hilbert Nullstellensatz decision problem $\mathrm{QPS}_{\mathbb{K}}$ studied in previous sections. To show its membership in $\mathrm{NP}_{\mathbb{K}}$ one can guess a potential solution $y \in \mathbb{K}^n$, plug it into the polynomials of the system and verify whether all equations are satisfied by y. Clearly, this verification algorithm in general has to inspect all components of y. So the above question for the discrete satisfiability problem as well makes perfect sense here: Can we give another verification proof for solvability of such a system that is much more stable in the sense of detecting errors with high probability by inspecting only a small amount of proof components?

This kind of question is made more precise by defining the corresponding verification procedures as well as the languages in $\mathbb{K}^*$ which are accepted by such verifiers.

Definition 5.1. Let $r, q : \mathbb{N} \mapsto \mathbb{N}$ be two functions. An $(r(n), q(n))$-restricted verifier V in the BSS model over $\mathbb{K}, \mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$ is a randomized BSS algorithm over $\mathbb{K}$ working as follows. For an input $x \in \mathbb{K}^*$ of algebraic size n and another vector $y \in \mathbb{K}^*$ representing a potential membership proof of x in a certain set $L \subseteq \mathbb{K}^*$, the verifier in a first phase generates non-adaptively a sequence of $O(r(n))$ many random bits (under the uniform distribution on $\{0,1\}^{O(r(n))}$). Given x and these $O(r(n))$ many random bits V in the next phase computes in a deterministic manner the indices of $O(q(n))$ many components of y. This again is done non-adaptively, i.e., the choice of components does not depend on previously seen values of other components. Finally, in the decision phase V uses the input x together with the random string and the values of the chosen components of y in order to perform a deterministic polynomial time algorithm in the BSS model. At the end of this algorithm V either accepts (result 1) or rejects (result 0) the input x. For an input x, a guess y and a sequence of random bits ρ we denote by $V(x, y, \rho) \in \{0, 1\}$ the result of V in case the random sequence generated for (x, y) was ρ.

The time used by the verifier in the decision phase 3 is also called its decision-time. It should be polynomially bounded in the size of x.

An easy example of such a verifier is given below in the proof of Lemma 5.10.

Remark 5.2. Concerning the running time of a verifier the following has to be pointed out. In general, generating a random bit is assumed to take one time unit,

and the same applies when the verifier asks for the value of a proof component. Below in relation to long transparent proofs we need more than polynomially many random bits. In such a situation the time for generating a random string would be superpolynomial. We then assume that the entire random string can be generated at unit cost. Note however that this is of no concern since the existence of long transparent proofs will be used in the proof of the full PCP theorem only for instances of constant size and thus the number of random bits is constant as well. We comment on this point once more after Theorem 5.7 below.

Using the above notion of a verifier it is immediate to define the languages accepted by verifiers.

DEFINITION 5.3. ($\mathrm{PCP}_{\mathbb{K}}$-classes) Let $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$ and let $r, q : \mathbb{N} \mapsto \mathbb{N}$; a decision problem $L \subseteq \mathbb{K}^*$ is in class $\mathrm{PCP}_{\mathbb{K}}(r(n), q(n))$ iff there exists an $(r(n), q(n))$-restricted verifier V such that conditions a) and b) below hold:

a) For all $x \in L$ there exists a $y \in \mathbb{K}^*$ such that for all randomly generated strings $\rho \in \{0,1\}^{O(r(size_{\mathbb{K}}(x)))}$ the verifier accepts. In other words:
$$\Pr_{\rho}\{V(x, y, \rho) = 1\} = 1 \quad .$$

b) If $x \notin L$, then for all $y \in \mathbb{K}^*$
$$\Pr_{\rho}\{V(x, y, \rho) = 0\} \geq \frac{3}{4} \quad .$$

In both cases the chosen probability distribution is the uniform one over all strings $\rho \in \{0,1\}^{O(r(size_{\mathbb{K}}(x)))}$.

In this section we will discuss the existence of transparent long proofs for problems in $\mathrm{NP}_{\mathbb{K}}$ in detail. Our exposition and the given proof below basically follow [**71**], where this existence was shown for $\mathrm{NP}_{\mathbb{R}}$. Note however that though the almost same analysis is used it seems that a longer verification proof is needed than the one given there; so we adapt the required arguments accordingly. This change nevertheless is of no concern with respect to the role long transparent proofs play in the full PCP Theorem 5.14 below. There, they are applied to constant size inputs only, so the length of a long transparent proof a verifier wants to inspect is constant anyway. The more important aspect is the structure of the verification proof, see below.

We shall construct such a verifier for the $\mathrm{NP}_{\mathbb{K}}$-complete problem $\mathrm{QPS}_{\mathbb{K}}$. The construction is described for $\mathbb{K} := \mathbb{R}$, always pointing out where some care has to be taken when $\mathbb{K} = \mathbb{C}$ is considered instead.

The system's coefficients can be arbitrary real numbers. The verifier will receive the following three objects as input: A family of degree two polynomials, a (possibly incorrect) proof of the existence of an assignment under which the polynomials evaluate to zero, and a sequence of random bits. The verifier outputs either "accept" if it believes the proof to be correct or "reject" otherwise. The polynomials and the proof will be in the form of a sequence of real numbers whereas the random string is a sequence over $\{0,1\}$. Randomness is used to decide which locations in the proof to query. Since the corresponding addresses can be coded discretely only discrete randomness is needed.

Throughout this subsection let n denote the number of variables of the input polynomials. Let $\mathcal{P} := \{p_1, \ldots, p_m\}$ denote the system. All polynomials p_i are of

degree at most two and depend on at most three variables. For $r \in \{0,1\}^m$ define $P(x,r) := \sum_{i=1}^{m} p_i(x) \cdot r_i$. The following is easy to see: Let $x \in \mathbb{R}^n$ be fixed. If x is a common zero of all $p_i(x)$, then $P(x,r) = 0$ for all r. And if x is no common zero the probability for uniformly taken r that $P(x,r) = 0$ is at most $\frac{1}{2}$. We work with $P(x,r)$ in order to capture both the real and the complex case in common.

Of course, if we want to verify whether an $a \in \mathbb{R}^n$ solves the system it does not make sense to plug it into $P(a,r)$ and evaluate because this requires again reading all components of a. We therefore rewrite $P(a,r)$ as follows:

$$P(a,r) = E(r) + A \circ L_A(r) + B \circ L_B(r), \tag{5.2}$$

where functions E, A, B, L_A, and L_B have the following properties. A and B are linear functions with n and n^2 many inputs, respectively. The coefficient vectors that represent these mappings depend on the chosen a only. More precisely,

$$A : \mathbb{R}^n \mapsto \mathbb{R} \text{ such that } A(x_1, \ldots, x_n) = \sum_{i=1}^{n} a_i \cdot x_i \ \forall\ x \in \mathbb{R}^n;$$
$$B : \mathbb{R}^{n^2} \mapsto \mathbb{R} \text{ such that } B(y_{11}, \ldots, y_{nn}) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_i \cdot a_j \cdot y_{ij} \ \forall\ y \in \mathbb{R}^{n^2}.$$

The functions E, L_A and L_B are linear as well. They take as arguments inputs from $\mathbb{Z}_2^m := \{0,1\}^m$ and give results in the spaces $\mathbb{R}, \mathbb{R}^n$ and $\mathbb{R}^{n^2}$, respectively. It is important to note that these mappings do only depend on the coefficients of the polynomials $p_1, \ldots, p_m$ but not on a. Therefore, given the system and a random vector $r \in \mathbb{Z}_2^m$ these functions can be evaluated deterministically without inspecting a component of the verification proof. As an immediate consequence of equation (1), for evaluating $P(a,r)$ it is sufficient to know two function values of certain linear functions, namely the value of A in $L_A(r) \in \mathbb{R}^n$ and that of B in $L_B(r)$. The verifier expects from the verification proof to contain these two real values.

More precisely, the proof is expected to contain so-called linear function encodings of the coefficient vectors defining A and B. This means that instead of expecting the proof to just write down those vectors we do the following. We define a finite subset $\mathfrak{D}$ of $\mathbb{R}^n$ and require the proof to contain all values of $A(x) := a^t \cdot x$ for all $x \in \mathfrak{D}$; similarly for B and a subset of $\mathbb{R}^{n^2}$. In order to work out this idea several problems have to be handled. First, though A in principle is a linear function over all $\mathbb{R}^n$ the verification proof must be finite. It can only contain finitely many components representing values of A. Among these components we of course must find those values in arguments that arise as images $L_A(r)$ for $r \in \mathbb{Z}_2^m$. Secondly, the verifier cannot trust the proof to represent a linear function which maps $\mathfrak{D}$ to $\mathbb{R}$. All it can do is to interpret the proof as giving just a function A from $\mathfrak{D}$ to $\mathbb{R}$ and try to find out if it is linear. Thirdly, even if the functions A and B are indeed linear on their corresponding domains and encode coefficient vectors a and b the verifier has to find out whether b is consistent with a, i.e., whether the coefficient vector $\{b_{ij}\}$ defining B satisfies $b_{ij} = a_i \cdot a_j$.

To verify all requirements within the necessary resources and error bounds the verifier tries to realize the following tasks: It expects the proof to provide two function value tables representing A and B on suitable domains (to be specified). Then first it checks whether both tables with high probability represent a linear function on the respective domains and if 'yes' how to compute the correct values of those functions in a given argument with high probability. In a second part

the verifier checks consistency of the two involved coefficient vectors with high probability. Finally, it evaluates (1) to check whether the result equals 0.

A correct proof will provide the tables of two linear functions on the appropriate domains of form $A(x) = a^t \cdot x$ and $B(x) = b^t \cdot x$ with vectors $a \in \mathbb{R}^n, b \in \mathbb{R}^{n^2}$ such that $b_{ij} = a_i \cdot a_j, 1 \leq i, j \leq n$. In this ideal case, equation (1) can be evaluated by reading only two components of the entire proof, namely one value of A and one of B. If the proof is correct the verifier will always accept.

Suppose then that the given $\mathrm{QPS}_{\mathbb{R}}$ instance has no solution. The verifier has to detect this for any proof with high probability. There are different cases to consider where in the proof errors can occur. The first such case is the one in which one of the two functions which the proof provides is in a certain sense far from being linear. The verifier will be able to detect this with high probability by making only a few queries into the function value table and then reject. A more difficult situation occurs when the given function is not linear but close to linear. In this case the verifier's information about the proof is not sufficient to conclude that it is not completely correct. To get around this problem a procedure that aims to self-correct the values which the proof gives is invoked. For A and B as given in the tables we shall define self-corrections f_A, f_B. Assuming that the function value tables are almost linear will guarantee that these self-corrected functions are linear on the part of the domain which is important for us. Furthermore, the values of these self-corrected functions can be computed correctly with high probability at *any* argument in this part of the domain by making use of constantly many other values in the table only. In case A is linear f_A equals A on the domain on which it is defined.

We will now carry out the following plan:

(1) Define the domains on which we want the verification proof to define functions A and B;
(2) check linearity of these functions such that if they are far from linear it will be discovered with high probability;
(3) assuming no contradiction to linearity has been detected so far define the self-corrections f_A and f_B; use these to detect with high probability an error if consistency between the coefficient vectors of the two linear functions is violated;
(4) for random $r \in \mathbb{Z}_2^m$ obtain the correct values of $f_A(L_A(r))$ and $f_B(L_B(r))$ with high probability and use these values together with $E(r)$ to evaluate $P(a, r)$. Check whether the result is zero.

5.2.1. *Appropriate domains for linearity.* We will now describe the domain $\mathfrak{D}$ on which the values of A should be provided by the proof. The domain on which we want the proof to define the function B will be constructed analogously.

The function $L_A : \mathbb{Z}_2^m \to \mathbb{R}^n$ which generates the arguments in which A potentially has to be evaluated has a simple structure depending on the input coefficients of the polynomials p_i. Written as a matrix its entries are either 0 or such coefficients, i.e., real numbers that constitute the $\mathrm{QPS}_{\mathbb{R}}$ instance. Let $\Lambda := \{\lambda_1, \ldots, \lambda_K\}$ denote this set of entries in L_A, considered as a multiset. Since each p_i depends on at most 3 variables it is $K = O(m)$. In order to simplify some of the calculations below we assume without loss of generality that $m = O(n)$; if not we can add a polynomial number of dummy variables to the initial instance. Thus $K = O(n)$. Without loss of generality we also assume $\lambda_1 = 1$. The components of any vector

occurring as argument of A now are 0-1 linear combinations of elements in Λ. We therefore define

$$\mathcal{X}_0 := \{\sum_{i=1}^{K} s_i \cdot \lambda_i \mid s_i \in \{0,1\}\}^n.$$

This set contains $\mathbb{Z}_2^n$ and thus a basis of $\mathbb{R}^n$. If we could guarantee additivity on pairs taken from $\mathcal{X}_0$ as well as scalar multiplicativity with respect to all scalars taken from Λ we could be sure to work with a correct linear function for our purposes.

Here a first problem occurs: For getting almost surely a linear function A on $\mathcal{X}_0$ from a table for A we need to know and test values of A on a much larger domain $\mathcal{X}_1$. So a larger test domain is needed in order to get a much smaller safe domain, compare [**89**]. The idea behind constructing $\mathcal{X}_1$ is as follows: We want $\mathcal{X}_1$ to be almost closed under addition of elements from $\mathcal{X}_0$. With this we mean that for every fixed $x \in \mathcal{X}_0$, picking a random $y \in \mathcal{X}_1$ and adding x to it results with high probability again in an element in $\mathcal{X}_1$. Similarly, $\mathcal{X}_1$ should be almost closed under scalar multiplication with a factor $\lambda \in \Lambda$. These properties of $\mathcal{X}_1$ will be important in proving linearity of f_A on $\mathcal{X}_0$ if A satisfies the tests on $\mathcal{X}_1$ to be designed. Note that when we speak about linearity of f_A *on* $\mathcal{X}_0$ we mean that for all $x, y \in \mathcal{X}_0$ it holds $f_A(x) + f_A(y) = f_A(x + y)$, even though the sum $x + y$ in most cases does not belong to $\mathcal{X}_0$; similarly for arguments λx.

We remark that the above requirements are more difficult to be satisfied than in the corresponding construction of a long transparent proof in the Turing model. There, all domains are subsets of some $\mathbb{Z}_2^N$ and thus arguments are performed on a highly structured set with a lot of invariance properties of the uniform distribution. Secondly, there are no scalars other than 0 and 1, so additivity implies linearity. In the BSS setting some difficulties arise because some of the elements in Λ can be algebraically independent.

The above motivates the following definition. Let two sets M and M^+ be defined as $M := \{\prod_{i=1}^{K} \lambda_i^{t_i} | t_i \in \{0, \ldots, n^2\}\}$, $M^+ := \{\prod_{i=1}^{K} \lambda_i^{t_i} | t_i \in \{0, \ldots, n^2+1\}\}$ and let

$$\mathcal{X}_1 := \left\{ \frac{1}{\alpha} \sum_{\beta \in M^+} s_\beta \cdot \beta \mid s_\beta \in \{0, \ldots, n^3\}, \alpha \in M \right\}^n.$$

We now prove that $\mathcal{X}_1$ does indeed have the desired properties. To keep things simple we will think of elements in $\mathcal{X}_0$, $\mathcal{X}_1$ (and later also in $\mathfrak{D}$) as formal sums of products defining M^+. This means for example that we distinguish elements in $\mathcal{X}_1$ which have the same numerical value because some λ_i's in Λ could be the same, but arise from formally different sums. Such elements are counted twice below when talking about the uniform distribution on the respective domains. Doing it this way simplifies some counting arguments because we don't have to take algebraic dependencies between the λ_i's into account.

LEMMA 5.4. *Let $\epsilon > 0$ and let $n \in \mathbb{N}$ be large enough, then the following holds:*

a) For every fixed $x \in \mathcal{X}_0$ it is $\Pr_{y \in \mathcal{X}_1}\{y + x \in \mathcal{X}_1\} \geq 1 - \epsilon$.

Here, the probability distribution is the uniform one on $\mathcal{X}_1$, taking into account the above mentioned way how to count elements in $\mathcal{X}_1$.

b) Similarly, for fixed $\lambda_s \in \Lambda$ it is $\Pr_{y \in \mathcal{X}_1}\{\lambda_s \cdot y \in \mathcal{X}_1\} \geq 1 - \epsilon$.

c) For fixed $\lambda \in \Lambda$ it is $\Pr_{\alpha \in M}\{\alpha/\lambda \in M\} \geq 1 - \epsilon$.

PROOF. For part a) let us focus on a single coordinate j. Then x_j is a 0-1 sum of the λ_i's. We have y_j of the form $\frac{1}{\alpha}\sum_{\beta\in M^+} s_\beta\cdot\beta$ with $\alpha\in M$ and $s_\beta\le n^3$ for $\beta\in M^+$. If the sum for x_j contains a term $1\cdot\lambda_i$ and the corresponding coefficient of monomial λ_i in y_j is $< n^3$, then y_j+x_j also has the required form. Since $K=O(n)$ let $K\le cn$ for a suitable constant $c>0$. Thus for each of the at most K many addends in x_j there are n^3 out of n^3+1 choices for the coefficient of the corresponding monomial in y_j that imply x_j+y_j to be of the required form with respect to this monomial. Since this argument applies for all n components one obtains

$$\begin{aligned}\Pr_{y\in\mathcal{X}_1}\{y+x\in\mathcal{X}_1\} &= \left(\tfrac{n^3}{n^3+1}\right)^{K\cdot n}=\left(1-\tfrac{1}{n^3+1}\right)^{O(n^2)} \underbrace{\ge}_{Bernoulli} 1-\tfrac{O(n^2)}{n^3+1}\\ &\ge 1-\tfrac{c}{n}\ge 1-\epsilon.\end{aligned}$$

For part b) consider an arbitrary fixed $\lambda_s\in\Lambda$ together with a random $y\in\mathcal{X}_1$. Consider again a fixed component j of y. The α in the representation of this y_j has the form $\prod_{i=1}^K\lambda_i^{t_i}$ with $t_i\in\{0,\dots,n^2\}$. If the particular exponent t_s of λ_s in this α satisfies $t_s>0$, then $\lambda_s\cdot y$ will belong to $\mathcal{X}_1$ (and for some cases with $t_s=0$ as well). The probability that $t_s>0$ and thus $\lambda_s\cdot y\in\mathcal{X}_1$ is therefore bounded from below by

$$\Pr_{y\in\mathcal{X}_1}\{\lambda_s\cdot y\in\mathcal{X}_1\}=\left(\frac{n^2}{n^2+1}\right)^n\ge 1-\frac{c}{n}\ge 1-\epsilon.$$

Part c) is trivial. □

In order to verify (almost) linearity of A on $\mathcal{X}_0$ with respect to scalars from Λ a test is designed that works on arguments of the forms $x+y$, where $x,y\in\mathcal{X}_1$ and $\alpha\cdot x$ with $\alpha\in M, x\in\mathcal{X}_1$. The function value table expected from a proof therefore must contain values in all arguments from the set $\mathfrak{D}:=\{x+y|x,y\in\mathcal{X}_1\}\cup\{\alpha\cdot x|\alpha\in M,x\in\mathcal{X}_1\}$. In the next subsection a test is designed on $\mathfrak{D}$ that verifies with high probability linearity of A on $\mathcal{X}_0$.

5.2.2. *The linearity test and self-correction.* As in the previous section we will only describe how things work for the function $A:\mathfrak{D}\to\mathbb{R}$. In the ideal case this function A is linear and thus uniquely encodes the coefficient vector $a\in\mathbb{R}^n$ of the related linear function.

In order to make the formulas look a bit simpler we define the abbreviation $A_\alpha(x):=A(\alpha\cdot x)/\alpha$. We repeat the following test a constant number of times:

Linearity test:

- Uniformly and independently choose random x,y from $\mathcal{X}_1$ and random α,β from M;
- check if $A(x+y)=A_\alpha(x)+A_\beta(y)$.

If all checks were correct the test accepts. Otherwise the test rejects.

Each round will inspect at most three different proof components, namely $A(x+y), A(\alpha\cdot x)$ and $A(\beta\cdot y)$. Thus in finitely many rounds $O(1)$ components will be inspected.

Clearly the linearity test accepts any linear function A with probability 1. For any $\delta>0$ and $\epsilon>0$ we can choose the number of repetitions of the linearity test so large that if

$$\Pr_{x,y\in\mathcal{X}_1,\alpha,\beta\in M}\{A(x+y)=A_\alpha(x)+A_\beta(y)\}>1-\delta \tag{5.3}$$

does not hold, then the test rejects with probability $1-\epsilon$. The following cases have to be analyzed. If the linearity test rejects the verifier rejects the proof and nothing more is required. So suppose the linearity test does not give an error. If (5.3) is not satisfied, i.e., in particular the function value table does not come from a linear function, the verifier would err. Luckily it is easy to show that the probability for this to happen is small. And according to the definition of the $\mathrm{PCP}_{\mathbb{R}}$ classes we are allowed to accept incorrect proofs with small probability. It remains to deal with the only more difficult situation: The linearity test accepts and (5.3) holds. This of course does not mean that all values in the table necessarily are the correct ones. If the verifier asks for a particular such value we must therefore guarantee that at least with high probability we can extract the correct one from the table. One can get around this problem by defining a so-called self-correction f_A on $\mathcal{X}_1$ which can be shown to be linear on $\mathcal{X}_0$. This self-correction looks as follows: For $x \in \mathcal{X}_1$ define

$$f_A(x) = \mathrm{Majority}_{y\in\mathcal{X}_1,\alpha\in M}\{A_\alpha(x+y) - A_\alpha(x)\}.$$

Hence $f_A(x)$ is the value that occurs most often in the multiset $\{A_\alpha(x+y) - A_\alpha(x) | y \in \mathcal{X}_1, \alpha \in M\}$. It could be the case that $A_\alpha(x+y)$ is not defined. If this happens we just do not count this 'value'.

LEMMA 5.5. *Under the above assumptions the function f_A is linear on $\mathcal{X}_0$ with scalars from Λ, i.e., for all $v, w \in \mathcal{X}_0$ we have $f_A(v+w) = f_A(v) + f_A(w)$ and for all $x \in \mathcal{X}_0, \lambda \in \Lambda$ we have $f_A(\lambda \cdot x) = \lambda \cdot f_A(x)$.*

PROOF. For arbitrary fixed $v \in \mathcal{X}_0$ and random $x \in \mathcal{X}_1$ by Lemma 5.4 it is $x + v \in \mathcal{X}_1$ with probability $\geq 1-\epsilon$ assuming n is large enough. Since $x \mapsto x+v$ is injective and due to the use of the uniform distribution in (5.3) replacing x by $x+v$ in (5.3) gives

$$\Pr_{x,y\in\mathcal{X}_1,\alpha,\beta\in M}\{A(x+v+y) = A_\alpha(x+v) + A_\beta(y)\} > 1-\delta-\epsilon.$$

Doing the same with y instead of x yields

$$\Pr_{x,y\in\mathcal{X}_1,\alpha,\beta\in M}\{A(x+v+y) = A_\alpha(x) + A_\beta(v+y)\} > 1-\delta-\epsilon$$

and combining these two inequalities results in

$$\Pr_{x,y\in\mathcal{X}_1,\alpha,\beta\in M}\{A_\alpha(x+v) - A_\alpha(x) = A_\beta(v+y) - A_\beta(y)\} > 1-2\delta-2\epsilon.$$

From this it follows that

$$\Pr_{x\in\mathcal{X}_1,\alpha\in M}\{f_A(v) = A_\alpha(x+v) - A_\alpha(x)\} \geq 1-2\delta-2\epsilon. \tag{5.4}$$

Similarly, for a fixed $w \in \mathcal{X}_0$ one obtains

$$\Pr_{x\in\mathcal{X}_1,\alpha\in M}\{f_A(w) = A_\alpha(x+w) - A_\alpha(x)\} \geq 1-2\delta-2\epsilon$$

and using again the fact that shifting a random $x \in \mathcal{X}_1$ by a fixed $v \in \mathcal{X}_0$ does not change the distribution too much we obtain

$$\Pr_{x\in\mathcal{X}_1,\alpha\in M}\{f_A(w) = A_\alpha(x+v+w) - A_\alpha(x+v)\} \geq 1-2\delta-3\epsilon. \tag{5.5}$$

Using the above argument a third time, now with $v+w$ instead of v (and thus 2ϵ instead of ϵ) we get

$$\Pr_{x\in\mathcal{X}_1,\alpha\in M}\{f_A(v+w) = A_\alpha(x+v+w) - A_\alpha(x)\} \geq 1-2\delta-4\epsilon. \tag{5.6}$$

Combining (5.4), (5.5) and (5.6) it follows

$$\Pr_{x\in\mathcal{X}_1,\alpha\in M}\{f_A(v+w)=f_A(v)+f_A(w)\}\geq 1-6\delta-9\epsilon.$$

This is independent of both x and α, so the probability is either 0 or 1. Hence, choosing δ and ϵ small enough it will be 1 and the first part of the linearity condition is proved.

Concerning scalar multiplicativity let $e_i\in\mathbb{R}^n$ be a unit vector and $\lambda\in\Lambda$. Since $\lambda\cdot e_i\in\mathcal{X}_0$ one can apply Lemma 5.4 together with (5.4) to get

$$\Pr_{x\in\mathcal{X}_1,\alpha\in M}\{f_A(\lambda\cdot e_i)=A_{\alpha/\lambda}(\lambda\cdot e_i+\lambda\cdot x)-A_{\alpha/\lambda}(\lambda\cdot x)\}\geq 1-2\delta-4\epsilon.$$

Since $A_{\alpha/\lambda}(\lambda\cdot e_i+\lambda\cdot x)-A_{\alpha/\lambda}(\lambda\cdot x)=\lambda(A_\alpha(e_i+x)-A_\alpha(x))$ and by (5.4)

$$\Pr_{x\in\mathcal{X}_1,\alpha\in M}\{f_A(e_i)=A_\alpha(e_i+x)-A_\alpha(x)\}\geq 1-2\delta-2\epsilon$$

it follows that

$$\Pr_{x\in\mathcal{X}_1,\alpha\in M}\{f_A(\lambda\cdot e_i)=\lambda f_A(e_i)\}\geq 1-4\delta-6\epsilon.$$

This is again independent of x and α, so choosing δ and ϵ small enough yields $f_A(\lambda\cdot e_i)=\lambda f_A(e_i)$. Finally, given additivity on $\mathcal{X}_0$ and scalar multiplicativity for scalars $\lambda\in\Lambda$ on the standard basis the claim follows. □

5.2.3. *Checking consistency.* If the function value tables for both A and B have been tested with high probability to be close to unique linear functions f_A and f_B it remains to deal with consistency of these two functions. If $a\in\mathbb{R}^n, b\in\mathbb{R}^{n^2}$ are the corresponding coefficient vectors consistency means that $b_{ij}=a_i\cdot a_j$. In this subsection it is outlined how to test it.

For any $x\in\mathcal{X}_0$ and $\epsilon>0$ it has been shown how to compute the correct value of $f_A(x)$ with probability $1-\epsilon$ by making only a constant number of queries. We can therefore from now on pretend to simply get the correct values of $f_A(x)$ and $f_B(z)$. The probabilities of obtaining an incorrect value at the places where these functions are used are added to the small probability with which we are allowed to accept incorrect proofs.

For $x\in\mathbb{R}^n$, let $x\otimes x$ denote the vector $y\in\mathbb{R}^{n(n+1)/2}$ for which $y_{i,j}=x_i\cdot x_j$, $1\leq i\leq j\leq n$. Now a is consistent with b if and only if for all $x\in\mathbb{Z}_2^n$ it is the case that $f_A(x)^2=f_B(x\otimes x)$.[7] This is the property that will be tested. Repeat the following consistency test a constant number of times:

Consistency test:

- Uniformly choose random x from $\mathbb{Z}_2^n$;
- check if $f_A(x)^2=f_B(x\otimes x)$.

If in every round of the test the check is correct the verifier accepts, otherwise it rejects.

As with the linearity test the interesting case to deal with is when the verifier accepts the consistency test with high probability.

LEMMA 5.6. *With the above notations if the consistency test accepts with probability $>\frac{3}{4}$, then consistency of a and b holds.*

[7]The appropriate domain on which f_B can be shown to be linear in particular contains $x\otimes x$ for all $x\in\mathbb{Z}_2^n$.

PROOF. The proof basically relies on the fact that if two vectors in some $\mathbb{R}^N$ are different multiplying both with a random $x \in \mathbb{Z}_2^N$ will give different results with probability at least $\frac{1}{2}$. This is applied to the two linear functions on $\mathbb{R}^{n^2}$ resulting from $a \otimes a$ and b. The same is true over $\mathbb{C}$. For details see [**71**]. □

5.2.4. *Putting everything together.* The linearity and consistency tests together ensure that any proof for which the self-corrections f_A and f_B are not linear on $\mathcal{X}_0$ or are not consistent are rejected with high probability. So the only thing left to do is to verify whether a is indeed a zero of the polynomial system. This is done by evaluating equation (1). If it evaluates to zero the verifier accepts, otherwise not.

Summarizing the results of this section we finally get the following theorem. Due to the fact that the verifier uses a proof of doubly exponential length in the theorem's statement we slightly deviate from the properties of a verifier as given in Definition 5.1. This is of no major concern as will be commented on after the theorem.

THEOREM 5.7. *For every problem $L \in \mathrm{NP}_{\mathbb{R}}$ there is a verifier working as follows: Given an instance w of size n the verifier expects a proof of length $f(n)$, where f is doubly exponential in n. The verifier generates uniformly a finite number of random strings. Using those strings it computes the addresses of finitely many proof-components it wants to read. This computation is done without reading the input w, i.e., the components to be seen only depend on the random strings generated. In its decision phase the verifier uses input w together with the finitely many components and accepts L according to the requirements of Definition* 5.1. *It has a decision time that is polynomially bounded in the input size n.*

The according statement holds for $\mathrm{NP}_{\mathbb{C}}$.

Let us comment on the theorem in view of Remark 5.2 above. Recall that the size of $\mathfrak{D}$ in our proof is doubly exponential in the input size n. Therefore, the random strings used in the proof above are exponential in length. In the verification procedure they are used to compute the proof-components which the verifier wants to see. In contrast to Definition 5.1 these components are computed independently of the concrete input w (but dependent on n). The reason to require this is that we want to forbid the verifier to potentially use exponential time in the query phase in order to decide the input. After having read the values of the finitely many components the verifier uses the input and the values of those components (and not any longer the random string) in order to make its decision after a running time being polynomial in the size of the input. The verifier constructed above thus is more restricted than general verifiers because it is limited with respect to how it computes the components to be seen.

Note however that the decisive point behind Theorem 5.7 is the structure of the verification proof. In the next section we shall see that for the full $\mathrm{PCP}_{\mathbb{R}}$ theorem transparent long proofs are invoked in a situation where inputs are of constant size. In this situation of course also the length of each random string remains constant. Then the structure of the verification procedure is more important than the parameter values; the latter automatically are constant. Therefore, when used in the framework of the full PCP theorem the verifier in Theorem 5.7 can again be chosen according to Definition 5.1.

The proof of Theorem 5.7 can be adapted word by word for the complex number BSS model. There is no argument involved that uses the presence of an ordering,

except that in the definition of $P(x, r)$ we avoided to use instead the sum of the squared single polynomials of the system as could be done over the reals. This would save some small amount of randomness, introducing at the same time a more complicated polynomial of degree 4.

5.3. The full PCP theorem. In 2005 Dinur [**38**] gave an alternative proof for the classical PCP theorem. This proof was much more combinatorial than the original one by Arora et al. In this subsection we outline how to transform Dinur's proof to the BSS model both over $\mathbb{R}$ and $\mathbb{C}$. We only sketch the main ideas and refer to [**5**] for full proof details. The next subsection then discusses our current knowledge concerning a potential proof that closer follows the lines of the original one by Arora et al.

Central aspect of Dinur's proof is the design of a very particular reduction between instances of a Constraint Satisfiability Problem CSP. The latter is a generalization of the 3-Satisfiability problem. An instance of CSP consists of a collection of constraints over a finite alphabet and the question is whether all can be satisfied in common by an assignment for variables taken from the underlying alphabet. The reduction we are looking for creates a gap in the following sense. If a given CSP instance is satisfiable so is the one generated by the reduction. But if the given instance is not satisfiable, then for the one obtained by the reduction at least a constant fraction of constraints cannot be satisfied in common. This constant fraction is the gap. It has been well known early that the existence of such an efficient gap-creating reduction is equivalent to the PCP theorem. However, before Dinur's proof it could only be designed using the PCP theorem.

It is easy to see that for a suitable variant of the QPS problem the existence of a gap reduction as well would imply the PCP theorem over both $\mathbb{R}$ and $\mathbb{C}$. So the strategy is to adapt Dinur's proof to the BSS framework. This in fact turns out to be possible. Below we describe the main ideas for the real number model. Over $\mathbb{C}$ nothing changes significantly.

5.3.1. *The problem to consider.* A problem in the real number model which is similar to the above mentioned CSP problem is the following variant of the $\mathrm{QPS}_{\mathbb{R}}$ problem. Here, the way to look upon a system of polynomial equations is slightly changed.

Definition 5.8. Let $m, k, q, s \in \mathbb{N}$. An instance of the $\mathrm{QPS}_{\mathbb{R}}(\mathrm{m}, \mathrm{k}, \mathrm{q}, \mathrm{s})$ problem is a set of m constraints. Each constraint consists of at most k polynomial equations each of degree at most two. The polynomials in a single constraint depend on at most q variable arrays which have dimension s, i.e., they range over $\mathbb{R}^s$.

Hence, a single constraint in a $\mathrm{QPS}_{\mathbb{R}}(m, k, q, s)$-instance depends on at most qs variables in $\mathbb{R}$. So if there are m constraints the whole instance contains at most qm arrays and at most qsm variables. For what follows parameters q and s are most important; q will be chosen to be 2, i.e., each constraint will depend on 2 variable arrays. Controlling s so that it remains constant is a crucial goal during the different design steps of the gap reduction. Note that the problem is $\mathrm{NP}_{\mathbb{R}}$-complete for most values of (q, s), for example if $q \geq 2, s \geq 3$.

Definition 5.9. A $\mathrm{QPS}_{\mathbb{R}}(m, k, q, s)$-instance ϕ is satisfiable if there exists an assignment in $\mathbb{R}^{mqs}$ which satisfies all of its constraints. A constraint is satisfied by an assignment if all polynomials occurring in it evaluate to zero. The minimum

fraction of unsatisfied constraints, where the minimum is taken over all possible assignments, is denoted by $\mathrm{UNSAT}(\phi)$. So if ϕ is satisfiable $\mathrm{UNSAT}(\phi) = 0$ and if ϕ is unsatisfiable, then $\mathrm{UNSAT}(\phi) \geq 1/m$.

With a gap reduction we mean an algorithm which in polynomial time transforms a $\mathrm{QPS}_{\mathbb{R}}(m, k, q, s)$-instance ϕ into a $\mathrm{QPS}_{\mathbb{R}}(m', k', q, s)$-instance ψ such that there exists a fixed constant $\epsilon > 0$ and

- if ϕ is satisfiable, then ψ is satisfiable and
- if ϕ is not satisfiable, then $\mathrm{UNSAT}(\psi) \geq \epsilon$.

Thus either all constraints in the output instance ψ are satisfiable or at least an ϵ-fraction is violated, no matter which values are assigned to the variables. Most important, ϵ is a fixed constant not depending on the size of the given instances.

The following easy lemma shows the importance of gap-reductions for the PCP theorem:

LEMMA 5.10. *Suppose for an* $\mathrm{NP}_{\mathbb{R}}$*-complete* $\mathrm{QPS}_{\mathbb{R}}(\mathrm{m}, \mathrm{k}, \mathrm{q}, \mathrm{s})$ *there exists a gap-reduction with a fixed* $\epsilon > 0$*. Then the* $\mathrm{PCP}_{\mathbb{R}}$ *theorem holds, i.e.,* $\mathrm{NP}_{\mathbb{R}} = \mathrm{PCP}_{\mathbb{R}}(O(\log n), O(1))$.

PROOF. The task is to construct a $(O(\log n), O(1))$-verifier V for the problem $\mathrm{QPS}_{\mathbb{R}}(m, k, q, s)$. Supposing the existence of a gap-reduction the verifier works as follows on an instance ϕ. First, it applies the reduction and computes ψ. As proof of satisfiability of ψ (and thus of ϕ) it expects an assignment for the variables of ψ. Then, finitely many times the following is repeated: V selects at random a constraint in ψ and evaluates it in the given assignment. Since each constraint of ψ depends on at most qs variables this bounds the number of proof components V reads in a single round. In case that ϕ is not satisfiable each assignment violates an ϵ-fraction of clauses in ψ. Thus V randomly picks with probability $\geq \frac{1}{\epsilon}$ such a constraint for the assignment given by the proof. Repeating this procedure constantly many times the error probability can be made arbitrarily small, thus proving the $\mathrm{PCP}_{\mathbb{R}}$ theorem. □

5.3.2. *Outline for creating a gap reduction.* The goal now is to design such a gap reduction following Dinur's original construction. This is done in a number of rounds. Each of them increases the gap by a factor of at least 2 if the instance on which the round was performed still had a small gap below a suitable constant ϵ_{final}. Since for the original instance ϕ it holds that either ϕ is satisfiable or at least a $\frac{1}{m}$-fraction of its constraints is always unsatisfied (i.e., one constraint), in principle it suffices to perform a logarithmic in m number of such rounds in order to create an instance which has a gap of at least ϵ_{final}. The number of rounds not being constant it is important that in a single round the size of the instance grows linearly only. That way a logarithmic number of rounds creates a polynomial growth in size only.

Each round consists of three steps, a preprocessing, a gap amplification step and a dimension reduction step. These steps use $\mathrm{QPS}_{\mathbb{R}}$-instances with different values for the number q of variable arrays the constraints of an instance depend on. The two important values for this q are 2 in the amplification step and a constant Q coming from the constant query complexity of long transparent proofs.

A round starts on an instance of the problem $\mathrm{QPS}_{\mathbb{R}}(m, k, Q, 1)$, where Q denotes the number of queries the verifier of Section 5.2 needs to verify the long transparent

proofs. So in these instances every constraint depends on at most Q arrays of dimension 1, i.e., variables ranging over $\mathbb{R}^1$. The first main step in a single round is the amplification step which amplifies the gap. However, this step requires the array dimension to be $q = 2$ as well as some nice structure on its input instances. To fulfil these requirements a preprocessing step is necessary. Though amplification increases the gap it has the disadvantage of enlarging the dimension of variable arrays. To get finally back arrays of dimension 1 it is necessary to continue after amplification with a dimension reduction step. For this step long transparent proofs are crucial.

Both the preprocessing step and the dimension reduction step decrease the gap. Since the amplification factor can be taken large enough in comparison to the two factors by which the other steps reduce the gap, in total there will be a sufficient increase of the gap after logarithmically many rounds.

Preprocessing consists of a number of relatively simple constructions, so we omit this technical step and just summarize its outcome.

PROPOSITION 5.11. *There exist a constant $d \in \mathbb{N}$ and a polytime computable reduction from* $\mathrm{QPS}_{\mathbb{R}}$ *instances to* $\mathrm{QPS}_{\mathbb{R}}$ *instances such that the following holds. The reduction maps an instance ϕ in* $\mathrm{QPS}_{\mathbb{R}}(\mathrm{m}, \mathrm{k}, \mathrm{q}, \mathrm{s})$*-instance to a* nice *instance ψ in* $\mathrm{QPS}_{\mathbb{R}}(3\mathrm{qd}^2\mathrm{m}, \mathrm{k} + \mathrm{qs}, 2, \mathrm{qs})$ *such that*

- *if ϕ is satisfiable, then ψ is satisfiable;*
- *if ϕ is not satisfiable, then $UNSAT(\psi) \geq UNSAT(\phi)/(240qd^2)$.*

The term *nice* in the above statement refers to a special structure the resulting instances exhibit. This structure is related to so-called expander graphs which are heavily used in the amplification step. Expanders in particular are regular graphs and the parameter d used in the statement denotes this regularity. Without being too technical below it will be pointed out what kind of properties of expanders are needed.

5.3.3. *The amplification step.* As already mentioned amplification requires a $\mathrm{QPS}_{\mathbb{R}}(m, k, 2, s)$-instance ψ as input. To such an instance one can canonically attach a constraint graph in which vertices correspond to variable arrays and edges correspond to the constraints depending on at most two arrays each. Constraints depending on a single array only give loops in the constraint graph.

Starting with an instance ψ obtained at the end of preprocessing a new instance ψ^t is constructed as follows. The ultimate goal is to amplify the occurence of a constraint in ψ in such a way that it influences much more constraints in ψ^t. Since constraints correspond to edges in the constraint graph this amplification is obtained by Dinur invoking deep results about expander graphs. Very roughly, constraints of the new instance collect several constraints (edges) of ψ in such a way that each violated old constraint forces violation of many of the new constraints in which it occurs. In order to make this idea working the structure of d-regular expander graphs is important.

Here is a brief outline of how the new instance ψ^t is obtained. Its construction depends both on the regularity d and an additional constant parameter $t \in \mathbb{N}$ that can be chosen arbitrarily. Both determine the factor with which the gap is amplified.

The new instance ψ^t will have the same number of variable arrays but they will be of larger dimension. For every vertex in the constraint graph of the input instance ψ a new array will be defined. The dimension of these new arrays will

be so large that they can claim values for all old arrays which can be reached in the constraint graph within at most $t + \sqrt{t}$ steps. By this we mean that blocks of components of suitable size in such a new array are identified with the variable components of a particular old array. Due to the increased dimension each single new array that way will cover several old arrays.

Since one of the conditions on the input instance is that its constraint graph is regular of degree d for some constant d the size of the new arrays is bounded by the constant $d^{t+\sqrt{t}+1} \cdot s$. So for every array in the old instance ψ there will be lots of arrays in the new instance ψ^t that claim a value for it. Of course these claimed values can be different. In the proof for the classical case there is the guarantee that at least a certain fraction of the claims will be equal because of the finite alphabet. In the real number case we do not have this guarantee because there the "alphabet" is of course infinite. However, this technical problem can easily be circumvented by adding some consistency requirements to the constraints in the new instance ψ^t.

The constraints in ψ^t will be sets of constraints of the old instance ψ together with consistency requirements as mentioned above. For every path of length $2t+1$ in the constraint graph of ψ a constraint will be added to the new instance ψ^t. This constraint will depend on the two arrays corresponding to the endpoints of the path. The new arrays claim values for all old arrays in a $t + \sqrt{t} + 1$-neighbourhood of the vertex. Therefore, all old arrays related to vertices in a certain middle segment of such a path get values from the new arrays corresponding to both end-points. The constraint that we add will express that these claimed values are consistent and that they satisfy the constraints of the old instance ψ. This finishes the rough description of how to construct the new instance.

It is easy to see that the construction transfers satisfiability from ψ to ψ^t. The hard part of Dinur's proof is to show that if the input instance ψ is not satisfiable, then the fraction of unsatisfied constraints in the new instance ψ^t is increased by a constant factor depending on t. This is shown by Dinur as follows and basically can be done similarly in the real and complex number framework. Take any assignment for the new arrays. From this assignment a plurality assignment is defined for the old arrays. More precisely, perform a random walk of t steps on the constraint graph starting in the vertex of the old array. Its plurality value is the assignment that most frequently is claimed for the old array by those new arrays that occur as an endpoint of such a walk. The further analysis now exploits both the expander properties of the constraint graph together with an additional structural requirement called niceness before. It basically addresses the number of loops each vertex in the constraint graph has. This in a suitable way makes random walks of length t basically look like walks that either have a slightly shorter or longer length. It finally guarantees the quantity $UNSAT(\psi^t)$ to grow proportionally in $\sqrt{t} \cdot UNSAT(\psi)$.

The formal statement resulting from the above ideas reads

THEOREM 5.12. *There exists an algorithm which works in polynomial time that maps a nice* $\mathrm{QPS}_{\mathbb{R}}(\mathrm{m}, \mathrm{k}, 2, \mathrm{s})$*-instance* ψ *to a* $\mathrm{QPS}_{\mathbb{R}}(\mathrm{d}^{2\mathrm{t}}\mathrm{m}, 2\sqrt{\mathrm{t}}\mathrm{k} + (2\sqrt{\mathrm{t}} + 1)\mathrm{s}, 2, \mathrm{d}^{\mathrm{t}+\sqrt{\mathrm{t}}+1}\mathrm{s})$*-instance* ψ^t *and has the following properties:*

- *If* ψ *is satisfiable, then* ψ^t *is satisfiable.*
- *If* ψ *is not satisfiable and* $UNSAT(\psi) < \frac{1}{d\sqrt{t}}$*, then* $UNSAT(\psi^t) \geq \frac{\sqrt{t}}{3520d} \cdot UNSAT(\psi)$.

5.3.4. *Dimension reduction.* Amplification increases the gap but also the dimension of variable arrays. However, the latter in the end has to remain constant because it is directly related to the query complexity. Thus in the final step of a single round the array dimension has to be reduced. Actually, it can be put down to 1 before the next round starts.

To achieve this aim the structure of transparent long proofs as explained above plays the decisive role. First, there is a natural close relation between the computation of verifiers and sets of constraints. To each string of random bits a verifier generates one can associate a constraint. This constraint expresses the verifier's computation after the random string has been generated. It is satisfied if the verifier accepts the proof with the corresponding random bits. If the input instance of the verifier is satisfiable, then there exists an assignment (i.e., a proof) which satisfies all of these constraints; and if the input instance is not satisfiable every assignment violates at least half of the constraints.

Now we apply this viewpoint to the instance ψ^t generated after amplification. The idea is to view every constraint in ψ^t as an input instance for the long transparent proof verifier and replace this constraint with the set of constraints which we described in the lines above. Note that a single constraint in ψ^t still has constant size. It depends on two arrays of dimension $s(t) := d^{t+\sqrt{t}+1} \cdot s$. Since the verifier checks this for a concrete assignment within a time bound depending on the constraint size, all of the derived constraints described above also are constant in size. It therefore is of no concern that the verifier needs long transparent proofs; they all still have constant size. More important is the structure of the verifier.

In order to realize this idea for dimension reduction first parts of the preprocessing step are applied once more; to do so the original constraints in ψ^t have to be decoupled. This means that different constraints have to depend on different arrays. It is achieved by using formal copies. Of course, the intended reduction must carry over satisfiability, so the decoupling of variables has to be repaired afterwards by introducing consistency constraints.

Now the particular structure of the long transparent proof guarantees that both the original constraints in ψ and the consistency constraints can be replaced by constraints that depend on at most Q variable arrays of dimension 1 each. Here, Q is the query complexity of the long transparent verifier.

Dimension reduction thus gives

THEOREM 5.13. *There exists a reduction which works in polynomial time and maps a* $\mathrm{QPS}_{\mathbb{R}}(\mathrm{m(t)}, \mathrm{k(t)}, 2, \mathrm{s(t)})$*-instance* ψ^t *to a* $\mathrm{QPS}_{\mathbb{R}}(\widehat{\mathrm{m}}(\mathrm{t}), \mathrm{C}, \mathrm{Q}, 1)$*-instance* $\widehat{\psi}^t$*, where* C, Q *are constants,* $\widehat{m}(t)$ *is linear in* $m(t)$ *(the multiplication factor being double exponential in* $s(t)$*) and the following holds:*

- *If* ψ^t *is satisfiable, then so is* $\widehat{\psi}^t$ *and*
- *if* ψ^t *is unsatisfiable, then* $UNSAT(\widehat{\psi}^t) \geq UNSAT(\psi^t)/(160(d+1)^2)$.

The final argument is to apply the above steps a logarithmic in m number of times for a given $\mathrm{QPS}_{\mathbb{R}}(m, k, q, s)$-instance. A suitable choice of t guarantees that the amplification factor in each round is at least 2. So starting with a fraction of $\frac{1}{m}$ unsatisfied constraints the gap is increased to a constant fraction. We finally obtain

THEOREM 5.14 (PCP theorem for $\mathrm{NP}_{\mathbb{R}}$, [**5**]). *The PCP theorem holds both in the real and the complex number model, i.e.,*

$$\mathrm{NP}_{\mathbb{R}} = \mathrm{PCP}_{\mathbb{R}}(O(\log n), O(1)) \text{ and } \mathrm{NP}_{\mathbb{C}} = \mathrm{PCP}_{\mathbb{C}}(O(\log n), O(1)).$$

All proof details are given in the full version of [**5**]. None of the arguments rely on the ordering available over the real numbers and so the statement holds as well for the complex number BSS model.

5.4. Almost transparent short proofs. Though we have seen in the previous sections that Dinur's proof of the PCP theorem can be adapted to the BSS model another interesting question remains open. Can the real number PCP theorem be proved as well along the lines of the classical proof by Arora et al.? In this final section we briefly indicate what is currently known concerning this problem.

The classical proof uses long transparent proofs as well as two additional constructions. Another verifier is designed that uses a logarithmic amount of randomness and inspects a polylogarithmic number of components. The 'almost transparent short' proof that this verifier requires in addition must obey a certain structure in order to make the final step applicable. This is a composition step of the two verifiers resulting in the final verifier whose existence implies the PCP theorem. So far it is possible to construct an almost transparent short proof for $\mathrm{NP}_{\mathbb{R}}$. However, at the time being it is not clear to the authors how to put this verifier into a more specific structure in order to make the final step working. We comment on this point at the end.

Let us shortly explain the main ideas in designing this verifier following [**72**]. Instead of using tables of linear functions as coding objects for a zero of a polynomial system now multivariate polynomials of a not too large degree are employed. They are usually called low-degree polynomials in this framework.

5.4.1. *The problem setting.* Starting point once again is the $\mathrm{QPS}_{\mathbb{R}}$ problem in the version of Definition 3.7. For it the verifier is constructed. Given the $\mathrm{NP}_{\mathbb{R}}$-completeness proof of $\mathrm{QPS}_{\mathbb{R}}$ in [**19**] an instance system $\mathcal{P}$ in variables $x_1, \ldots, x_n$ can be further assumed to be of the following particular form. Each polynomial has one the types below:

Type 1: $x_{i_1} - c_\ell$, where c_ℓ is one among finitely many fixed real constants,
Type 2: $x_{i_1} - (x_{i_2} - x_{i_3})$,
Type 3: $x_{i_1} - (x_{i_2} + x_{i_3})$ or
Type 4: $x_{i_1} - (x_{i_2} \cdot x_{i_3})$.

Here the i_1, i_2 and i_3 do not have to be different. As with the linear encodings used before we change a bit the viewpoint on an assignment for the system's variables.

To do this the index set of the variables in $\mathcal{P}$ is coded differently. Choose integers h, k such that $h^k \geq n$ and set $H := \{1, \ldots, h\}$. Now H^k is used as index set instead of $\{1, \ldots, n\}$. Thus a real assignment $a \in \mathbb{R}^n$ to the variables is a function $f_a : H^k \to \mathbb{R}$. Next, the way to look upon the system $\mathcal{P}$ is altered. More precisely, for each polynomial in $\mathcal{P}$ its type is extracted by means of using certain characteristic functions for the types.

Towards this end $\mathcal{P}$ is seen as a subset of some universe U to be specified; now identify $\mathcal{P}$ with the function $\chi : U \to \{0, 1\}$ which maps elements, i.e., polynomials in $\mathcal{P}$ to 1 and elements outside $\mathcal{P}$ to 0. Actually, we will first split $\mathcal{P}$ in four parts $\mathcal{P}^1$, $\mathcal{P}^2$, $\mathcal{P}^3$ and $\mathcal{P}^4$. Part $\mathcal{P}^1$ further splits into finitely many parts $\mathcal{P}^1_\ell$, one for each

real coefficient c_ℓ of the system introduced via a polynomial of type 1. A triple $(i_1, i_2, i_3) \in H^{3k}$ uniquely identifies a polynomial in each part. For example, in part $\mathcal{P}^1_\ell$ it identifies the polynomial $x_{i_1} - c_\ell$, in part $\mathcal{P}^2$ it identifies the polynomial $x_{i_1} - (x_{i_2} - x_{i_3})$, and so on. Hence, each part of $\mathcal{P}$ can be identified with a subset of H^{3k}. The characteristic functions of the respective parts are denoted by χ^1_ℓ, χ^2, χ^3, and χ^4, respectively.

The solvability question this way is transformed into the question of the existence of a function $f_a : H^k \to \mathbb{R}$ such that for all $(i_1, i_2, i_3) \in H^{3k}$ the following equations hold:

$$\begin{aligned}
\chi^1_\ell(i_1, i_2, i_3) \cdot (f(i_1) - c_\ell) &= 0 \quad \text{for all } \ell, \\
\chi^2(i_1, i_2, i_3) \cdot (f(i_1) - (f(i_2) - f(i_3))) &= 0, \\
\chi^3(i_1, i_2, i_3) \cdot (f(i_1) - (f(i_2) + f(i_3))) &= 0, \\
\chi^4(i_1, i_2, i_3) \cdot (f(i_1) - (f(i_2) \cdot f(i_3))) &= 0.
\end{aligned}$$

Squaring and adding lead to $\sum_{(i_1,i_2,i_3)\in H^{3k}} g(i_1, i_2, i_3) = 0$, where $g : H^{3k} \to \mathbb{R}$ is defined as

$$\begin{aligned}
g(i_1, i_2, i_3) \;:=\; & \textstyle\sum_\ell \left[\chi^{(1)}_\ell(i_1, i_2, i_3) \cdot (f(i_1) - c_\ell)\right]^2 \\
& + \left[\chi^{(2)}(i_1, i_2, i_3) \cdot (f(i_1) - (f(i_2) - f(i_3)))\right]^2 \\
& + \left[\chi^{(3)}(i_1, i_2, i_3) \cdot (f(i_1) - (f(i_2) + f(i_3)))\right]^2 \\
& + \left[\chi^{(4)}(i_1, i_2, i_3) \cdot (f(i_1) - (f(i_2) \cdot f(i_3)))\right]^2.
\end{aligned}$$

Summarizing, an assignment $a \in \mathbb{R}^n$ is a zero of the given system $\mathcal{P}$ if and only if the sum $\sum_{(i_1,i_2,i_3)\in H^{3k}} g(i_1, i_2, i_3) = 0$, where g is defined as above using an encoding $f_a : H^k \to \mathbb{R}$ for a. The degree of g in each of its variables is $d := O(h)$.

5.4.2. *Sum check and low degree extensions.* At the moment not much is gained. If the sum is evaluated term by term there are at least $|H|^k \geq n$ many terms that depend on at least one value of f_a, thus such a direct evaluation would inspect too many proof components. However, the particular form allows to proceed differently in order to circumvent this problem. The first step is to apply a well-known technique called sum-check procedure [**62**] in order to evaluate the above huge sum more efficiently using randomization. The idea is to express the sum recursively as iterated sum of univariate polynomials and including an encoding of those univariate polynomials in the verification proof. More precisely, for $1 \leq i \leq 3k$ one defines partial-sum polynomials of g as

$$g_i(x_1, \ldots, x_i) := \sum_{y_{i+1}\in H} \sum_{y_{i+2}\in H} \cdots \sum_{y_{3k}\in H} g(x_1, \ldots, x_i, y_{i+1}, \ldots, y_{3k}).$$

Note that $\sum_{r\in H^{3k}} g(r) = \sum_{x_1\in H} g_1(x_1)$ and $g_i(x_1, \ldots, x_i) = \sum_{y\in H} g_{i+1}(x_1, \ldots, x_i, y)$ for all i.

In order to make the probability analysis of the following procedure working it turns out that the polynomials g and g_i have to be defined on a larger set F^{3k}, where $H \subset F$ (even though the sum to be computed still ranges over H^{3k}). The verifier expects a proof to contain for each $1 \leq i \leq 3k, (r_1, \ldots, r_{i-1}) \in F^{i-1}$ a univariate polynomial $x \to g'_i(r_1, \ldots, r_{i-1}, x)$ of degree at most d. The proof is required to represent such a polynomial by specifying its $d+1$ many real coefficients. An ideal proof is supposed to use the corresponding restriction $x \to g_i(r_1, \ldots, r_{i-1}, x)$ of the

partial-sum polynomial g_i as $g_i'(r_1, \ldots, r_{i-1}, x)$. The basis of the sum-check procedure now is to verify for all i the relation $g_i(x_1, \ldots, x_i) = \sum_{y \in H} g_{i+1}(x_1, \ldots, x_i, y)$ together with $\sum_{y \in H} g_1(y) = 0$. In the corresponding test this is done finitely many times using a random choice $(r_1, \ldots, r_{3k}) \in F^{3k}$ of points in F for the x_i's. It can be shown that this test when accepted guarantees with high probability that the pairs (g_i, g_{i+1}) are consistent and that the entire sum evaluates to 0. Most important, this part of the verifier needs the following resources. Choosing $3k$ many points from F randomly requires $O(k \cdot \log |F|)$ many random bits. For each of the $O(k)$ many equations tested the verifier reads $d + 1$ many proof components representing the univariate polynomial $y \mapsto g_i(r_1, \ldots, r_{i-1}, y)$. For the final check $\sum_{y \in H} g_1(y) = 0$ a constant number of values from f_a is required.

The analysis (not given here) then guarantees everything to work fine when the involved parameters are chosen according to $k = O(\log n), h = O(\log n), |F| = poly \log n$.

There is one major new problem that has been tacitly introduced in the above sum-check procedure. Its probability analysis makes it necessary to consider the g_i on a larger domain F^{3k}. But f_a originally is defined on H^k only. So if in the sum-check part a value of f_a in an argument outside H^k has to be inspected, we must first extend f_a consistently to domain F^k. Consistency of course is crucial here in order to make sure that still the same f_a, and thus the same assignment $a \in \mathbb{R}^n$, is used.

Dealing with this problem is the main task for obtaining the desired verifier. By interpolation every function $f : H^k \to \mathbb{R}$ can in a unique way be seen as a polynomial in k variables that ranges over H and with degree at most $h - 1$ in each variable. This polynomial of course is defined as well on any larger set F^k since we consider both H and F as subsets of $\mathbb{R}$. It is this low-degree extension that should be used in the sum-check. Note that the above g in fact is obtained in a similar way using as well the low-degree extensions of the $\chi^{(i)}$ functions in its definition.

But then we are left with a question that is very similar to what has been discussed in relation to long transparent proofs. The verifier expects a function value table of a function $\tilde{f} : F^k \to \mathbb{R}$. As part of its test it first has to convince itself that the table with high probability is close to a unique low-degree polynomial $f : F^k \to \mathbb{R}$. This polynomial is identical to the low-degree extension of a function f_a. It is this a which then is expected by the verifier to solve the given polynomial system.

So it is necessary to design a test which checks such a function $\tilde{f}$ for being close to a low-degree polynomial with high probability. Once again, a problem for doing it is that the domains H and F cannot be taken to have a nice structure such as finite fields which are used in the Turing setting. However, based on work by Friedl et al. [**42**] it is shown in [**72**] that such a test can be developed and additionally respects the required resource bounds. Putting this low-degree test and the sum-check procedure together one obtains

THEOREM 5.15 ([**72**]). $\mathrm{NP}_{\mathbb{R}} = \mathrm{PCP}_{\mathbb{R}}(O(\log n), poly \log n)$.

It remains open whether the full $\mathrm{PCP}_{\mathbb{R}}$ theorem can be obtained pushing the above ideas forward by combining the two verifiers using a long transparent and a short almost transparent proof, respectively. The reason why we are doubtful is that in order to apply a real version of verifier composition - a technique introduced

by [**3**] and similar to the use of the long transparent proof in Dinur's approach - the verifier of Theorem 5.15 needs to obey an improved structure. In the classical proof this better structure is obtained by designing yet another low-degree test which considers the *total* degree of polynomials. In contrast, the low-degree test used above is dealing with the maximal degree of each variable. What is the problem here? It seems that designing a better structured total degree test over the reals might use a much larger domain to be tested, thus leading to a higher amount of randomness necessary. We do not know at the time being whether such a test can be designed respecting the required resourse bounds. This certainly is an interesting research question of independent interest since it deals with a typical example of property testing in real domains.

Acknowledgement

Thanks are due to an anonymous referee for his/her careful reading and a lot of comments helping to improve the presentation.

References

[1] Sanjeev Arora and Boaz Barak, *Computational complexity*, Cambridge University Press, Cambridge, 2009. A modern approach. MR2500087 (2010i:68001)

[2] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy, *Proof verification and the hardness of approximation problems*, J. ACM **45** (1998), no. 3, 501–555, DOI 10.1145/278298.278306. MR1639346 (99d:68077b)

[3] Sanjeev Arora and Shmuel Safra, *Probabilistic checking of proofs: a new characterization of NP*, J. ACM **45** (1998), no. 1, 70–122, DOI 10.1145/273865.273901. MR1614328 (99d:68077a)

[4] Giogio Ausiello, Pierluigi Crescenzi, Giorgio Gambosi, Viggo Kann, Alberto Marchetti-Spaccamela, and Marco Protasi, *Complexity and approximation*, Springer-Verlag, Berlin, 1999. Combinatorial optimization problems and their approximability properties; With 1 CD-ROM (Windows and UNIX). MR1734026 (2001f:68002)

[5] Baartse, Martijn, Meer, Klaus: The PCP theorem for NP over the reals. Extended abstract in: Proc. 30th Symposium on Theoretical Aspects of Computer Science STACS 2013, Leibniz International Proceedings in Informatics Schloss Dagstuhl, LIPICS Vol. 20, 104–115, 2013. http://dx.doi.org/10.4230/LIPIcs.STACS.2013.104
Full version available from the authors.

[6] Bank, Benrd, Giusti, Marc, Heintz, Joos: Polar, bipolar and copolar varieties: Real solving of algebraic varieties with intrinsic complexity, Preprint 2012.

[7] Bank, Benrd, Giusti, Marc, Heintz, Joos: Point searching in real singular complete intersection varieties - algorithms of intrinsic complexity. To appear in Mathematics of Computation, American Mathematical Society.

[8] Saugata Basu, *A complex analogue of Toda's theorem*, Found. Comput. Math. **12** (2012), no. 3, 327–362, DOI 10.1007/s10208-011-9105-5. MR2915565

[9] Saugata Basu, Richard Pollack, and Marie-Françoise Roy, *On the combinatorial and algebraic complexity of quantifier elimination*, J. ACM **43** (1996), no. 6, 1002–1045, DOI 10.1145/235809.235813. MR1434910 (98c:03077)

[10] Saugata Basu and Thierry Zell, *Polynomial hierarchy, Betti numbers, and a real analogue of Toda's theorem*, Found. Comput. Math. **10** (2010), no. 4, 429–454, DOI 10.1007/s10208-010-9062-4. MR2657948 (2011j:68047)

[11] Carlos Beltrán and Luis Miguel Pardo, *Smale's 17th problem: average polynomial time to compute affine and projective solutions*, J. Amer. Math. Soc. **22** (2009), no. 2, 363–385, DOI 10.1090/S0894-0347-08-00630-9. MR2476778 (2009m:90147)

[12] Carlos Beltrán and Luis Miguel Pardo, *Efficient polynomial system-solving by numerical methods*, J. Fixed Point Theory Appl. **6** (2009), no. 1, 63–85, DOI 10.1007/s11784-009-0113-x. MR2558484 (2010j:65071)

[13] Carlos Beltrán and Luis Miguel Pardo, *Fast linear homotopy to find approximate zeros of polynomial systems*, Found. Comput. Math. **11** (2011), no. 1, 95–129, DOI 10.1007/s10208-010-9078-9. MR2754191 (2011m:65111)

[14] Shai Ben-David, Klaus Meer, and Christian Michaux, *A note on non-complete problems in* $\mathrm{NP}_{\mathbf{R}}$, J. Complexity **16** (2000), no. 1, 324–332, DOI 10.1006/jcom.1999.0537. Real computation and complexity (Schloss Dagstuhl, 1998). MR1762408 (2001g:68029)

[15] D. N. Bernstein, *The number of roots of a system of equations*, Funkcional. Anal. i Priložen. **9** (1975), no. 3, 1–4 (Russian). MR0435072 (55 #8034)

[16] Lenore Blum, *Computing over the reals: where Turing meets Newton*, Notices Amer. Math. Soc. **51** (2004), no. 9, 1024–1034. MR2089092 (2005e:68044)

[17] Lenore Blum, Felipe Cucker, Mike Shub, and Steve Smale, *Algebraic settings for the problem "*$\mathrm{P} \neq \mathrm{NP}$*?"*, The mathematics of numerical analysis (Park City, UT, 1995), Lectures in Appl. Math., vol. 32, Amer. Math. Soc., Providence, RI, 1996, pp. 125–144. MR1421332 (98a:68064)

[18] Lenore Blum, Felipe Cucker, Michael Shub, and Steve Smale, *Complexity and real computation*, Springer-Verlag, New York, 1998. With a foreword by Richard M. Karp. MR1479636 (99a:68070)

[19] Lenore Blum, Mike Shub, and Steve Smale, *On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines*, Bull. Amer. Math. Soc. (N.S.) **21** (1989), no. 1, 1–46, DOI 10.1090/S0273-0979-1989-15750-9. MR974426 (90a:68022)

[20] Lenore Blum and Steve Smale, *The Gödel incompleteness theorem and decidability over a ring*, From Topology to Computation: Proceedings of the Smalefest (Berkeley, CA, 1990), Springer, New York, 1993, pp. 321–339. MR1246131 (96b:03056)

[21] William W. Boone, *The word problem*, Proc. Nat. Acad. Sci. U.S.A. **44** (1958), 1061–1065. MR0101267 (21 #80)

[22] Mark Braverman and Stephen Cook, *Computing over the reals: foundations for scientific computing*, Notices Amer. Math. Soc. **53** (2006), no. 3, 318–329. MR2208383 (2006m:68019)

[23] Peter Bürgisser, *On the structure of Valiant's complexity classes*, Discrete Math. Theor. Comput. Sci. **3** (1999), no. 3, 73–94 (electronic). MR1734899 (2000m:68068)

[24] Peter Bürgisser, *Completeness and reduction in algebraic complexity theory*, Algorithms and Computation in Mathematics, vol. 7, Springer-Verlag, Berlin, 2000. MR1771845 (2001g:68030)

[25] Peter Bürgisser and Michael Clausen, *Algebraische Komplexitätstheorie. I. Eine Einführung*, Sém. Lothar. Combin. **36** (1996), Art. S36a, approx. 18pp. (electronic) (German, with German summary). MR1429584 (98d:68109)

[26] Peter Bürgisser and Felipe Cucker, *Variations by complexity theorists on three themes of Euler, Bézout, Betti, and Poincaré*, Complexity of computations and proofs, Quad. Mat., vol. 13, Dept. Math., Seconda Univ. Napoli, Caserta, 2004, pp. 73–151. MR2131406 (2006c:68053)

[27] Peter Bürgisser and Felipe Cucker, *Counting complexity classes for numeric computations. II. Algebraic and semialgebraic sets*, J. Complexity **22** (2006), no. 2, 147–191, DOI 10.1016/j.jco.2005.11.001. MR2200367 (2007b:68059)

[28] Peter Bürgisser and Felipe Cucker, *On a problem posed by Steve Smale*, Ann. of Math. (2) **174** (2011), no. 3, 1785–1836, DOI 10.4007/annals.2011.174.3.8. MR2846491

[29] Wesley Calvert, Ken Kramer, and Russell Miller, *Noncomputable functions in the Blum-Shub-Smale model*, Log. Methods Comput. Sci. **7** (2011), no. 2, 2:15, 20, DOI 10.2168/LMCS-7(2:15)2011. MR2804634 (2012g:03116)

[30] Olivier Chapuis and Pascal Koiran, *Saturation and stability in the theory of computation over the reals*, Ann. Pure Appl. Logic **99** (1999), no. 1-3, 1–49, DOI 10.1016/S0168-0072(98)00060-8. MR1708145 (2001c:03101)

[31] George E. Collins, *Quantifier elimination for real closed fields by cylindrical algebraic decomposition*, Automata theory and formal languages (Second GI Conf., Kaiserslautern, 1975), Springer, Berlin, 1975, pp. 134–183. Lecture Notes in Comput. Sci., Vol. 33. MR0403962 (53 #7771)

[32] Felipe Cucker, *The arithmetical hierarchy over the reals*, J. Logic Comput. **2** (1992), no. 3, 375–395, DOI 10.1093/logcom/2.3.375. MR1177970 (93k:03043)

[33] Felipe Cucker, $\mathrm{P}_{\mathbf{R}} \neq \mathrm{NC}_{\mathbf{R}}$, J. Complexity **8** (1992), no. 3, 230–238, DOI 10.1016/0885-064X(92)90024-6. MR1187416 (93m:03068)

[34] Felipe Cucker and Pascal Koiran, *Computing over the reals with addition and order: higher complexity classes*, J. Complexity **11** (1995), no. 3, 358–376, DOI 10.1006/jcom.1995.1018. MR1349264 (96h:68068)

[35] Felipe Cucker and Klaus Meer, *Logics which capture complexity classes over the reals*, J. Symbolic Logic **64** (1999), no. 1, 363–390, DOI 10.2307/2586770. MR1683914 (2000f:03118)

[36] Felipe Cucker and Francesc Rosselló, *On the complexity of some problems for the Blum, Shub & Smale model*, LATIN '92 (São Paulo, 1992), Lecture Notes in Comput. Sci., vol. 583, Springer, Berlin, 1992, pp. 117–129, DOI 10.1007/BFb0023823. MR1253351

[37] Jean-Pierre Dedieu, Gregorio Malajovich, and Mike Shub, *On the curvature of the central path of linear programming theory*, Found. Comput. Math. **5** (2005), no. 2, 145–171, DOI 10.1007/s10208-003-0116-8. MR2149414 (2006a:90053)

[38] Irit Dinur, *The PCP theorem by gap amplification*, J. ACM **54** (2007), no. 3, Art. 12, 44, DOI 10.1145/1236457.1236459. MR2314254 (2008f:68037b)

[39] Noaï Fitchas, André Galligo, and Jacques Morgenstern, *Precise sequential and parallel complexity bounds for quantifier elimination over algebraically closed fields*, J. Pure Appl. Algebra **67** (1990), no. 1, 1–14, DOI 10.1016/0022-4049(90)90159-F. MR1076744 (91j:03010)

[40] Hervé Fournier and Pascal Koiran, *Are lower bounds easier over the reals?*, STOC '98 (Dallas, TX), ACM, New York, 1999, pp. 507–513. MR1715598

[41] Hervé Fournier and Pascal Koiran, *Lower bounds are not easier over the reals: inside PH*, Automata, languages and programming (Geneva, 2000), Lecture Notes in Comput. Sci., vol. 1853, Springer, Berlin, 2000, pp. 832–843, DOI 10.1007/3-540-45022-X_70. MR1795939 (2001h:68041)

[42] Katalin Friedl, Zsolt Hátsági, and Alexander Shen, *Low-degree tests*, Algorithms (Arlington, VA, 1994), ACM, New York, 1994, pp. 57–64. MR1285151 (95d:68055)

[43] Pierluigi Frisco, *Computing with cells*, Oxford University Press, Oxford, 2009. Advances in membrane computing. MR2761790 (2012c:68002)

[44] Michael R. Garey and David S. Johnson, *Computers and intractability*, W. H. Freeman and Co., San Francisco, Calif., 1979. A guide to the theory of NP-completeness; A Series of Books in the Mathematical Sciences. MR519066 (80g:68056)

[45] Christine Gaßner, *A hierarchy below the halting problem for additive machines*, Theory Comput. Syst. **43** (2008), no. 3-4, 464–470, DOI 10.1007/s00224-007-9020-y. MR2461280 (2010d:68033)

[46] Erich Grädel and Klaus Meer, *Descriptive complexity theory over the real numbers*, The mathematics of numerical analysis (Park City, UT, 1995), Lectures in Appl. Math., vol. 32, Amer. Math. Soc., Providence, RI, 1996, pp. 381–403. MR1421346 (98c:03086)

[47] Dima Yu. Grigor′ev, *Complexity of deciding Tarski algebra*, J. Symbolic Comput. **5** (1988), no. 1-2, 65–108, DOI 10.1016/S0747-7171(88)80006-3. MR949113 (90b:03054)

[48] Haykin, Simon: Neural Networks - A Comprehensive Foundation. Prentice Hall, 2nd edition, 1999.

[49] Heintz, J., Roy, M.F., Solerno, P.: *On the complexity of semialgebraic sets.* Proceedings IFIP 1989, San Fracisco, North-Holland 293–298 , 1989.

[50] Stefan Hougardy, Hans Jürgen Prömel, and Angelika Steger, *Probabilistically checkable proofs and their consequences for approximation algorithms*, Discrete Math. **136** (1994), no. 1-3, 175–223, DOI 10.1016/0012-365X(94)00112-V. Trends in discrete mathematics. MR1313286 (96k:68073)

[51] Jansen, Klaus, Margraf, Margraf: Approximative Algorithmen und Nichtapproximierbarkeit. de Gruyter, 2008.

[52] Hubertus Th. Jongen, Klaus Meer, and Eberhard Triesch, *Optimization theory*, Kluwer Academic Publishers, Boston, MA, 2004. MR2069118 (2005b:90001)

[53] Ker-I Ko, *Complexity theory of real functions*, Progress in Theoretical Computer Science, Birkhäuser Boston Inc., Boston, MA, 1991. MR1137517 (93i:03057)

[54] Pascal Koiran, *A weak version of the Blum, Shub & Smale model*, 34th Annual Symposium on Foundations of Computer Science (Palo Alto, CA, 1993), IEEE Comput. Soc. Press, Los Alamitos, CA, 1993, pp. 486–495, DOI 10.1109/SFCS.1993.366838. MR1328445

[55] Pascal Koiran, *Computing over the reals with addition and order*, Theoret. Comput. Sci. **133** (1994), no. 1, 35–47, DOI 10.1016/0304-3975(93)00063-B. Selected papers of the Workshop on Continuous Algorithms and Complexity (Barcelona, 1993). MR1294424 (95m:68040)

[56] Pascal Koiran, *Elimination of constants from machines over algebraically closed fields*, J. Complexity **13** (1997), no. 1, 65–82, DOI 10.1006/jcom.1997.0433. MR1449762 (98d:68079)

[57] Pascal Koiran, *The real dimension problem is* $\mathrm{NP}_{\mathbf{R}}$*-complete*, J. Complexity **15** (1999), no. 2, 227–238, DOI 10.1006/jcom.1999.0502. MR1693880 (2000d:68041)

[58] Pascal Koiran, *The complexity of local dimensions for constructible sets*, J. Complexity **16** (2000), no. 1, 311–323, DOI 10.1006/jcom.1999.0536. Real computation and complexity (Schloss Dagstuhl, 1998). MR1762407 (2002b:14083)

[59] Bernhard Korte and Jens Vygen, *Combinatorial optimization*, 5th ed., Algorithms and Combinatorics, vol. 21, Springer, Heidelberg, 2012. Theory and algorithms. MR2850465

[60] Richard E. Ladner, *On the structure of polynomial time reducibility*, J. Assoc. Comput. Mach. **22** (1975), 155–171. MR0464698 (57 #4623)

[61] Tien-Yien Li, *Numerical solution of multivariate polynomial systems by homotopy continuation methods*, Acta numerica, 1997, Acta Numer., vol. 6, Cambridge Univ. Press, Cambridge, 1997, pp. 399–436, DOI 10.1017/S0962492900002749. MR1489259 (2000i:65084)

[62] Carsten Lund, Lance Fortnow, Howard Karloff, and Noam Nisan, *Algebraic methods for interactive proof systems*, J. Assoc. Comput. Mach. **39** (1992), no. 4, 859–868, DOI 10.1145/146585.146605. MR1187215 (94j:68268)

[63] Roger C. Lyndon and Paul E. Schupp, *Combinatorial group theory*, Springer-Verlag, Berlin, 1977. Ergebnisse der Mathematik und ihrer Grenzgebiete, Band 89. MR0577064 (58 #28182)

[64] Gregorio Malajovich, *Nonlinear equations*, Publicações Matemáticas do IMPA. [IMPA Mathematical Publications], Instituto Nacional de Matemática Pura e Aplicada (IMPA), Rio de Janeiro, 2011. With an appendix by Carlos Beltrán, Jean-Pierre Dedieu, Luis Miguel Pardo and Mike Shub; 28º Colóquio Brasileiro de Matemática. [28th Brazilian Mathematics Colloquium]. MR2798351 (2012j:65148)

[65] Gregorio Malajovich and Klaus Meer, *On the structure of* $\mathcal{NP}_{\mathbf{C}}$, SIAM J. Comput. **28** (1999), no. 1, 27–35 (electronic), DOI 10.1137/S0097539795294980. MR1630421 (2000b:68086)

[66] Gregorio Malajovich and Klaus Meer, *Computing minimal multi-homogeneous Bézout numbers is hard*, Theory Comput. Syst. **40** (2007), no. 4, 553–570, DOI 10.1007/s00224-006-1322-y. MR2305377 (2009a:68028)

[67] Matijasevich, Yuri: Enumerable sets are diophantine. Dokl. Acad. Nauk 191, 279–282, 1970.

[68] Klaus Meer, *A note on a* $\mathrm{P} \neq \mathrm{NP}$ *result for a restricted class of real machines*, J. Complexity **8** (1992), no. 4, 451–453, DOI 10.1016/0885-064X(92)90007-X. MR1195263 (94b:03072)

[69] Klaus Meer, *On the complexity of quadratic programming in real number models of computation*, Theoret. Comput. Sci. **133** (1994), no. 1, 85–94, DOI 10.1016/0304-3975(94)00070-0. Selected papers of the Workshop on Continuous Algorithms and Complexity (Barcelona, 1993). MR1294427 (95e:90066)

[70] Klaus Meer, *Counting problems over the reals*, Theoret. Comput. Sci. **242** (2000), no. 1-2, 41–58, DOI 10.1016/S0304-3975(98)00190-X. MR1769145 (2002g:68041)

[71] Klaus Meer, *Transparent long proofs: a first PCP theorem for* $\mathrm{NP}_{\mathbb{R}}$, Found. Comput. Math. **5** (2005), no. 3, 231–255, DOI 10.1007/s10208-005-0142-1. MR2168677 (2006g:68075)

[72] Klaus Meer, *Almost transparent short proofs for* $\mathrm{NP}_{\mathbb{R}}$, Fundamentals of computation theory, Lecture Notes in Comput. Sci., vol. 6914, Springer, Heidelberg, 2011, pp. 41–52, DOI 10.1007/978-3-642-22953-4_4. MR2886893

[73] Klaus Meer, *On Ladner's result for a class of real machines with restricted use of constants*, Inform. and Comput. **210** (2012), 13–20, DOI 10.1016/j.ic.2011.11.001. MR2878798

[74] Meer, Klaus: Some initial thoughts on bounded query computations over the reals. To appear in: Special Issue on 'Frontier between Decidability and Undecidability and Related Problem', International Journal of Foundations of Computer Science.

[75] Klaus Meer and Martin Ziegler, *An explicit solution of Post's problem over the reals*, J. Complexity **24** (2008), no. 1, 3–15, DOI 10.1016/j.jco.2006.09.004. MR2386927 (2009k:03059)

[76] Klaus Meer and Martin Ziegler, *Real computational universality: the word problem for a class of groups with infinite presentation*, Found. Comput. Math. **9** (2009), no. 5, 599–609, DOI 10.1007/s10208-009-9048-2. MR2534405 (2011d:20063)

[77] Friedhelm Meyer auf der Heide, *A polynomial linear search algorithm for the n-dimensional knapsack problem*, J. Assoc. Comput. Mach. **31** (1984), no. 3, 668–676, DOI 10.1145/828.322450. MR819161

[78] Friedhelm Meyer auf der Heide, *Fast algorithms for n-dimensional restrictions of hard problems*, J. Assoc. Comput. Mach. **35** (1988), no. 3, 740–747, DOI 10.1145/44483.44490. MR963170 (89m:68051)

[79] Christian Michaux, P $\neq$ NP *over the nonstandard reals implies* P $\neq$ NP *over* $\mathbf{R}$, Theoret. Comput. Sci. **133** (1994), no. 1, 95–104, DOI 10.1016/0304-3975(94)00067-0. Selected papers of the Workshop on Continuous Algorithms and Complexity (Barcelona, 1993). MR1294428 (95h:03099)

[80] Alexander Morgan and Andrew Sommese, *A homotopy for solving general polynomial systems that respects m-homogeneous structures*, Appl. Math. Comput. **24** (1987), no. 2, 101–113, DOI 10.1016/0096-3003(87)90063-4. MR914806 (88j:65110)

[81] Michael A. Nielsen and Isaac L. Chuang, *Quantum computation and quantum information*, Cambridge University Press, Cambridge, 2000. MR1796805 (2003j:81038)

[82] Erich Novak and Henryk Woźniakowski, *Tractability of multivariate problems. Vol. 1: Linear information*, EMS Tracts in Mathematics, vol. 6, European Mathematical Society (EMS), Zürich, 2008. MR2455266 (2009m:46037)

[83] Petr Sergeevich Novikov, *On the algorithmic insolvability of the word problem in group theory*, American Mathematical Society Translations, Ser 2, Vol. 9, American Mathematical Society, Providence, R. I., 1958, pp. 1–122. MR0092784 (19,1158b)

[84] Piergiorgio Odifreddi, *Classical recursion theory*, Studies in Logic and the Foundations of Mathematics, vol. 125, North-Holland Publishing Co., Amsterdam, 1989. The theory of functions and sets of natural numbers; With a foreword by G. E. Sacks. MR982269 (90d:03072)

[85] Emil L. Post, *Recursively enumerable sets of positive integers and their decision problems*, Bull. Amer. Math. Soc. **50** (1944), 284–316. MR0010514 (6,29f)

[86] Emil L. Post, *A variant of a recursively unsolvable problem*, Bull. Amer. Math. Soc. **52** (1946), 264–268. MR0015343 (7,405b)

[87] Jaikumar Radhakrishnan and Madhu Sudan, *On Dinur's proof of the PCP theorem*, Bull. Amer. Math. Soc. (N.S.) **44** (2007), no. 1, 19–61 (electronic), DOI 10.1090/S0273-0979-06-01143-8. MR2265009 (2008f:68036)

[88] James Renegar, *On the computational complexity and geometry of the first-order theory of the reals. I. Introduction. Preliminaries. The geometry of semi-algebraic sets. The decision problem for the existential theory of the reals*, J. Symbolic Comput. **13** (1992), no. 3, 255–299, DOI 10.1016/S0747-7171(10)80003-3. MR1156882 (93h:03011a)

[89] Ronitt Rubinfeld and Madhu Sudan, *Self-testing polynomial functions efficiently and over rational domains*, Algorithms (Orlando, FL, 1992), ACM, New York, 1992, pp. 23–32. MR1173877 (93f:68067)

[90] Marcus Schaefer, *Complexity of some geometric and topological problems*, Graph drawing, Lecture Notes in Comput. Sci., vol. 5849, Springer, Berlin, 2010, pp. 334–344, DOI 10.1007/978-3-642-11805-0_32. MR2680464

[91] Uwe Schöning, *A uniform approach to obtain diagonal sets in complexity classes*, Theoret. Comput. Sci. **18** (1982), no. 1, 95–103, DOI 10.1016/0304-3975(82)90114-1. MR650242 (83b:68055)

[92] Harold Schreiber, Klaus Meer, and Burkhard J. Schmitt, *Dimensional synthesis of planar Stephenson mechanisms for motion generation using circlepoint search and homotopy methods*, Mech. Mach. Theory **37** (2002), no. 7, 717–737, DOI 10.1016/S0094-114X(02)00016-2. MR1912977 (2003d:70008)

[93] Michael Shub, *Complexity of Bezout's theorem. VI. Geodesics in the condition (number) metric*, Found. Comput. Math. **9** (2009), no. 2, 171–178, DOI 10.1007/s10208-007-9017-6. MR2496558 (2010f:65103)

[94] Michael Shub and Steve Smale, *Complexity of Bézout's theorem. I. Geometric aspects*, J. Amer. Math. Soc. **6** (1993), no. 2, 459–501, DOI 10.2307/2152805. MR1175980 (93k:65045)

[95] Michael Shub and Steve Smale, *On the intractability of Hilbert's Nullstellensatz and an algebraic version of "*NP $\neq$ P*?"*, Duke Math. J. **81** (1995), no. 1, 47–54 (1996), DOI 10.1215/S0012-7094-95-08105-8. A celebration of John F. Nash, Jr. MR1381969 (97h:03067)

[96] Steve Smale, *Some remarks on the foundations of numerical analysis*, SIAM Rev. **32** (1990), no. 2, 211–220, DOI 10.1137/1032043. MR1056052 (91k:00007)

[97] Steve Smale, *Complexity theory and numerical analysis*, Acta numerica, 1997, Acta Numer., vol. 6, Cambridge Univ. Press, Cambridge, 1997, pp. 523–551, DOI 10.1017/S0962492900002774. MR1489262 (99d:65385)
[98] Steve Smale, *Mathematical problems for the next century*, Mathematics: frontiers and perspectives, Amer. Math. Soc., Providence, RI, 2000, pp. 271–294. MR1754783 (2001i:00003)
[99] Robert I. Soare, *Recursively enumerable sets and degrees*, Perspectives in Mathematical Logic, Springer-Verlag, Berlin, 1987. A study of computable functions and computably generated sets. MR882921 (88m:03003)
[100] Andrew J. Sommese and Charles W. Wampler II, *The numerical solution of systems of polynomials*, World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2005. Arising in engineering and science. MR2160078 (2007a:14065)
[101] Apostolos Syropoulos, *Hypercomputation*, Springer, New York, 2008. Computing beyond the Church-Turing barrier. MR2450722 (2009j:68065)
[102] Alfred Tarski, *A decision method for elementary algebra and geometry*, University of California Press, Berkeley and Los Angeles, Calif., 1951. 2nd ed. MR0044472 (13,423a)
[103] Seinosuke Toda, *PP is as hard as the polynomial-time hierarchy*, SIAM J. Comput. **20** (1991), no. 5, 865–877, DOI 10.1137/0220053. MR1115655 (93a:68047)
[104] Alan M. Turing, *Rounding-off errors in matrix processes*, Quart. J. Mech. Appl. Math. **1** (1948), 287–308. MR0028100 (10,405c)
[105] Leslie G. Valiant, *The complexity of computing the permanent*, Theoret. Comput. Sci. **8** (1979), no. 2, 189–201, DOI 10.1016/0304-3975(79)90044-6. MR526203 (80f:68054)
[106] Leslie G. Valiant, *Completeness classes in algebra*, Computing (Atlanta, Ga., 1979), ACM, New York, 1979, pp. 249–261. MR564634 (83e:68046)
[107] Klaus Weihrauch, *Computable analysis*, Texts in Theoretical Computer Science. An EATCS Series, Springer-Verlag, Berlin, 2000. An introduction. MR1795407 (2002b:03129)
[108] Henryk Woźniakowski, *Why does information-based complexity use the real number model?*, Theoret. Comput. Sci. **219** (1999), no. 1-2, 451–465, DOI 10.1016/S0304-3975(98)00300-4. Computability and complexity in analysis (Castle Dagstuhl, 1997). MR1694443 (2000c:65129)
[109] Yoshimi Yonezawa, *The Turing degrees for some computation model with the real parameter*, J. Math. Soc. Japan **60** (2008), no. 2, 311–324. MR2421978 (2009b:03117)
[110] Martin Ziegler, *(Short) survey of real hypercomputation*, Computation and logic in the real world, Lecture Notes in Comput. Sci., vol. 4497, Springer, Berlin, 2007, pp. 809–824, DOI 10.1007/978-3-540-73001-9_86. MR2646295
[111] Richard Zippel, *Probabilistic algorithms for sparse polynomials*, Symbolic and algebraic computation (EUROSAM '79, Internat. Sympos., Marseille, 1979), Lecture Notes in Comput. Sci., vol. 72, Springer, Berlin, 1979, pp. 216–226. MR575692 (81g:68061)

Computer Science Institute, BTU Cottbus, Platz der Deutschen Einheit 1, D-03046 Cottbus, Germany
E-mail address: baartse@tu-cottbus.de

Computer Science Institute, BTU Cottbus, Platz der Deutschen Einheit 1, D-03046 Cottbus, Germany
E-mail address: meer@informatik.tu-cottbus.de

Contemporary Mathematics
Volume **604**, 2013
http://dx.doi.org/10.1090/conm/604/12068

Polar, bipolar and copolar varieties: Real solving of algebraic varieties with intrinsic complexity

Bernd Bank, Marc Giusti, and Joos Heintz

Abstract. This survey covers a decade and a half of joint work with L. Lehmann, G. M. Mbakop, and L. M. Pardo. We address the problem of finding a smooth algebraic sample point for each connected component of a real algebraic variety, being only interested in components which are generically smooth locally complete intersections. The complexity of our algorithms is essentially polynomial in the degree of suitably defined generalized polar varieties and is therefore intrinsic to the problem under consideration.

1. Introduction

The modern concept of polar varieties was introduced in the 1930's by F. Severi ([**35**], [**34**]) and J. A. Todd ([**38**], [**37**]), while the intimately related notion of a reciprocal curve goes back to the work of J.-V. Poncelet in the period of 1813–1829. As pointed out by Severi and Todd, generic polar varieties have to be understood as being organized in certain equivalence classes which embody relevant geometric properties of the underlying algebraic variety S. This view led to the consideration of rational equivalence classes of the generic polar varieties. For historical details we refer to [**30**, **36**].

About 16 years ago (classic) polar varieties became our fundamental tool to tackle the task of real equation solving with a new view. We used them for the design of a pseudo-polynomial computer procedure with an intrinsic complexity bound which finds for a given complete intersection variety S with a smooth compact real trace $S_{\mathbb{R}}$ algebraic sample points for each connected component of $S_{\mathbb{R}}$ if there are such points ([**1**, **2**]).

Actually the geometric resolution of polar varieties led directly to a good pseudo–polynomial complexity, thanks to the algoritm `Kronecker` developed by the TERA-group [**16**, **18**, **19**, **24**].

Then we dropped successively the hypothesis on compactness of $S_{\mathbb{R}}$ (leading to dual polar varieties [**3**, **4**]) and eventually the hypothesis on smoothness of $S_{\mathbb{R}}$.

The presence of real singularities of $S_{\mathbb{R}}$ led us to the introduction of copolar incidence and bipolar varieties ([**6**, **7**]).

2010 *Mathematics Subject Classification.* Primary 68W30, 14P05, 14B05, 14B07, 68W10.
Digiteo DIM 2009–36HD "Magix", ANR-2010-BLAN-0109-04 "LEDA", MTM2010-16051. CONICET PIP 2461/01, UBACYT 20020100100945, PICT–2010–0525.

2. Notations and statement of results

2.1. Notations. Let $\mathbb{Q}$, $\mathbb{R}$ and $\mathbb{C}$ be the fields of rational, real and complex numbers, respectively, let $X := (X_1, \ldots, X_n)$ be a vector of indeterminates over $\mathbb{C}$ and let $F_1, \ldots, F_p$ be a regular sequence of polynomials in $\mathbb{Q}[X]$ defining a closed, $\mathbb{Q}$–definable subvariety S of the n–dimensional complex affine space $\mathbb{A}^n := \mathbb{C}^n$. Thus S is a non–empty equidimensional affine variety of dimension $n-p$, i.e., each irreducible component of S is of dimension $n-p$. Said otherwise, S is a closed subvariety of $\mathbb{A}^n$ of pure codimension p (in $\mathbb{A}^n$).

Let $\mathbb{A}^n_{\mathbb{R}} := \mathbb{R}^n$ be the n–dimensional real affine space. We denote by $S_{\mathbb{R}} := S \cap \mathbb{A}^n_{\mathbb{R}}$ the real trace of the complex variety S. Moreover, we denote by $\mathbb{P}^n$ the n–dimensional complex projective space and by $\mathbb{P}^n_{\mathbb{R}}$ its real counterpart. We shall use also the following notations:

$$\{F_1 = 0, \ldots, F_p = 0\} := S \text{ and } \{F_1 = 0, \ldots, F_p = 0\}_{\mathbb{R}} := S_{\mathbb{R}}.$$

We call the regular sequence $F_1, \ldots, F_p$ *reduced* if the ideal $(F_1, \ldots, F_p)$ generated in $\mathbb{Q}[X]$ is the ideal of definition of the affine variety S, i.e., if $(F_1, \ldots, F_p)$ is radical. We call $(F_1, \ldots, F_p)$ *strongly reduced* if for any index $1 \leq k \leq p$ the ideal $(F_1, \ldots, F_k)$ is radical. Thus, a strongly reduced regular sequence is always reduced.

A point x of $\mathbb{A}^n$ is called $(F_1, \ldots, F_p)$–*regular* if the Jacobian $J(F_1, \ldots, F_p) := \left[\frac{\partial F_j}{\partial X_k}\right]_{\substack{1 \leq j \leq p \\ 1 \leq k \leq n}}$ has maximal rank p at x. Observe, that for each *reduced* regular sequence $F_1, \ldots, F_p$ defining the variety S, the locus of $(F_1, \ldots, F_p)$–regular points of S is the same. In this case we call an $(F_1, \ldots, F_p)$–regular point of S simply *regular* (or *smooth*) or we say that S is regular (or smooth) at x. The set S_{reg} of regular points of S is called the *regular locus*, whereas $S_{sing} := S \setminus S_{reg}$ is called the *singular locus* of S. Remark that S_{reg} is a non–empty open and S_{sing} a proper closed subvariety of S. We say that a connected component C of $S_{\mathbb{R}}$ is *generically smooth* if C contains at least one smooth point.

We are going to use the expression *generic* according to Thom's terminology. A property that depends on parameters belonging to a certain configuration space Ω is called *generic* if there exists an Zariski open and dense subset of Ω, where the parameters are taken from, to insure the property.

We suppose now that there are given natural numbers d, L and an essentially division–free arithmetic circuit β in $\mathbb{Q}[X]$ with p output nodes such that the following conditions are satisfied.

- The degrees $\deg F_1, \ldots, \deg F_p$ of the polynomials $F_1, \ldots, F_p$ are bounded by d.
- The p output nodes of the arithmetic circuit β represent the polynomials $F_1, \ldots, F_p$ by evaluation.
- The size of the arithmetic circuit β is bounded by L.

For the terminology and basic facts concerning arithmetic circuits we refer to [**10**, **12**, **18**].

2.2. Statement of the results. For the sake of simplicity we suppose that the variables $X_1, \ldots, X_n$ are in generic position with respect to the variety S. Observe that we allow $S_{\mathbb{R}}$ to have singular points.

In this paper we comment a series of complexity results which concern the computational task to find in each, in the sense of Section 2.1 generically smooth, connected component of $S_{\mathbb{R}}$ at least one suitably encoded smooth point.

The most general result we are going to present is the following statement about the *existence* of an algorithm with certain properties (see Theorem 6.2 below).

For each $1 \leq i \leq n-p$ there exists a, by a sequence of algebraic computation trees (for this terminology we refer to [**10**]) realized, *non–uniform deterministic* or *uniform probabilistic* procedure Π_i over $\mathbb{Q}$ and an invariant δ_i satisfying the following specification.

- (*i*) The invariant δ_i is a positive integer depending on $F_1, \ldots, F_p$ and having asymptotic order not exceeding $(n\,d)^{O(n)}$. We call δ_i the *degree of the real interpretation of the equation system* $F_1 = 0, \ldots, F_p = 0$.
- (*ii*) The algorithm Π_i decides on input β whether the variety S contains a smooth real point and, if it is the case, produces for each generically smooth connected component of S a suitably encoded smooth real algebraic sample point.
- (*iii*) In order to achieve this goal, the algorithm Π_i performs on input β a computation in $\mathbb{Q}$ with $\binom{n}{p} L (n\,d)^{O(1)} \delta_i^2$ arithmetic operations (additions, subtractions, multiplications and divisions) and comparisons.

The worst case complexity of the procedure Π_i meets the already known extrinsic bound of $(n\,d)^{O(n)}$ for the elimination problem under consideration (compare the original papers [**8**, **11**, **20**, **25**–**27**, **31**, **32**] and the comprehensive book [**9**]).

The complexity of the procedure Π_i depends polynomially on the *extrinsic* parameters L, n, $\binom{n}{p}$ and d and on the degree δ_i of the real interpretation of the equation system $F_1 = 0, \ldots, F_p = 0$ which represents an *intrinsic* parameter measuring the input size of our computational task. In this sense we say that the procedure Π_i is of *intrinsic complexity*.

Since for fixed p the complexity $\binom{n}{p} L (n\,d)^{O(1)} \delta_i^2$ is polynomial in all its parameters, including the intrinsic parameter δ_i, we say that the procedure Π_i is *pseudo-polynomial*.

The lower complexity bounds of [**22**, **23**] for different elimination problems suggest that intrinsic complexity and pseudo-polynomiality constitute the best runtime behavior of Π_i which can be expected.

The above result is the consequence of a reduction to the case that $S_{\mathbb{R}}$ is smooth, where a similar, but somewhat simpler, complexity statement is true (see Theorem 4.1 below). For this reduction we considered in [**7**] a new type of geometrical objects, called *copolar incidence* and *bipolar* varieties.

First complexity results in this direction were obtained for the case that $S_{\mathbb{R}}$ is smooth and compact using *classic polar varieties* [**1**, **2**]. In order to treat the smooth unbounded case we introduced in [**3**, **4**] the concept of *dual polar varieties*.

In the present paper we put emphasis on the geometrical ideas which together with the `Kronecker` algorithm [**16**, **18**, **19**, **24**], that solves polynomial equation systems over the complex numbers, lead to our complexity statements.

3. Polar varieties

Let notations be as in Subsection 2.1. Let $F_1, \ldots, F_p \in \mathbb{Q}[X]$ be a reduced regular sequence defining a (non–empty) subvariety S of $\mathbb{A}^n$ of pure codimension p.

Let $1 \leq i \leq n-p$ and let $a := [a_{k,l}]_{\substack{1\leq k\leq n-p-i+1\\ 0\leq l\leq n}}$ be a complex $((n-p-i+1)\times(n+1)$–matrix and suppose that $[a_{k,l}]_{\substack{1\leq k\leq n-p-i+1\\ 1\leq l\leq n}}$ has maximal rank $n-p-i+1$. In case $(a_{1,0},\ldots,a_{n-p-i+1,0}) = 0$ we denote by $\underline{K}(a) := \underline{K}^{n-p-i}(a)$ and in case $(a_{1,0},\ldots,a_{n-p-i+1,0}) \neq 0$ by $\overline{K}(a) := \overline{K}^{n-p-i}(a)$ the $(n-p-i)$–dimensional linear subvarieties of the projective space $\mathbb{P}^n$ which for $1 \leq k \leq n-p-i+1$ are spanned by the points $(a_{k,0} : a_{k,1} : \cdots : a_{k,n})$.

The hyperplane at infinity of $\mathbb{P}^n$ is the set of points whose first coordinate is zero. It determines an embedding of $\mathbb{A}^n$ into $\mathbb{P}^n$. The classic and the dual ith polar varieties of S associated with the linear varieties $\underline{K}(a)$ and $\overline{K}(a)$, respectively, are geometrically defined as the Zariski closures of the set of points of S, where the tangent space of S is not transversal to the affine traces of $\underline{K}(a)$ and $\overline{K}(a)$, respectively.

Algebraically, the classic and the dual ith polar varieties of S associated with the linear varieties $\underline{K}(a)$ and $\overline{K}(a)$, respectively, can be described as the closures of the loci of the smooth points of S where all $(n-i+1)$–minors of the respective polynomial $((n-i+1)\times n)$–matrix

$$\begin{bmatrix} \frac{\partial F_1}{\partial X_1} & \cdots & \frac{\partial F_1}{\partial X_n} \\ \vdots & \vdots & \vdots \\ \frac{\partial F_p}{\partial X_1} & \cdots & \frac{\partial F_p}{\partial X_n} \\ a_{1,1} & \cdots & a_{1,n} \\ \vdots & \vdots & \vdots \\ a_{n-p-i+1,1} & \cdots & a_{n-p-i+1,n} \end{bmatrix}$$

and

$$\begin{bmatrix} \frac{\partial F_1}{\partial X_1} & \cdots & \frac{\partial F_1}{\partial X_n} \\ \vdots & \vdots & \vdots \\ \frac{\partial F_p}{\partial X_1} & \cdots & \frac{\partial F_p}{\partial X_n} \\ a_{1,1} - a_{1,0}X_1 & \cdots & a_{1,n} - a_{1,0}X_n \\ \vdots & \vdots & \vdots \\ a_{n-p-i+1,1} - a_{n-p-i+1,0}X_1 & \cdots & a_{n-p-i+1,n} - a_{n-p-i+1,0}X_n \end{bmatrix}$$

vanish.

If a is a real $((n-p-i+1)\times(n+1)$–matrix, we denote the real traces of the polar varieties $W_{\underline{K}(a)}(S)$ and $W_{\overline{K}(a)}(S)$ by

$$W_{\underline{K}(a)}(S_{\mathbb{R}}) := W_{\underline{K}^{n-p-i}(a)}(S_{\mathbb{R}}) := W_{\underline{K}(a)}(S) \cap \mathbb{A}^n_{\mathbb{R}}$$

and

$$W_{\overline{K}(a)}(S_{\mathbb{R}}) := W_{\overline{K}^{n-p-i}(a)}(S_{\mathbb{R}}) := W_{\overline{K}(a)}(S) \cap \mathbb{A}^n_{\mathbb{R}}$$

and call them the real polar varieties.

Observe that this definition of classic and dual polar varieties may be extended to the case that there is given a Zariski open subset O of $\mathbb{A}^n$ such that the equations $F_1 = 0,\ldots,F_p = 0$ intersect transversally at any of their common solutions in O and that S is now the locally closed subvariety of $\mathbb{A}^n$ given by

$$S := \{F_1 = 0,\ldots,F_p = 0\} \cap O,$$

which is supposed to be non-empty.

In Section 6 we shall need this extended definition of polar varieties in order to establish the notion of a bipolar variety of a given reduced complete intersection. For the moment let us suppose again that S is the closed subvariety of $\mathbb{A}^n$ defined by the reduced regular sequence $F_1, \ldots, F_p$.

In [**3**] and [**4**] we have introduced the notion of dual polar varieties of S (and $S_{\mathbb{R}}$) and motivated by geometric arguments the calculatory definition of these objects. Moreover, we have shown that, for a complex $((n-p-i+1)\times(n+1))$–matrix $a = [a_{k,l}]_{\substack{1\le k\le n-p-i+1\\ 0\le l\le n}}$ with $[a_{k,l}]_{\substack{1\le k\le n-p-i+1\\ 1\le l\le n}}$ *generic*, the polar varieties $W_{\underline{K}(a)}(S)$ and $W_{\overline{K}(a)}(S)$ are either empty or of pure codimension i in S. As mathematical facts, we have shown that $W_{\underline{K}(a)}(S)$ and $W_{\overline{K}(a)}(S)$ are normal and Cohen–Macaulay (but for $1 < p \le n$ not necessarily smooth) at any of their $(F_1, \ldots, F_p)$–regular points (see [**5**], Corollary 2 and Section 3.1). This motivates the consideration of the so–called *generic* polar varieties $W_{\underline{K}(a)}(S)$ and $W_{\overline{K}(a)}(S)$, associated with complex $((n-p-i+1)\times(n+1))$–matrices a which are generic in the above sense, as invariants of the complex variety S (independently of the given equation system $F_1 = 0, \ldots, F_p = 0$). However, when a generic $((n-p-i+1)\times(n+1))$–matrix a is real, we cannot consider $W_{\underline{K}(a)}(S_{\mathbb{R}})$ and $W_{\overline{K}(a)}(S_{\mathbb{R}})$ as invariants of the real variety $S_{\mathbb{R}}$, since for suitable real generic $((n-p-i+1)\times(n+1))$–matrices these polar varieties may turn out to be empty, whereas for other real generic matrices they may contain points (see [**5**], Theorem 1 and Corollary 2 and [**6**], Theorem 8 and Corollary 9).

In case that $S_{\mathbb{R}}$ is smooth and a is a real $((n-p-i+1)\times(n+1))$–matrix, the real dual polar variety $W_{\overline{K}(a)}(S_{\mathbb{R}})$ contains at least one point of each connected component of $S_{\mathbb{R}}$, whereas the classic (complex or real) polar varieties $W_{\underline{K}(a)}(S)$ and $W_{\underline{K}(a)}(S_{\mathbb{R}})$ may be empty (see [**3**] and [**4**], Proposition 2).

4. The smooth case

In this section we suppose that $S_{\mathbb{R}}$ is smooth. We choose a generic rational $((n-p)\times n)$–matrix $a := [a_{k,l}]_{\substack{1\le k\le n-p\\ 1\le l\le n}}$. For $1 \le i \le n-p$ we consider the $((n-p-i+1)\times(n+1))$–matrices $\underline{a}^{(i)} := \left[\underline{a}^{(i)}_{k,l}\right]_{\substack{1\le k\le n-p-i+1\\ 0\le l\le n}}$ and $\overline{a}^{(i)} := \left[\overline{a}^{(i)}_{k,l}\right]_{\substack{1\le k\le n-p-i+1\\ 0\le l\le n}}$ with $\underline{a}^{(i)}_{k,l} = \overline{a}^{(i)}_{k,l} = a^{(i)}_{k,l}$ for $1 \le k \le n-p-i+1$ and $1 \le l \le n$ and $\underline{a}^{(i)}_{1,0} = \cdots = \underline{a}^{(i)}_{n-p-i+1,0} = 0$ and $\overline{a}^{(i)}_{1,0} = \cdots = \overline{a}^{(i)}_{n-p-i+1,0} = 1$. Then

$$W_{\underline{K}(\underline{a}^{(n-p)})}(S) \subset \cdots \subset W_{\underline{K}(\underline{a}^{(1)})}(S) \subset S$$

and

$$W_{\overline{K}(\overline{a}^{(n-p)})}(S) \subset \cdots \subset W_{\overline{K}(\overline{a}^{(1)})}(S) \subset S$$

form two flags of generic classic and dual polar varieties of S.

If $S_{\mathbb{R}}$ is compact, then, for $1 \le i \le n-p$, the classic real polar variety $W_{\underline{K}(\underline{a}^{i})}(S_{\mathbb{R}})$ contains a point of each connected component of $S_{\mathbb{R}}$ and, in particular, $W_{\underline{K}(\underline{a}^{i})}(S)$ is of pure codimension i in S. The inclusion relations in the first flag are therefore strict and $W_{\underline{K}(\underline{a}^{(n-p)})}(S)$ is a zero–dimensional algebraic variety. Mutatis mutandis the same statement holds true for the second flag without the assumption that $S_{\mathbb{R}}$ is compact.

Let

$$\underline{\delta} := \max\{\max\{\deg\{F_1 = 0, \ldots, F_s = 0 | 1 \leq s \leq p\}\}, \max\{W_{\underline{K}(\underline{a}^i)} | 1 \leq i \leq n-p\}\}$$

and

$$\overline{\delta} := \max\{\max\{\deg\{F_1 = 0, \ldots, F_s = 0 | 1 \leq s \leq p\}\}, \max\{W_{\overline{K}(\overline{a}^i)} | 1 \leq i \leq n-p\}\}.$$

We call $\underline{\delta}$ and $\overline{\delta}$ the degrees of the real interpretation of the equation system

$$F_1 = 0, \ldots, F_p = 0.$$

Our most general complexity result for the case that $S_{\mathbb{R}}$ is smooth is the following.

THEOREM 4.1 ([**4**]). *Let n, p, d, δ, L be natural numbers. Let $X_1, \ldots, X_n$ and Z be indeterminates over $\mathbb{Q}$ and let $X := (X_1, \ldots, X_n)$.*

There exists an algebraic computation tree $\mathcal{N}$ over $\mathbb{Q}$, depending on certain parameters and having depth

$$L\,(n\,d)^{O(1)}\,\delta^2 = (n\,d)^{O(n)}$$

such that $\mathcal{N}$ satisfies the following condition:

Let $F_1, \ldots, F_p \in \mathbb{Q}[X]$ be polynomials of degree at most d and assume that $F_1, \ldots, F_p$ are given by an essentially division–free arithmetic circuit β in $\mathbb{Q}[X]$ of size L. Suppose that $F_1, \ldots, F_p$ form a strongly reduced regular sequence in $\mathbb{Q}[X]$, that $\{F_1 = 0, \ldots, F_p = 0\}_{\mathbb{R}}$ is empty or smooth and that $\overline{\delta} \leq \delta$ holds.

Then the algorithm represented by the algebraic computation tree $\mathcal{N}$ starts from the circuit β as input and decides whether the variety $\{F_1 = 0, \ldots, F_p = 0\}$ contains a real point. If this is the case, the algorithm produces a circuit representation of the coefficients of $n+1$ polynomials $P, G_1, \ldots, G_n \in \mathbb{Q}[Z]$ satisfying for $G := (G_1, \ldots, G_n)$ the following conditions:

- *P is monic and separable,*
- *$\deg G < \deg P \leq \overline{\delta}$,*
- *the zero–dimensional complex affine variety $\{G(z) \mid z \in \mathbb{A}^1, P(z) = 0\}$ contains at least one smooth real algebraic sample point for each connected component of $\{F_1 = 0, \ldots, F_p = 0\}_{\mathbb{R}}$.*

*In order to represent these sample points the algorithm returns an encoding "à la Thom" (see e.g. [**13**]) of the real zeros of the polynomial P.*

The parameters of $\mathcal{N}$ may be chosen randomly. This yields an uniform bounded error probabilistic algorithm which works in time $L\,(n\,d)^{O(1)}\delta^2 = (n\,d)^{O(n)}$ (counting arithmetic operations and comparisons in $\mathbb{Q}$ at unit costs).

In the mainly algebraic statement of Theorem 4.1 the relation to the real zeros of $F_1 = 0, \ldots, F_p = 0$ becomes established by the fact that the zero–dimensional variety $\{G(z) \,|\, z \in \mathbb{A}^1,\, P(z) = 0\}$ coincides with the generic dual polar variety $W_{\overline{K}(\overline{a}^{(n-p)})}(S)$ which contains a point of each connected component of $S_{\mathbb{R}}$.

The complexity result has an interpretation in the non–uniform deterministic as well as in the uniform probabilistic computational model. If we add the condition that $\{F_1 = 0, \ldots, F_p = 0\}_{\mathbb{R}}$ must be compact the statement of Theorem 4.1 holds true for $\overline{\delta}$ replaced by $\underline{\delta}$ ([**1**] and [**2**]).

Interpretation of Theorem 4.1 in the hypersurface case. We are going to comment Theorem 4.1 in the case of a smooth compact real hypersurface given by a regular polynomial equation. So let $p := 1$ and $F := F_1 \in \mathbb{Q}[X]$ be a squarefree polynomial of positive degree d and $S := \{F = 0\}$. For sake of simplicity we assume that the variables $X_1, \ldots, X_n$ are in generic position with respect to S and that $S_{\mathbb{R}}$ is non-empty, smooth and compact (see Section 2.1 for our notion of genericity).

Let F be given by an essentially division–free arithmetic circuit β in $\mathbb{Q}[X]$ of size L. The algebraic version of the Bertini–Sard Theorem (see [**14**]) and our assumptions imply that for each $1 \leq i < n$ the polynomials $F, \frac{\partial F}{\partial X_1}, \ldots, \frac{\partial F}{\partial X_i}$ form a strongly reduced regular sequence in the ring of fractions $\mathbb{Q}[X]_{\frac{\partial F}{\partial X_{i+1}}}$. It is not hard to see that the set

$$F = 0, \frac{\partial F}{\partial X_1} = 0, \ldots, \frac{\partial F}{\partial X_i} = 0, \frac{\partial F}{\partial X_{i+1}} \neq 0$$

is the locus of a generic classic polar variety of S where $\frac{\partial F}{\partial X_{i+1}}$ does not vanish. Therefore, the degree of the Zariski closure of this set is bounded by $\underline{\delta}$. For the same reason

$$\left\{F = 0, \frac{\partial F}{\partial X_1} = 0\,, \ldots, \frac{\partial F}{\partial X_{n-1}} = 0\,, \frac{\partial F}{\partial X_n} \neq 0\right\}$$

is a finite set that contains a point of each connected component of $S_{\mathbb{R}}$.

We are now in conditions to apply the `Kronecker` algorithm to the given circuit β in order to find the complex solutions of the system

$$F = 0, \frac{\partial F}{\partial X_1} = 0\,, \ldots, \frac{\partial F}{\partial X_{n-1}} = 0\,, \frac{\partial F}{\partial X_n} \neq 0$$

Between these solutions we filter out the real ones. We control the complexity of the algoritm computing for $1 \leq i < n$ at its $(i+1)$th step a lifting fiber [**19**] of the system $F = 0$, $\frac{\partial F}{\partial X_1} = 0\,, \ldots, \frac{\partial F}{\partial X_i} = 0\,, \frac{\partial F}{\partial X_{i+1}} \neq 0$. This can be done performing $L\,(n\,d)^{O(1)}\,\underline{\delta}^2 = (n\,d)^{O(n)}$ arithmetic operations in $\mathbb{Q}$.

5. Tools to handle the singular case

In this section we consider the algorithmic problem of finding for each geometrically smooth connected component of $S_{\mathbb{R}}$ an $(F_1, \ldots, F_p)$–regular point when $S_{\mathbb{R}}$ may be singular. In the next two sections we are going to prepare the geometrical tools for this task.

5.1. Two families of copolar incidence varieties. Let i be a natural numbers with $1 \leq i \leq n - p$ and let $B := [B_{k,l}]_{\substack{1 \leq k \leq n-i \\ 1 \leq l \leq n}}$, $\Lambda := [\Lambda_{r,s}]_{1 \leq r,s \leq p}$ and $\Theta := [\Theta_{k,r}]_{\substack{1 \leq k \leq n-i \\ 1 \leq r \leq p}}$ be matrices of indeterminates over $\mathbb{C}$.

We denote by $F := (F_1, \ldots, F_p)$ the sequence of the given polynomials and by $J(F) := \left[\frac{\partial F_s}{\partial X_l}\right]_{\substack{1 \leq s \leq p \\ 1 \leq l \leq n}}$ the Jacobian of F. Observe that the rank of $J(F)$ is generically p on any irreducible component of the complex variety $S := \{F_1 = \cdots = F_p = 0\}$. We write $J(F)^T$ for the transposed matrix of $J(F)$ and for any point $x \in \mathbb{A}^n$ we denote by $\operatorname{rk} J(F)(x)$ the rank of the complex matrix $J(F)(x)$.

We are now going to introduce two families of varieties which we shall call *copolar incidence varieties*. In order to define the first one we consider in the

ambient space

$$\mathbb{T}_i := \mathbb{A}^n \times \mathbb{A}^{(n-i)\times n} \times \mathbb{A}^{p\times p} \times \mathbb{A}^{(n-i)\times p}$$

the $\mathbb{Q}$–definable locally closed incidence variety

$$H_i := \{(x, b, \lambda, \vartheta) \in \mathbb{T}_i |\ x \in S,\ \mathrm{rk}\, b = n - i,\ \mathrm{rk}\, \vartheta = p,\ J(F)(x)^T\lambda + b^T\vartheta = 0\}.$$

Observe that the isomorphy class of H_i does not depend on the choice of the generators $F_1, \ldots, F_p$ of the vanishing ideal of S. The canonical projection of $\mathbb{T}_i$ onto $\mathbb{A}^n$ maps H_i into S.

Let us state three facts, namely Lemma 5.1 and Propositions 5.1 and 5.2 below, which will be fundamental in the sequel.

LEMMA 5.1. *Let $(x, b, \lambda, \vartheta)$ be a point of H_i. Then x belongs to S_{reg} and λ is a regular complex $(p \times p)$–matrix. Moreover, the canonical projection of $\mathbb{T}_i$ onto $\mathbb{A}^n$ maps H_i onto S_{reg} and $(H_i)_{\mathbb{R}}$ onto $(S_{\mathbb{R}})_{reg}$.*

PROPOSITION 5.1. *Let D_i be the closed subvariety of $\mathbb{T}_i$ defined by the conditions $rk\, B < n - i$ or $rk\, \Theta < p$. Then the polynomial equations*

$$F_1(X) = \cdots = F_p(X) = 0,$$

$$\sum_{1\le s\le p} \Lambda_{r,s} \frac{\partial F_s}{\partial X_l}(X) + \sum_{1\le k\le n-i} B_{k,l}\, \Theta_{k,r} = 0, \tag{5.1}$$

$$1 \le r \le p,\ 1 \le l \le n,$$

intersect transversally at any of their common solutions in $\mathbb{T}_i \setminus D_i$. Moreover, H_i is exactly the set of solutions of the polynomial equation system (5.1) outside of the locus D_i.

In particular, H_i is an equidimensional algebraic variety which is smooth and of dimension $n(n - i + 1) + p(p - i - 1) \ge 0$.

For algorithmic applications Proposition 5.1 contains too many open conditions, namely the conditions $\mathrm{rk}\, B = n - i$ and $\mathrm{rk}\, \Theta = p$. By means of a suitable specialization of the matrices B and Θ we are going to eliminate these open conditions. However, we have to take care that these specialization process does not exclude to many smooth points of the variety S. The following result, namely Proposition 5.2 below seems to represent a fair compromise. We shall need it later for the task of finding smooth points of S. For the formulation of this proposition we need some notations.

Let $\mathbf{B}$ and $\mathbf{\Theta}$ be the following matrices

$$\mathbf{B} := \begin{bmatrix} B_{1,n-i+1} & \cdots & B_{1,n} \\ \vdots & \ldots & \vdots \\ B_{p,n-i+1} & \cdots & B_{p,n} \end{bmatrix} \quad \text{and} \quad \mathbf{\Theta} := \begin{bmatrix} \Theta_{p+1,1} & \cdots & \Theta_{p+1,p} \\ \vdots & \ldots & \vdots \\ \Theta_{n-i,1} & \cdots & \Theta_{n-i,p,} \end{bmatrix}.$$

Let σ be a permutation of the set $\{1, \ldots, n\}$ (in symbols, $\sigma \in \mathrm{Sym}\,(n)$) and apply σ to the columns of the $((n - i) \times n)$–matrix

$$\begin{bmatrix} I_p & O_{p\times(n-p-i)} & \mathbf{B} \\ O_{(n-p-i)\times p} & I_{n-p-i} & O_{(n-p-i)\times i} \end{bmatrix}.$$

In this way we obtain a $((n-i)\times n)$–matrix which we denote by $\mathbf{B}_{i,\sigma}$. Furthermore, let

$$\mathbf{\Theta}_i := \begin{bmatrix} I_p \\ \mathbf{\Theta} \end{bmatrix} \qquad \text{and} \qquad \Delta_\sigma := \det \left[\frac{\partial F_s}{\partial X_{\sigma(r)}}\right]_{1\le s,r\le p}.$$

If we specialize in $\mathbf{B}_{i,\sigma}$ the submatrix $\mathbf{B}$ to $b \in \mathbb{A}^{p\times i}$ and in $\mathbf{\Theta}_i$ the submatix $\mathbf{\Theta}$ to $\vartheta \in \mathbb{A}^{(n-p-i)\times p}$ then the resulting complex matrices become denoted by $b_{i,\sigma}$ and ϑ_i, respectively.

We consider now in the ambient space

$$\mathbb{F}_i := \mathbb{A}^n \times \mathbb{A}^{p\times i} \times \mathbb{A}^{p\times p} \times \mathbb{A}^{(n-p-i)\times p}$$

a copolar incidence variety of more restricted type, namely

$$H_{i,\sigma} := \{(x, b, \lambda, \vartheta) \in \mathbb{F}_i \mid x \in S,\ J(F)(x)^T\lambda + b_{i,\sigma}^T\vartheta_i = 0\}.$$

Observe that $H_{i,\sigma}$ is a $\mathbb{Q}$–definable closed subvariety of $\mathbb{F}_i$ whose isomorphy class does not depend on the choice of the polynomials $F_1, \ldots, F_p$ of the vanishing ideal of S.

In the statement of the next result we make use of the Kronecker symbol $\delta_{r,l}$, $1 \le r, l \le p$ which is defined by $\delta_{r,l} := 0$ for $r \ne l$ and $\delta_{r,r} := 1$.

PROPOSITION 5.2. *Let notations and definitions be as before. For the sake of simplicity assume that σ is the identity permutation of Sym (n). Then the polynomial equations*

$$\begin{aligned} &F_1 = 0, \ldots, F_s = 0,\\ &\sum_{1\le s\le p} \Lambda_{r,s}\frac{\partial F_s}{\partial X_l}(X) + \delta_{r,l} = 0, \quad 1 \le r \le p,\ 1 \le l \le p,\\ &\sum_{1\le s\le p} \Lambda_{r,s}\frac{\partial F_s}{\partial X_l}(X) + \Theta_{l,r} = 0,\ 1 \le r \le p,\ p < l \le n-i,\\ &\sum_{1\le s\le p} \Lambda_{r,s}\frac{\partial F_s}{\partial X_l}(X) + B_{r,l} = 0,\ \ 1 \le r \le p,\ n-i < l \le n \end{aligned} \tag{5.2}$$

intersect transversally at any of their common solutions in $\mathbb{F}_i$. Moreover, $H_{i,\sigma}$ is exactly the set of solutions of the equation system (5.2). In particular, $H_{i,\sigma}$ is a closed equidimensional algebraic variety which is empty or smooth and of dimension $n-p$.

The image of $H_{i,\sigma}$ under the canonical projection of $\mathbb{F}_i$ onto $\mathbb{A}^n$ is the set of (smooth) points of S where Δ_σ does not vanish. For each real point $x \in S$ with $\Delta_\sigma(x) \ne 0$ there exists a real point $(x, b, \lambda, \vartheta)$ of $H_{i,\sigma}$.

In the sequel we shall refer to H_i and $H_{i,\sigma}$ as the *copolar incidence varieties* of $S := \{F_1 = \cdots = F_p = 0\}$ associated with the indices $1 \le i \le n-p$ and $\sigma \in \mathrm{Sym}\,(n)$.

The notion of a copolar incidence variety is inspired by the Room-Kempf canonical desingularization of determinantal varieties [**28**, **33**].

5.2. Copolar varieties. Let notations and assumptions be as in previous section and let $b \in \mathbb{A}^{(n-i)\times n}$ be a full rank matrix. We observe that the set

$$\widetilde{V}_b(S) := \{x \in S \mid \exists\,(\lambda, \vartheta) \in \mathbb{A}^{p\times p} \times \mathbb{A}^{(n-p)\times p} \ :\ \mathrm{rk}\ \vartheta = p \ \text{ and } \ (x, b, \lambda, \vartheta) \in H_i\}$$

does not depend on the choice of the generators $F_1, \ldots, F_p$ of the vanishing ideal of S. We call the Zariski closure in $\mathbb{A}^n$ of $\widetilde{V}_b(S)$ the *copolar variety* of S associated with the matrix b and we denote it by $V_b(S)$. Obviously we have $\widetilde{V}_b(S) = V_b(S) \cap S_{\mathrm{reg}}$.

Observe that a point x of S belongs to $\widetilde{V}_b(S)$ if and only if there exist p rows of the $((n-i)\times n)$–matrix b which generate the same affine linear space as the

rows of the Jacobian $J(F)$ at x. In case $p := 1$ and $F := F_1$ the copolar variety $V_b(\{F = 0\})$ coincides with the ith classic polar variety $W_{\underline{K}^{n-1-i}(\underline{b})}(\{F = 0\})$ of the complex hypersurface $\{F = 0\}$ (here $\underline{b}$ denotes the $((n-i) \times (n+1))$–matrix whose column number zero is a null-vector, whereas the columns numbered $1, \ldots, n$ are the corresponding columns of b).

PROPOSITION 5.3. *If $b \in \mathbb{A}^{(n-i)\times n}$ is a generic matrix, then the copolar variety $V_b(S)$ is empty or an equidimensional closed subvariety which is smooth at any point of $V_b(S) \cap S_{reg}$ and has (non-negative) dimension $n - (i+1)p$.*

Observe that for a generic $b \in \mathbb{A}^{(n-i)\times n}$ the emptiness or non-emptiness and in the latter case also the geometric degree of the copolar variety $V_b(S)$ is an invariant of the variety S. The incidence varieties H_i and $H_{i,\sigma}$ may be interpreted as suitable algebraic families of copolar varieties. In [**6**] we considered in the case $p := 1$ three analogous incidence varieties which turned out to be algebraic families of dual polar varieties. Here we have a similar situation since in the hypersurface case, namely in the case $p := 1$, the copolar varieties are classic polar varieties.

6. Bipolar varieties and real point finding in the singular case

In order to measure the complexity of the real point finding procedures of this paper for complete intersection varieties, we consider for $1 \leq p \leq n$, $1 \leq i \leq n-p$ and $\sigma \in \mathrm{Sym}\,(n)$ the generic dual polar varieties of the copolar incidence varieties H_i and $H_{i,\sigma}$. In analogy to the hypersurface case tackled in [**6**], we call them the *large* and the *small* bipolar varieties of S.

DEFINITION 6.1. The bipolar varieties $\mathfrak{B}_{(i,j)}$ and $\mathcal{B}_{(i,\sigma,j)}$ are defined as follows:

- for $1 \leq j \leq n(n-i+1) + p(p-i-1)$ let $\mathfrak{B}_{(i,j)}$ a $(n(n-i+1) + p(p-i-1) - j + 1)$th generic dual polar variety of H_i and,
- for $1 \leq j \leq n-p$ and $\sigma \in \mathrm{Sym}\,(n)$ let $\mathcal{B}_{(i,\sigma,j)}$ a $(n-p-j+1)$th generic dual polar variety of $H_{i,\sigma}$.

We call $\mathfrak{B}_{(i,j)}$ the *large* and $\mathcal{B}_{(i,\sigma,j)}$ the *small bipolar variety of* S, respectively.

The bipolar varieties $\mathfrak{B}_{(i,j)}$ and $\mathcal{B}_{(i,\sigma,j)}$ are well defined geometric objects which depend on the equation system $F_1(X) = \cdots F_p(X) = 0$, although the copolar incidence variety H_i is not closed (compare the definition of the notion of polar variety in Section 3, where we have taken care of this situation). Moreover, our notation is justified because we are only interested in invariants like the dimension and the degree of our bipolar varieties and these are independent of the particular (generic) choice of the linear projective varieties used to define the bipolar varieties.

Observe that the large bipolar varieties of S form a chain of equidimensional varieties

$$\overline{H_i} \supsetneq \mathfrak{B}_{(i,n(n-i+1)+p(p-i-1))} \supset \cdots \supset \mathfrak{B}_{(i,1)}.$$

The variety $\mathfrak{B}_{(i,1)}$ is empty or zero–dimensional. If $\mathfrak{B}_{(i,1)}$ is nonempty, then the chain is strictly decreasing.

Similarly the small bipolar varieties $\mathcal{B}_{(i,\sigma,j)}$ of S form also a chain of equidimensional varieties

$$\overline{H_{i,\sigma}} \supsetneq \mathcal{B}_{(i,\sigma,n-p)} \supset \cdots \supset \mathcal{B}_{(i,\sigma,1)}.$$

The variety $\mathcal{B}_{(i,\sigma,1)}$ is empty or zero–dimensional. If $\mathcal{B}_{(i,\sigma,1)}$ is nonempty, then the chain is strictly decreasing.

We denote by $\deg \mathfrak{B}_{(i,j)}$ and $\deg \mathcal{B}_{(i,\sigma,j)}$ the geometric degrees of the respective bipolar varieties in their ambient spaces $\mathbb{T}_i$ and $\mathbb{F}_i$ (see [**21**] for a definition and properties of the geometric degree of a subvariety of an affine space).

Observe that $\deg \mathfrak{B}_{(i,j)}$ remains invariant under linear transformations of the coordinates $X_1, \ldots, X_n$ by unitary complex matrices.

From [**6**], Lemma 1 and [**5**], Theorem 3 we deduce that for $1 \leq j \leq n-p$

$$\deg \mathcal{B}_{(i,\sigma,j)} \leq \deg \mathfrak{B}_{(i,n(n-i))+p(p-i)+j)} \tag{6.1}$$

holds.

Suppose that S contains a regular real point x. The there exists a permutation $\sigma \in \text{Sym } (n)$ with $\Delta_\sigma(x) \neq 0$. From Proposition 5.2 we deduce that $(H_{i,\sigma})_{\mathbb{R}}$ is nonempty. This implies that $H_{i,\sigma}$ is given by a reduced regular sequence of polynomials, namely the polynomials in the equation system (5.2). Moreover, the real variety $(H_{i,\sigma})_{\mathbb{R}}$ is smooth. Therefore we may apply [**3**, **4**], Proposition 2 to conclude that $(\mathcal{B}_{(i,\sigma,j)})_{\mathbb{R}}$ contains for each connected component of $(H_{i,\sigma})_{\mathbb{R}}$ at least one point. This implies

$$1 \leq \deg \mathcal{B}_{(i,\sigma,1)} \leq \deg \mathfrak{B}_{(i,n(n-i))+p(p-i)+1)}.$$

For $1 \leq r \leq p$, $1 \leq l \leq n$ and $\sigma \in \text{Sym } (n)$ we are going to analyze in the following closed subvarieties $S^{(i)}_{(r,l)}$ and $S^{(i,\sigma)}_{(r,l)}$ of the affine subspaces $\mathbb{T}_i$ and $\mathbb{F}_i$, respectively. For this purpose we consider the lexicographical order $<$ of the set of all pairs (r,l) with $1 \leq r \leq p$, $1 \leq l \leq n$.

Let $S^{(i)}_{(r,l)}$ be the Zariski closure of the locally closed subset of $\mathbb{T}_i$ defined by the conditions

$$\begin{gathered}
F_1(X) = \cdots = F_p(X) = 0 \\
\sum_{1 \leq s \leq p} \Lambda_{r',s} \frac{\partial F_s}{\partial X_{l'}} + \sum_{1 \leq k \leq n-i} B_{k,l'} \Theta_{k,r} = 0, \\
1 \leq r' \leq p, \;\; 1 \leq l' \leq n, \;\; (r',l') \leq (r,l) \;\text{ and} \\
\text{rk } B = n-i, \; \text{rk } \Theta = p \;\text{ and }\; \text{rk } J(F) = p.
\end{gathered} \tag{6.2}$$

Observe that the particular structure of the Jacobian of the equations of system (6.2) implies that the corresponding polynomials form a reduced regular sequence at any of their common zeros outside of the closed locus given by the conditions

$$\text{rk } B < n-i, \; \text{rk } \Theta < p \;\text{ or }\; \text{rk } J(F) < p.$$

Furthermore, let $S^{(i,\sigma)}_{(r,l)}$ be the locally closed subset of $\mathbb{F}_i$ defined by the conditions

$$\begin{gathered}
F_1(X) = \cdots = F_p(X) = 0, \\
\sum_{1 \leq s \leq p} \Lambda_{r',s} \frac{\partial F_s}{\partial X_{l'}} + \delta_{r',l'} = 0, \;\; 1 \leq r' \leq r, \quad 1 \leq l' \leq p, \qquad (r',l') \leq (r,l), \\
\sum_{1 \leq s \leq p} \Lambda_{r',s} \frac{\partial F_s}{\partial X_{l'}} + \Theta_{l',r'} = 0, \;\; 1 \leq r' \leq r, \;\; p < l' \leq n-i, \;\; (r',l') \leq (r,l), \\
\sum_{1 \leq s \leq p} \Lambda_{r',s} \frac{\partial F_s}{\partial X_{l'}} + B_{r',l'} = 0, \quad 1 \leq r' \leq r, \;\; n-i < l \leq n, \;\; (r',l') \leq (r,l) \\
\text{and} \quad \Delta_\sigma(X) \neq 0.
\end{gathered} \tag{6.3}$$

Again the particular structure of the Jacobian of the equations of system (6.3) implies that the corresponding polynomials form a reduced regular sequence at any of their common zeros outside of the closed locus given by the condition $\Delta_\sigma(X) = 0$.

In conclusion, the polynomials of the systems (5.1) and (5.2) form *strongly reduced* regular sequences at any of their common zeros outside of the corresponding closed loci.

For the next statement recall that the degree of the polynomials $F_1, \ldots, F_p$ is bounded by d (see Section 2.1).

PROPOSITION 6.1. *Let $1 \le r \le p$ and $1 \le l \le n$. Then we have the extrinsic estimate*

$$\deg S^{(i)}_{(r,l)} = (n\,d)^{O(n)}.$$

This bound relies on the multi-homogenous Bézout Inequality [**29**]. Simpler to prove is the following result.

PROPOSITION 6.2. *Let $1 \le r \le p$ and $1 \le l \le n$. Then we have the estimate*

$$\deg S^{(i,\sigma)}_{(r,l)} = (nd)^{O(n)}.$$

Let $1 \le i \le n - p$. We proceed now to state two extrinsic estimates for the degrees of the bipolar varieties $\mathfrak{B}_{(i,j)}$, $1 \le j \le n(n-i+1) + p(p-i+1)$, and $\mathcal{B}_{(i,\sigma,j)}, \sigma \in \mathrm{Sym}\,(n)$, $1 \le j \le n-p$.

PROPOSITION 6.3. *For $1 \le j \le n(n-i+1)+p(p-i-1)$ one has the extrinsic estimate $\deg \mathfrak{B}_{(i,j)} = (n\,d)^{O(n^2)}$. In particular, for $n(n-i)+p(p-i) < j \le n(n-i+1)+p(p-i-1)$ one has the estimate $\deg \mathfrak{B}_{(i,j)} = (nd)^{O(n)}$.*

PROPOSITION 6.4. *The extrinsic estimate $\deg \mathcal{B}_{(i,\sigma,j)} = (nd)^{O(n)}$ is valid for any $\sigma \in Sym\,(n)$ and $1 \le j \le n-p$.*

We associate now with $1 \le i \le n-p$, $\sigma \in \mathrm{Sym}\,(n)$ and the polynomial equation system $F_1 = \cdots = F_p = 0$ the following discrete parameters, namely

$$\begin{aligned}\delta_i := \max\{&\max\{\deg\{F_1 = 0 \cdots = F_s = 0\} \mid 1 \le s \le p\},\\ &\max\{\deg S^{(i)}_{(r,l)} \mid 1 \le r \le p,\ 1 \le l \le n\},\\ &\max\{\deg \mathfrak{B}_{i,n(n-i)+p(p-i)+j} \mid 1 \le j \le n-p\}\}\end{aligned}$$

and

$$\begin{aligned}\delta_{i,\sigma} := \max\{&\max\{\deg\{F_1 = 0 \cdots = F_s = 0\} \mid 1 \le s \le p\},\\ &\max\{\deg S^{(i,\sigma)}_{(r,l)} \mid 1 \le r \le p,\ 1 \le l \le n\},\\ &\max\{\deg \mathcal{B}_{(i,\sigma,j)} \mid 1 \le j \le n-p\}\}.\end{aligned}$$

Adapting the terminology of [**6**], Section 4.2 and taking into account that for $1 \le j \le n-p$ the degree of $\mathfrak{B}_{(i,n(n-i)+p(p-i)+j)}$ remains invariant under linear transformations of the coordinates $X_1, \ldots, X_n$ by unitary complex matrices, we call δ_i and $\delta_{i,\sigma}$ the *unitary-independent* and the *unitary-dependent degree of the real interpretation* of the equation system $F_1 = \cdots = F_p = 0$ associated with i and σ.

Observe that (6.1) and the Bézout Inequality imply

$$\delta_{i,\sigma} \le \delta_i \quad \text{for any} \quad \sigma \in \mathrm{Sym}\,(n). \tag{6.4}$$

From Propositions 6.2, 6.3 and 6.4 and the Bézout Inequality we deduce the following extrinsic estimates

$$\delta_i = (nd)^{O(n)} \tag{6.5}$$

and

$$\delta_{i,\sigma} = (n\,d)^{O(n)} \tag{6.6}$$

(compare for the case $p := 1$ the estimates (16) and (17) given in [**6**], Section 4.2).

For the rest of the paper we fix a family $\{\sigma_1, \ldots, \sigma_{\binom{n}{p}}\}$ of permutations from Sym (n) such that for any choice $1 \leq k_1 < \cdots < k_p \leq n$ there exists an index $1 \leq k \leq \binom{n}{p}$ with $\sigma_k(1) = k_1, \ldots, \sigma_k(p) = k_p$.

For each $1 \leq k \leq \binom{n}{p}$ the varieties $\{F_1 = 0, \ldots, F_r = 0\},\ 1 \leq r \leq p,\ S^{(i,\sigma_k)}_{(r,l)}, 1 \leq r \leq p,\ 1 \leq l \leq n$ and $\mathcal{B}_{(i,\sigma_k,j)}, 1 \leq j \leq n-p$ form a descending chain which is strict in case that there exists a real point x of S with $\Delta_{\sigma_k}(x) \neq 0$. We may now apply a suitably adapted version of the `Kronecker` algorithm (see [**7**], Section 5) to this chain in order to determine the points of the complex variety $\mathcal{B}_{(i,\sigma_k,1)}$ which is empty or zero-dimensional.

Observe that $\mathcal{B}_{(i,\sigma_k,1)}$ contains a point of each connected component of $S_{\mathbb{R}}$ where Δ_{σ_k} does not vanish identically. Therefore we obtain for each such component at least one point.

All this can be done using $L\,(n\,d)^{O(1)}\delta^2_{i,\sigma_k}$ arithmetic operations and comparisons in $\mathbb{Q}$. Repeating this procedure for each $1 \leq k \leq \binom{n}{p}$ and taking into account the estimate 6.4 and 6.6 we obtain the following result.

THEOREM 6.2. *Let n, p, d, i, δ, L be natural numbers with $d \geq 1$, $1 \leq i \leq n-p$. Let $X_1, \ldots, X_n$ and Z be indeterminates over $\mathbb{Q}$ and let $X := (X_1, \ldots, X_n)$.*

There exists an algebraic computation tree $\mathcal{N}$ over $\mathbb{Q}$, depending on certain parameters and having depth

$$\binom{n}{p} L\,(n\,d)^{O(1)}\,\delta^2 = (n\,d)^{O(n)}$$

such that $\mathcal{N}$ satisfies the following condition:

Let $F_1, \ldots, F_p \in \mathbb{Q}[X]$ be polynomials of degree at most d and assume that $F_1, \ldots, F_p$ are given by an essentially division–free arithmetic circuit β in $\mathbb{Q}[X]$ of size L. Suppose that $F_1, \ldots, F_p$ form a strongly reduced regular sequence in $\mathbb{Q}[X]$ and that $\delta_i \leq \delta$ holds.

Then the algorithm represented by the algebraic computation tree $\mathcal{N}$ starts from the circuit β as input and decides whether the variety $\{F_1 = 0, \ldots, F_p = 0\}$ contains a smooth real point. If this is the case, the algorithm produces a circuit representation of the coefficients of $n+1$ polynomials $P, G_1, \ldots, G_n \in \mathbb{Q}[Z]$ satisfying for $G := (G_1, \ldots, G_n)$ the following conditions:

- *P is monic and separable,*
- *$\deg G < \deg P \leq \delta$,*
- *the zero–dimensional complex affine variety $\{G(z) \mid z \in \mathbb{A}^1, P(z) = 0\}$ contains a smooth real algebraic sample point for each generically smooth connected component of $\{F_1 = 0, \ldots, F_p = 0\}_{\mathbb{R}}$.*

In order to represent these sample points the algorithm returns an encoding "à la Thom" of the real zeros of the polynomial P.

The parameters of $\mathcal{N}$ may be chosen randomly. This yields a uniform bounded error probabilistic algorithm which works in time $\binom{n}{p} L\,(n\,d)^{O(1)}\delta^2 = (n\,d)^{O(n)}$.

Interpretation of Theorem 6.2 in the hypersurface case. We are going to comment the geometric aspects of the method which leads to Theorem 6.2 in the case of a hypersurface. Let $p := 1$ and $F := F_1 \in \mathbb{Q}[X]$ be a squarefree polynomial of degree d and $S := \{F = 0\}$. Suppose that F is given by an essentially division–free arithmetic circuit β in $\mathbb{Q}[X]$ and, for sake of simplicity, that the variables $X_1, \ldots, X_n$ are in generic position with respect to S. For each generically smooth connected component of $S_{\mathbb{R}}$ we wish to find a representative point.

Let $1 \le i \le n-1$, $B := [B_{k,l}]_{\substack{1\le k\le n-i\\ 1\le l\le n}}$ be a matrix and $(B_{n-i+1},,\ldots,B_n)$ and $\Theta = (\Theta_1, \ldots, \Theta_{n-i})$ row vectors of indeterminates over $\mathbb{C}$. Furthermore let Λ be a single indeterminate over $\mathbb{C}$ and let $J(F) = (\frac{\partial F}{\partial X_1}, \ldots, \frac{\partial F}{\partial X_n})$ be the gradient (i.e., the Jacobian) of F. Let $\mathbb{T}_i := \mathbb{A}^n \times \mathbb{A}^{(n-i)\times n} \times \mathbb{A}^1 \times \mathbb{A}^{n-i}$ and $\mathbb{F}_i := \mathbb{A}^n \times \mathbb{A}^i \times \mathbb{A}^1 \times \mathbb{A}^{n-i-1}$. The equations

$$F(X) = 0,$$
$$\Lambda \frac{\partial F}{\partial X_l}(X) + \sum_{1\le k\le n-i} B_{k,l}\,\Theta_k = 0,\ 1 \le l \le n,$$

define outside of the locus given by the condition $\operatorname{rk} B < n-i$ or $\Theta = \mathbf{0}$ in $\mathbb{T}_i$ the copolar incidence variety H_i of S and intersect transversally at any point of H_i. In particular, H_i is smooth and of dimension $(n-i)(n+1)$.

Since the variables $X_1, \ldots, X_n$ are in generic position with respect to S, the partial derivative $\frac{\partial F}{\partial X_1}$ does not vanish identically on any generically smooth connected component of $S_{\mathbb{R}}$. It suffices therefore to consider $H_{i,\sigma}$ only for the identity permutation σ of $\{1, \ldots, n\}$.

The equations

$$F(X) = 0,$$
$$\Lambda \frac{\partial F}{\partial X_1}(X) + 1 = 0,$$
$$\Lambda \frac{\partial F}{\partial X_l}(X) + \Theta_l = 0,\ 2 \le l \le n-i,$$
$$\Lambda \frac{\partial F}{\partial X_l}(X) + B_l\Theta_1 = 0,\ n-i < l \le n$$

define in $\mathbb{F}_i$ the copolar incidence variety $H_{i,\sigma}$. In particular $H_{i,\sigma}$ is smooth and of dimension $n-1$. For δ_i and $\delta_{i,\sigma}$ we obtain the estimates $\delta_{i,\sigma} \le \delta_i = (n\,d)^{O(n)}$.

The algorithmic considerations are now similar as in the general complete intersection case and yield the statement of Theorem 6.2 for $p := 1$ and $\delta_i \le \delta$.

References

[1] B. Bank, M. Giusti, J. Heintz, and G. M. Mbakop, *Polar varieties, real equation solving, and data structures: the hypersurface case*, J. Complexity **13** (1997), no. 1, 5–27, DOI 10.1006/jcom.1997.0432. MR1449757 (98h:68123)

[2] B. Bank, M. Giusti, J. Heintz, and G. M. Mbakop, *Polar varieties and efficient real elimination*, Math. Z. **238** (2001), no. 1, 115–144, DOI 10.1007/PL00004896. MR1860738 (2002g:14084)

[3] Bernd Bank, Marc Giusti, Joos Heintz, and Luis M. Pardo, *Generalized polar varieties and an efficient real elimination procedure*, Kybernetika (Prague) **40** (2004), no. 5, 519–550. MR2120995 (2006e:14078)

[4] B. Bank, M. Giusti, J. Heintz, and L. M. Pardo, *Generalized polar varieties: geometry and algorithms*, J. Complexity **21** (2005), no. 4, 377–412, DOI 10.1016/j.jco.2004.10.001. MR2152713 (2006f:14068)

[5] Bernd Bank, Marc Giusti, Joos Heintz, Mohab Safey El Din, and Eric Schost, *On the geometry of polar varieties*, Appl. Algebra Engrg. Comm. Comput. **21** (2010), no. 1, 33–83, DOI 10.1007/s00200-009-0117-1. MR2585564 (2011c:68065)

[6] Bernd Bank, Marc Giusti, Joos Heintz, Lutz Lehmann, and Luis Miguel Pardo, *Algorithms of intrinsic complexity for point searching in compact real singular hypersurfaces*, Found. Comput. Math. **12** (2012), no. 1, 75–122, DOI 10.1007/s10208-011-9112-6. MR2886157

[7] B. Bank, M. Giusti, and J. Heintz, *Point searching in real singular complete intersection varieties - algorithms of intrinsic complexity*, under revision in Math. Comp. (2012)

[8] Saugata Basu, Richard Pollack, and Marie-Françoise Roy, *On the combinatorial and algebraic complexity of quantifier elimination*, J. ACM **43** (1996), no. 6, 1002–1045, DOI 10.1145/235809.235813. MR1434910 (98c:03077)

[9] Saugata Basu, Richard Pollack, and Marie-Françoise Roy, *Algorithms in real algebraic geometry*, 2nd ed., Algorithms and Computation in Mathematics, vol. 10, Springer-Verlag, Berlin, 2006. MR2248869 (2007b:14125)

[10] Peter Bürgisser, Michael Clausen, and M. Amin Shokrollahi, *Algebraic complexity theory*, Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], vol. 315, Springer-Verlag, Berlin, 1997. With the collaboration of Thomas Lickteig. MR1440179 (99c:68002)

[11] J. F. Canny, *Some algebraic and geometric computations in PSPACE,* ACM Symposium on Theory of Computing (STOC) (1988), 460-467.

[12] D. Castro, M. Giusti, J. Heintz, G. Matera, and L. M. Pardo, *The hardness of polynomial equation solving*, Found. Comput. Math. **3** (2003), no. 4, 347–420, DOI 10.1007/s10208-002-0065-7. MR2009683 (2004k:68056)

[13] M. Coste and M.-F. Roy, *Thom's lemma, the coding of real algebraic numbers and the computation of the topology of semi-algebraic sets*, J. Symbolic Comput. **5** (1988), no. 1-2, 121–129, DOI 10.1016/S0747-7171(88)80008-7. MR949115 (89g:12002)

[14] M. Demazure, *Catastrophes et bifurcations,* Ellipses, Paris 1989.

[15] M. Giusti, J. Heintz, J. E. Morais, and L. M. Pardo, *When polynomial equation systems can be "solved" fast?*, (Paris, 1995), Lecture Notes in Comput. Sci., vol. 948, Springer, Berlin, 1995, pp. 205–231, DOI 10.1007/3-540-60114-7_16. MR1448166 (98a:68106)

[16] M. Giusti, J. Heintz, K. Hägele, J. E. Morais, L. M. Pardo, and J. L. Montaña, *Lower bounds for Diophantine approximations*, J. Pure Appl. Algebra **117/118** (1997), 277–317, DOI 10.1016/S0022-4049(97)00015-7. Algorithms for algebra (Eindhoven, 1996). MR1457843 (99d:68106)

[17] M. Giusti, J. Heintz, J. E. Morais, J. Morgenstern, and L. M. Pardo, *Straight-line programs in geometric elimination theory*, J. Pure Appl. Algebra **124** (1998), no. 1-3, 101–146, DOI 10.1016/S0022-4049(96)00099-0. MR1600277 (99d:68128)

[18] Marc Giusti and Joos Heintz, *Kronecker's smart, little black boxes*, Foundations of computational mathematics (Oxford, 1999), London Math. Soc. Lecture Note Ser., vol. 284, Cambridge Univ. Press, Cambridge, 2001, pp. 69–104. MR1836615 (2002e:65075)

[19] Marc Giusti, Grégoire Lecerf, and Bruno Salvy, *A Gröbner free alternative for polynomial system solving*, J. Complexity **17** (2001), no. 1, 154–211, DOI 10.1006/jcom.2000.0571. MR1817612 (2002b:68123)

[20] D. Yu. Grigor′ev and N. N. Vorobjov Jr., *Solving systems of polynomial inequalities in subexponential time*, J. Symbolic Comput. **5** (1988), no. 1-2, 37–64, DOI 10.1016/S0747-7171(88)80005-1. MR949112 (89h:13001)

[21] Joos Heintz, *Definability and fast quantifier elimination in algebraically closed fields*, Theoret. Comput. Sci. **24** (1983), no. 3, 239–277, DOI 10.1016/0304-3975(83)90002-6. MR716823 (85a:68062)

[22] J. Heintz, B. Kuipers, A. Rojas Paredes, *Software Engineering and complexity in effective algebraic geometry,* J. Complexity 28 (2012), to appear.

[23] J. Heinz, B. Kuipers, A. Rojas Paredes, *On the intrinsic complexity of elimination problems in effective algebraic geometry,* this volume.
[24] Joos Heintz, Guillermo Matera, and Ariel Waissbein, *On the time-space complexity of geometric elimination procedures*, Appl. Algebra Engrg. Comm. Comput. **11** (2001), no. 4, 239–296, DOI 10.1007/s002000000046. MR1818975 (2002c:68108)
[25] J. Heintz, M.-F. Roy, and P. Solernó, *On the complexity of semialgebraic sets,* in IFIP Information Processing 89 (G. X. Ritter , ed.), Elsevier, 1989, pp. 293-298.
[26] Joos Heintz, Marie-Françoise Roy, and Pablo Solernó, *Complexité du principe de Tarski-Seidenberg*, C. R. Acad. Sci. Paris Sér. I Math. **309** (1989), no. 13, 825–830 (French, with English summary). MR1055203 (92c:12012)
[27] Joos Heintz, Marie-Françoise Roy, and Pablo Solernó, *Sur la complexité du principe de Tarski-Seidenberg*, Bull. Soc. Math. France **118** (1990), no. 1, 101–126 (French, with English summary). MR1077090 (92g:03047)
[28] George Kempf, *On the geometry of a theorem of Riemann*, Ann. of Math. (2) **98** (1973), 178–185. MR0349687 (50 #2180)
[29] Alexander Morgan and Andrew Sommese, *A homotopy for solving general polynomial systems that respects m-homogeneous structures*, Appl. Math. Comput. **24** (1987), no. 2, 101–113, DOI 10.1016/0096-3003(87)90063-4. MR914806 (88j:65110)
[30] Ragni Piene, *Polar classes of singular varieties*, Ann. Sci. École Norm. Sup. (4) **11** (1978), no. 2, 247–276. MR510551 (80j:14051)
[31] J. Renegar, *A faster PSPACE algorithm for the existential theory of the reals,* in Proc. 29th Annual IEEE Symposium on the Foundation of Computer Science, 1988, pp. 291-295.
[32] James Renegar, *On the computational complexity and geometry of the first-order theory of the reals. I. Introduction. Preliminaries. The geometry of semi-algebraic sets. The decision problem for the existential theory of the reals*, J. Symbolic Comput. **13** (1992), no. 3, 255–299, DOI 10.1016/S0747-7171(10)80003-3. MR1156882 (93h:03011a)
[33] T. G. Room, *The geometry of determinantal loci,* Cambridge Univ. Press (1938).
[34] F. Severi, *Sulle intersezioni delle varieta algebriche e sopra i loro caratteri e singolarita proiettive,* Torino Mem. (2) 52 (1903), 61-118.
[35] Francesco di Severi, *La serie canonica e la teoria delle serie principali di gruppi di punti sopra una superficie algebrica*, Comment. Math. Helv. **4** (1932), no. 1, 268–326, DOI 10.1007/BF01202721 (Italian). MR1509461
[36] Bernard Teissier, *Quelques points de l'histoire des variétés polaires, de Poncelet à nos jours*, Séminaire d'Analyse, 1987–1988 (Clermont-Ferrand, 1987), Univ. Clermont-Ferrand II, Clermont, 1990, pp. Exp. No. 4, 12 (French). MR1088966 (91m:14001)
[37] J. A. Todd, *The Geometrical Invariants of Algebraic Loci*, Proc. London Math. Soc. **S2-43**, no. 2, 127, DOI 10.1112/plms/s2-43.2.127. MR1575589
[38] J. A. Todd, *The Arithmetical Invariants of Algebraic Loci*, Proc. London Math. Soc. **S2-43**, no. 3, 190, DOI 10.1112/plms/s2-43.3.190. MR1575915

Humboldt-Universität zu Berlin, Institut für Mathematik, 10099 Berlin, Germany
E-mail address: bank@math.hu-berlin.de

CNRS, École Polytechnique, Lab. LIX, 91228 Palaiseau Cedex, France
E-mail address: marc.giusti@polytechnique.fr

Departamento de Computación, Universidad de Buenos Aires and CONICET, Ciudad Univ., Pab.I, 1428 Buenos Aires, Argentina, and Departamento de Matemáticas, Estadística y Computación, Facultad de Ciencias, Universidad de Cantabria, Avda. de los Castros, s/n, E-39005 Santander, Spain
E-mail address: joos@dc.uba.ar

Contemporary Mathematics
Volume **604**, 2013
http://dx.doi.org/10.1090/conm/604/12069

The complexity and geometry of numerically solving polynomial systems.

Carlos Beltrán and Michael Shub

This paper is dedicated to the memory of our beloved friend and colleague Jean Pierre Dedieu.

Abstract. These pages contain a short overview on the state of the art of efficient numerical analysis methods that solve systems of multivariate polynomial equations. We focus on the work of Steve Smale who initiated this research framework, and on the collaboration between Stephen Smale and Michael Shub, which set the foundations of this approach to polynomial system–solving, culminating in the more recent advances of Carlos Beltrán, Luis Miguel Pardo, Peter Bürgisser and Felipe Cucker.

1. The modern numerical approach to polynomial system solving

In this paper we survey some of the recent advances in the solution of polynomial systems. Such a classical topic has been studied by hundreds of authors from many different perspectives. We do not intend to make a complete historical description of all the advances achieved during the last century or two, but rather to describe in some detail the state of the art of what we think is the most successful (both from practical and theoretical perspectives) approach. Homotopy methods are used to solve polynomial systems in real life applications all around the world.

The key ingredient of homotopy methods is a one–line thought: given a goal system to be solved, choose some other system (similar in form, say with the same degree and number of variables) with a known solution ζ_0, and move this new system to the goal system, tracking how the known solution moves to a solution of the goal. Before stating any notation, we can explain briefly why this process is reasonable: if for every $t \in [0,1]$ we have a system of equations f_t (f_0 is the system with a known solution, f_1 is the one we want to solve), then we are looking for a path ζ_t, $t \in [0,1]$, such that $f_t(\zeta_t) = 0$. As long as the derivative $df_t(\zeta_t)$ is invertible for all t we can continue the solution from f_0 to f_1, by the implicit function theorem. Now we have various methods to accomplish this continuation. We can slowly increment t and use iterative numerical solution methods such as Newton's method to track the solution or we may differentiate the expresion $f_t(\zeta_t) = 0$ and solve for

2010 *Mathematics Subject Classification.* Primary 65H10, 14Q20, 68Q25.

The first author was partially Supported by MTM2010-16051, Spanish Ministry of Science.

The second author was partially supported by a CONICET grant PIP0801 2010–2012 and by ANPCyT PICT 2010–00681.

We thank an anonymous referee for his detailed reading of the manuscript.

$d/(dt)(\zeta_t) = \dot{\zeta}_t$. Then, we can write our problem as an initial value problem:

$$\begin{cases} \dot{\zeta}_t = -Df_t(\zeta_t)^{-1} f_t(\zeta_t) \\ \zeta_0 \text{ known} \end{cases} \tag{1.1}$$

Systems of ODEs have been much studied and hence this is an interesting idea: we have reduced our original problem to a very much studied one. One can just plug in a standard numerical ODE solver such as backward Euler or a version of Runge–Kutta's method. Even then, in practice, it is desirable to, from time to time, perform some steps of Newton's method $z \to x - Df_t(x)^{-1} f_t(x)$ to our approximation z_t of ζ_t, to get closer to the path (f_t, ζ_t). After some testing and adjustment of parameters, this naïve idea can be made to work with impressive practical performance and there are several software packages which attain spectacular results (solving systems with many variables and high degree) in a surprisingly short running time, see for example [**7**, **41**, **42**, **63**]

From a mathematical point of view, there are several things in the process we have just described that need to be analyzed: will there actually exist a path ζ_t (maybe it is only defined for, say, $t < 1/2$)? what is the expected complexity of the process (in particular, can we expect average polynomial running time in some sense)? what "simple system with a known solution" should we start at? how should we join f_0 and f_1, that is what should be the path f_t?

In the last few decades a lot of progress has been made in studying these questions. This progress is the topic of this paper.

2. A technical description of the problem

We will center our attention in Smale's 17–th problem, which we recall now.

Problem 2.1. *Can a zero of n complex polynomial equations in n unknowns be* **found approximately**, **on the average, in polynomial time** *with a* **uniform algorithm***?*

We have written in bold the technical terms that need to be clarified.

In order to understand the details of the problem and the solution suggested in Section 1, we need to describe some important concepts and notation in detail. Maybe the first one is our understanding of what a "solution" is: clearly, one cannot expect solutions of polynomial systems to be rational numbers, so one can only search for "quasi–solutions" in some sense. There are several definitions of such a thing, the most stable being the following one (introduced in [**57**], see also [**23**, **39**, **40**]):

Definition 2.2. Given a polynomial system, understood as a mapping $f : \mathbb{C}^n \to \mathbb{C}^n$, an approximate zero of f with associated (exact) zero ζ is a vector $z_0 \in \mathbb{C}^n$ such that

$$\|z_k - \zeta\| \leq \frac{1}{2^{2^k - 1}} \|z_0 - \zeta\|, \quad k \geq 0,$$

where z_k is the result of applying k times Newton's operator $z \mapsto z - Df(z)^{-1} f(z)$ (note that the definition of approximate zero implicitly assumes that z_k is defined for all $k \geq 0$.)

The power of this definition is that, as we will see below, given any polynomial system f and any exact zero $\zeta \in \mathbb{C}^n$, approximate zeros of f with associated zero ζ exist whenever $Df(\zeta)$ is an invertible matrix.

Recall that our first goal is to transform the problem of polynomial system solving into an implicit function problem or an ODE system like that of (1.1). There exist two principal reasons why the solution of such a system can fail to be defined for all $t > 0$: that the function defining the derivative is not everywhere defined (this corresponds naturally to $Df_t(\zeta_t)$ not being invertible), and that the solution escapes to infinity. The first problem seems to be more delicate and difficult to solve, but the second one is actually very easily dealt with: we just need to define our ODE in a compact manifold, instead of just in $\mathbb{C}^n$. The most similar compact manifold to $\mathbb{C}^n$ is $\mathbb{P}(\mathbb{C}^{n+1})$, and the way to take the problem into $\mathbb{P}(\mathbb{C}^{n+1})$ is just homogenizing the equations.

DEFINITION 2.3. Let $f : \mathbb{C}^n \to \mathbb{C}^n$ be a polynomial system, that is $f = (f_1, \ldots, f_n)$ where $f_i : \mathbb{C}^n \to \mathbb{C}$ is a polynomial of degree some d_i,

$$f(x_1, \ldots, x_n) = \sum_{\alpha_1 + \cdots + \alpha_n \leq d_i} a^{(i)}_{\alpha_1, \ldots, \alpha_n} x_1^{\alpha_1} \cdots x_n^{\alpha_n}.$$

The homogeneous counterpart of f is $h : \mathbb{C}^{n+1} \to \mathbb{C}^n$ defined by $h = (h_1, \ldots, h_n)$ where

$$h(x_0, x_1, \ldots, x_n) = \sum_{\alpha_1 + \cdots + \alpha_n \leq d_i} a^{(i)}_{\alpha_1, \ldots, \alpha_n} x_0^{d_i - \sum_{i=1}^n \alpha_i} x_1^{\alpha_1} \cdots x_n^{\alpha_n}.$$

We will talk about such a system h simply as a homogeneous system.

Note that if ζ is a zero of f then $(1, \zeta)$ is a zero of the homogeneous counterpart h of f. Reciprocally, if $\zeta = (\zeta_0, \zeta_1, \ldots, \zeta_n)$ is a zero of h and if $\zeta_0 \neq 0$, then $(\zeta_1/\zeta_0, \ldots, \zeta_n/\zeta_0)$ is a zero of f. Thus, the zeros of f and h are in correspondence and we can think of solving h and then recovering the zeros of f (this is not a completely obvious process when we only have approximate zeros, see [**15**].) Moreover, it is clear that for any complex number $\lambda \in \mathbb{C}$ and for $x \in \mathbb{C}^{n+1}$ we have

$$h(\lambda x) = Diag(\lambda^{d_1}, \ldots, \lambda^{d_n}) h(x),$$

and thus the zeros of h lie naturally in the projective space $\mathbb{P}(\mathbb{C}^{n+1})$.

As we will be working with homogeneous systems and projective zeros, we need a definition of approximate zero in the spirit of Definition 2.2 which is amenable to a projective setting. The following one, which uses the projective version [**50**] of Newton's operator, makes the work. Here and throughout the paper, given a matrix or vector A, by A^* we mean the complex conjugate transpose of A, and by $d_R(x, y)$ we mean the Riemannian distance from x to y, where x and y are elements in some Riemannian manifold.

DEFINITION 2.4. Given a homogeneous system h, an approximate zero of h with associated (exact) zero $\zeta \in \mathbb{P}(\mathbb{C}^{n+1})$ is a vector $z_0 \in \mathbb{P}(\mathbb{C}^{n+1})$ such that

$$d_R(z_k, \zeta) \leq \frac{1}{2^{2^k - 1}} d_R(z_0, \zeta), \quad k \geq 0,$$

where z_k is the result of applying k times the projective Newton operator $z \mapsto z - Dh(z) \mid_{z^\perp}^{-1} h(z)$ (again, the definition of approximate zero implicitly assumes that z_k is defined for all $k \geq 0$.) Here, by $Df(z) \mid_{z^\perp}$ we mean the restriction of the derivative of h at z, to the (complex) orthogonal subspace $z^\perp = \{y \in \mathbb{C}^{n+1} : y^* z = 0\}$.

It is a simple exercise to verify that (projective) Newton's method is well defined, that is the point it defines in projective space does not depend on the representative $z \in \mathbb{C}^{n+1}$ chosen for a point in projective space.

A (projective) approximate zero of h is thus a projective point such that the successive iterates of the projective Newton operator quickly approach an exact zero of h. Thus finding an approximate zero is an excellent output of a numerical zero–finding algorithm to solve h.

Because we are going to consider paths of systems $\{h_t\}_{t\in[a,b]}$, it is convenient to fix a framework where one can define these nicely. To this end, we consider the vector space of homogeneous polynomials of fixed degree $s \geq 1$:

$$\mathcal{H}_s = \{h \in \mathbb{C}[x_0, \ldots, x_n] : h \text{ is homogeneous of degree } s\}.$$

It is convenient to consider an Hermitian product (and the associated metric) on $\mathcal{H}_s$. A desirable property of such a metric is the unitary invariance, namely, we would like to have an Hermitian product such that

$$\langle h, g\rangle_{\mathcal{H}_s} = \langle h \circ U, g \circ U\rangle_{\mathcal{H}_s}, \quad \forall\, U \in \mathcal{U}_{n+1},$$

where $\mathcal{U}_{n+1}$ is the group of unitary matrices of size $n+1$. Such property was studied in detail in [**52**]. It turns out that there exists a unique (up to scalar multiplication) Hermitian product that satisfies it, the one defined as follows:

$$\Big\langle \sum_{\alpha_0+\cdots+\alpha_n=s} a_{\alpha_0,\ldots,\alpha_n} x_0^{\alpha_0}\cdots x_n^{\alpha_n}, \sum_{\alpha_0+\cdots+\alpha_n=s} b_{\alpha_0,\ldots,\alpha_n} x_0^{\alpha_0}\cdots x_n^{\alpha_n}\Big\rangle_{\mathcal{H}_s} =$$

$$\sum_{\alpha_0+\cdots+\alpha_n=s} \frac{\alpha_0!\cdots\alpha_n!}{s!} a_{\alpha_0,\ldots,\alpha_n}\overline{b_{\alpha_0,\ldots,\alpha_n}},$$

where $\bar{\cdot}$ just means complex conjugation. Note that this is just a weighted version of the standard complex Hermitian product in complex affine space.

Then, given a list of degrees $(d) = (d_1, \ldots, d_n)$, we consider the vector space

$$\mathcal{H}_{(d)} = \prod_{i=1}^{n} \mathcal{H}_{d_i}.$$

Note that an element h of $\mathcal{H}_{(d)}$ can be seen both as a mapping $h : \mathbb{C}^{n+1} \to \mathbb{C}^n$ or as a polynomial system, and can be identified by the list of coefficients of $h_1, \ldots, h_n$. We denote by $\mathbb{P}(\mathcal{H}_{(d)})$ the projective space associated to $\mathcal{H}_{(d)}$, by N the complex dimension of $\mathbb{P}(\mathcal{H}_{(d)})$ (so the dimension of $\mathcal{H}_{(d)}$ is $N+1$) and we consider the following Hermitian structure in $\mathcal{H}_{(d)}$:

$$\langle h, g\rangle = \sum_{i=1}^{n} \langle h_i, g_i\rangle_{H_{d_i}}, \quad \|h\| = \langle h, h\rangle^{1/2}.$$

This Hermitian product (and the associate Hermitian structure and metric) is also called the Bombieri–Weyl or the Kostlan product (structure, metric). As usual, this Hermitian product in $\mathcal{H}_{(d)}$ defines an associated Riemannian structure given by the real part of $\langle\cdot,\cdot\rangle$. We can thus consider integrals of functions defined on $\mathcal{H}_{(d)}$.

We denote by $\mathbb{S}$ the unit sphere in $\mathcal{H}_{(d)}$, and we endow $\mathbb{S}$ with the inherited Riemannian structure from that of $\mathcal{H}_{(d)}$. Then, $\mathbb{P}(\mathcal{H}_{(d)})$ has a natural Riemannian structure, the unique one making the projection $\mathbb{S} \to \mathbb{P}(\mathcal{H}_{(d)})$ a Riemannian submersion. That is the derivative of the projection restricted to the normal to the

fibers is an isometry. We can thus also consider integrals of functions defined in $\mathbb{S}$ or $\mathbb{P}(\mathcal{H}_{(d)})$. We can now talk about probabilities in $\mathbb{S}$ or $\mathbb{P}(\mathcal{H}_{(d)})$: given a measurable (nonnegative or integrable) mapping X defined in $\mathbb{S}$ or $\mathbb{P}(\mathcal{H}_{(d)})$, we can consider its expected value:

$$\mathrm{E}_{\mathbb{S}}(X) = \frac{1}{\nu(\mathbb{S})}\int_{\mathbb{S}} X(h)\,dh \quad \text{or} \quad \mathrm{E}_{\mathbb{P}(\mathcal{H}_{(d)})}(X) = \frac{1}{\nu(\mathbb{P}(\mathcal{H}_{(d)}))}\int_{\mathbb{P}(\mathcal{H}_{(d)})} X(h)\,dh,$$

where we simply denote by $\nu(E)$ the volume of a Riemannian manifold E. Similarly, one can talk about probabilities in $\mathcal{H}_{(d)}$ according to the standard Gaussian distribution compatible with $\langle\cdot,\cdot\rangle$: given a measurable (nonnegative or integrable) mapping X defined in $\mathcal{H}_{(d)}$, its expected value is:

$$\mathrm{E}_{\mathcal{H}_{(d)}}(X) = \frac{1}{(2\pi)^{N+1}}\int_{\mathcal{H}_{(d)}} X(h)e^{-\|h\|^2/2}\,dh.$$

We can now come back to Problem 2.1 and see what do each of the terms in that problem mean: Smale himself points out that one can just solve homogeneous systems (as suggested above). We still have a few terms to clarify:

- found approximately. This means finding an approximate zero in the sense of Definition 2.4.
- on the average, in polynomial time. This now means that, if $X(h)$ is the time needed by the algorithm to output an approximate zero of the input system h, then the expected value of X is a quantity polynomial in the input size, that is polynomial in N. The number of variables, n, and the maximum of the degrees, d, are smaller than N, and hence one attempts to get a bound on the expected value of X, as a polynomial in n, d, N.
- uniform algorithm. Smale demands an algorithm in the Blum–Shub–Smale model [**20**, **21**], that is exact operations and comparisons between real numbers are assumed. This assumption departs from the actual performance of our computers, but it is close enough to be translated to performance in many situations. Uniform means that the same algorithm works for all (d) and n.

3. Geometry and condition number

We can now set up a geometric framework for homotopy methods. Consider the following set, usually called the solution variety:

$$\mathcal{V} = \{(h,\zeta) \in \mathbb{P}(\mathcal{H}_{(d)}) \times \mathbb{P}(\mathbb{C}^{n+1}) : h(\zeta) = 0\}. \tag{3.1}$$

This set is actually a smooth complex submanifold (as well as a complex algebraic subvariety) of $\mathbb{P}(\mathcal{H}_{(d)}) \times \mathbb{P}(\mathbb{C}^{n+1})$, see [**20**], and is clearly compact. It will be useful to consider the following diagram.

$$\begin{array}{ccc} & \mathcal{V} & \\ \pi_1 \swarrow & & \searrow \pi_2 \\ \mathbb{P}(\mathcal{H}_{(d)}) & & \mathbb{P}(\mathbb{C}^{n+1}) \end{array} \tag{3.2}$$

It is clear that $\pi_1^{-1}(h)$ is a copy of the zero set of h. Reciprocally, for fixed $\zeta \in \mathbb{P}(\mathbb{C}^{n+1})$, the set $\pi_2^{-1}(\zeta)$ is the vector space of polynomial systems that have ζ as a zero.

Let $\Sigma' \subseteq \mathcal{V}$ be the set of critical points of π_1 and $\Sigma = \pi_1(\Sigma') \subseteq \mathbb{P}(\mathcal{H}_{(d)})$ the set of critical values of π_1. It is not hard to prove that:

- π_1 restricted to the set $\mathcal{V} \setminus \pi_1^{-1}(\Sigma)$ is a (smooth) $\mathcal{D}$–fold covering map, where $\mathcal{D} = d_1 \cdots d_n$ is the Bezóut number.
- $\Sigma' = \{(h, \zeta) \in \mathcal{V} : Dh(\zeta) \mid_{\zeta^\perp} \text{ has non–maximal rank}\}$. In that case, we say that ζ is a singular zero of h. Otherwise, we say that ζ is a regular zero of h.

This means, in particular, that the homotopy process described above can be carried out whenever the path of systems lies outside of Σ:

THEOREM 3.1. *Let $\{h_t : t \in [a, b]\}$ be a C^1 curve in $\mathbb{P}(\mathcal{H}_{(d)}) \setminus \Sigma$ and let ζ be a zero of h_a. Then, there exists a unique lift of h_t through π_1, that is a C^1 curve $(h_t, \zeta_t) \in \mathcal{V}$ such that $\zeta_a = \zeta$. In particular, ζ_b is a zero of h_b. Moreover, the lifted curve satisfies:*

$$\frac{d}{dt}(h_t, \zeta_t) = \left(\dot{h}_t, -Dh_t(\zeta_t) \mid_{\zeta_t^\perp}^{-1} \dot{h}_t(\zeta_t)\right). \tag{3.3}$$

Finally, the set $\Sigma \subseteq \mathbb{P}(\mathcal{H}_{(d)})$ is a complex projective algebraic variety, thus it has real codimension 2 and the projection of most real lines in $\mathcal{H}_{(d)}$ to $\mathbb{P}(\mathcal{H}_{(d)})$ does not intersect Σ.

The last claim of Theorem 3.1 must be understood as follows. Let $g, f \in \mathcal{H}_{(d)}$ be chosen at random. Then, with probability one, the projection to $\mathbb{P}(\mathcal{H}_{(d)})$ of the line containing g and f does not intersect Σ.

In the case the thesis of Theorem 3.1 holds we just say that ζ_a can be continued to a zero ζ_b of h_a. One can be even more precise:

THEOREM 3.2. *Let $\{h_t : t \in [a, b]\}$ be a C^1 curve in $\mathbb{P}(\mathcal{H}_{(d)}) \setminus \Sigma$ and let ζ be a zero of h_a. Then, every zero ζ of h_a can be continued to a zero of h_b, defining a bijection between the $\mathcal{D}$ zeros of h_a and those of h_b.*

REMARK 3.3. Even if h_t crosses Σ some solutions may be able to be continued while others may not.

The (normalized) condition number [**52**] is a quantity describing "how close to singular" a zero is. Given $h \in \mathcal{H}_{(d)}$ and $z \in \mathbb{P}(\mathbb{C}^{n+1})$, let

$$\mu(f, z) = \|f\| \|(Dh(z) \mid_{z^\perp})^{-1} Diag(\|z\|^{d_i - 1} d_i^{1/2})\|_2, \tag{3.4}$$

and $\mu(f, z) = +\infty$ if $Dh(z) \mid_{z^\perp}$ is not invertible. Sometimes μ is denoted μ_{norm} or μ_{proj} but we prefer to keep the more simple notation here. One of the most important properties of μ is that it is an upper bound for the norm of the (locally defined) implicit function related to π_1 in (3.2). Namely, let $(\dot{h}, \dot{\zeta}) \in T_{(h,\zeta)}\mathcal{V}$ where $(h, \zeta) \in \mathcal{V}$ is such that $\mu(h, \zeta) < +\infty$. Then,

$$\|\dot{\zeta}\| \leq \mu(h, \zeta) \|\dot{h}\|, \quad \mu(h, \zeta) \geq \sqrt{n}. \tag{3.5}$$

We also have the following result.

THEOREM 3.4 (Condition Number Theorem,[**52**]).

$$\mu(h, \zeta) = \frac{1}{\sin\left(d_R(h, \Sigma_\zeta)\right)},$$

where d_R is the Riemannian distance in $\mathbb{P}(\mathcal{H}_{(d)})$ and

$$\Sigma_\zeta = \{h \in \mathbb{P}(\mathcal{H}_{(d)}) \ : \ h(\zeta) = 0, \text{ and } Dh(\zeta) \mid_{\zeta^\perp} \text{ is not invertible}\}.$$

Note that this is a version of the classical Condition Number Theorem of linear algebra (see Theorem 6.5 below). The existence of approximate zeros in the sense of Definition 2.4 above is also guaranteed by this condition number, as was noted in [**52**]. More precisely:

THEOREM 3.5 (μ–Theorem, [**52**]). *There exists a constant $u_0 > 0$ ($u_0 = 0.17586$ suffices) with the following property. Let $(h, \zeta) \in \mathcal{V}$ and let $z \in \mathbb{P}(\mathbb{C}^{n+1})$ satisfy*

$$d_R(z, \zeta) \leq \frac{u_0}{d^{3/2}\,\mu(h, \zeta)}.$$

Then, z is an approximate zero of h with associated zero ζ.

4. The complexity of following a homotopy path

The sentence "can be continued" in the discussion of Section 3 can be made much more precise, by defining an actual path–following method. It turns out that the unique method that has actually been proved to correctly follow the homotopy paths and at the same time achieve some known complexity bound is the most simple one, which only uses the projective Newton operator, and not an ODE solver step.

PROBLEM 4.1. *It would be an interesting project to compare the overall cost of using a higher order ODE solver to the projective Newton–based method we describe below. Higher order methods or even predictor–corrector methods may require fewer steps but be more expensive at each step so a total cost comparison is in order. Some experience indicates that higher order methods are rarely cheaper, if ever. See* [**39**, **40**].

More precisely, the projective Newton–based homotopy method is as follows. Given a C^1 path $\{h_t : a \leq t \leq b\} \subseteq \mathbb{P}(\mathcal{H}_{(d)})$, and given z_a an approximate zero of h_a with associated (exact) zero ζ_a, let $t_0 > 0$ be "small enough" and let

$$z_{a+t_0} = z_a - (Dh_{a+t_0}(z_a) \mid_{z_a^\perp})^{-1} h_{a+t_0}(z_a),$$

that is z_{a+t_0} is the result of one application of the projective Newton operator based on h_{a+t_0} to the point z_a. If z_a is an approximate zero of h_a and t_0 is small enough, then z_a can be close enough to the actual zero ζ_{a+t_0} of h_{a+t_0} to satisfy Theorem 3.5 and thus be an approximate zero of h_{a+t_0} as well. Then, by definition of approximate zero, z_{a+t_0} will be half–closer to ζ_{a+t_0} than z_a. This leads to an inductive process (choosing t_1, then t_2, etc. until h_b is reached) that, analysed in detail, can be made to work and actually programmed. The details on how to choose t_0 would take us too far apart from the topic, so we just give an intuitive explanation: if we are to move from (h_a, ζ_a) to $(h_{a+t_0}, \zeta_{a+t_0})$ we must be sure that we are far enough from Σ' to have our algorithm behaving properly. As the condition number essentially measures the distance to Σ', it should be clear that the bigger the condition number, the smaller step t_0 we can take. This idea lead to the following result (see [**56**] for a weaker, earlier result):

THEOREM 4.2 ([**51**]). *Let $(h_t, \zeta_t) \subseteq \mathcal{V} \setminus \Sigma'$, $t \in [a, b]$ be a C^1 path. If the steps $t_0, t_1, \ldots$ are correctly chosen, then an approximate zero of h_b is reached at some point, namely there is a $k \geq 1$ such that $\sum_{i=0}^{k} t_i = b - a$ (k is the number of steps in the inductive process above.) Moreover, one can bound*

$$k \leq \lceil C d^{3/2} L_\kappa \rceil,$$

where d is the maximum of the degrees in (d), C is some universal constant, and

$$L_\kappa = \int_a^b \mu(h_t, \zeta_t) \|(\dot{h}_t, \dot{\zeta}_t)\| \, dt \tag{4.1}$$

is called the condition length of the path (h_t, ζ_t). Moreover, the amount of arithmetic operations needed in each step is polynomial in the input size N, and hence the total complexity of the path–following procedure is a quantity polynomial in N and linear in L_κ

There exist several ways to algorithmically produce the steps $t_0, t_1, \ldots$ in this theorem (and indeed the process has been programmed in two versions [**12**, **13**],) but the details are too technical for this report, see [**8**, **27**, **31**]. We also point out that, if the path we are following is linear, i.e. $h_t = (1-t)h_0 + th_1$, and if the input coordinates are (complex) rational numbers, then all the operations can be carried out over the rationals without a dramatic increase of the bit size of intermediate results, see [**13**].

Note that since L_κ is a length it is independent of the C^1 parametrization of the path. If we specify a path of polynomial systems in $\mathcal{H}_{(d)}$ then we project the path of polynomials and solutions into $\mathcal{V}$ to calculate the length. We may project from $\mathcal{H}_{(d)}$ to $\mathbb{S}$ first and reparametrize if we wish. For example, we project the straight line segment $h_t = (1-t)g + th$ for $0 \leq t \leq 1$ into $\mathbb{S}$ and reparametrize by arc–length. If $\|g\| = \|h\| = 1$ the resulting curve is

$$h_t = g\cos(t) + \frac{h - \langle h, g\rangle g}{\|h - \langle h, g\rangle g\|} \sin(t)$$

which is an arc of great circle through g and h. If $0 \leq t \leq d_R(g,h)$, then the arc goes from g to h. Here $d_R(g,h)$ is the Riemannian distance in $\mathbb{S}$ between g and h which is the angle between them.

5. The problem of good starting points

We now come back to the original question in Smale's 17-th problem. Our plan is to analyse the complexity of an algorithm that we could call "linear homotopy": choose some $g \in \mathbb{S}, \zeta \in \mathbb{P}(\mathbb{C}^{n+1})$ such that $g(\zeta) = 0$ (we will call (g, ζ) a "starting pair"). For input $h \in \mathbb{S}$, consider the path contained in the great circle :

$$h_t = g\cos(t) + \frac{h - \langle h, g\rangle g}{\|h - \langle h, g\rangle g\|} \sin(t), \quad t \in [0, d_R(g,h)]. \tag{5.1}$$

Then, use the method described in Theorem 4.2 to track how ζ_0 moves to $\zeta_{d_R(g,h)}$, a zero of $h_{d_R(g,h)} = h$, thus producing an approximate zero of h. We call this linear homotopy (maybe a more appropriate name would be "great circle homotopy") because great circles are projections on $\mathbb{S}$ of segments in $\mathcal{H}_{(d)}$.

Assuming that the input h is uniformly distributed on $\mathbb{S}$, we can give an upper bound for the average number of arithmetic operations needed for this task (that is, the average complexity of the linear homotopy method) by a polynomial in N multiplied by the following quantity:

$$\frac{1}{\nu(\mathbb{S})} \int_{h\in\mathbb{S}} \int_0^{d_R(g,h)} \mu(h_t, \zeta_t) \|(\dot{h}_t, \dot{\zeta}_t)\| \, dt \, d\mathbb{S},$$

where h_t is defined by (5.1) and ζ_t is defined by continuation (the fact that $h_t \cap \Sigma = \emptyset$, and thus the existence of such ζ_t, is granted by Theorem 3.1 for most choices of g, h). It is convenient to replace this last expected value by a similar upper bound:

$$\mathcal{A}_1(g,\zeta) = \frac{1}{\nu(\mathbb{S})} \int_{h\in\mathbb{S}} \int_0^{\pi} \mu(h_t,\zeta_t) \|(\dot{h}_t,\dot{\zeta}_t)\| \, dt \, d\mathbb{S}.$$

Note that we are just replacing the integral from 0 to $d_R(g,h)$ by the distance from 0 to π.

We thus have:

THEOREM 5.1. *Let $(g,\zeta) \in \mathcal{V}$. The average complexity of linear homotopy with starting pair (g,ζ) is bounded above by a polynomial in N multiplied by $\mathcal{A}_1(g,\zeta)$.*

This justifies the following definition:

DEFINITION 5.2. Fix some polynomial[1] $p \in \mathbb{R}[x,y,z]$. We say that (g,ζ) is a good starting pair w.r.t. $p(x,y,z)$ if $\mathcal{A}_1(g,\zeta) \leq p(n,d,N)$ (which implies that the average number of steps of the linear homotopy is $O(d^{3/2}p(n,d,N))$.) From now on, if nothing is said, we assume $p(x,y,z) = \sqrt{2}\pi xz$. Thus, $(g,\zeta) \in \mathcal{V}$ is a good initial pair if $\mathcal{A}_1(g,\zeta) \leq \sqrt{2}\pi nN$.

So, if a good sequence of initial pair is known for all choices of n and the list of degrees (d), then the total average complexity of linear homotopy is polynomial in N. In other words, finding good starting pairs for every choice of n and (d) gives a satisfactory solution to Problem (2.1).

In [**56**] the following pair[2] was conjectured to be a good starting pair (for some polynomial $p(x,y,z)$) :

$$g(z) = \begin{cases} d_1^{1/2} z_0^{d_1-1} z_1, \\ \quad \vdots \\ d_n^{1/2} z_0^{d_n-1} z_n \end{cases}, \qquad \zeta = (1,0,\ldots,0). \tag{5.2}$$

To this date, proving this conjecture is still an open problem. Some experimental data supporting this conjecture was shown in [**12**].

5.1. Choosing initial pairs at random: an Average Las Vegas algorithm for problem (2.1). One can study the average value of the quantity $\mathcal{A}_1(g,\zeta)$ described above. Most of the results in this section are based on the fact that the expected value of the square of the condition number is relatively small. This was first noted in [**53**], then this expected value was computed exactly in [**16**]:

THEOREM 5.3. *Let $h \in \mathbb{S}$ be chosen at random, and let ζ be chosen at random, with the uniform distribution, among the zeros of h. Then, the expected value of $\mu^2(h,\zeta)$ is at most nN. More exactly:*

$$\mathrm{E}_{h\in\mathbb{S}} \left(\frac{1}{\mathcal{D}} \sum_{\zeta : h(\zeta)=0} \mu(h,\zeta)^2 \right) = N \left(n \left(1 + \frac{1}{n} \right)^{n+1} - 2n - 1 \right) \leq nN.$$

[1]Because $n, d \leq N$, we could just talk about a one variable polynomial $p(x)$ and change $p(n,d,N)$ to $p(N)$ in the following definition. However, we prefer here to be a bit more precise.

[2]The pair conjectured in [**56**] does not contain the extra $d_i^{1/2}$ factors. There is, however, some consensus that these extra factors should be added, for with these factors the condition number $\mu(g,\zeta) = n^{1/2}$ is minimal.

In particular, in the case of one homogeneous polynomial of degree d (i.e. $n = 1$,) we have:

$$\mathrm{E}_{h\in\mathbb{S}}\left(\sum_{\zeta:h(\zeta)=0}\mu(h,\zeta)^2\right) = d(d+1).$$

Now we use some arguments which are very much inspired by ideas from integral geometry, one of the main contributions of Lluis Santaló to XX century mathematics. We can try to compute the expected value of $\mathcal{A}_1(g,\zeta)$. Although this can be done directly (see [**18**],) it is easier to first consider an upper bound of $\mathcal{A}_1$: let us note from (3.5) that

$$\mathcal{A}_1(g,\zeta) \leq \frac{\sqrt{2}}{\nu(\mathbb{S})}\int_{h\in\mathbb{S}}\int_0^\pi \mu(h_t,\zeta_t)^2\,dt\,d\mathbb{S}. \tag{5.3}$$

So, we have

$$\mathrm{E}_{g\in\mathbb{S}}\left(\sum_{\zeta:g(\zeta)=0}\mathcal{A}_1(g,\zeta)\right) \leq \mathrm{E}_{g\in\mathbb{S}}\left(\sum_{\zeta:h(\zeta)=0}\frac{\sqrt{2}}{\nu(\mathbb{S})}\int_{h\in\mathbb{S}}\int_0^\pi \mu(h_t,\zeta_t)^2\,dt\,d\mathbb{S}.\right) =$$

$$\sqrt{2}\,\mathrm{E}_{(g,h)\in\mathbb{S}\times\mathbb{S}}\left(\int_{f\in L_{g,h}}\sum_{\zeta:f(\zeta)=0}\mu(f,\zeta)^2\right),$$

where $L_{g,h}$ is the half–great circle in $\mathbb{S}$ containing g, h, starting at g and going to $-g$ (we have to remove from this argument the case $h = -g$ but this is unimportant for integration purposes.) Note that we can define a measure and more generally a concept of integral in $\mathbb{S}$ as follows: given any measurable function $q : \mathbb{S} \to [0,\infty)$, its integral is

$$\mathrm{E}_{(g,h)\in\mathbb{S}\times\mathbb{S}}\left(\int_{f\in L_{g,h}} q(f)\right). \tag{5.4}$$

Now, this last formula describes an invariant (with respect to the group of symmetries of $\mathbb{S}$, that can be identified with the unitary group of size $N+1$ or with the orthogonal group of size $2N+2$) measure in $\mathbb{S}$ and is thus equal to a multiple of the usual measure in $\mathbb{S}$. In words, averaging over $\mathbb{S}$ or over great circles in $\mathbb{S}$ is the same, up to a constant. The constant is easy to compute by considering the constant function $q \equiv 1$. What we get is:

$$\mathrm{E}_{g\in\mathbb{S}}\left(\sum_{\zeta:g(\zeta)=0}\mathcal{A}_1(g,\zeta)\right) \leq \frac{\pi}{\sqrt{2}}\mathrm{E}_{h\in\mathbb{S}}\left(\sum_{\zeta:h(\zeta)=0}\mu(h,\zeta)^2\right).$$

After this argument is made rigorous, we have (see [**14**, **15**] for earlier versions of the following result:)

THEOREM 5.4 ([**16**]). *Let $g \in \mathbb{S}$ be chosen at random with the uniform distribution, and let ζ be chosen at random, with the uniform (discrete) distribution among the roots of g. Then, the expected value of $\mathcal{A}_1(g,\zeta)$ is at most $\frac{\pi}{\sqrt{2}}nN$. In particular, for such a randomly chosen pair (g,ζ), with probability at least $1/2$ we have $\mathcal{A}_1(g,\zeta) \leq \sqrt{2}\pi nN$, that is, (g,ζ) is a good starting pair*[3].

[3]Note that we are computing there the average of $\mathcal{A}_1$ not that of the integral of μ^2 as in [**16**]. From (5.3), the constant $\sqrt{2}$ has to be added to the formula in [**16**] in this context.

The previous result would be useless for describing an algorithm (because choosing a random zero of a randomly chosen $g \in \mathbb{S}$ might be a difficult problem) without the following one.

THEOREM 5.5 ([**16**]). *The process of choosing a random $g \in \mathbb{S}$ and a random zero ζ of g can be emulated by a simple linear algebra procedure.*

The details of the linear algebra procedure of Theorem 5.5 require the introduction of too much notation. We just describe the process in words: one has to choose a random $n \times (n+1)$ matrix M with complex entries, compute its kernel (a projective point $\zeta \in \mathbb{P}(\mathbb{C}^{n+1})$) and consider the system $g \in \mathbb{S}$ that has ζ as a zero and whose linear part is given by M. A random higher–degree term has to be added to g, and then linear and higher–degree terms must be correctly weighted. This whole process has running time polynomial in N. We thus have:

COROLLARY 5.6. *The linear homotopy algorithm with the starting pair obtained as in Theorem* 5.5 *has average complexity*[4] $\tilde{O}(N^2)$.

The word "average" in Corollary 5.6 must be understood as follows. For an input system h, let $T(h)$ be the expected running time of the linear homotopy algorithm, when (g, ζ) is randomly chosen following the procedure of Theorem 5.5. Then, the average value of $T(h)$ for random h is $\tilde{O}(N^2)$. This kind of algorithm is called Average Las Vegas, the "Las Vegas" term coming from the fact that a random choice has to be done. The user of the algorithm plays the role of a Las Vegas casino, not of a Las Vegas gambler: the chances of winning (i.e. getting a fast answer to our problem) are much higher than those of loosing (i.e. waiting for a long time before getting an answer.)

Some of the higher moments of $\mathcal{A}_1(g, \zeta)$ have also been proved to be small. For example, the second moment (thus, also the variance) of $\mathcal{A}_1(g, \zeta)$ is polynomial in N, as the following result shows:

THEOREM 5.7 ([**18**]). *Let $2 \leq k < 3$. Then, the expectation of $\mathcal{A}_1(g, \zeta)^k$ satisfies*

$$\mathrm{E}\left(\mathcal{A}_1(g, \zeta)^k\right) < \infty.$$

Moreover, let $2 \leq k < 3 - \frac{1}{2 \ln \mathcal{D}}$. Then, the expectation $\mathrm{E}\left(\mathcal{A}_1(g, \zeta)^k\right)$ satisfies,

$$\mathrm{E}\left(\mathcal{A}_1(g, \zeta)^k\right) \leq 2^{2k+k/2+4}\, e\, \pi^k n^{3k-4} N^2 \mathcal{D}^{4k-8} \ln \mathcal{D}.$$

In particular, $\mathrm{E}(\mathcal{A}_1(g, \zeta)^2) \leq 512 e \pi^2 n^2 N^2 \ln \mathcal{D}$.

We have been concentrating on finding one zero of a polynomial system. But we could find k zeros $0 \leq k \leq \mathcal{D}$ by choosing k different random initial pairs using Theorem 5.5. This process is known from [**16**] to output every zero of the goal system h with the same probability $1/\mathcal{D}$, if $h \notin \Sigma$. Another option is to choose some initial system g which has k known zeros, and simultaneously continuing the k homotopy paths with the algorithm of Theorem 4.2. In the case of finding all zeros the sum of the number of steps to follow each path, is by Theorem 4.2 and (3.5), bounded above by a constant times

$$d^{3/2} \int_0^{d_R(g,h)} \sum_{\zeta_t : h_t(\zeta_t) = 0} \mu(h_t, \zeta_t)^2 \, dt.$$

[4]We use here the $\tilde{O}(X)$ notation: this is the same as $O(X \log(X)^c)$ for some constant c, that is logarithmic factors are cleaned up to make formulas look prettier.

So for the great circle homotopies we have been discussing an analogue of Theorem 5.4 holds:

THEOREM 5.8 ([**16**]). *Let $g \in \mathbb{S}$ be chosen at random with the uniform distribution. Then, the expected value of $\int_0^{d_R(g,h)} \sum_{\zeta_t : h_t(\zeta_t)=0} \mu(h_t,\zeta_t)^2\, dt$ is at most $\frac{\pi}{\sqrt{2}} nN\mathcal{D}$. In particular, for such a randomly chosen g, with probability at least $1/2$ we have $\int_0^{d_R(g,h)} \sum_{\zeta_t : h_t(\zeta_t)=0} \mu(h_t,\zeta_t)^2\, dt \leq \sqrt{2}\pi nN\mathcal{D}$, that is, the linear homotopy for finding all zeros starting at g takes at most a constant times $d^{3/2}nN\mathcal{D}$ steps to output all zeros of h, on the average.*

Note that in general, one cannot write down all the $\mathcal{D}$ zeros of g to begin with, so Theorem 5.8 does not immediately yield a practical algorithm.

We point out that, even for the case $n = 1$, no explicit descriptions of pairs (g,ζ) satisfying $\mathcal{A}_1(g,\zeta) \leq d^{O(1)}$ are known. Of course, no explicit polynomial $g \in \mathbb{S}$ is known in that case satisfying the claim of Theorem 5.8. An attempt to determine such a polynomial has led to some progress in the understanding of elliptic Fekete points, see Section 8.

5.2. The roots of unity combined with a method of Renegar: a quasi–polynomial time deterministic algorithm for problem (2.1). One can also ask for an algorithm for Problem (2.1) which does not rely on random choices (a deterministic algorithm). The search of a deterministic algorithm with polynomial running time for Problem (2.1) is still open, but a quasi–polynomial algorithm is known since [**27**].

This algorithm is actually a combination of two: on one hand, we consider the initial pair

$$g = \begin{cases} \frac{1}{\sqrt{2n}}(x_0^{d_1} - x_1^{d_1}) \\ \vdots \\ \frac{1}{\sqrt{2n}}(x_0^{d_1} - x_n^{d_n}) \end{cases}, \quad \zeta = (1,\dots,1) \tag{5.5}$$

Then, we have:

THEOREM 5.9 ([**27**]). *The projective Newton–based homotopy method with initial pair (5.5) has average running time polynomial in N and n^d (recall that d is the maximum of the degrees).*

Theorem 5.9 is a consequence of the following stronger result:

THEOREM 5.10 ([**27**]). *The projective Newton–based homotopy method with initial pair $(g,\zeta) \in \mathcal{V}$ has average running time polynomial in N and in $\max\{\mu(g,\eta) : g(\eta) = 0\}$.*

Theorem 5.9 follows from Theorem 5.10 and the fact that $\mu(g,\eta) \leq 2(n+1)^d$ for g given by (5.5) for every zero η of g.

For small (say, bounded) values of d, the quantity n^d is polynomial in n and thus polynomial in N, but for big values of d the quantity n^d is not bounded by a polynomial in N, and thus Theorem 5.9 does not claim the existence of a polynomial running time algorithm. However, it turns out that there is a previously known algorithm, based on the factorization of the u–resultant, that has exponential running time for small degrees, but polynomial running time for high degrees (this

may seem contradictory, but it is not: when the degrees are very high, the input size is big, and thus bounding the running time by a polynomial in the input size is sometimes possible in this case.) More precisely:

THEOREM 5.11 ([**27**, **48**]). *There is an algorithm with average running time polynomial in N and $\mathcal{D}$ that, on input $h \in \mathbb{P}(\mathcal{H}_{(d)}) \setminus \Sigma$, outputs an approximate zero associated to every single exact zero of h.*

Note that $\mathcal{D}$ is usually exponential in n, but as suggested above, if the degrees are very high compared to n, then $\mathcal{D}$ can be bounded above by a polynomial in the input size N and thus the algorithm of Theorem 5.11 becomes a polynomial running time algorithm.

An appropriate combination of theorems 5.9 and 5.11, using the homotopy method of Theorem 5.9 for moderately low degrees and the symbolic–numeric method of Theorem 5.11 for moderately high degrees turns out to be quasipolynomial for every choice of n and (d) . Indeed:

THEOREM 5.12 ([**27**]). *The average (for random $h \in \mathbb{S}$) running time of the following procedure is $O(N^{\log\log N})$: on every input $h \in \mathbb{P}(\mathcal{H}_{(d)}) \setminus \Sigma$, run simultaneously the algorithms of theorems* 5.9 *and* 5.11*, stopping the computation whenever one of the two algorithms gives an output.*

Note that the running time of this algorithm is thus quasi–polynomial in N. Moreover, the algorithm is deterministic because it does not involve random choices.

5.3. Homotopy paths based on the evaluation at one point. Another approach to construct homotopies was considered in [**57**] and generalized in [**4**]. Given $h \in \mathcal{H}_{(d)}$ and $\zeta \in \mathbb{P}(\mathbb{C}^{n+1})$, consider $g = h - \hat{h}_\zeta$, where $\hat{h}_\zeta \in \mathcal{H}_{(d)}$ is defined as

$$\hat{h}_\zeta(z) = Diag\left(\frac{\langle z,\zeta\rangle^{d_i}}{\langle \zeta,\zeta\rangle^{d_i}}\right) h(\zeta).$$

Then, $g(\zeta) = 0$. So, we consider the homotopy $h_t = (1-t)g + th = h - (1-t)\hat{h}_\zeta$. We continue the zero ζ from $h_0 = g$ to $h_1 = h$. For any fixed ζ, for example $\zeta = e_0 = (1, 0, \ldots, 0)$, the homotopy may be continued for almost all $h \in \mathcal{H}_{(d)}$. Let

$$K(h,\zeta) = \text{ number of steps sufficient to continue } \zeta \text{ to a zero of } h,$$

and

$$K(h) = \mathrm{E}_{\zeta\in\mathbb{P}(\mathbb{C}^{n+1})}(K(h,\zeta)).$$

Then,

THEOREM 5.13 ([**4**]).

$$\mathrm{E}_{h\in\mathcal{H}_{(d)}}(K(h)) \leq \frac{Cd^{3/2}\Gamma(n+1)2^{n-1}}{(2\pi)^N\pi^n} \int_{h\in\mathcal{H}_{(d)}} \left(\sum_{\eta:h(\eta)=0} \frac{\mu(h,\eta)^2}{\|h\|^2}\Theta(h,\eta)\right) e^{-\|h\|^2/2}\, dh,$$

where

$$\Theta(h,\eta) = \int_{\zeta\in B(h,\eta)} \frac{(\|h\|^2 - T^2)1/2}{T^{2n-1}}\Gamma(T^2/2, n)e^{T^2/2}\, d\zeta,$$

$$T = \|Diag(\|\zeta\|^{-d_i}))h(\zeta)\|,$$

and $\Gamma(\alpha, n) = \int_\alpha^{+\infty} t^{n-1}e^{-t}\, dt$ is the incomplete gamma function.

In Theorem 5.13, $B(h,\eta)$ is the basin of η, which we now define. Suppose η is a non–degenerate zero of $h \in \mathcal{H}_{(d)}$. We define the basin of η, $B(h,\eta)$, as those $\zeta \in \mathbb{P}(\mathbb{C}^{n+1})$ such that the zero ζ of $g = h - \hat{h}_\zeta$ continues to η for the homotopy $h_t = (1-t)g + th$. We observe that the basins are open sets.

Not much is known about $\mathrm{E}(K)$. See [**4**] for precise questions and motivations. Here is one:

PROBLEM 5.14. *Is* $\mathrm{E}(K)$ *a quantity polynomial in* N*?*

6. The condition Lipschitz–Riemannian structure

Let us know turn our sight back to (4.1). If we drop the condition number $\mu(h_t,\zeta_t)$ from that formula, we get

$$L = \int_a^b \|\dot{h}_t, \dot{\zeta}_t)\|\, dt,$$

that is simply the length of the path (h_t,ζ_t) in the solution variety $\mathcal{V}$, taking on $\mathcal{V}$ the natural metric: the one inherited from that of the product $\mathbb{P}(\mathcal{H}_{(d)}) \times \mathbb{P}(\mathbb{C}^{n+1})$. The formula in (4.1) can now be seen under a geometrical perspective: L_κ is just the length of the path (h_t,ζ_t) when $\mathcal{V}$ is endowed with the conformal metric obtained by multiplying the natural one by the square of the condition number. Note that this new metric is only defined on $\mathcal{W} = \mathcal{V} \setminus \Sigma'$. We call this new metric the condition metric in $\mathcal{W}$. This justifies the name condition length we have given to L_κ. Theorem 4.2 now reads simply as follows: the complexity of following a homotopy path (h_t,ζ_t) is at most a small constant $cd^{3/2}$ times the length of (h_t,ζ_t) in the condition metric. This makes the condition metric an interesting object of study: which are the theoretical properties of that metric? given $p, q \in \mathcal{W}$, what is the condition length of the shortest path joining p and q?

The first thing to point out is that μ is not a C^1 function, as it involves a matrix operator norm. However, μ is locally Lipschitz. Thus, the condition metric is not a Riemannian metric (usually, one demands smoothness or at least C^1 for Riemannian metrics,) but rather we may call it a Lipschitz–Riemannian structure. This departs from the topic of most available books and papers dealing with geometry of manifolds, but there are still some things we can say. It is convenient to take a tour to a slightly more general kind of problems; that's the reason for the following section.

6.1. Conformal Lipschitz–Riemann structures and self–convexity. Let M be a finite–dimensional Riemannian manifold, that is a smooth manifold with a smoothly varying inner product defined at the tangent space to each point $x \in \mathrm{M}$, let us denote it $\langle\cdot,\cdot\rangle_x$. Let $\alpha : \mathrm{M} \to [0,\infty)$ be[5] a Lipschitz function, that is, there exists some constant $K \geq 0$ such that

$$|\alpha(x) - \alpha(y)| \leq K d_R(x,y), \quad \forall\, x, y \in \mathrm{M},$$

where $d_R(x,y)$ is the Riemannian distance from x to y. Then, consider on each point $x \in \mathrm{M}$ the inner product $\langle\cdot,\cdot\rangle_{\alpha,x} = \alpha(x)\langle\cdot,\cdot\rangle_x$. Note that this need no longer be smoothly varying with x, for $\alpha(x)$ is just Lipschitz. We call such a structure

[5]The reader may have in mind the case $\alpha(h,\zeta) = \mu(h,\zeta)^2$ defined in $\mathrm{M} = \mathcal{V}$.

a (conformal) Lipschitz–Riemannian structure in $\mathcal{M}$, and call it the α–structure. The condition length of a C^1 path $\gamma(t) \subseteq \mathrm{M}$, $a \leq t \leq b$, is just

$$L_\alpha(\gamma) = \int_a^b \|\dot\gamma(t)\|_{\alpha,\gamma(t)}\, dt = \int_a^b \langle \dot\gamma(t), \dot\gamma(t)\rangle^{1/2}_{\alpha,\gamma(t)}\, dt$$

The distance between any to points $p, q \in \mathrm{M}$ in this α–structure is defined as

$$d_\alpha(p,q) = \inf_{\gamma(t)\subseteq\mathcal{V}} L_\alpha(\gamma), \quad p,q \in \mathrm{M}, \tag{6.1}$$

where the infimum is over all C^1 paths with $\gamma(0) = p, \gamma(1) = q$.

A path $\gamma(t)$, $a \leq t \leq b$ is called a minimizing geodesic if $L_\alpha(\gamma) = d_\alpha(\gamma(a), \gamma(b))$ and $\|\dot\gamma(t)\|_{\alpha,\gamma(t)} \equiv 1$, that is, if it minimizes the length of curves joining its extremal points and if it is parametrized by arc–length. Then, a curve $\gamma(t) \subseteq \mathrm{M}$, for t in some (possibly unbounded) interval I is called a geodesic if it is locally minimizing, namely if for every t in the interior of I there is some interval $[a,b] \subseteq I$ containing t and such that $\gamma\,|_{[a,b]}$ is a minimizing geodesic.

Each connected component of the set M with the metric given by d_α is a path metric space, and it is locally compact because M is a smooth finite–dimensional manifold. We are in a position to use Gromov's version of the classical Hopf–Rinow theorem [**36**, Th.1.10], and we have:

THEOREM 6.1. *Let* M *and* α *be as in the discussion above. Assume additionally that* M *is connected and that* (M, d_α) *is a complete metric space. Then:*

- *each closed, bounded subset is compact,*
- *each pair of points can be joined by a minimizing geodesic.*

Theorem 6.1 gives us sufficient conditions for conformal Lipschitz–Riemannian structures to be "well defined" in the sense that the infimum of (6.1) becomes a minimum. We can go further:

THEOREM 6.2 ([**11**]). *In the notation above, any geodesic is of class* C^{1+Lip}*, that is it is* C^1 *and has a Lipschitz derivative.*

See [**22**] for an early version of Theorem 6.2 and for experiments related to this problem.

One often thinks of the function α as some kind of "squared inverse of the distance to a bad set", so for each connected component of M the set (M, d_α) will actually be complete.

A natural property to ask about is the following: given $p, q \in \mathrm{M}$, and given a geodesic $\gamma(t)$ such that $\gamma(a) = p$, $\gamma(b) = q$, does α attain its maximum on γ in the extremes? Namely, if we think on α as some kind of squared inverse to a bad set, do we have to get closer to the bad set than what we are in the extremes?

EXAMPLE 6.3. A model to think of is Poincaré half–plane with the metric given by the usual scalar product in $\mathbb{R}^2 \cap \{y > 0\}$, multiplied by $\alpha(x,y) = y^{-2}$. Geodesics then become just portions of vertical lines or half–circles with center at the axis $y = 0$. It is clear that, to join any two points, the geodesic does not need to become closer to the bad set $\{y = 0\}$.

We can ask for more: we say that α is self–convex (an abbreviation for self–log–convex) if for any geodesic $\gamma(t)$, the following is a convex function:

$$t \mapsto \log(\alpha(\gamma(t))).$$

Note that this condition is stronger than just asking for $t \mapsto \alpha(\gamma(t))$ to be convex, and thus stronger than asking for the maximum of α on γ to be at the extremal points.

6.2. Convexity properties of the condition number. We have the following result:

THEOREM 6.4 ([**10**]). *Let $k \geq 1$ and let $\mathrm{N} \subseteq \mathbb{R}^k$ be a C^2 submanifold without boundary of $\mathbb{R}^2$. Let $U \subseteq \mathbb{R}^n \setminus \mathrm{N}$ be the biggest open set all of whose points have a unique closest point in* N. *Then, the function $\alpha(x) = \mathit{distance}(x, \mathrm{N})^{-2}$ is self–convex in U.*

Note that Theorem 6.4 is a more general version of Example 6.3, where the horizontal line $\{y = 0\}$ is changed to a submanifold N.

A well–known result usually attributed to Eckart and Young [**35**] and to Schmidt and Mirsky (see [**61**]) relates the usual condition number of a full rank rectangular matrix to the inverse distance to the set of rank–deficient matrices:

THEOREM 6.5 (Condition Number Theorem of linear algebra). *Let $A \in \mathbb{C}^{mn}$ be a $m \times n$ matrix for some $1 \leq m \leq n$. Let $\sigma_1(A), \ldots, \sigma_m(A)$ be its singular values. Then,*

$$\sigma_m(A) = \mathit{distance}(A, \{\textit{rank–deficient matrices}\}).$$

In particular, in the case of square maximal rank matrices, we can rewrite this as $\|A^{-1}\| = \mathit{distance}(A, \{\textit{rank–deficient matrices}\})^{-1}$, that is the (unscaled) condition number $\|A^{-1}\|$ equals the inverse of the distance from A to the set of singular matrices. We more generally call $\sigma_m^{-1}(A)$ the unscaled condition number of a (possibly rectangular) full–rank matrix A.

One feels tempted to conclude from theorems 6.4 and 6.5 that the function sending a full–rank complex matrix A to the squared inverse of its smallest singular value (i.e. to the square of its unscaled condition number) should be self–convex. Indeed, one cannot apply Theorem 6.4 because the set of rank–deficient matrices is not a C^2 manifold, and because the distance to it is for many matrices (more precisely: whenever the multiplicity of the smallest singular value is greater than 1) not attained in a single point. It takes a considerable effort to prove that the result is still true:

THEOREM 6.6 ([**11**]). *The function defined in the space of full–rank $m \times n$ matrices, $1 \leq m \leq n$, as the squared inverse of the unscaled condition number, is self–convex.*

Note that this implies that, given any two complex matrices A, B of size $m \times n$, and given any geodesic $\gamma(t)$, $a \leq t \leq b$ in the α–structure defined in

$$\mathbb{C}^{mn} \setminus \{ \text{ rank–deficient matrices}\}$$

by $\alpha(C) = \sigma_m(C)^{-2}$ such that $\gamma(a) = A$, $\gamma(b) = B$, the maximum of α along γ is $\alpha(A)$ or $\alpha(B)$.

Note that, if a similar result could be stated for the α–structure defined by $(h, \zeta) \mapsto \mu(h, \zeta)^2$ in $\mathcal{W}$, we would have quite a nice description of how geodesics in the condition metric of $\mathcal{W}$ are. Proving this is still an open problem:

PROBLEM 6.7. *Prove or disprove μ^2 is a self–convex function in $\mathcal{W}$.*

Note that from Theorem 3.4, the function μ^2 is not exactly the squared inverse of the distance to a submanifold, but it is still something similar to that. This makes it plausible to believe that Problem 6.7 has an affirmative answer. A partial answer is known:

THEOREM 6.8 ([**11**]). *The function $h \mapsto \mu^2(h, e_0)$ defined in the set $\{h \in \mathbb{P}(\mathcal{H}_{(d)}) : h(e_0) = 0\}$ is self–convex. Here, $e_0 = (1, 0, \ldots, 0)$.*

7. Condition geodesics and the geometry of $\mathcal{W}$

Although we do not have an answer to Problem 6.7, we can actually state some bounds that give clues on the properties of the geodesics in the condition structure in $\mathcal{W}$. More precisely:

THEOREM 7.1 ([**17**]). *For every two pairs $(h_1, \zeta_1), (h_2, \zeta_2) \in \mathcal{W}$, there exists a curve $\gamma_t \subseteq \mathcal{W}$ joining (h_1, ζ_1) and (h_2, ζ_2), and such that*

$$L_\kappa(\gamma_t) \leq 2cnd^{3/2} + 2\sqrt{n} \ln \left(\frac{\mu(h_1, \zeta_1)\mu(h_2, \zeta_2)}{n} \right),$$

c a universal constant.

In the light of Theorem 4.2, this means that if one can find geodesics in the condition structure in $\mathcal{W}$, one would be able to follow these paths in very few steps: just logarithmic in the condition number of the starting pair and the goal pair.

COROLLARY 7.2. *A sufficient number of projective Newton steps to follow some path in $\mathcal{W}$ starting at the pair (g, e_0) of (5.2) to find an approximate zero associated to a solution ζ of a given system $h \in \mathbb{P}(\mathcal{H}_{(d)})$ is*

$$cd^{3/2} \left(nd^{3/2} + \sqrt{n} \ln \left(\frac{\mu(h, \zeta)}{\sqrt{n}} \right) \right),$$

c a universal constant.

Note that only the logarithm of the condition number appears in Corollary 7.2. Thus, if one could find an easy way to describe condition geodesics in $\mathcal{W}$, the average complexity of approximating them using Theorem 4.2 would involve just the expectation of the average of $\ln(\mu)$, not that of μ^2 as in Theorem 5.3. As a consequence, the average number of steps needed by such an algorithm would be $O(nd^3 \ln N)$. See [**18**, Cor. 3] for a more detailed statement of this fact. At this point we ask a rather naive, vague question:

PROBLEM 7.3. *May homotopy methods be useful in solving linear systems of equations? Might using geodesics help as in Corollary* 7.2 *and the comments above?*

Large sparse systems are frequently solved by iterative methods and the condition number plays a role in the error estimates. So Problem (7.3) has some plausibility.

REMARK 7.4. There is an exponential gap between the average number of steps needed by linear homotopy $O(d^{3/2}nN)$ and those promised by the condition geodesic–based homotopy (which stays at a theoretical level by now, because one cannot easily describe those geodesics). This exponential gap occurs frequently in theoretical computer science. For example NP–complete problems are solvable in

simply exponential time but polynomial with a witness. The estimates for homotopies with condition geodesics may likely serve as a lower bound for what can be achieved. Also, properties of geodesics as we learn them can inform the design of homotopy algorithms.

There is more we can say about the geometry (and topology) of $\mathcal{W}$, by studying the Frobenius condition number in W, which is defined as follows:

$$\widetilde{\mu}(h,\zeta) = \|h\| \, \|Dh(\zeta)^\dagger \, Diag(\|\zeta\|^{d_i-1} d_i^{1/2})\|_F, \quad \forall (h,\zeta) \in W,$$

where $\|\cdot\|_F$ is Frobenius norm (i.e. $Trace(L^*L)^{1/2}$ where L^* is the conjugate transpose of L) and † is Moore-Penrose pseudoinverse.

REMARK 7.5. The Moore-Penrose pseudoinverse $L^\dagger : \mathbb{F} \to \mathbb{E}$ of a linear operator $L : \mathbb{E} \to \mathbb{F}$ of finite dimensional Hilbert spaces is defined as the composition

$$L^\dagger = i_{\mathbb{E}} \circ (L \mid_{Ker(L)^\perp})^{-1} \circ \pi_{Image(L)}, \tag{7.1}$$

where $\pi_{Image(L)}$ is the orthogonal projection on image L, $Ker(L)^\perp$ is the orthogonal complement of the nullspace of L, and $i_{\mathbb{E}}$ is the inclusion. If A is a $m \times (n+1)$ matrix and $A = UDV^*$ is a singular value decomposition of A, $D = Diag(\sigma_1, \ldots, \sigma_k, 0, \ldots, 0)$ then we can write

$$A^\dagger = VD^\dagger U^*, \qquad D^\dagger = Diag(\sigma_1^{-1}, \ldots, \sigma_k^{-1}, 0, \ldots, 0). \tag{7.2}$$

In [**19**] we prove that $\widetilde{\mu}$ is an equivariant Morse function defined in $\mathcal{W}$ with a unique orbit of minima given by the orbit $\mathcal{B}$ of the pair of (5.2) under the action of the unitary group $(U, (h,\zeta)) \mapsto (h \circ U^*, U\zeta)$.

The function $\mathcal{A}_1(g,\zeta)$ or even its upper bound (up to a $\sqrt{2}$ factor) estimate [6]

$$\mathcal{B}_1(g,\zeta) = \frac{1}{\nu(\mathbb{S})} \int_{h \in \mathbb{S}} \int_0^\pi \mu(h_t, \zeta_t)^2 \, dt \, d\mathbb{S}$$

is an average of μ^2 in great circles. This remark motivates the following

PROBLEM 7.6. *Is $\mathcal{A}_1(g,\zeta)$ or $\mathcal{B}_1(g,\zeta)$ also an equivariant Morse function whose only critical point set is a unique orbit of minima?*

If so, due to symmetry considerations, it is the orbit through the conjectured good starting point (5.2). Here, one may want to replace the condition number μ in the definition of $\mathcal{B}_1$ with a smooth version such as the Frobenius condition number. A positive solution to this problem solves our main problem: the conjectured good initial pair (5.2) is not only good but even best.

Because the Frobenius condition number is an equivariant Morse function, the homotopy groups of $\mathcal{W}$ are equal to those of $\mathcal{B}$, that can be studied with standard tools from algebraic topology. In the case that $n > 1$, for example, we get:

$$\begin{aligned} \pi_0(\mathcal{W}) &= \{0\} \\ \pi_1(\mathcal{W}) &= \mathbb{Z}/a\mathbb{Z} \\ \pi_2(\mathcal{W}) &= \mathbb{Z} \\ \pi_3(\mathcal{W}) &= \pi_k(\mathcal{SU}_{n+1}) \ (k \geq 3), \end{aligned}$$

where $\mathcal{SU}_{n+1}$ is the set of special unitary matrices of size $n+1$, $a = gcd(n, d_1 + \cdots + d_n - 1)$ and $\mathbb{Z}/a\mathbb{Z}$ is the finite cyclic group of a elements.

[6] see (5.3).

In particular, we see that if all the $d_i's$ are equal then $a = 1$ and $\mathcal{W}$ is simply connected; in particular, any curve can be continously deformed into a minimizing geodesic. See [**19**] for more results concerning the geometry of $\mathcal{W}$. We can also prove a lower bound similar to the upper bound of Theorem 7.1:

THEOREM 7.7. *let $\alpha : [a, b] \to \mathcal{W}$ be a C^1 curve. Then, its condition length is at least*

$$\frac{1}{d^{3/2}\sqrt{n+1}} \left| \ln \left(\frac{\mu(\alpha(a))}{\mu(\alpha(b))} \right) - \ln \sqrt{n+1} \right|.$$

REMARK 7.8. We have written Theorem 7.7 using the condition metric as defined in this paper. The original result [**19**, Prop. 11] was written for the so–called smooth condition length, obtained by changing μ to $\widetilde{\mu}$ in the definition of the condition length. This change produces the $\sqrt{n+1}$ factors in Theorem 7.7.

In his article [**59**], Smale suggests that the input size of an instance of a numerical analysis problem should be augmented by $\log W(y)$ where $W(y)$ is a weight function "... to be chosen with much thought..." and he suggests that " the weight is to resemble the reciprocal of the distance to the set of ill–posed problems." That is the case here. The condition numbers we have been using are comparable to the distance to the ill–posed problems and figure in the cost estimates. It would be good to develop a theory of computation which incorporates the distance to ill–posedness, or condition number and distance to ill-posedness in case they may not be comparable, (and precision in the case of round–off error) more systematically so that a weight function will not require additional thought. For the case of linear programming Renegar [**49**] accomplished this. It is our main motivating example as well as the work we have described on polynomial systems. The book [**28**] is the current state of the art. The geometry of the condition metric will to our mind intervene in the analysis. If floating point arithmetic is the model of arithmetic used then ill-posedness will include points where the output is zero as well as points where the output is not Lipschitz.

8. The univariate case and elliptic Fekete points

Let us now center our attention in the univariate case, that, once homogenized, is the case of degree d homogeneous polynomials in two variables. Then,

$$\mu(h, \zeta) = d^{1/2} \|h\| \|(Dh(\zeta)\mid_{\zeta^\perp})^{-1}\| \|\zeta\|^{d-1}.$$

If we are given a univariate polynomial $f(x)$ and a complex zero z of f, we can also use the following more direct (and equivalent) formula for $\mu(h, \zeta)$ where h is the homogeneous counterpart of f and $\zeta = (1, z)$:

$$\mu(h, \zeta) = \frac{d^{1/2}(1 + |z|^2)^{\frac{d-2}{2}}}{|f'(z)|} \|h\|.$$

It was noted in [**54**] that the condition number is related to the classical problem of finding elliptic Fekete points, which we recall now in its computational form (see [**9**] for a survey on the state of art of this problem.)

Given d different points $x_1, \ldots, x_d \in \mathbb{R}^3$, let $X = (x_1, \ldots, x_d)$ and

$$\mathcal{E}(X) = \mathcal{E}(x_1, \ldots, x_d) = -\sum_{i<j} \log \|x_i - x_j\|$$

be its logarithmic potential. Sometimes $\mathcal{E}(X)$ is denoted by $\mathcal{E}_0(X)$, $\mathcal{E}(0,X)$ or $V_N(X)$. Let $S(1/2)$ be the Riemann sphere in $\mathbb{R}^3$, that is the sphere of radius $1/2$ centered at $(0,0,1/2)$, and let

$$m_d = \min_{x_1,\dots,x_d \in S(1/2)} \mathcal{E}(x_1,\dots,x_d)$$

be the minimum value of $\mathcal{E}$. A minimising d–tuple $X = (x_1,\dots,x_d)$ is called a set of elliptic Fekete points [7].

The computational problem of finding elliptic Fekete points is another of the problems in Smale's list [8].

Smale's 7th problem [**60**]: Can one find $X = (x_1,\dots,x_d)$ such that

$$\mathcal{E}(X) - m_d \leq c \log d, \qquad c \text{ a universal constant.} \tag{8.1}$$

The first clue that this problem is hard comes from the fact that the value of m_d is not known, even to $O(d)$. A general technique (valid for Riemannian manifolds) given by Elkies shows that

$$m_d \geq \frac{d^2}{4} - \frac{d \log d}{4} + O(d).$$

Wagner [**64**] used the stereographic projection and Hadamard's inequality to get another lower bound. His method was refined by Rakhmanov, Saff and Zhou [**45**], who also proved an upper bound for m_d using partitions of the sphere. The lower bound was subsequently improved upon by Dubickas and Brauchart [**34**], [**24**]. The following result summarizes the best known bounds:

THEOREM 8.1. *Let C_d be defined* [9] *by*

$$m_d = \frac{d^2}{4} - \frac{d \log d}{4} + C_d d.$$

Then,

$$-0.4375 \leq \liminf_{d \mapsto \infty} C_d \leq \limsup_{d \mapsto \infty} C_d \leq -0.3700708...$$

The relation of this problem to the condition number relies on the fact that sets of elliptic Fekete points are naturally "well separated", and are thus good candidates to be the zeros of a "well–conditioned" polynomial, that is a polynomial all of whose zeros have a small condition number. In [**54**] Shub and Smale proved the following relation between the condition number and elliptic Fekete points.

THEOREM 8.2 ([**54**]). *Let $\zeta_1,\dots,\zeta_d \in \mathbb{P}(\mathbb{C}^2)$ be a set of projective points, and consider them as points in the Riemann sphere $S(1/2)$ with the usual identification $\mathbb{P}(\mathbb{C}^2) \equiv S(1/2)$. Let h be a degree d homogeneous polynomial such that its zeros are $\zeta_1,\dots,\zeta_d$. Then,*

$$\max\{\mu(h,\zeta_i) : 1 \leq i \leq d\} \leq \sqrt{d(d+1)}\, e^{\mathcal{E}(\zeta_1,\dots,\zeta_d) - m_d}.$$

[7] Such a d–tuple can also be defined as a set of d points in the sphere which maximize the product of their mutual distances.

[8] Smale thinks on points in the unit sphere, but we may think on points in the Riemann sphere, as the two problems are equivalent by sending $(a,b,c) \in S(1/2)$ to $2(a,b,c) - (0,0,1)$.

[9] The result in the original sources is written for the unit sphere, we translate it here to the Riemann sphere.

In particular, is $x_1, \ldots, x_d$ are a set of elliptic Fekete points, then

$$\max\{\mu(h, \zeta_i) : 1 \leq i \leq d\} \leq \sqrt{d(d+1)}.$$

REMARK 8.3. Let $\mathfrak{Re}$ and $\mathfrak{Im}$ be, respectively, the real and complex part of a complex number. Here is alternative, equivalent definition for h and the ζ_i. Instead of considering projective points in $\mathbb{P}(\mathbb{C}^2)$ we may just consider a set of complex numbers $z_1, \ldots, z_d \in \mathbb{C}$. Then, for $1 \leq i \leq d$, we can define $\zeta_i \in \mathbb{S}$ as

$$\zeta_i = \left(\frac{\mathfrak{Re}(z_i)}{1+|z_i|^2}, \frac{\mathfrak{Im}(z_i)}{1+|z_i|^2}, \frac{1}{1+|z_i|^2} \right)^T \in S(1/2), \qquad 1 \leq i \leq d, \tag{8.2}$$

f as the polynomial whose zeros are $z_1, \ldots, z_d$, and h as the homogeneous counterpart of f.

There exists no explicit known way of describing a sequence of polynomials satisfying $\max\{\mu(h, \zeta) : h(\zeta) = 0\} \leq d^c$, for any fixed constant c and $d \geq 1$. Theorem 8.2 implies that, if a d–tuple satisfying (8.1) can be described for any d, then such a sequence of polynomials can also be generated. From Theorem 5.10, such $h's$ are good starting points for the linear homotopy method, both for finding one root and for finding all roots. So, solving the elliptic Fekete points problem solves the starting point problem for $n = 1$. The reciprocal question is: does solving the starting point problem for $n = 1$ help with the Fekete point problem?

PROBLEM 8.4. *Suppose $n = 1$ and $g \in \mathbb{S}$ minimizes $\sum_{\zeta : g(\zeta)=0} \mu(g, \zeta)^2$. Do $\zeta_1, \ldots, \zeta_d$ (the zeros of g, seen as points in $S(1/2)$) solve Smale's 7–th problem?*

We have seen in Theorem 5.3 that the condition number of (h, ζ) where h is chosen at random and ζ is uniformly chosen at random among the zeros of h, grows polynomially in d. Then, Theorem 8.2 suggests that spherical points associated with zeros of random polynomials might produce small values of $\mathcal{E}$. We can actually put some numbers to this idea. First, one can easily compute the average value of $\mathcal{E}$ when $x_1, \ldots, x_d$ are chosen at random in $S(1/2)$, uniformly and independently with respect to the probability distribution induced by Lebesgue measure in $S(1/2)$:

$$\mathrm{E}_{X \in S(1/2)^d} \mathcal{E}(X) = \frac{d^2}{4} - \frac{d}{4}.$$

By comparing this with Theorem 8.1, we can see that random choices of points in the sphere already produce pretty low values of the minimal energy. One can prove that random polynomials actually produce points which behave better with respect to $\mathcal{E}$:

THEOREM 8.5 ([**3**]). *Let $n = 1$ and $h \in \mathbb{S}$ be chosen at random w.r.t. the uniform distribution in $\mathbb{S}$. Let $\zeta_1, \ldots, \zeta_d \in S(1/2)$ be the zeros of h. Then, the expected value of $\mathcal{E}(\zeta_1, \ldots, \zeta_d)$ equals*

$$\frac{d^2}{4} - \frac{d \log d}{4} - \frac{d}{4}.$$

By comparing this with Theorem 8.1, we conclude that spherical points coming from zeros of random polynomials agree with the minimal value of $\mathcal{E}$, to order $O(d)$.

This result fits into a more general[10] result related to random sections on Riemann surfaces, see [**65**, **66**].

9. The algebraic eigenvalue problem

The double fibration scheme proposed in (3.2) has been – at least partly – successfully used in other contexts. For example, in [**1**] a similar projection scheme (9.1)

$$\begin{array}{ccc} & \mathcal{V}_{eig} & = \{((A,\lambda),v) \in \mathbb{P}(\mathbb{C}^{n^2+1}) \times \mathbb{P}(\mathbb{C}^n) : Av = \lambda v\} \\ \pi_1 \swarrow & & \searrow \pi_2 \\ \mathbb{P}(\mathbb{C}^{n^2+1}) & & \mathbb{P}(\mathbb{C}^{n+1}) \end{array}$$

was used to study the complexity of a homotopy–based eigenvalue algorithm, obtaining the following:

THEOREM 9.1. *A homotopy algorithm can be designed that continues an eigenvalue–eigenvector pair* (λ_0, v_0) *of a* $n \times n$ *matrix* A_0 *to one* (λ_1, v_1) *of another matrix* A_1*, the number of steps bounded above by*

$$c \int_0^1 \|(\dot{A}, \dot{\lambda}, \dot{v})\| \mu_{eig}(A, \lambda, v)\, dt,$$

c a universal constant. Here, μ_{eig} *is the condition number* [11] *for the algebraic eigenvalue problem , defined as*

$$\mu_{eig}(A, \lambda, v) = \max\left\{1, \|A\|_F \|\pi_{v^\perp}(\lambda I_n - A)\,|_{v^\perp}^{-1}\,\|\right\}, \tag{9.2}$$

where $\|A\|_F = trace(A^*A)^{1/2}$ *is the Frobenius norm of A.*

Of course, we do not intend to summarize here the enormous amount of methods and papers dealing with the eigenvalue problem (see [**61**] for example). We just point out that there exists no proven polynomial–time algorithm for approximating eigenvalues (although different numerical methods achieve spectacular results in practice.) See [**44**] for some statistics about the QR (and Toda) algorithms for symmetric matrices. We don't know a good reference for the more difficult general case. Unshifted QR is not the fast algorithm of choice. The QR algorithm with Francis double shift executed on upper Hermitian matrices should be the gold standard.

PROBLEM 9.2. *Does the QR algorithm with Francis double shift fail to attain convergence on an open subset of upper Hessenberg matrices?*

See [**6**] for open sets where Rayleigh quotient iteration fails, and [**5**] for a proof of convergence for normal matrices as well as a good introduction to the dynamics involved.

Theorem 9.1 can probably be used in an analysis similar to that of Section 5 to complete a complexity analysis. Note that the integral in Theorem 9.1 is very similar in spirit to that of (4.1). This allows to introduce a condition metric in

[10] Steve Zelditch tells us that "the relation between the special case of the round metric on S(1/2) and the general metric on any Riemann surface is that the expansion terminates on S(1/2) because the Fubini-Study metric is balanced,i.e. the szego kernel is constant on the diagonal. For general metrics it will not terminate."

[11] A quantity similar in spirit to the condition number μ for the polynomial system solving problem.

$\mathcal{V}_{eig}$. Some of the results in previous sections can be adapted to this new case. For example, an analogue of Theorem 7.1 holds (i.e. short geodesics exist,) see [**2**].

The eigenvalue problem and the problem of finding roots of a polynomial in one variable are, of course, connected. Given an $n \times n$ matrix A we may compute the characteristic polynomial of A, $p(z) = det(zI - A)$ and then solve $p(z)$. The zeros of $p(z)$ are the eigenvalues of A. Trefethen and Bau [**62**] write "This algorithm is not only backward unstable but unstable and should not be used". Indeed when presented with a univariate polynomial $p(z)$ to solve, numerical linear algebra packages may convert the problem to an eigenvalue problem by considering the companion matrix of $p(z)$ and then solve the eigenvalue problem. If $p(z) = z^d + a_{d-1}z^{d-1} + \cdots + a_0$ the companion matrix is

$$\begin{pmatrix} 0 & 0 & 0 & \cdots & 0 & -a_0 \\ 1 & 0 & 0 & \cdots & 0 & -a_1 \\ 0 & 1 & 0 & \cdots & 0 & -a_2 \\ \vdots & & \ddots & \ddots & \vdots & \vdots \\ \vdots & & \ddots & \ddots & 0 & -a_{d-2} \\ 0 & \cdots & \cdots & 0 & 1 & -a_{d-1} \end{pmatrix},$$

which is already in upper Hessenberg form. So conceivably Francis double shifted QR may fail to converge on an open set of companion matrices?

Let us recall that the condition number of a polynomial and root is a property of the output map as a function of the input. So it doesn't depend on the algorithms to solve the problem. This motivates the following

PROBLEM 9.3. *What might explain the experience of numerical analysts, relating the polynomial solving methods versus that of eigenvalue solving? Might the condition number of the eigenvalue problem have small average over the set of $n \times n$ matrices with a given characteristic polynomial?*

Finally, we can consider the problem $Av = \lambda v$ as a system of n quadratic equations in n unknowns. By Bezout's theorem, after we homogenize, we expect 2^n roots counted with multiplicity. But there are only n eigenvalues. In [**1**, **2**] it is shown that the use of multihomogeneous Bezóut theorem yields the correct zero count for this problem. Thus, a reasonable thing to do is to introduce a new variable α and consider the bilinear equation $A\alpha v = \lambda v$ which is bilinear in (α, λ) and v.

PROBLEM 9.4 (see [**32**]). *Prove an analogue of Theorem* 9.1 *in the general multihomogeneous setting.*

Appendix A. A model of computation for machines with round-off and input errors

This section has been developed in discussions with Jean Pierre Dedieu and his colleagues Paola Boito and Guillaume Chèze. We thank Felipe Cucker for helpful comments.

A.1. Introduction. During the second half of the 20th century, with the emergence of computers, algorithms have taken a spectacular place in mathematics, especially numerical algorithms (linear algebra, ode's, pde's, optimization), but

also symbolic computation. In this context, complexity studies give a better understanding of the intrinsic difficulty of a problem, and describe the performance of algorithms which solve such problems. One can associate the classical Turing model to symbolic computation based on integer arithmetic, and the BSS model to scientific computation on real numbers. However this ideal picture suffers from an important defect. Scientific computation does not use the exact arithmetic of real numbers but floating-point numbers and a finite precision arithmetic. Thus, a numerical algorithm designed on real numbers and the same algorithm running in finite precision arithmetic give a priori two different results. Any numerical analysis undergraduate book has at least one chapter dealing with the precision of numerical computations. See for example [**62**] or [**38**]. Yet, there is no solid approach to the definition and study of a model of computation including this aspect, as well as the role that conditioning of problems should play in the complexity estimates.

Besides linear algebra problems and iterative processes, a key point to bear in mind is that we sometimes use floating point computers to answer decision (i.e Yes/No) problems, as is this matrix singular? or does this polynomial have a real zero?. The first attempts to use round-off machines to study decision problems are [**30**], and [**29**]. The authors consider questions like: under which conditions is the decision taken by a BSS machine the same as the decision taken by the corresponding round-off machine? Or, under which conditions is the decision taken by a round-off machine for a given input the same as the decision taken by the BSS machine on a nearby input?

In these pages we point towards the development of a theory of finite precision computation via a description of round–off machines, size of an input, cost of a computation, single (resp. multiple) precision computations (a computation is "single precision" when a sufficient round-off unit δ to reach relative precision u for any input x in the considered range is proportional to u), finite precision computability and finite precision decidability. These concepts have to be related to the intrinsic characteristics of the problem: its condition number (the local Lipschitz constant of the solution map), and its posedness (the distance to ill-posed problems).

The model we propose is inspired by the BSS model but it stays close to real-life numerical computation. We prefer relative errors to absolute ones (this is the basis of the usual floating-point arithmetic.) We mention two papers of interest about the foundations of scientific computing, [**25**, **26**], with a point of view different than ours.

A.2. Round–off machine. A round-off machine is an implementation of a BSS-machine accounting for input error and round-off error of computations. These errors may mimic a particular floating point arithmetic but are designed to be more general. In particular, they are not tied down to a particular floating point model. Let $\mathbb{R}^\infty$ be the disjoint union of the sets $\mathbb{R}^n$, $n \geq 0$. For given $x \in \mathbb{R}^\infty$ we define $\|x\| = \max_i |x_i|$. A subset $U \subseteq \mathbb{R}^\infty$ is open if it is the disjoint union of U_n with $U_n \subseteq \mathbb{R}^n$ an open set. For this topology, a mapping $f : \mathbb{R}^\infty \to \mathbb{R}$ is continuous iff each restriction $f_n = f \mid_{\mathbb{R}^n}$ is continuous.

A (real number) BSS machine M is a directed graph with with several kinds of nodes including an input node, with input $x \in \mathbb{R}^\infty$, output nodes, computation nodes where rational functions are generally computed but here we restrict ourselves without loss of generality to the standard arithmetic operations, branching nodes (we branch on an inequality of the form $y \geq 0$.) A machine is a decision machine

when the output is -1 or 1. The halting set $\mathcal{H}$ of M is the set of inputs giving rise to an output. We denote by $\mathcal{O} : \mathcal{H} \to \mathbb{R}^\infty$ the output map. There are a few technical concepts (mainly the input map $I_M(x)$ and the computing endomorphism H_M) associated to M, the nonfamiliar reader may find formal definitions in [**20**, Chapters 2 and 3].

Given a BSS machine M with nodes $\{1, \ldots, N\}$ and state space $\mathbb{R}_\infty$, we augment the state space $\mathbb{R}_\infty$ by an extra copy of $\mathbb{R}$ so the new state space is $\mathbb{R} \times \mathbb{R}_\infty$. The state space component of the input map is $(1, I_M(x))$. We define a new next node next state map $\hat{H}_M$ by

$$\hat{H}_M(\eta, k, x) = (\pi_1(H_M(\eta, x)), k + 1, \pi_2(H_M(\eta, x)),$$

so the first coordinate acts as a counter (of the number of nodes visited by M). We say that the machine defined $\hat{H}_M$ is a *counting BSS machine*. A little programming shows that adding this extra coordinate does nothing to change the computability or complexity theory of real BSS machines (indeed because $\mathbb{R} \times \mathbb{R}_\infty \equiv \mathbb{R}_\infty$, one can easily see that this newly defined machine is actually a BSS machine). We will moreover assume that our BSS machines are *elementary*, that is that the computation nodes of our machines contain only elementary operations, that is operations of the form $a \circ b$ where $a, b \in \mathbb{R}$ and $\circ \in \{+, -, \times, /\}$. It is a routine task to convert any given BSS machine into a counting elementary machine (this process can be done in many ways, because there are many different ways to compute a polynomial).

Definition A.1 (Round–off machine associated to a given BSS machine). Given a counting, elementary BSS machine M defined over the real numbers and $0 \leq \delta \leq 1$, a round-off machine associated to M and δ is another machine (i.e. a directed graph with the same type of nodes as a BSS machine) denoted (M, δ). The nodes and state space of (M, δ) are the same as for M. The input map $I_{(M,\delta)}$ of (M, δ) satisfies $|I_{(M,\delta)}(x)_j - I_M(x)_j| < \delta |I_M(x)_j|$ that is to say the relative error of the input is less than δ for every coordinate j. The next node next state of (M, δ) at a computation node has the same next node component as H_M, and the jth components of the next states satisfy $|H_{(M,\delta),state}(x)_j - H_{M,state}(x)_j| < \delta |H_{M,state}(x)_j|$, unless $H_{M,state}(x)_j = x_j$ in which case there is no error (i.e. $H_{(M,\delta),state}(x)_j = x_j$). The next node next state map is unchanged at a branch node or at a shift node.

Given any BSS machine M defined over the real numbers and $0 \leq \delta \leq 1$, a round–off machine associated to M and δ is a round–off machine $(\tilde{M}, \delta)$ associated to $\tilde{M}$ and δ where $\tilde{M}$ is some counting, elementary version of M.

Remark A.2. The rounding error introduced at each computation node depends on the whole state and, because M is assumed to be a counting machine, the rounding error may thus depend on the counter. Thus, the rounding error introduced at a given node visited twice may be different (because the counter may be different). Note that the counter is also affected by rounding errors.

Note that a round off machine is not necessarily a BSS machine, and that given M and δ, there are many machines satisfying this definition. For example, M itself satisfies this definition for every δ. The power of the definition is that certain claims will hold for every such a round–off machine, allowing us to use just the defining properties and not the particular structure of a given round–off machine.

DEFINITION A.3. Given a BSS machine M and $0 < \delta < 1$, a δ pseudo–computation with input x is the sequence of pairs $(node, state)$ generated by *some* round–off machine associated to some counting, elementary version of M.

We also point out that not every BSS machine can be (reasonably) converted into a round–off machine. For example, assume that a BSS machine performs the operation $x = (x_1, \dots, x_N) \mapsto x_1 + x_N$. This machine must contain a loop counting up to N. If the form of the *if* node defining the loop is $k \geq 0$ (k the counter which is, say, diminished by 1 at each step) then an arbitrarily small error in the counter of the loop may produce that an associated round–off machine on input x outputs $x_1 + x_{N-1}$ instead of $x_1 + x_N$. A clear way out is to consider the slightly different BSS machine which checks if $-1/2 \leq k \leq 1/2$ instead of $k \geq 0$. Then, a round–off machine with reasonable precision $\delta = O(1/N)$ will do the job. Note that this fits perfectly into the definition of single precision computation (A.7) below. This also reflects the fact, known to every numerical analyst or programmer, that not every program is suitable for floating point conversion: a little care needs to be taken!

A.3. Computability. In the sequel, we will only consider functions $f : \Omega \subseteq \mathbb{R}^\infty \to \mathbb{R}^\infty$ such that, for each n, the restriction f_n of f to $\Omega_n = \Omega \cap \mathbb{R}^n$ takes its values in $\mathbb{R}^m$ for an m depending only on n.

Such a function is round-off computable when there exists a BSS machine M such that for any $x \in \Omega$ and any $0 < \epsilon < 1$, there exists a $\delta(x, \epsilon)$ such that any round–off machine $(M, \delta(x, \epsilon))$ outputs $\tilde{O}(x)$ with

$$|\tilde{O}(x)_j - f(x)_j| \leq \epsilon |f(x)_j|,$$

that is the output of $(M, \delta(x, \epsilon))$ is coordinatewise equal to $f(x)$ up to relative error ϵ. Equivalently, we say that M round–off computes f if given $x \in \Omega$ and $0 < \epsilon < 1$, there is $\delta(x, \epsilon)$ such that all $\delta(x, \epsilon)$ pseudo–computations of M on input x output $f(x)$ with relative precision ϵ.

EXAMPLE A.4. The function $f : \mathbb{R}^2 \to \mathbb{R}$, $f(x, y) = xy$ (we can let it be zero in $\mathbb{R}^\infty \setminus \mathbb{R}^2$) is round–off computable. Indeed, let $x, y \neq 0$ and $0 < \epsilon < 1$. The output of a round–off machine (M, δ) associated to the natural BSS machine for computing $f(x, y)$ is a number

$$z = xy(1 + \delta_1)(1 + \delta_2)(1 + \delta_3),$$

for some $\delta_1, \delta_2, \delta_3$ bounded in absolute value by δ. It is useful to note the elementary inequality

$$\left| \left(1 + \frac{u}{n}\right)^n - 1 \right| \leq 2u, \quad \forall\, 0 \leq |u| \leq 1. \tag{A.1}$$

From this, we obviously have $|z - xy| \leq \epsilon |xy|$ by taking

$$\delta((x, y), \epsilon) = \frac{\epsilon}{6}, \tag{A.2}$$

The output of any round–off machine if $x = 0$ or $y = 0$ is clearly 0, and hence the same value for ϵ of (A.2) suffices to satisfy the definition of computability.

EXAMPLE A.5. The same argument proves that the function $f : \mathbb{R}^\infty \to \mathbb{R}$ given by $f(x_1, \dots, x_n) = x_1 \cdots x_n$ is round–off computable (say, we compute first $x_1 x_2$ then $x_1 x_2 x_3$ and so on) with

$$\delta((x, y), \epsilon) = \frac{\epsilon}{4n - 2}, \tag{A.3}$$

EXAMPLE A.6. A longer computation shows that the function $f : \{(x,y) \in \mathbb{R}^2 : x+y \neq 0\} \to \mathbb{R}$, $f(x,y) = x+y$ (again, we let it be zero in $\mathbb{R}^\infty \setminus \mathbb{R}^2$) is also round–off computable. It suffices to take

$$\delta((x,y),\epsilon) = \frac{\epsilon}{2\max\left(1, \left|\frac{x}{x+y}\right|, \left|\frac{y}{x+y}\right|\right)}.$$

A more simple and still valid formula is

$$\delta((x,y),\epsilon) = \frac{|x+y|}{3\sqrt{2}\sqrt{x^2+y^2}}\epsilon. \tag{A.4}$$

EXAMPLE A.7. Let us now see that $f(x) = x_1 + \ldots + x_n$ is round–off computable in the set $\Omega = \{x \in \mathbb{R}^\infty : x_i \geq 0 \ \forall i\}$. Indeed, let $0 < \epsilon < 1$ and let us consider the most simple BSS machine which computes first $x_1 + x_2$, then adds x_3 and so on[12] A round–off machine with precision δ will produce, on input $x = (x_1, \ldots, x_n)$, a number

$$x_1\left(\prod_{k=1}^{n}(1+\delta_1^{(k)})\right) + x_2\left(\prod_{k=1}^{n}(1+\delta_2^{(k)})\right) + \cdots + x_n\left(\prod_{k=n-1}^{n}(1+\delta_n^{(k)})\right),$$

for some $\delta_i^{(k)}$ bounded in absolute value by δ. This follows from the fact that, in addition to the input error on each coordinate, x_1 and x_2 go through $n-1$ additions (which generate $n+1$ errors), x_3 goes though $n-2$ additions and so on. Note that

$$x_1(1-\delta)^n \leq x_1\left(\prod_{k=1}^{n}(1+\delta_1^{(k)})\right) \leq x_1(1+\delta)^n.$$

Choosing $\delta = \alpha\epsilon/(2n)$, $0 < \alpha \leq 1$ and using (A.1) we conclude that

$$\left|x_1\left(\prod_{k=1}^{n}(1+\delta_1^{(k)})\right) - x_1\right| \leq \alpha\epsilon x_1,$$

and the same formula holds for $x_2, \ldots, x_n$. The output of a round–off machine thus satisfies

$$\tilde{O}(x) = \sum_{i=1}^{n} x_i(1+\alpha\epsilon_i), \quad 0 \leq |\epsilon_i| \leq \epsilon.$$

That is,

$$\left|\tilde{O}(x) - \sum_{i=1}^{n} x_i\right| = \sum_{i=1}^{n} x_i\alpha\epsilon_i \leq \sum_{i=1}^{n} x_i\alpha|\epsilon_i| \leq \alpha\epsilon\sum_{i=1}^{n} x_i,$$

proving that $f(x)$ is round–off computable in that set (just take $\alpha = 1$).

EXAMPLE A.8. Let us now see that $f(x) = x_1 + \ldots + x_n$ is round–off computable in the set $\Omega = \{x \in \mathbb{R}^\infty : \sum x_i \neq 0\}$. We consider the BSS machine that first adds all the nonnegative numbers, call a the result, then adds all the negative numbers, call b the result, and then computes $a - b$. Let $0 < \epsilon < 1$. We note that from Example A.7 by choosing $\delta = \alpha\epsilon/(2n)$ (some $0 < \alpha \leq 1$) the round–off computation of the sum of positive (resp. negative) terms will be

$$\tilde{a} = a(1+\alpha\epsilon_1), \qquad \tilde{b} = b(1+\alpha\epsilon_2), \text{ for some } 0 \leq |\epsilon_1|, |\epsilon_2| \leq \epsilon.$$

[12]This is not the algorithm of choice in practical programming but is sufficient for our purposes here.

From Example A.6, if we let

$$\alpha = \frac{|a+b|}{3\sqrt{2}\sqrt{a^2+b^2}},$$

that is if we let

$$\delta(x,\epsilon) \leq \frac{|a+b|}{3\sqrt{2}\sqrt{a^2+b^2}}\frac{\epsilon}{2n},$$

then $\tilde{O}(x) = \sum_i x_i$ up to relative precision ϵ. Using that $a^2+b^2 \leq n\sum x_i^2$, we can also use the formula

$$\delta(x,\epsilon) = \frac{|\sum_{i=1}^n x_i|}{6\sqrt{2}n^{3/2}\sqrt{\sum_{i=1}^n x_i^2}}\,\epsilon. \tag{A.5}$$

Example A.9. Combining examples A.5 and A.8 we see that the evaluation map of any multivariate polynomial $p(x_1,\ldots,x_n)$ is round–off computable in the complement of its zero set (just compute first the monomials and them add all the results).

A.4. Ill–conditioned instances, condition number, posedness. Let us think of a function $f : \Omega \subseteq \mathbb{R}^\infty \to \mathbb{R}^\infty$ as the solution map associated with some problem to be solved. The condition number associated with f and x measures the first–order (relative) componentwise or normwise variations of $f(x)$ in terms of the first–order (relative) variations of x.

First assume that $f : \Omega \to \mathbb{R}$, that is the function is real–valued. We say that $x \in \bar{\Omega}$ (the topological closure of Ω) is well–conditioned when:

- Either $\|x\| \neq 0$, and f can be extended to a Lipschitz function defined in a neighborhood of x in $\bar{\Omega}$ with $|f(x)| \neq 0$. In that case we define the componentwise condition number by
$$\kappa_f(x) = \limsup_{x' \mapsto x, x' \in \bar{\Omega}} \frac{\frac{|f(x')-f(x)|}{|f(x)|}}{\frac{\|x'-x\|}{\|x\|}},$$
- or f is constant in a neighborhood of x with $|f(x)| = 0$. In this later case we define the condition number by $\kappa_f(x) = 0$.

Otherwise, we say that $x \in \bar{\Omega}$ is ill–conditioned. The set of ill–conditioned instances is denoted by Σ_f, and for $x \in \Sigma_f$, we let $\kappa_f(x) = \infty$.

For a general $f : \Omega \to \mathbb{R}^\infty$, we define

$$\kappa_f(x) = \sup_j \kappa_{f_j}(x) \quad \text{(componenwise condition number)}$$

that is the condition number of f is the supremum of the condition numbers of its coordinates. Sometimes it is more useful to consider the normwise condition number, that we denote by the same letter as the context should make clear which one is used on each problem:

$$\kappa_f(x) = \limsup_{x' \mapsto x, x' \in \bar{\Omega}} \frac{\frac{\|f(x')-f(x)\|}{\|f(x)\|}}{\frac{\|x'-x\|}{\|x\|}} \quad \text{(normwise condition number)},$$

We define the posedness of a problem instance x with $\|x\| \neq 0$ as the distance to ill–posed problems:

$$\pi_f(x) = \frac{d(x,\Sigma_f)}{\|x\|}.$$

Here, $d(x, \Sigma_f) = \inf\{d(x, y) : y \in \Sigma_f\}$. The relation between condition number and posedness is an important but unclear problem. Following [**33**], we may expect a relation of the type

$$\pi_f(x) \approx \kappa_f(x)^{-1}$$

(condition number theorem) or at least inequalities like

$$C_1 \pi_f(x)^{\rho_1} \leq \kappa_f(x)^{-1} \leq C_w \pi_f(x)^{\rho_2}$$

for suitable positive constants C_i, ρ_i (cf. Lojasiewicz's inequality.) To get such a relation ill–posed problems should correspond to infinite condition numbers, but this is not always the case. Consider for example the decision problem: Is $x^2+y^2 \leq \Pi$? The problem is well conditioned except on the circle $x^2+y^2 = \Pi$, but the distance to this circle determines the precision we need in the computation.

Let $K_f(x) = max(\kappa_f(x), \pi_f(x)^{-1})$.

EXAMPLE A.10. For $f(x) = x_1 \cdots x_n$ defined in $\mathbb{R}^\infty$, it is easy to see that

$$\kappa_f(x) = \sqrt{x_1^2 + \cdots + x_n^2}\sqrt{\frac{1}{x_1^2} + \cdots + \frac{1}{x_n^2}},$$

whenever $x_1, \ldots, x_n \neq 0$. If $x_i = 0$ for any i then $\kappa_f(x) = \infty$.

On the other hand,

$$\pi_f(x) = \frac{\min(|x_1|, \ldots, |x_n|)}{\sqrt{x_1^2 + \cdots + x_n^2}}.$$

Thus, we have

$$\kappa_f(x) \leq \sqrt{x_1^2 + \cdots + x_n^2}\sqrt{\frac{n}{\min(|x_1|, \ldots, |x_n|)^2}} = \sqrt{n}\pi_f(x)^{-1},$$

and

$$\kappa_f(x) \geq \sqrt{x_1^2 + \cdots + x_n^2}\sqrt{\frac{1}{\min(|x_1|, \ldots, |x_n|)^2}} = \pi_f(x)^{-1}.$$

Namely,

$$\pi_f(x)^{-1} \leq \kappa_f(x) \leq \sqrt{n}\pi_f(x)^{-1}.$$

EXAMPLE A.11. For $f(x) = x_1 + \cdots + x_n$ defined in $\Omega = \{x \in \mathbb{R}^\infty : \sum x_i \neq 0\}$, we have:

- For $x \in \Omega$, a simple computation shows that
$$\kappa_f(x) = \frac{\sqrt{n}\sqrt{\sum x_i^2}}{|\sum x_i|}.$$
- For $x \in \partial\Omega$, that is $\sum x_i = 0$, we have $\kappa_f(x) = \infty$.

Thus, we have

$$\pi_f(x) = \frac{d(x, \{x : \sum x_i = 0\})}{\sqrt{\sum x_i^2}} = \frac{|\sum x_i|}{\sqrt{n}\sqrt{\sum x_i^2}} = \kappa_f(x, y)^{-1}.$$

Namely,

$$K_f(x) = \frac{\sqrt{n}\sqrt{\sum x_i^2}}{|\sum x_i|}. \tag{A.6}$$

A.5. Single, multiple precision. Let f be a round–off computable function, and let M be a BSS machine satisfying the definition of round–off computability above. This computation is single precision when for every $0 < \epsilon < 1$ there is a $\delta = \delta(\epsilon)$ such that any round–off machine (M, δ) attains relative precision ϵ for any input $x \in \Omega$, and such that

$$\delta \geq \frac{c_0 \epsilon}{K_f(x)^{c_2} \dim(x)^{c_3}} \tag{A.7}$$

for some positive constants c_0, c_2, c_3. This computation is multiple precision when there exists δ such that

$$\delta \geq \frac{c_0 \epsilon^{c_1}}{K_f(x)^{c_2} \dim(x)^{c_3}}, \tag{A.8}$$

for some $c_1 > 1$. We say that the computation is strictly multiple precision when it is multiple precision but not single precision.

Example A.12. The inductive, naive algorithm for computing the round–off computable function $f(x_1, \ldots, x_n) = x_1 \cdots x_n$ defined in $\mathbb{R}^\infty$ is single precision, from (A.3). The algorithm given in Example A.8 for computing the round–off computable function $f(x) = x_1 + \cdots + x_n$ defined in $\{x \in \mathbb{R}^\infty : \sum x_i \neq 0\}$ is single precision from (A.5) and (A.6).

A.6. Size of an input. In many practical problems, we want to specify an output precision ϵ. From our definition of round–off computable function, given $x \in \Omega$ and $0 < \epsilon < 1$ some $\delta(x, \epsilon)$ will exist guaranteeing the desired precision, although it may be very hard to compute this δ in some cases. Moreover, from (A.8), the number $K_f(x)$ will in general play a role in the value of $\delta(x, \epsilon)$ needed for any machine solving the problem. This dependence suggests that maybe the input should be considered as (x, ϵ) and not just as x. These thoughts justify our definition of the size of an input, which includes a term related to ϵ and another related to $K_f(x)$:

$$\dim(x) + |\log \epsilon| + \log(K_f(x) + 1). \tag{A.9}$$

A.7. Cost of a computation. The cost of a computation on a round-off machine (M, δ) which outputs $\tilde{y}$ on input x is

$$T(x, \delta) \cdot \left(\max_i \dim(y^{(i)}) + |\log \delta| \right),$$

where $T(x, \delta)$ is the time for the computation to halt and

$$x = y^{(0)}, \ldots, y^{T(x,\delta)} = \tilde{y}$$

are the different vectors computed by (M, δ) on input x.

We say that a function $f : \Omega \to \mathbb{R}^\infty$ is polynomial cost computable if there exists a BSS machine M such that for every $x \in \Omega$ and $0 < \epsilon < 1$ there exists $\delta(x, \epsilon)$ such that any round–off machine $(M, \delta(x, \epsilon))$ computes $\tilde{y}$ which equals $f(x)$ to relative error ϵ, with cost polynomially bounded by the input size (A.9).

The most important cases of polynomial cost computability will be in the cases where we restrict the space of functions to single (multiple) precision functions, for example in the case of single precision to the definition of polynomial cost we add the restriction that $\delta(x, \epsilon)$ must satisfy (A.7). These two possibilities (single or multiple precision) will give us two theories, both of which deserve to be worked out.

Now that we have the notion of polynomial cost the classes P and NP may be defined and the problem: Does $P = NP$? stated.

References

[1] D. Armentano. Complexity of path-following methods for the eigenvalue problem . To appear.

[2] D. Armentano. PH. D. Thesis. Universidad de la República, Uruguay, and Université Paul Sabatier, France.

[3] Diego Armentano, Carlos Beltrán, and Michael Shub, *Minimizing the discrete logarithmic energy on the sphere: the role of random polynomials*, Trans. Amer. Math. Soc. **363** (2011), no. 6, 2955–2965, DOI 10.1090/S0002-9947-2011-05243-8. MR2775794 (2012f:31009)

[4] D. Armentano, M. Shub. Smale's fundamental theorem of algebra reconsidered. To appear in Foundations of Computational Mathematics. DOI: 10.1007/s10208-013-9155-y.

[5] Steve Batterson, *Convergence of the Francis shifted QR algorithm on normal matrices*, Linear Algebra Appl. **207** (1994), 181–195, DOI 10.1016/0024-3795(94)90010-8. MR1283957 (95h:65028)

[6] Steve Batterson and John Smillie, *Rayleigh quotient iteration fails for nonsymmetric matrices*, Appl. Math. Lett. **2** (1989), no. 1, 19–20, DOI 10.1016/0893-9659(89)90107-9. MR989851 (90b:65086)

[7] D. J. Bates, J. D. Hauenstein, A. J. Sommese, and C. W. Wampler. Bertini: software for numerical algebraic geometry. Available at http://www.nd.edu/∼sommese/bertini.

[8] Carlos Beltrán, *A continuation method to solve polynomial systems and its complexity*, Numer. Math. **117** (2011), no. 1, 89–113, DOI 10.1007/s00211-010-0334-3. MR2754220 (2011m:65102)

[9] C. Beltrán. The state of the art in Smale's 7–th problem. In Foundations of Computational Mathematics, Budapest 2011. London Mathematical Society. Lecture notes series 403. F. Cucker, T. Krick, A. Pinkus, A. Szanto editors.

[10] Carlos Beltrán, Jean-Pierre Dedieu, Gregorio Malajovich, and Mike Shub, *Convexity properties of the condition number*, SIAM J. Matrix Anal. Appl. **31** (2009), no. 3, 1491–1506, DOI 10.1137/080718681. MR2587788 (2011c:65071)

[11] Carlos Beltrán, Jean-Pierre Dedieu, Gregorio Malajovich, and Mike Shub, *Convexity properties of the condition number II*, SIAM J. Matrix Anal. Appl. **33** (2012), no. 3, 905–939, DOI 10.1137/100808885. MR3023457

[12] Carlos Beltrán and Anton Leykin, *Certified numerical homotopy tracking*, Exp. Math. **21** (2012), no. 1, 69–83, DOI 10.1080/10586458.2011.606184. MR2904909

[13] Carlos Beltrán and Anton Leykin, *Robust Certified Numerical Homotopy Tracking*, Found. Comput. Math. **13** (2013), no. 2, 253–295, DOI 10.1007/s10208-013-9143-2. MR3032682

[14] Carlos Beltrán and Luis Miguel Pardo, *On Smale's 17th problem: a probabilistic positive solution*, Found. Comput. Math. **8** (2008), no. 1, 1–43, DOI 10.1007/s10208-005-0211-0. MR2403529 (2009h:65082)

[15] Carlos Beltrán and Luis Miguel Pardo, *Smale's 17th problem: average polynomial time to compute affine and projective solutions*, J. Amer. Math. Soc. **22** (2009), no. 2, 363–385, DOI 10.1090/S0894-0347-08-00630-9. MR2476778 (2009m:90147)

[16] Carlos Beltrán and Luis Miguel Pardo, *Fast linear homotopy to find approximate zeros of polynomial systems*, Found. Comput. Math. **11** (2011), no. 1, 95–129, DOI 10.1007/s10208-010-9078-9. MR2754191 (2011m:65111)

[17] Carlos Beltrán and Michael Shub, *Complexity of Bezout's theorem. VII. Distance estimates in the condition metric*, Found. Comput. Math. **9** (2009), no. 2, 179–195, DOI 10.1007/s10208-007-9018-5. MR2496559 (2010f:65100)

[18] Carlos Beltrán and Michael Shub, *A note on the finite variance of the averaging function for polynomial system solving*, Found. Comput. Math. **10** (2010), no. 1, 115–125, DOI 10.1007/s10208-009-9054-4. MR2591841 (2011b:65075)

[19] Carlos Beltrán and Michael Shub, *On the geometry and topology of the solution variety for polynomial system solving*, Found. Comput. Math. **12** (2012), no. 6, 719–763, DOI 10.1007/s10208-012-9134-8. MR2989472

[20] Lenore Blum, Felipe Cucker, Michael Shub, and Steve Smale, *Complexity and real computation*, Springer-Verlag, New York, 1998. With a foreword by Richard M. Karp. MR1479636 (99a:68070)

[21] Lenore Blum, Mike Shub, and Steve Smale, *On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines*, Bull. Amer. Math. Soc. (N.S.) **21** (1989), no. 1, 1–46, DOI 10.1090/S0273-0979-1989-15750-9. MR974426 (90a:68022)

[22] Paola Boito and Jean-Pierre Dedieu, *The condition metric in the space of rectangular full rank matrices*, SIAM J. Matrix Anal. Appl. **31** (2010), no. 5, 2580–2602, DOI 10.1137/08073874X. MR2740622 (2012e:65078)

[23] Allan Borodin and Ian Munro, *The computational complexity of algebraic and numeric problems*, American Elsevier Publishing Co., Inc., New York-London-Amsterdam, 1975. Elsevier Computer Science Library; Theory of Computation Series, No. 1. MR0468309 (57 #8145)

[24] J. S. Brauchart, *Optimal logarithmic energy points on the unit sphere*, Math. Comp. **77** (2008), no. 263, 1599–1613, DOI 10.1090/S0025-5718-08-02085-1. MR2398782 (2010e:31004)

[25] M. Braverman. On the complexity of real functions, FOCS 2005.

[26] Mark Braverman and Stephen Cook, *Computing over the reals: foundations for scientific computing*, Notices Amer. Math. Soc. **53** (2006), no. 3, 318–329. MR2208383 (2006m:68019)

[27] Peter Bürgisser and Felipe Cucker, *On a problem posed by Steve Smale*, Ann. of Math. (2) **174** (2011), no. 3, 1785–1836, DOI 10.4007/annals.2011.174.3.8. MR2846491

[28] P. Bürgisser and F. Cucker. *Condition: The Geometry of Numerical Algorithms*. Grundlehren der mathematischen Wissenschaften, 349. ISBN-10:3642388957 — ISBN-13: 978-3642388958.

[29] Felipe Cucker and Steve Smale, *Complexity estimates depending on condition and round-off error*, J. ACM **46** (1999), no. 1, 113–184, DOI 10.1145/300515.300519. MR1692497 (2000f:68040)

[30] F. Cucker and J.-P. Dedieu, *Decision problems and round-off machines*, Theory Comput. Syst. **34** (2001), no. 5, 433–452. MR1862890 (2002h:68050)

[31] Jean-Pierre Dedieu, Gregorio Malajovich, and Michael Shub, *Adaptive step-size selection for homotopy methods to solve polynomial equations*, IMA J. Numer. Anal. **33** (2013), no. 1, 1–29, DOI 10.1093/imanum/drs007. MR3020948

[32] Jean-Pierre Dedieu and Mike Shub, *Multihomogeneous Newton methods*, Math. Comp. **69** (2000), no. 231, 1071–1098 (electronic), DOI 10.1090/S0025-5718-99-01114-X. MR1752092 (2000m:65072)

[33] James Weldon Demmel, *On condition numbers and the distance to the nearest ill-posed problem*, Numer. Math. **51** (1987), no. 3, 251–289, DOI 10.1007/BF01400115. MR895087 (88i:15014)

[34] A. Dubickas, *On the maximal product of distances between points on a sphere*, Liet. Mat. Rink. **36** (1996), no. 3, 303–312, DOI 10.1007/BF02986850 (English, with English and Lithuanian summaries); English transl., Lithuanian Math. J. **36** (1996), no. 3, 241–248 (1997). MR1455810 (98e:52015)

[35] C. Eckart and G. Young. The approximation of one matrix by another of lower rank, Psychometrika 1 (1936), 211-218.

[36] Misha Gromov, *Metric structures for Riemannian and non-Riemannian spaces*, Progress in Mathematics, vol. 152, Birkhäuser Boston Inc., Boston, MA, 1999. Based on the 1981 French original [MR0682063 (85e:53051)]; With appendices by M. Katz, P. Pansu and S. Semmes; Translated from the French by Sean Michael Bates. MR1699320 (2000d:53065)

[37] G.H. Hardy, J.E. Littlewood, G. Pólya. Inequalities, Cambridge University Press, 1934.

[38] Nicholas J. Higham, *Accuracy and stability of numerical algorithms*, 2nd ed., Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2002. MR1927606 (2003g:65064)

[39] M.H. Kim. Computational complexity of the Euler type algorithms for the roots of complex polynomials. PhD thesis, The City University of New York, 1985.

[40] Myong-Hi Kim, *On approximate zeros and rootfinding algorithms for a complex polynomial*, Math. Comp. **51** (1988), no. 184, 707–719, DOI 10.2307/2008771. MR958638 (90f:65073)

[41] T. L. Lee, T. Y. Li, and C. H. Tsai. Hom4ps-2.0: A software package for solving polynomial systems by the polyhedral homotopy continuation method. Available at http://hom4ps.math.msu.edu/HOM4PS_soft.htm.

[42] Anton Leykin, *Numerical algebraic geometry*, J. Softw. Algebra Geom. **3** (2011), 5–10. MR2881262

[43] Gregorio Malajovich, *Nonlinear equations*, Publicações Matemáticas do IMPA. [IMPA Mathematical Publications], Instituto Nacional de Matemática Pura e Aplicada (IMPA), Rio de

Janeiro, 2011. With an appendix by Carlos Beltrán, Jean-Pierre Dedieu, Luis Miguel Pardo and Mike Shub; 28º Colóquio Brasileiro de Matemática. [28th Brazilian Mathematics Colloquium]. MR2798351 (2012j:65148)

[44] C.W. Pfrang, P. Deift and G. Menon. How long does it take to compute the eigenvalues of a random symmetric matrix? arXiv 1203.4635.

[45] E. A. Rakhmanov, E. B. Saff, and Y. M. Zhou, *Minimal discrete energy on the sphere*, Math. Res. Lett. **1** (1994), no. 6, 647–662. MR1306011 (96e:78011)

[46] James Renegar, *On the cost of approximating all roots of a complex polynomial*, Math. Programming **32** (1985), no. 3, 319–336, DOI 10.1007/BF01582052. MR796429 (87a:65083)

[47] James Renegar, *On the worst-case arithmetic complexity of approximating zeros of polynomials*, J. Complexity **3** (1987), no. 2, 90–113, DOI 10.1016/0885-064X(87)90022-7. MR907192 (89a:68107)

[48] James Renegar, *On the worst-case arithmetic complexity of approximating zeros of systems of polynomials*, SIAM J. Comput. **18** (1989), no. 2, 350–370, DOI 10.1137/0218024. MR986672 (90j:68021)

[49] James Renegar, *Incorporating condition measures into the complexity theory of linear programming*, SIAM J. Optim. **5** (1995), no. 3, 506–524, DOI 10.1137/0805026. MR1344668 (96c:90048)

[50] Michael Shub, *Some remarks on Bezout's theorem and complexity theory*, From Topology to Computation: Proceedings of the Smalefest (Berkeley, CA, 1990), Springer, New York, 1993, pp. 443–455. MR1246139 (95a:14002)

[51] Michael Shub, *Complexity of Bezout's theorem. VI. Geodesics in the condition (number) metric*, Found. Comput. Math. **9** (2009), no. 2, 171–178, DOI 10.1007/s10208-007-9017-6. MR2496558 (2010f:65103)

[52] Michael Shub and Steve Smale, *Complexity of Bézout's theorem. I. Geometric aspects*, J. Amer. Math. Soc. **6** (1993), no. 2, 459–501, DOI 10.2307/2152805. MR1175980 (93k:65045)

[53] M. Shub and S. Smale, *Complexity of Bezout's theorem. II. Volumes and probabilities*, Computational algebraic geometry (Nice, 1992), Progr. Math., vol. 109, Birkhäuser Boston, Boston, MA, 1993, pp. 267–285. MR1230872 (94m:68086)

[54] Michael Shub and Steve Smale, *Complexity of Bezout's theorem. III. Condition number and packing*, J. Complexity **9** (1993), no. 1, 4–14, DOI 10.1006/jcom.1993.1002. Festschrift for Joseph F. Traub, Part I. MR1213484 (94g:65152)

[55] Michael Shub and Steve Smale, *Complexity of Bezout's theorem. IV. Probability of success; extensions*, SIAM J. Numer. Anal. **33** (1996), no. 1, 128–148, DOI 10.1137/0733008. MR1377247 (97k:65310)

[56] M. Shub and S. Smale, *Complexity of Bezout's theorem. V. Polynomial time*, Theoret. Comput. Sci. **133** (1994), no. 1, 141–164, DOI 10.1016/0304-3975(94)90122-8. Selected papers of the Workshop on Continuous Algorithms and Complexity (Barcelona, 1993). MR1294430 (96d:65091)

[57] Steve Smale, *The fundamental theorem of algebra and complexity theory*, Bull. Amer. Math. Soc. (N.S.) **4** (1981), no. 1, 1–36, DOI 10.1090/S0273-0979-1981-14858-8. MR590817 (83i:65044)

[58] Steve Smale, *Newton's method estimates from data at one point*, computational mathematics (Laramie, Wyo., 1985), Springer, New York, 1986, pp. 185–196. MR870648 (88e:65076)

[59] S. Smale. The fundamental theorem of algebra and complexity theory, SIAM Rev. 32 (1990), no. 2, 211–220.

[60] Steve Smale, *Mathematical problems for the next century*, Mathematics: frontiers and perspectives, Amer. Math. Soc., Providence, RI, 2000, pp. 271–294. MR1754783 (2001i:00003)

[61] G. W. Stewart and Ji Guang Sun, *Matrix perturbation theory*, Computer Science and Scientific Computing, Academic Press Inc., Boston, MA, 1990. MR1061154 (92a:65017)

[62] Lloyd N. Trefethen and David Bau III, *Numerical linear algebra*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997. MR1444820 (98k:65002)

[63] J. Verschelde. Algorithm 795: PHCpack: A general-purpose solver for polynomial systems by homotopy continuation. ACM Trans. Math. Softw., 25 (1999), no. 2, 251–276. Available at http://www.math.uic.edu/∼jan.

[64] Gerold Wagner, *On the product of distances to a point set on a sphere*, J. Austral. Math. Soc. Ser. A **47** (1989), no. 3, 466–482. MR1018975 (90j:11080)

[65] Qi Zhong, *Energies of zeros of random sections on Riemann surfaces*, Indiana Univ. Math. J. **57** (2008), no. 4, 1753–1780, DOI 10.1512/iumj.2008.57.3329. MR2440880 (2009k:58051)
[66] Steve Zelditch and Qi Zhong, *Addendum to "Energies of zeros of random sections on Riemann surfaces". Indiana Univ. Math. J.* **57** *(2008), No. 4, 1753–1780 [MR 2440880]*, Indiana Univ. Math. J. **59** (2010), no. 6, 2001–2005, DOI 10.1512/iumj.2010.59.59073. MR2919745

Depto. de Matemáticas, Estadística y Computación, Universidad de Cantabria, Santander, Spain.
E-mail address: carlos.beltran@unican.es

CONICET, IMAS, Universidad de Buenos Aires, Argentina and CUNY Graduate School, New York, NY, USA.
E-mail address: shub.michael@gmail.com

Contemporary Mathematics
Volume **604**, 2013
http://dx.doi.org/10.1090/conm/604/12070

Multiplicity hunting and approximating multiple roots of polynomial systems

M. Giusti and J.-C. Yakoubsohn

Abstract. The computation of the multiplicity and the approximation of isolated multiple roots of polynomial systems is a difficult problem. In recent years, there has been an increase of activity in this area. Our goal is to translate the theoretical background developed in the last century on the theory of singularities in terms of computation and complexity. This paper presents several different views that are relevant to address the following issues: predict the multiplicity of a root and/or determine the number of roots in a ball, approximate fast a multiple root and give complexity results for such problems. Finally, we propose a new method to determine a regular system, called equivalent but deflated, i.e., admitting the same root as the initial singular one.

1. Introduction

Let $x \in \mathbf{C}^n$ and $f(x) = (f_1(x), \ldots, f_m(x)) \in \mathbf{C}[x]^m$. We denote by I the ideal generated by f. A multiple isolated root w of $f(x)$ is by definition the only root w of $f(x)$ in a certain ball at which its Jacobian matrix $Df(w)$ is not full rank. We use equally in the text singular root and multiple root. It is well known that the quadratic convergence of the Newton's method is lost in the neighbourhood of a multiple root. From starting points close to such roots, Newton's method is found to converge linearly or to diverge. For example the behaviour of the Newton sequence associated to the system $x - y^2 = 0$, $2cy^3 - 2xy = 0$ studied by Griewank and Osborne in **[23]** close to the root $(0, 0)$ of multiplicity 3 depends on the parameter c. For $c = 5/32$ there is linear convergence and for $c = 29/32$ we can observe

2010 *Mathematics Subject Classification.* Primary .

This work has been supported by the French ANR-10-BLAN 0109 and DIGITEO DIM 2009-36 HD "MAGIX"..

divergence (see Fig. 1 and Fig. 2).

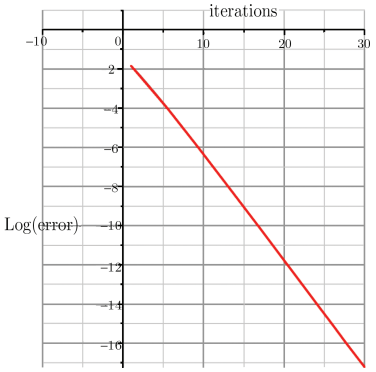


Fig. 1. Linear convergence of Newton sequence from $(0.1, -0.2)$ with $c = 5/32$.

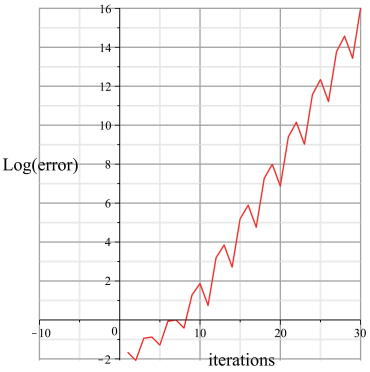


Fig. 2. Divergence of Newton sequence from $(0.1, -0.2)$ with $c = 29/32$.

Our purpose is to recover this quadratic convergence. In the example above, it is easy to determine a regular system admitting the same root as the initial one (we say an equivalent system). For that we remark the gradient of $2cy^3 - 2xy$ is zero at $(0, 0)$. Hence we can replace the polynomial $2cy^3 - 2xy$ by the two partial derivatives : y and $3cy^2 - x$. It turns out that the system $(x - y^2, y, 3cy^2 - x)$ is now regular at $(0, 0)$. We will develop this idea in section 6 to propose a new method to compute an equivalent system. More formally, from the initial system we compute a sequence of systems and stop when appears a regular system. A step in this iterative method consists of two operations called respectively *deflating* and *kerneling* [**42**]. The deflating operation replaces the polynomials by their gradient when the latter vanishes at the root. After the deflating operation we have ensured that all the rows of the Jacobian matrix evaluated at the root are non-zero. If this Jacobian matrix is not full rank, the kerneling operation consists to add the numerators of coefficients of a formal Schur complement of this Jacobian matrix. The multiplicity of the root obtained after a step decreases in the number of distinct polynomials added by the deflating and kerneling operations.

The goal of hunting the multiplicity is ambitious. This is a long standing challenge in many areas as optimization, dynamical systems, computer algebra and numerical algorithms dealing with polynomial or analytic systems. The univariate case is well understood : the Taylor series is a useful tool to describe the multiplicity of a root. For instance two iterations of Newton's method close to a multiple root are enough to predict the multiplicity. In fact the Newton sequence converges to the multiple root following a quasi straight line. More precisely, if $N_f(x) = x - \dfrac{f(x)}{f'(x)}$ is the Newton operator associated to a univariate function f, the iterate $x_{k+1} = N_f(x_k)$, $(k \geq 0)$, defining the Newton sequence starting from an initial point x_0, it is easy to see that

$$x_{k+1} - w = \left(1 - \frac{1}{m}\right)^k (x_0 - w) + O((x_0 - w)^2), \quad k \geq 0.$$

Schröder points out in [**51**] that the quadratic convergence is recovered using the generalized Newton operator

$$S_{f,m}(x) = x - m\frac{f(x)}{f'(x)}.$$

This has been hugely studied in the literature see Ostrowski [**45**], Rall [**47**], Householder [**26**], Traub [**59**]. α-theory in the spirit of Smale [**55**] for multiple roots in the univariate case has been done by Giusti-Lecerf-Salvy-Yakoubsohn in [**18**] and Yakoubsohn in [**62**], [**63**] : the links between Rouché's theorem and Schröder-Newton's method for multiple roots are precisely studied. To sum up, the order of Taylor series at the neighbourhood of the root defines the multiplicity in the univariate case. But unfortunately, Taylor series are not sufficient to determine the multiplicity in the multivariate case. In order to recover the quadratic convergence, the behaviour of Newton's method has been extensively investigated by Reddien [**48**], [**49**], Decker-Keller-Kelley in [**12**], [**13**], [**11**], Griewank in [**20**], [**21**], Griewank-Osborne in [**22**], [**23**], Rabier-Reddien [**46**]. These papers give characterizations of certain singular points and assumptions to get convergence. Sometimes the authors propose modifications to accelerate the convergence. In areas other than numerical analysis, the question of the multiplicity theory has also been intensively studied. There are many different way to introduce the concept of multiple root but, this is a more complicated matter than it is in one dimension : this requires background from algebra and analysis. The elimination theory provides algebraic objects like standard bases and the introduction of local rings reduces the multiplicity to the dimension of a quotient space. From an algebraic point of view, Fulton [**16**] chapter 7 gives a more general framework and explain different approaches. Milnor in Appendix B of [**37**] defines the multiplicity as the degree of a certain map. Using a similar approach Arnold, Varchenko, Gusein-Zade [**5**] rely the multiplicity to the index of a holomorphic germ. Another presentation is treated by Aizenberg and Yuzhakov in [**1**] where the multiplicity is defined via a perturbation of an analytic map. This last definition is directly linked to homotopy continuation methods which can be a reliable and an efficient way to numerically approximate isolated roots. After these theoretical studies on the multiplicity, we don't forget the heuristic book of Stetter, *Numerical Polynomial Algebra*, [**57**] and especially the chapter nine including the work of Thallinger.

The paper is organized as follows, first a survey part: in section 2 we present the algebraic geometric point of view on the multiplicity. Next, via the notion of duality, we give relationship to linear algebra where the multiplicity appears as the dimension of the kernel of a Macaulay matrix. In section 3, we explain how the multiplicity comes numerically from Rouché's theorem and recall some results. We also state an open problem concerning an efficient Rouché's theorem. In section 4, we justify why the homotopy methods work in the regular case and discuss the complexity of the linear homotopy in the singular case. The section 5 is devoted to describe the theoretical background of some deflation methods which are implemented in ApaTools of Zhonggang Zeng (recently upgraded to NAClab) `http://www.neiu.edu/~zzeng/NAClab.html` [**64**] and, PHCpack of Jan Verschelde `http://www.math.uic.edu/~jan/download.html` [**60**].

Section 6 is original. We propose a new way to determine an equivalent regular system from an initial singular system. We end by examples to show how this new method works.

2. Multiplicity. Algebraic geometric point of view

This theoretical material belongs to folklore. An exposition can be found e.g. in Cox, Little, O'Shea in **[8]**, among others.

2.1. Number of roots and dimension. Let $x = (x_1, \ldots, x_n) \in \mathbf{C}^n$ and I be the ideal generated by the polynomials $f_1(x), \ldots f_m(x)$ of $\mathbf{C}[x]$. The first question is the number of isolated roots of a polynomial system. This is given by the following Bézout's Theorem which is the equivalent of the fundamental theorem of algebra for univariate polynomials:

THEOREM 1. *The number of isolated roots of a polynomial system is less than the product of degrees of each polynomial.*

We refer to Heintz **[25]** for a proof using the dimension theory. Evidently the bound of theorem 1 is reached. If $V(I)$ means the variety associated to I then the following theorem gives a necessary and sufficient condition for $V(I)$ to be a set of isolated points. In this case the cardinal of $V(I)$ is the dimension of a quotient space. More precisely :

THEOREM 2. *Under the previous notations we have :*

1– *The dimension of $\mathbf{C}[x]/I$ is finite if and only if the dimension of $V(I)$ is zero.*

2– *In the finite dimension case we have :*

$$\dim \mathbf{C}[x]/I \geq \#V(I)$$

where $\#V(I)$ is the number of distinct points of $V(I)$. This equality holds if and only if the ideal I is radical.

In fact we will see below that when the ideal I is not radical we can associate a multiplicity at each point of $V(I)$ so that the sum of multiplicities equals the dimension of $C[x]/I$. A way to determine $\dim \mathbf{C}[x]/I$ is to compute a Gröbner basis of the ideal I.

THEOREM 3. *Let G a Gröbner basis of an ideal I. Let $LT(G)$ the ideal generated by the leading terms of G. Define $SM(G) = \{$ monomials $\notin LT(G)\ \}$. Then*

$$\dim \mathbf{C}[x]/I = \#SM(G).$$

EXAMPLE 1. *Let $f_1(x, y) = x^2 + x^3$, $f_2(x, y) = x^3 + y^2$. Then $V(I) = \{(0, 0), (-1, 1), (-1, -1)\}$. Let us choose the lexicographic ordering induced by $x > y$; the leading term is the Sup. A Gröbner basis of I is $\{y^4 - y^2, xy^2 + y^2, x^2 - y^2\}$ and $SM(G) = \{1, x, y, y^2, y^3, xy\}$. We deduce $\dim \mathbf{C}[x]/I = 6$. We will see that the root $(0, 0)$ has multiplicity 4.* ∘

Some computer algebra systems compute Gröbner bases, among them Maple, Magma, Singular. For instance, most classical algorithms are implemented in Maple.

2.2. Multiplicity and dimension. A way to define the multiplicity at a point of $w = (w_1, \ldots, w_n) \in V(I)$ is to consider the local ring $\mathbf{C}\{x - w\}$ of convergent series in n variables with the maximal ideal generated by $x_1 - w_1, \ldots x_n - w_n$. We denote by $I\mathbf{C}\{x - w\}$ the ideal generated by I in $\mathbf{C}\{x - w\}$. Finally we consider the local quotient space $A_w = \mathbf{C}\{x - w\}/I\mathbf{C}\{x - w\}$. The link between the local

quotient spaces associated to points of $V(I)$ and the quotient space $C[x]/I$ is given by the :

THEOREM 4. *Let $V(I) = \{w^{(1)}, \ldots, w^{(N)}\}$. Then*

1– $\mathbf{C}[x]/I \sim A_{w^{(1)}} \times \ldots \times A_{w^{(N)}}$.

2– $\dim \mathbf{C}[x]/I = \sum_{i=1}^{N} \dim A_{w^{(i)}}$.

We then can define the algebraic multiplicity.

DEFINITION 1. *Let $w \in V(I)$. The dimension of local quotient space A_w is the algebraic multiplicity of w.*

To determine the dimension of A_w, a similar way to the affine global setting is to compute a standard basis of A_w. We then have an equivalent result to the theorem 3.

THEOREM 5. *Let S a standard basis of the ideal $I\mathbf{C}\{x - w\}$. Let $LT(S)$ the ideal generated by the leading terms of S. Define $SM(S) = \{\mathsf{monomials} \notin LT(S)\ \}$. Then*

$$\dim A_w = \#SM(S).$$

EXAMPLE 2. *Let $f_1(x, y)$ and $f_2(x, y)$ be as the example 1. We are interested first in the root $(0, 0)$. Let us choose an ordering refining the valuation; the leading term will be the Inf. A standard basis of $I\mathbf{C}\{(x, y)\}$ is $S = \{x^2, y^2\}$. Hence $SM(S) = \{1, x, y, xy\}$ and $\dim A_{(0,0)} = 4$.*

In the same way a standard basis of $I\mathbf{C}\{(x, y) - (-1, 1)\}$ (respectively $I\mathbf{C}\{(x, y) - (-1, -1)\}$) is $S = \{x, y\}$. Hence $SM(S) = \{1\}$ and $\dim A_{(-1,1)} = \dim A_{(-1,-1)} = 1$. The identity $\dim \mathbf{C}[x]/I = \dim A_{(0,0)} + \dim A_{(-1,1)} + \dim A_{(-1,-1)}$ is satisfied. ∘

The tangent cone algorithm [**38**] allows to compute standard bases. An improved version of this algorithm is implemented in Singular by Greuel and Pfister [**19**].

2.3. Multiplicity and Duality. The link between multiplicity and duality is described first by Macaulay in [**34**] and perhaps also Gröbner [**24**]. A modern exposition is done by Emsalem [**15**]. More recent developments are given by Marinara, Möller, Mora in [**36**], Alonso, Marinari, Mora in [**3**], [**4**]. Also improvements concerning complexity are proposed by Mantzaflaris, Mourrain [**35**], [**41**]. For a multiple index $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbf{N}^n$, we denote by ∂^α the differential operator $g \to \dfrac{\partial^\alpha g(x)}{\partial x^\alpha}$. The operator ∂_w^α is the evaluation operator of ∂^α at a point w of $\mathbf{C}^n$. Also, if $L = \sum_{|\alpha| \le k} L_\alpha \partial^\alpha$ then $L_w = \sum_{|\alpha| \le k} L_\alpha \partial_w^\alpha$.

It is classical that there is an isomorphism between the dual space $\mathbf{C}[x]^*$ of $\mathbf{C}[x]$ and the set of formal series in ∂_w. Macaulay in [**34**] introduce the inverse system of the ideal I

$$I^\perp = \{L \in \mathbf{C}[x]^* \ : \ \forall g \in I,\ L(g) = 0\}$$

The result is that we can identify $I^\perp$ and the dual of $\mathbf{C}[x]/I$:

THEOREM 6. *There is a canonical $\mathbf{C}$-isomorphism between $I^\perp$ and the dual of $\mathbf{C}[x]/I$.*

The link between the duality and the multiplicity is explained by the relation between the quotient rings A_w and the subspaces

$$D_w^k(I) = \{L = \sum_{|\alpha|\leq k} L_\alpha \partial^\alpha \ : \ \forall g \in I,\ L_w(g) = 0\}.$$

We will write D_w^k for $D_w^k(I)$. We have :

THEOREM 7. *A root w of f is isolated if and only if there exists an integer δ satisfying $\mathcal{D}_w^{\delta-1} = \mathcal{D}_w^\delta$. In this case D_w^δ is the dual space of A_w and the dimension of $\mathcal{D}_w^\delta$ is equal to the multiplicity of w. In other words*

$$\dim A_w = \dim D_w^\delta.$$

We call δ the thickness of the multiple root w.

REMARK 1.

We adopt the term *thickness* which is the translation of the french word *épaisseur* introduced by Ensalem in [**15**] rather than the term *depth* more recently used by Mourrain, Matzaflaris in [**35**] or Dayton, Li, Zeng [**10**], [**9**]. ∘

To compute the dimension of the vector space D_w^k, let us introduce the Macaulay matrices

$$S_k = (\ \partial_\alpha[w]((x-w)^\alpha f_i(x))\ )_{\substack{|\alpha|\leq k-1 \\ 1\leq i\leq m}}$$

THEOREM 8. *The vector space D_w^k is isomorphic to the kernel of S_k.*

Consequently the multiplicity μ of w satisfies $\mu = \dim Ker(S_{\delta-1}) = \dim Ker(S_\delta)$.

EXAMPLE 3. *Let $f_1 = x^2 + y^2 - 2$, $f_2 = xy - 1$. $w = (1,1)$. Let us construct the Macaulay matrices in $w = (1,1)$:*

		∂_{00}	∂_{10}	∂_{01}	∂_{20}	∂_{11}	∂_{02}
S_0	f_1	0 |	2	2 |	2	0	2
S_1	f_2	0 |	1	1 |	0	1	0
	− − −	−	−	−			
	$(x-1)f_1$	0	0	0	4	2	0
S_2	$(x-1)f_2$	0	0	0	2	1	0
	$(y-1)f_1$	0	0	0	0	2	4
	$(y-1)f_2$	0	0	0	0	1	2

We have successively $\mathsf{rank}(S_0) = 0$, $\mathsf{rank}(S_1) = 1$, $\mathsf{rank}(S_2) = 4$. Hence $\mathsf{corank}(S1) = \mathsf{corank}(S_2) = 2$. It follows the multiplicity of $(1,1)$ is 2. ∘

We now explain how the knowledge of the structure of the dual space permits to find a regular system at w. Let μ the dimension of D_w^k and $\Lambda = \{\Lambda_1, \ldots, \Lambda_\mu\}$ a basis of D_w^k. We introduce the polynomial system of $m\mu$ equations and n variables :

$$\Lambda(f) = (\Lambda_1(f), \ldots, \Lambda_\mu(f))$$

with $\Lambda_k(f) = (\Lambda_k(f_1), \ldots, \Lambda_k(f_m))$. Mantzaflaris and Mourrain state the following :

THEOREM 9 ([**35**]). *The polynomial system $\Lambda(f)$ is regular at w.*

EXAMPLE 4. *A basis of the kernel of the Macaulay matrix S_2 of the example 3 is*

$\{(1,0,0,0,0,0),(0,1,-1,0,0,0)\}$. *Hence the set* $\{\partial^{(0,0)}, \partial^{(1,0)} - \partial^{(0,1)}\}$ *is a basis of* $D^2_{(1,1)}$. *Consequently*

$$\Lambda(f_1, f_2) = (x^2 + y^2 - 2, xy - 1, 2x - 2y, y - x).$$

It is easy to see the Jacobian of $\Lambda(f_1, f_2)$ *has rank* 2.

3. Multiplicity. Numerical point of view

3.1. Multiplicity and perturbation. From a numerical point of view an exact multiple root makes no sense. We must think of a cluster of roots which comes from perturbations of the data. In this way we can consider the initial system as close to another system which admits an exact multiple root.

DEFINITION 2. *A root* w *of* $f = (f_1, \ldots, f_m)$ *is regular if the Jacobian matrix* $Df(w)$ *has full rank (in the opposite case* w *is a singular root).*

The link to the algebraic multiplicity is given by the following.

PROPOSITION 1. *The algebraic multiplicity of a regular root is equal to* 1.

PROOF. We denote by $Df(w)^*$ the adjoint of $Df(w)$. Let I the ideal generated by f. Since $Df(w)$ has full rank $Df(w)^*Df(w)$ is invertible. Hence the ideal generated by $g(x) = (Df(w)^*Df(w))^{-1}f(x)$ is equal to I. But

$$(Df(w)^*Df(w))^{-1}f(x) = x - w + \sum_{k\geq 2} \frac{1}{k!}(Df(w)^*Df(w))^{-1}D^k f(w)(x-w)^k.$$

Consequently $LT(g)$ is generated by $x - w$. Its follows that $\dim A_w = 1$. □

A very useful result is the Rouché's theorem **[50]** which links a perturbation of analytic functions to the number of roots in a ball, see also Lojasiewicz for a version in several variables **[33]**.

THEOREM 10. *Let* f *and* g *two analytic functions defined in a real ball* $B(x,r) \subset \mathbf{C}^n$. *If for all* $z \in \partial B(x,r)$ *we have*

$$||f(z) - g(z)|| < ||f(z)||$$

then f *and* g *have the same number of roots in* $B(x,r)$ *where each root is counted as many times as its multiplicity.*

PROPOSITION 2. w *is a singular isolated root of* f *if and only if the multiplicity of* w *is strictly greater than 1.*

PROOF. Since w is an isolated root there exists a ball $B(w,r)$ where f admits only this root. There exists $z_0 \in \partial B(w,r)$ such that for all $z \in \partial B(w,r)$ one has $||f(z)|| \geq ||f(z_0)||$. Then the function $g(z) = f(z) + y$ with $||y|| < ||f(z_0)||/2$ satisfies the inequality of Rouché's theorem on $\partial B(w,r)$. Consequently the number of roots of g in $B(w,r)$ is the multiplicity, say μ, of w. Moreover for almost every y, Sard's theorem insures that $Dg(z)$ has full rank at each of the roots. Hence the roots of g, say $w^{(1)}, \ldots, w^{(\mu)}$, are regular in the ball $B(w,r)$. Let us consider the homotopy

$$h(z,t) = (1-t)g(z) + tf(z) = f(z) - (1-t)y.$$

We have $h(w^{(k)}, 0) = 0$ for every k and $h(w,1) = 0$. For almost every y, from implicit function theorem there exists μ regular curves $x^{(k)}(t)$: $[0,1[\to B(w,r)$ such

that $f(x^{(k)}(t)) = (1-t)y$ and $x^{(k)}(t)' = -Df(x^{(k)}(t))^{-1}y$. Hence if $\mu > 1$ the quantities $x^{(k)}(1)'$ make no sense and the root w is singular. □

The link between Rouché's theorem and the local ring theory can be summarized by the identity

$$\dim A_w^f = \sum_{\bar{w}\in B(w,r)\cap g^{-1}(0)} \dim A_{\bar{w}}^g$$

where A_w^f (respectively $A_{\bar{w}}^g$) is the local quotient ring associated to f (respectively g). Here we find again the classical idea from a numerical point of view that we deal with clusters of roots rather than exact multiple roots.

In the case where the system has no root or only one regular root in a ball, it is possible to give an effective version of Rouché's theorem : this is obtained from the Taylor series of f. It is also valid when the system f is analytic.

THEOREM 11 ([**17**]). *Let us consider a ball $B(x,r)$.*

1– *If*

$$||f(x)|| > \sum_{k\geq 1} \frac{1}{k!}||D^k f(x)||r^k$$

there is no root in $B(x,r)$.

2– *Let r be a positive real number smaller than the radius of convergence of $\sum_{k\geq 0} \frac{1}{k!}||D^k f(x)||r^k$. If*

$$||Df(x)^{-1}f(x)|| < r - \sum_{k\geq 2} \frac{1}{k!}||Df(x)^{-1}D^k f(x)||r^k$$

there is only one regular root of f in $B(x,r)$.

The case of a simple double root has been studied by Dedieu-Shub [**14**].

THEOREM 12. *Let $c = 0.19830\ldots$. For $v, x \in \mathbf{C}^n$, $||v|| = 1$, we define the linear operator :*

$$A(x,f,v) = Df(x) + \frac{1}{2}D^2 f(x)(v,\Pi_v)$$

where Π_v is the projection on the space spanned by v. Let L be the linear operator defined by $L(v) = Df(x)v$ and $L(w) = 0$ if w is orthogonal at v. Let $B(x,f,v) = A(x,f,v) - L$. We introduce the quantity

$$\gamma_2(f,x,v) = \max\left(1, \sup_{k\geq 2}\left|\left|\frac{1}{k!}B(f,x,v)^{-1}D^k f(x)\right|\right|^{\frac{1}{k-1}}\right).$$

If we have

$$||f(x)|| + ||Df(x)v||\frac{c}{2\gamma_2(f,x,v)^2} < \frac{c^3}{4||B(f,x,v)^{-1}||\gamma_2(f,x,v)^4}$$

then f has two zeros (counting multiplicities) in the ball of radius $\dfrac{c}{2\gamma_2(f,x,v)^2}$ around x.

In fact the previous case describes double roots of corank one : they are clusters of two roots of embedding dimension one. A quantitative version of Rouché's theorem in the embedding dimension 1 case is given by Giusti, Lecerf, Salvy, Yakoubsohn in [**17**] but, the statement is technically too difficult to appear here.

Open problem 1.

Find a qualitative version of Rouché's theorem for clusters of roots of analytic systems. ∘

Let us remark that the theorem 11 applied to the dual system $\Lambda(f)$ of theorem 9 can prove the existence of a (regular) root of $\Lambda(f)$.

4. Multiplicity and homotopy methods

Homotopy methods consist to deform smoothly a system with known roots to the initial system with unknown roots. These methods are currently used to solve systems of equations : the textbook of Allgower and Georg [**2**] or Morgan [**39**] are classical references. The homotopy used in this section is the linear homotopy h $:[0,1]\times\mathbf{C}^n\to\mathbf{C}^n$ defined by

$$h(x,t)=(1-t)g_{a,b}(x)+tf(x)$$

where $g_{a,b}(x)=(a_1x_1^{d_1}-b_1,\dots,a_nx_n^{d_n}-b_n)$. There are three kinds of curves $x(t)$ solutions of $h(t,x(t))=0$. First, the regular curves defined on $[0,1]$ which correspond to a regular root of $f(x)$. Next, the curves which are only regular on $[0,1[$ due to the existence of a multiple root of $f(x)$. Finally, the curves which go to infinity as $t\to 1$ and which correspond to infinite roots of $f(x)$. Infinite roots are explicitly described using complex projective space $\mathbf{CP}^n$. Wright in [**61**] give a proof of Bézout's theorem using the linear homotopy. More precisely

Theorem 13 ([**61**]). *Let* $F(x_0,x)=\left(x_0^{d_1}f_1(x/x_0),\dots,x_0^{d_n}f_n(x/x_0)\right)$, $G_{a,b}(x_0,x)=(a_1x_1^{d_1}-b_1x_0^{d_1},\dots,a_nx_n^{d_n}-b_nx_0^{d_n})$ *and*

$$H_{a,b}(t,x_0,x)=(1-t)G_{a,b}(x_0,x)+tF(x_0,x).$$

Let $Z_{a,b}=\{(t,x_0,x)\in[0,1[\times\mathbf{CP}^n\ :\ H_{a,b}(t,x_0,x)=0\}$. *For almost* $(a,b)\in\mathbf{C}^{2n}$ *we have :*

1– $0\in\mathbf{C}^n$ *is a regular value of* $H_{a,b}(t,1,x)=0$, *i.e,* $D_xH(t,1,x)$ *has full rank of for all* $(t,x)\in[0,1[\times\mathbf{C}^n$ *such that* $H_{a,b}(t,1,x)=0$.

2– $Z_{a,b}$ *consists of* $d_1\dots d_n$ *disjoint half-open arcs in* $\mathbf{CP}^n\times[0,1)$, *where the endpoint of each arc is a known root of* $G_{a,b}(x_0,x)$ *in* $\mathbf{CP}^n\times\{0\}$, *and where the limit of the other end of the arc is a root of* $F(x_0,x)$.

In fact linear homotopy methods are useful to prove Bézout's theorem : see Blum, Cucker, Shub, Smale [**7**] page 199 and references inside.

A straightforward consequence of this result is the multiplicity can be computed thanks to homotopy methods. More precisely

Corollary 1. *Let us consider the linear homotopy of the theorem* 13. *Each isolated root (respectively root at infinity) of multiplicity* μ *generates* μ *homotopy paths* $x(t)$ *converging towards it.*

To find one regular root, the complexity and the analysis of this homotopy method is studied by Shub and Smale in [**53**] and [**54**]. A better complexity bound is given by Shub [**52**]. We give a simplified version of this complexity result in the linear homotopy case.

THEOREM 14 ([**52**], [**6**]). *The number of numerical homotopy steps performed by the projective Newton's method to yield an approximate zero of the initial system is bounded by*

$$71d^{3/2}L$$

where d is the maximum of degrees of $f_i's$ and L is the condition length of the linear homotopy (see the references above for this definition).

The paper of T.Y Li [**32**] gives a good review on homotopy continuation methods and their improvement for deficient polynomial systems, i.e., for which the isolated solutions are fewer than the Bézout's number.

OPEN PROBLEM 2.

Estimate the complexity to approximate a multiple root using linear homotopy. ∘

In the chapter 10 of [**56**], Sommese and Wampler give some numerical heuristics to deal with *singular end games* based on power series, Cauchy integral and trace theorem. In the same vein, Huber and Verschelde in [**27**] explore links between *polyhedral end game* and power series to give some refinements. Another interesting way is proposed by Kobayashi, Suzuki and Sakai in [**28**] using Zeuthen's rule but unfortunately without study of complexity.

5. Recovering the quadratic convergence

The idea is to compute from the initial system another one which is regular at the singularity. The theorem 9 gives an augmented system computed from the kernel of the Macaulay matrices S_k. But the size of S_k is very huge i.e., $m\sum_{j=0}^{k}\binom{n+j-1}{j}\times\binom{n+k+1}{n}$. In the sequel, we describe two kinds of what is called a *deflation* method.

5.1. Lecerf deflation method. [**29**] The idea is to differentiate well chosen equations and to select new equations at each step of the method in order to obtain a regular system at the root w.

From now we adopt the Matlab notation : $x_{i:j}$ is the vector $(x_i,\dots x_j)$.

Initial Step : the system $f=(f_1,\dots,f_m)$ is considered as a subset of $\mathbf{C}\{x-w\}$. We set $\Phi_1=f$ and $R_1=1$.

Step $k\geq 1$. We compute a new system Φ_{k+1} and a new integer R_{k+1} from Φ_k and R_k. Let m_k be the valuation of Φ_k and

$$\tilde{\Phi}_k=\frac{\partial^{m_k-1}}{\partial x_{R_k}^{m_k-1}}\Phi_k:=\left\{\frac{\partial^j}{\partial x_{R_k}^j}\Phi_k\ :\ 1\leq j<m_k\right\}$$

Let r_k the rank of Jacobian of $\tilde{\Phi}_k$ with respect to the variables $x_{R_k:n}$ evaluated at $w_{R_k:n}$. Then we set $R_{k+1}=r_k+R_k$. Next we extract a subset Ω_k from $\tilde{\Phi}_k$ such that the gradient of Ω_k has rank r_k at $w_{R_k:n}$.

Finally, thanks to the implicit function theorem, there exist r_k power series $y_{R_k:R_{k+1}-1}$ in $\mathbf{C}\{x_{R_{k+1}:n}-w_{R_{k+1}:n}\}$ expressing $x_{R_k:R_{k+1}-1}$ in terms of $x_{R_{k+1}:n}$ such that $\Omega_k(y_{R_k:R_{k+1}-1},x_{R_{k+1}:n})=0$. Then

$$\Phi_{k+1}(x_{R_{k+1}:n})=\Phi_k(y_{R_k:R_{k+1}-1},x_{R_{k+1}:n}).$$

Stopping criterion. The above construction stops when $R_{k+1}=n+1$.

Output of the method. Let us suppose that there are ν steps. The output is the system $\Omega = (\Omega_1(x_{R_1:n}, \ldots, \Omega_\nu(x_{R_\nu:n}))$. The properties of this deflation sequence are given by

THEOREM 15. *Without loss of generality we can assume that at each step of the deflation process the variable x_{R_k} is in Weierstrass position with respect to the ideal generated by Φ_k (i.e. there exists an element of this ideal of valuation m_k having $x_{R_k}^{m_k}$ in its support). The construction above works up to a permutation of the variables. Moreover :*

1– $1 \leq r_k \leq n - R_k + 1$.
2– $1 \leq m_k \dim\left(\mathbf{C}\{x_{R_k:n} - w_{R_k:n}\}/\tilde{\Phi}_k\right) \leq \dim\left(\mathbf{C}\{x_{R_k:n} - w_{R_k:n}\}/\Phi_k\right)$.
3– *The system Ω is regular at the root w.*
4– $m_1 \ldots m_\mu \leq \dim A_w$.

EXAMPLE 5. *Let* $f := (f_1, f_2, f_3) = (x^2+x+y+z, y^2+y+x+z, z^2+z+x+y)$. *The root* $w = (0, 0, 0)$ *has multiplicity* 4.

We denote by O_k a generic power series $\sum_{|\alpha|\geq k} a_\alpha (x-w)^\alpha$.

Let $\Phi_1 = \{f_1, f_2, f_3\}$ and $R_1 = 1$. The rank of the Jacobian matrix of f is $r_1 = 1$ at x. We find $m_1 = 1$ and $\tilde{\Phi}_1 = \Phi_1$ and $R_2 = 2$. We choose $\Omega_1 = \{f_1\}$. The power series solution of $f_1(y_1, y, z) = 0$ is

$$y_1(y, z) = -y - z - y^2 - 2\,zy - z^2 + O_3.$$

Substituting x by y_1 in Φ_1 we find

$$\Phi_2 = \{O_3, -2zy - z^2 + O_3, -y^2 - 2zy + O_3\}.$$

For the next step $m_2 = 2$ and

$$\tilde{\Phi}_2 := \frac{\partial \Phi_2}{\partial y} = \{\Phi_2, -2z + O_2, -2y - 2z + O_2\}.$$

The rank of the Jacobian matrix of $\tilde{\Phi}_2$ is $r_2 = 2$. We choose $\Omega_2 = \{-2z+O_2, -2y-2z+O_2\}$ Since $R_3 = R_2 + r_2 = 4$. The deflation construction stops. The regular system at w is

$$\Omega = \{f_1, -2z + O_2, -2y - 2z + O_2\}.$$

We refer to [**29**] for the study of the complexity of this construction. Another type of deflation method mixing symbolic and numerical computations have been considered by Ojika, Watanabe, and Mitsui in [**44**], [**43**] : the new equations are generated by symbolic Gaussian eliminations but it remains to perform the numerical analysis and to study the complexity of this *modified deflation* method.

5.2. Augmented systems and deflation methods. From the knowledge of the structure of the local quotient algebra, Mantzaflaris and Mourrain determine a regular system given in the theorem 9. We sketch now another construction of deflation sequence based on a augmentation of the number of equations and of the number of variables. First, one defines a deflation operator which associates to the initial system f, a new system $Defl(f, x, y)$ where $(x, y) \in \mathbf{C}^{n+j}$. Next, one iterates this operator to obtain the deflation sequence :

$$x^0 = x,\ y^0 = y,\ F_0 = f, \quad x^{k+1} = (x^k, y^k), \quad F_{k+1} = Defl(f_k, x^k, y^k),\ k \geq 0.$$

The *length* of the deflation is the vector $(n_0, \ldots, n_k, \ldots)$ where n_k is the dimension of the kernel of the Jacobian matrix $DF_k(x^k)$. The thickness of the deflation is the number N such that $n_{N+1} = 0$.

In this way such a type of deflation operator had been proposed by Leykin, Verschelde, Zhao in [**30**], and extended in [**31**]. From an original system $f = (f_1, \ldots, f_m)$ with $\mathsf{rank}\, Df(w) = r$ they define the following :

$$LVZ(f, x, y) := LVZ(f, B, h, x, y) = \begin{cases} f(x) \\ Df(x)By \\ h^*y - 1 \end{cases}$$

where B is a random $n \times (r+1)$ matrix and h a random $r+1$ vector. The matrix $Df(x)B$ has generically a rank equal to r and the dimension of $\mathsf{Kernel}\, Df(w)B$ is 1. Hence there exists a unique $\lambda \in \mathbf{C}^{r+1}$ such that $Df(w)B\lambda = 0$ and $h^*\lambda - 1 = 0$.

THEOREM 16 ([**30**], [**31**]). *The multiplicity of the root (w, λ) of the system $LVZ(f, x, y)$ is strictly less than the multiplicity of the root w of the system f.*

Unfortunately the deflated system $LVZ(f, x, y)$ is not regular at its root (w, λ). In this case the method consists to deflate more until to find a regular system. We have

THEOREM 17 ([**30**], [**31**]). *The number of deflation steps to obtain a regular system is bounded by the multiplicity of w. If N is the number of deflations, the regular system has $n + N + \sum_{k=1}^{N} r_k$ variables and $2^N(n+1) - 1$ equations.*

EXAMPLE 6 ([**11**]). *Let $f(x, y) = (x + y^3, x^2y - y^4)$ with $(0, 0)$ has multiplicity 3. The number of deflations steps is 3 and the coranks of the Jacobian matrices of the deflated systems are equal to 1. The regular system has 16 variables and 23 equations.* ∘

EXAMPLE 7 ([**10**]). *Let $f = (x^4 - yzt, y^4 - zxt, z^4 - xyt, t^4 - xyz)$. The root has multiplicity 131. Two steps of LVZ deflation are needed with length $(4, 4)$. The regular system has 7 variables and 19 equations.* ∘

Another way to construct deflated systems by adding variables and equations has been proposed par Dayton and Zeng in [**10**] for the polynomial case and Dayton, Li, Zeng in [**9**] for the analytic case.

The deflation operator proposed by these authors is

$$DLZ(f, x, y) := DLZ(f, R, e_1, x, y) = \begin{cases} f(x) \\ Df(x)y \\ Ry - e_1 \end{cases}$$

where R is $p \times n$ random matrix in order that $\begin{bmatrix} Df(w) \\ R \end{bmatrix}$ has full rank and $e_1 = (1, 0, \ldots, 1)^T$ with size p is the dimension of the kernel of $Df(x)$.

THEOREM 18 ([**10**], [**9**]). *The number of steps of the DLZ deflation is bounded by the thickness δ of the root w defined in theorem 7. The last deflated system has 2^δ variables and $2^\delta n + \sum_{k=0}^{\delta-1} 2^k p_k$ where p_k is the corank of DLZ system k.*

EXAMPLE 8 ([**10**]). *Let* $f = (x^4 - yzt, y^4 - zxt, z^4 - xyt, t^4 - xyz)$. *The root has multiplicity* 131. *Two steps of DLZ deflation are needed with length* $(4, 4)$. *The regular system has* 16 *variables and* 28 *equations.* ∘

The example 6 lies to the class of systems of "breadth one" as defined by Dayton and Zeng in [**10**], i.e., the length is $(1, \ldots, 1)$. Note that this notation corresponds to the embedding dimension 1 as introduced by Giusti, Lecerf, Salvy, Yakoubsohn in [**17**]. For this class the DLZ deflation can be modified in order to obtain μn variables and μm equations.

6. Deflating and kerneling

We propose a new construction to deflate a system without adding new variables. It is based on two operations we called deflating and kerneling in the introduction.

6.1. Deflating. This operation consists to replace an equation $g(x) = 0$ by the n equations $\partial_i g(x) = 0, i = 1 : n$ when we have simultaneously $g(w) = 0$ and $\partial_i g(w) = 0$, $i = 1 : n$. We then can define the following recursive algorithm.

deflating$(f, \bar{w}, \epsilon)$

- Input : $f = (f_1, \ldots, f_m)$, $\bar{w}$ a point close to a multiple root w of f, and ϵ a precision.
- Let $J := Df(x)$ and $J_{\bar{w}} := Df(\bar{w})$.
- Let m_J the number of lines of J.
- $f_{deflated} = \emptyset$
- for $k = 1 : m_J$
- if $\max_{1 \leq j \leq n} |J_{\bar{w}}(k, j)| \leq \epsilon$ then
- deflating$(J(k, :), \bar{w}, \epsilon)$
- else
- $f_{deflated} = f_{deflated} \cup \{f_k / LT(f_k)\}$
- end if
- end for
- Output $f_{deflated}$

REMARK 2.

The assignment $f_{deflated} = f_{deflated} \cup \{f_k / LT(f_k)\}$ must be understood in the following way : the polynomial $f_k / LT(f_k)$ is added if it is not already an element of the set $f_{deflated}$. ∘

6.2. Kerneling. Let us consider a system $f = (f_1, \ldots, f_m)$ such that each line of $Df(w)$ is non zero and $Df(w)$ has a rank $r < n$. Without loss of generality we can write

$$Df(w) = \begin{pmatrix} A(w) & B(w) \\ C(w) & D(w) \end{pmatrix} \in \mathbf{C}^{m \times n}$$

where $A(w)$ is an invertible matrix of size $r \times r$. Then the Schur complement $D(w) - C(w)A^{-1}B(w)$ is zero. Hence w is a root of the system

$$D(x) - C(x)A^{-1}(x)B(x) = 0.$$

The kerneling operation consists of adding to the initial system at most the $(m - r) \times (n - r)$ polynomials given by the non zero numerators of the coefficients of the Schur complement. We then can define the following algorithm.

kerneling$(f, \bar{w}, \epsilon)$

- Input : $f = (f_1, \ldots, f_m)$, $\bar{w}$ a point close to a multiple root w of f, and ϵ a precision. Each line of $Df(\bar{w})$ is non zero.
- Determine r the numerical rank of $Df(\bar{w})$.
- Determine an invertible submatrix $A(\bar{w})$ of $Df(\bar{w})$ of size $r \times r$.
- Compute $S(x) = det(A(x))D(x) - det(A(x))C(x)A^{-1}B(x)$.
- $f_{deflated} = f \cup \{\text{elements of } S(x)\}$
- Output $f_{deflated}$

6.3. Equivalent system. Combining deflating and kerneling operations we compute a equivalent system of n variables and n equations.

equivalent$(f, \bar{w}, \epsilon)$

- Inputs : $f = (f_1, \ldots, f_m)$, $\bar{w}$ a point close to a multiple root w of f and ϵ a precision.
- $f_{deflated} = f$.
- while $Df_{delated}(\bar{w})$ is not numerically full rank
- $\quad f_{deflated} = \mathsf{deflating}(f_{deflated}, \bar{w}, \epsilon)$
- $\quad f_{deflated} = \mathsf{kerneling}(f_{deflated}, \bar{w}, \epsilon)$
- end while
- $f_{deflated} = \{n \text{ equations of full rank from } f_{deflated}\}$
- Output $f_{deflated}$

6.4. Example. Let us consider

$$f(x, y) = (x^3/3 + xy^2 + x^2 + 2xy + y^2,\ x^2y + x^2 + 2xy + y^2)$$

The point $(0, 0)$ is a root of $f(x, y) = 0$ with multiplicity 6. The deflating algorithm applied with $w = (0, 0)$ gives :

$$\begin{array}{cc|cc} \partial_1 & \partial_2 & \partial_1 & \partial_2 \\ x^2 + y^2 + 2x + 2y & 2yx + 2x + 2y & 2xy + 2x + 2y & x^2 + 2x + 2y \end{array}$$

All these previous quantities vanish at w. An additional step of deflating operation gives

$$\begin{array}{cc|cc||cc|cc} \partial_{11} & \partial_{12} & \partial_{21} & \partial_{22} & \partial_{11} & \partial_{12} & \partial_{11} & \partial_{12} \\ 2x + 2 & 2y + 2 & 2y + 2 & 2x + 2 & 2y + 2 & 2x + 2 & 2x + 2 & 2 \end{array}$$

All these quantities are non zero at w. Hence the deflated system is :

$$f_{deflated}(x, y) = (x^2 + y^2 + 2x + 2y, \quad xy + x + y, \quad x^2 + 2x + 2y)$$

Now we can use the kerneling algorithm of this new system.

$$Df_{deflated}(x, , y) = \begin{pmatrix} 2x + 2 & 2y + 2 \\ y + 1 & x + 1 \\ 2x + 2 & 2 \end{pmatrix}.$$

Then $Df_{deflated}(0, 0)$ has rank one. We can consider $A(x) = 2x + 2$. The Schur complement of $Df_{deflated}(x, y)$ associated to $2x + 2$ is

$$\begin{pmatrix} x + 1 \\ 2 \end{pmatrix} - \frac{2y + 2}{2x + 2} \begin{pmatrix} y + 1 \\ 2x + 2 \end{pmatrix} = \frac{1}{x + 1} \begin{pmatrix} x^2 + 2x - y^2 - 2y \\ -2xy - 2y \end{pmatrix}.$$

Finally from the system

$$(x^2 + y^2 + 2x + 2y, \quad xy + x + y, \quad x^2 + 2x + 2y,\ x^2 + 2x - y^2 - 2y,\ y)$$

we can choose

$$f_{deflated}(x,y) = (x + y + xy, y)$$

which is regular at w.

6.5. Why the multiplicity decreases? Let I be the ideal generated by $f_1, \ldots, f_m$ and w a multiple isolated root of $f_1 = \ldots = f_m = 0$. We deal with $\mathbf{C}\{x-w\}$ the local ring of convergent power series at w and $I\mathbf{C}\{x-w\}$ the ideal generated by I in $\mathbf{C}\{x-w\}$. Then the multiplicity of w is the dimension of the local quotient algebra $\mathbf{C}\{x-w\}/I\mathbf{C}\{x-w\}$. This dimension is finite if and only if the root w is isolated. We denote by $\{g_1, \ldots, g_p\}$ a local standard basis of $I\mathbf{C}\{x-w\}$. Let $\langle LT(I\mathbf{C}\{x-w\})\rangle$ the ideal generated by the leading monomials of IA. Then the multiplicity is the number of monomial that are not contained in $\langle LT(I\mathbf{C}\{x-w\})\rangle$. This number is independent of the chosen order on the monomials. We have the two classical results :

LEMMA 1. *Let h not in IA and $h(w) = 0$. Then the multiplicity of w as root of $f_1 = \ldots = f_m = 0$ is strictly greater than the multiplicity of w as root of $h = f_1 = f_2 = \ldots = f_m = 0$.*

PROOF. Since the leading term of h is not in $I\mathbf{C}\{x-w\}$ the lemma follows easily. □

The result we use to explain why the the multiplicity decreases under the action of the algorithm deflated is stated by Arnold, Gusein-Zade and Varchenko in **[5]** page 100 :

LEMMA 2. *Let $g = (g_1, \ldots, g_n) \in \mathbf{C}[x]^n$. Then the Jacobian $det(Dg(x))$ is not in the ideal $\langle g_1, \ldots, g_n\rangle$.*

The two lemmas below explain why the multiplicity decreases under the operations of deflating and kerneling.

LEMMA 3. *Let w a multiple root of a system $f_1 = \ldots = f_m = 0$ such that* grad $f_1(w) = 0$. *Then the multiplicity of w as root of $f_1 = \ldots = f_m = 0$ is strictly greater than the multiplicity of w as root of $\partial_1 f_1 = \ldots = \partial_n f_1 = f_2 = \ldots = f_m = 0$.*

PROOF. Let $g = (f_1, g_2 \ldots, g_n)$, the $g_i's$ being selected from the $f_2, \ldots, f_m$. Since the jacobian of g is not is the ideal generated by g, see lemma 2, then each line of the jacobian matrix of g has at least one element which is not in $\langle g\rangle$. In particular at least one of $\partial_i f_1$'s is not in $\langle g\rangle$. Following the lemma 1 we are done. □

LEMMA 4. *Let w a multiple root of $f_1 = \ldots = f_m = 0$ such that* grad $f_i(w) \neq 0$, *$i = 1 : m$. Let r be the rank of $Df(w)$ and*

$$Df(w) = \begin{pmatrix} A(w) & B(w) \\ C(w) & D(w) \end{pmatrix}$$

where $A(w)$ is an invertible matrix of size $r \times r$. Let $S(x) = det(A(x))D(x) - C(x)\Delta(x)B(x)$ where $\Delta(x) = det(A(x))A(x)^{-1}$. Then the multiplicity of w as root of $f_1 = \ldots = f_m = 0$ is strictly greater than the multiplicity of w as root of $S_{11} = \ldots = S_{m-r,n-r} = f_1 = f_2 \ldots = f_m = 0$.

PROOF. It is sufficient to prove that one of S_{ij}'s is not in the ideal $\langle f_1, \ldots, f_m \rangle$. Then, by lemma 1, the multiplicity of w as root of $f_1 = \ldots = f_m = 0$ is strictly greater than the multiplicity of w as root of $S_{ij} = f_1 = f_2 \ldots = f_m = 0$.

Let $F = (f_1, \ldots, f_r, h_1, \ldots, h_{n-r})$ with $h_i \in \{f_{r+1}, \ldots, f_m\}$.

We have $det(DF(x)) = det(A(x)\, det(S_F(x))$ where $S_F(x)$ is the Schur complement of $DF(x)$ associated to $A(x)$. From lemma 2, $det(DF(x))$ is not in the ideal $\langle F \rangle$. So it is the same for $det(A(x)$ and $det(S_F(x))$ which divide $det(DF(x))$. Hence there exists at least $n - r$ coefficients of the matrix $S_F(x)$ which are not in the ideal $\langle F \rangle$. Since the coefficients of $S_F(x)$ are also coefficients of the matrix $S(x)$ the conclusion follows. □

How much the multiplicity drops at each step of the equivalent algorithm ?

THEOREM 19. *For $k \geq 1$, let $F_0 = f$ and F_{k-1} the deflated system obtained at the step $k-1$ of equivalent algorithm and m_{k-1} the number of polynomials of F_{k-1}. Let p_k be the number of polynomials we add by deflating operation at the step k. We note by G_k the system F_k augmented by these p_k polynomials. Let r_k be the rank of the jacobian matrix of G_k at w. Then the number N of steps of the algorithm stops is equal to*

$$\min\{k \,:\, r_k = n \quad or \quad \sum_{k=1}^{N} s_k + t_k \leq \mu\}$$

where $\max(0, \min(1, p_k)) \leq s_k \leq p_k$ *and* $1 \leq t_k \leq p_k(n - r_k)$.

PROOF. From the lemmas 3 and 4 the multiplicity decreases at least by one. But we can be more precise. Let μ_k be the multiplicity of w as root of F_k. The deflating algorithm gives p_k polynomials. Then the multiplicity of the root w of G_k drops by $\mu_{k-1} - s_k$ where $\max(0, \min(1, p_k) \leq s_k \leq p_k$. Next, if the jacobian matrix of G_k at w has rank $r_k = n$ the equivalent algorithm stops. Otherwise, the multiplicity of w as root of F_k is $\mu_{k-1} - s_k - t_k$ where $1 \leq t_k \leq p_k(n - r_k)$. This bound is justified because all the polynomials of the Schur complement computed by the kerneling algorithm can be equal. □

7. Examples

We first treat three examples given in **[65]**. These examples show it is not necessary to know the complete structure of the local quotient algebra to determine a regular equivalent system from the initial one with a multiple root.

EXAMPLE 9. **[65]**

$$f_k(x_1, \ldots, x_n) = x_1 + \ldots + x_n + x_k^2, \quad k = 1 : n.$$

The jacobian matrix $Df(x) = \begin{pmatrix} 2x_1+1 & 1 & \ldots & 1 \\ 1 & 2x_2+1 & \ldots & 1 \\ \vdots & & & \\ 1 & 1 & \ldots & 2x_n+1 \end{pmatrix}$ *has rank one at* $(0, \ldots, 0)$. *The Schur complement associated to* $2x_1 + 1$ *gives the equations:*

$$x_1 = 0$$
$$(2x_1 + 1)(2x_k + 1) - 1 = 0, \quad k \geq 2.$$

EXAMPLE 10 ([**65**]).

$$\begin{aligned} f_k(x_1,\dots,x_n) &= x_k^3 - x_{k+1}x_{k+2}, \quad k = 1 : n-2 \\ f_{n-1}(x_1,\dots,x_n) &= x_{n-1}^3 - x_n x_1 \\ f_n(x_1,\dots,x_n) &= x_n^3 - x_1 x_2 \end{aligned}$$

A multiple root is $(0,\dots,0)$. In the first deflation step we replace the f_k's by their gradients. We obtain the equations :

$$x_1 = \dots = x_n = 0.$$

EXAMPLE 11 ([**65**]).

$$\begin{aligned} f_k(x_1,\dots,x_n) &= x_k + \dots + x_{n-2}, \quad k = 1 : n-2 \\ f_{n-1}(x_1,\dots,x_n) &= x_1 + \dots + x_{n-2} + x_{n-1}^5 + x_n^2 \\ f_n(x_1,\dots,x_n) &= x_1 + \dots + x_{n-2} + x_n^2 \end{aligned}$$

A multiple zero is $(0,\dots,0)$. The Jacobian matrix $Df(x) = \begin{pmatrix} I_{n-2} & 0 & 0 \\ 1\dots1 & 5x_{n-1}^4 & 2x_n \\ 1\dots1 & 0 & 2x_n \end{pmatrix}$ has rank $n-2$ at the multiple root $(0,\dots,0)$. The Schur complement associated to I_{n-2} furnishes the equations

$$5x_{n-1}^4 = 2x_n = 0.$$

After one step of deflation we obtain the system

$$f_1 = \dots = f_{n-2} = x_{n-1} = x_n = 0.$$

EXAMPLE 12 ([**58**, cmbs1]).

$$f(x,y,z) = (x^3 - yz,\ y^3 - xz,\ z^3 - xy).$$

A multiple root is $(0,0,0)$. A first of deflation gives the equations $x = y = z = 0$.

EXAMPLE 13 ([**58**, cmbs2]).

$$\begin{aligned} f(x,y,z) =& (x^3 - 3x^2y + 3xy^2 - y^3 - z^2, \\ & z^3 - 3z^2x + 3zx^2 - x^3 - y^2, \\ & y^3 - 3y^2z + 3yz^2 - z^3 - x^2). \end{aligned}$$

A multiple root is $(0,0,0)$. A first step of deflation gives the equations $x = y = z = x - y = x - z = y - z = 0$.

EXAMPLE 14 ([**40**]). caprasse

$$\begin{aligned} f(x,y,z,t) =& (-x^3z + 4\,xy^2z + 4\,x^2yt + 2\,y^3t + 4\,x^2 - 10\,y^2 + 4\,xz - 10\,yt + 2, \\ & -xz^3 + 4\,yz^2t + 4\,xzt^2 + 2\,yt^3 + 4\,xz + 4\,z^2 - 10\,yt - 10\,t^2 + 2 \\ & y^2z + 2\,xyt - 2\,x - z, \\ & 2\,yzt + xt^2 - x - 2\,z). \end{aligned}$$

The multiple root is $(2, -i\sqrt{3}, 2, i\sqrt{3})$. The gradient of each f_k is non zero at w and the jacobian matrix $Df(w)$ has rank 2. The step of kerneling adds the four polynomials before we get a regular system at w.

$$-10xt-5xy-5zt+\tfrac{17}{4}xyt^2z^2-7/2yt^2x^2z^3+11/4yt^4x^2z+\tfrac{17}{4}yt^2x^2z-2y^2txz^4+\tfrac{47}{8}y^2txz^2+\tfrac{49}{8}xt^3z^2y^2$$
$$-7x^2z^3t-3/4x^2zt^3+\tfrac{31}{4}x^2zt+\tfrac{37}{4}y^2z^3t-5y^2zt^3-25y^2zt+5xyt^4+\tfrac{103}{8}xz^2t+xyz^4+\tfrac{15}{4}xyz^2+15yzt^2-xt^3$$
$$+\tfrac{19}{4}z^3t+11zt^3-5/2yz^3+5/4y^3t^2z^3-y^3t^4z+11y^3t^2z-3y^2t^5x+7y^2t^3x+\tfrac{13}{4}yt^2z^3-yt^4z-1/2x^3t^3z^2$$
$$-\tfrac{7}{8}xt^3z^2-3/2x^3tz^2+4xty^2+x^3tz^4-xtz^4-3/2x^2yz^3+2x^2yz-\tfrac{9}{8}x^3t^5+3xt^5+3/4x^3t^3+3/8x^3t,$$

$$5/2xy^2z+\tfrac{25}{2}xzt^2+5/4y^3t-\tfrac{25}{4}y^2-\tfrac{25}{4}t^2-\tfrac{25}{2}yt+\tfrac{55}{4}yt^3+15yzxt+1/2yz^4x^2t-5/4yz^2x^2t^3-4yz^2x^2t$$
$$-19/2y^2zxt^2-5/4yz^3xt+3yzxt^3-\tfrac{13}{8}xt^2y^2z^3+\tfrac{25}{4}t^4-\tfrac{23}{4}t^2z^2x^2+15/2t^2z^2y^2+1/4y^3z^4t-3/2y^3z^2t^3$$
$$-3/2y^3z^2t+1/2y^2z^5x-3y^2z^3x-yz^2t^3+x^3t^2z^3-3/8x^3t^4z-1/8x^3t^2z+3/2x^2t^5y-11/2x^2t^3y-\tfrac{13}{8}xt^2z^3$$
$$+1/2xt^4z+5t^2z^2-3/2t^4x^2-15/2t^4y^2-5/2t^2x^2+\tfrac{55}{4}t^2y^2+15/2z^2y^2-5/4z^4y^2+y^3t^5-5/4yt^5-9/4y^3t^3,$$

$$-5xt-10xy-15yz-10zt+9/4xyt^2z^2+24yt^2x^2z+\tfrac{25}{4}y^2txz^2-7/2y^2x^2z^3t+\tfrac{43}{8}y^2x^2zt^3-\tfrac{7}{8}y^2x^2zt$$
$$+\tfrac{33}{4}y^3t^2xz^2-6yt^2x^3z^2-3/2x^2z^3t-\tfrac{11}{8}x^2zt^3+\tfrac{103}{8}x^2zt+5y^2z^3t+6y^2zt^3+2y^2zt+7xyt^4-15xyt^2$$
$$+\tfrac{31}{4}xz^2t-1/2xyz^4+\tfrac{43}{2}xyz^2+5yzt^2-5xt^3+3/8z^3t+3zt^3+11/4yz^3+15y^3t^2z+11y^2t^3x-7x^3tz^2$$
$$+23xty^2-9x^2yz^3+8x^2yz+\tfrac{21}{8}y^4z^3t-5y^4zt^3+4y^4zt-3y^3xt^4-3y^3xt^2-3/2y^3xz^4+2y^3xz^2-yz^2x^3$$
$$+\tfrac{25}{4}x^3t^3+\tfrac{19}{4}x^3t-5y^3z+\tfrac{21}{4}y^3z^3+\tfrac{13}{4}yt^2x^3-x^4tz+7/4yt^4x^3+yz^4x^3+x^4z^3t-zx^4t^3,$$

$$10xy^2z+10/3yz^2t+\tfrac{20}{3}xzt^2+10y^3t-\tfrac{25}{3}y^2-\tfrac{25}{3}t^2-\tfrac{50}{3}yt+10yt^3+20yzxt-\tfrac{47}{6}yz^2x^2t-6y^2zxt^2-4/3yz^3xt$$
$$-4yzxt^3+x^3yz^3t-5/6x^3yzt^3-7/6xy^3z^3t+2/3xy^3zt^3-2/3xy^3zt-5/4y^2t^2x^2z^2-4/3t^2z^2x^2-8/3t^2z^2y^2$$
$$-2/3y^3z^2t-7/2y^2z^3x-13/2x^3t^2z+4/3x^2t^3y-5/3t^4y^2+10t^2x^2+\tfrac{80}{3}t^2y^2+10/3z^2y^2-1/4z^4y^2-\tfrac{34}{3}y^3t^3$$
$$-5/3y^4t^2+2y^4z^2+4/3y^4t^4-3/4y^4z^4-2/3x^2y^2t^4+2/3x^2y^2z^4-\tfrac{23}{12}x^2y^2z^2+x^4t^2z^2$$

EXAMPLE 15 ([**11**, decker2]).

$$f(x,y) = (x+y^3, x^2y-y^4).$$

A multiple root is $(0,0,0)$. A first step of deflation gives the equations $x+y^3 = x = y = 0$.

EXAMPLE 16 ([**30**, mth191]).

$$f(x,y) = (x^3+y^2+z^2-1, x^2+y^3+z^2-1, x^2+y^2+z^3-1).$$

A multiple root is $w = (0,1,0)$. The gradients of each polynomials are non zero at w. The jacobian matrix has rank 1. The step of kerneling adds the four polynomials :

$$x\,(9\,xy-4)$$
$$z\,(3\,y-2)$$
$$x\,(3\,y-2)$$
$$z\,(9\,zy-4)$$

The system $f_1 = x\,(9\,xy-4) = z\,(3\,y-2) = 0$ is regular at w.

EXAMPLE 17 ([**10**, DZ1]).

$$f(x,y,z,t) = (x^4-yzt,\ y^4-xzt,\ z^4-xyt,\ t^4-xyz).$$

A multiple root is $w = (0, 0, 0, 0)$. A step of deflation gives $x = y = z = t = 0$.

EXAMPLE 18 ([**10**, DZ2]).

$$f(x, y, z) = (x^4,\ x^2y + y^4,\ z + z^2 - 7x^3 - 8x^2).$$

A multiple root is $w = (0, 0, -1)$. A step of deflation adds the equation $x = y = 0$.

EXAMPLE 19 ([**10**, DZ3]).

$$\begin{aligned} f(x,y) =& (14\,x+33\,y-3\,\sqrt{5}\left(x^2+4\,xy+4\,y^2+2\right)+\sqrt{7}+x^3+6\,x^2y+12\,xy^2+8\,y^3 \\ & \tfrac{41}{8}\,x-9/4\,y-1/8\,\sqrt{5}+x^3-3/2\,x^2y+3/4\,xy^2-1/8\,y^3+3/8\,\sqrt{7}\left(4\,xy-4\,x^2-y^2-2\right). \end{aligned}$$

A multiple root is $w = ((2\sqrt{7} + \sqrt{5})/5,\ (2\sqrt{5} - \sqrt{7})/5)$. The gradients of each polynomials are non zero. The step of kerneling adds the polynomial

$$\begin{aligned} & -360x^2\sqrt{5}y+630xy^2\sqrt{5}+240xy-180\sqrt{7}x^3+360\sqrt{7}y^3+1260x^2+1440y^2-360x^3\sqrt{5}+540x^3y+45x^2y^2 \\ & -540xy^3-180y^3\sqrt{5}+540\sqrt{7}x\sqrt{5}y+180x^4+180y^4+1605-960\sqrt{7}x+480\sqrt{7}y-600\sqrt{5}x-1200\sqrt{5}y \\ & +360\sqrt{7}\sqrt{5}x^2-630\sqrt{7}x^2y-360\sqrt{7}xy^2-360\sqrt{7}\sqrt{5}y^2 \end{aligned}$$

Its gradient is zero at w. The step of deflation replaces it by the two following polynomials:

$$\begin{aligned} & 1/3y+7/2x+x^3+3/4\sqrt{7}\sqrt{5}y-4/3\sqrt{7}-5/6\sqrt{5}-3/4\sqrt{7}x^2-3/2x^2\sqrt{5}+9/4x^2y+1/8xy^2-3/4y^3-x\sqrt{5}y \\ & +\tfrac{7}{8}y^2\sqrt{5}+\sqrt{7}\sqrt{5}x-7/4\sqrt{7}xy-1/2\sqrt{7}y^2 \end{aligned}$$

$$\begin{aligned} & 4/9x+16/3y+4/3y^3+\sqrt{7}\sqrt{5}x+\tfrac{8}{9}\sqrt{7}-\tfrac{20}{9}\sqrt{5}+2\sqrt{7}y^2+x^3+1/6x^2y-3xy^2-y^2\sqrt{5}-2/3x^2\sqrt{5}+7/3x\sqrt{5}y \\ & -7/6\sqrt{7}x^2-4/3\sqrt{7}xy-4/3\sqrt{7}\sqrt{5}y \end{aligned}$$

The system build from f_1, f_2 and from the two previous polynomials is regular at w.

EXAMPLE 20 ([**43**, Ojika2]).

$$f(x, y, z) = (x^2 + y + z - 1,\ x + y^2 + z - 1,\ x + y + z^2 - 1).$$

A multiple root is $w = (1, 0, 0)$. The rank of $Df(w)$ is 2. The step of kerneling adds the equation $4xyz - x - y - z + 1 = 0$.

EXAMPLE 21 ([**43**, Ojika3]).

$$f(x, y, z) = (x + y + z - 1,\ 2x^3 + 5y^2 - 10z + 5z^3 + 5,\ 2x + 2y + z^2 - 1).$$

A multiple root is $w = (-5/2, 5/2, 1)$. The rank of $Df(w)$ is 2. The step of kerneling adds the equation $3x^2z - 5yz + 5y - 3x^2 = 0$.

EXAMPLE 22 ([**29**, Lecerf]).

$$\begin{aligned} f(x,y,z) = (\quad & 2\,x + 2\,x^2 + 2\,y + 2\,y^2 + z^2 - 1, \\ & (x + y - z - 1)^3 - x^3, \\ & \left(2\,x^3 + 2\,y^2 + 10\,z + 5\,z^2 + 5\right)^3 - 1000\,x^5 \quad). \end{aligned}$$

A multiple root is $w = (0, 0, -1)$. The rank of $Df(w)$ is one. There is only one step of deflation to obtain the regular system

$$
\begin{aligned}
&x + x^2 + y + y^2 + 1/2\, z^2 - 1/2,\\
&y - z - 1,\\
&x + y - z - 1,\\
&\frac{9}{14}\, x^5 + \frac{5}{28}\, \left(2\, x^3 + 2\, y^2 + 10\, z + 5\, z^2 + 5\right) x^2 - \frac{625}{126}\, x,\\
&y,\\
&x,\\
&1 + z.
\end{aligned}
$$

8. Conclusion and future work

We have shown how to derive an equivalent regular system from a singular initial one, when we know the root. The stability of this process will be done in a future work and we describe briefly how to proceed. But from a numerical point of view a multiple root makes no sense and it is more realistic to speak of a cluster of roots : a m–cluster of roots is a open ball which contains m isolated regular roots of the system. Moreover we would hope for results with a "small" size of the cluster.

The operation of deflating is based on the evaluation of the gradient of a function, say $g(x)$, at given point $\bar{w}$. To decide whether there exists a root (or a cluster of roots) of this gradient closed to $\bar{w}$ we need to know if there exists $\bar{x}_1$ such that $(\bar{x}_1, \bar{w}_2, \ldots, \bar{w}_n)$ is closed to $\bar{w}$ and cancels the gradient of g. This can be done with the theoretical background developed in [**18**] where the words "closed to" and "small" are quantified.

The operation of kerneling requires more attention since we must discover the numerical rank of a jacobian matrix at a point $\bar{w}$ "closed to" the multiple root or the cluster of roots. The difficulty is that the rank drops only at the multiple root or in the cluster of roots. We propose to fix a coordinate, say x_1, and to perform a LU decomposition of the jacobian evaluated at $(x_1, \bar{w}_2, \ldots, \bar{w}_n)$. Each element of the diagonal of the matrix U of the LU decomposition is a polynomial in x_1. The numerical rank of the jacobian matrix is the number of these polynomials having a root "closed to" $\bar{w}_1$.

We illustrate these principles on Lecerf's example 22 [**29**]. We first show how to numerically discover that there is probably a point w near $(x_0, y_0, z_0) = (0.1, 0.09, -1.1 + 0.1i)$ where the jacobian matrix has a rank one. For that we determine the matrix U of the LU decomposition at (x_0, y_0, z). The diagonal of U is given by

$$
\begin{aligned}
&2.4\\
&2.28{+}4.51z{+}2.28z^2,\\
&3633.58{+}25322.98z{+}75771.82z^2{+}126177.32z^3{+}126276.08z^4{+}75944.48z^5{+}25413.69z^6\\
&{+}3650.4z^7.
\end{aligned}
$$

The Newton iteration (or more generally the Schröder iteration) initialized to z_0 and applied respectively to the polynomials $U_{22}(z)$ and $U_{33}(z)$ converges respectively to $-0.99 + 0.14i$ and $-0.98 + 0.05i$. The initial point z_0 is an approximated

zero of $U_{22}(z)$ and $U_{33}(z)$. This is the meaning given to the word "closed to". We will deduce that the numerical rank of the jacobian is one.

In this example we can numerically prove that there exists a point w where the two last lines of the jacobian matrix are zero. In fact the evaluation of the gradients of f_2 and f_3 at (x_0, y_0, z) gives

$$\begin{aligned}\nabla f_2(x_0,y_0,z) =&(2.91+5.94\,z+3.0\,z^2,\quad 0.0003\,(99+100\,z)^2,\quad -0.0003\,(99+100\,z)^2),\\ \nabla f_3(x_0,y_0,z) =&(4.03+18.06\,z+27.03\,z^2+18.0\,z^3+4.5\,z^4,\\ &-0.0000000432\,\left(25091+50000\,z+25000\,z^2\right)^2,\\ &0.0000012\,\left(25091+50000\,z+25000.\,z^2\right)^2(1+z)).\end{aligned}$$

Thanks to Newton iteration initialized at z_0 and applied successively to each polynomial coordinate of these two gradients we find a root closed to z_0. From this we can prove the existence of a perturbed system of the initial one with the two last lines of the jacobian matrix are zero. With this information we deflate the two corresponding equations of the initial system. This heuristic approach will be completely justified in a future work.

References

[1] Aizenberg, I.A. and Yuzhakov, A.P. *Integral Representations and Residues in Multidimensional Complex Analysis*, vol. 58. Providence,AMS, 1983.

[2] Eugene L. Allgower and Kurt Georg, *Numerical continuation methods*, Springer Series in Computational Mathematics, vol. 13, Springer-Verlag, Berlin, 1990. An introduction. MR1059455 (92a:65165)

[3] Maria Emilia Alonso, Maria Grazia Marinari, and Teo Mora, *The big mother of all dualities: Möller algorithm*, Comm. Algebra **31** (2003), no. 2, 783–818, DOI 10.1081/AGB-120017343. MR1968924 (2004b:13029)

[4] María Emilia Alonso, Maria Grazia Marinari, and Teo Mora, *The big mother of all dualities. II. Macaulay bases*, Appl. Algebra Engrg. Comm. Comput. **17** (2006), no. 6, 409–451, DOI 10.1007/s00200-006-0019-4. MR2270332 (2008d:13036)

[5] V. I. Arnol′d, S. M. Guseĭn-Zade, and A. N. Varchenko, *Singularities of differentiable maps. Vol. I*, Monographs in Mathematics, vol. 82, Birkhäuser Boston Inc., Boston, MA, 1985. The classification of critical points, caustics and wave fronts; Translated from the Russian by Ian Porteous and Mark Reynolds. MR777682 (86f:58018)

[6] Carlos Beltrán and Anton Leykin, *Certified numerical homotopy tracking*, Exp. Math. **21** (2012), no. 1, 69–83, DOI 10.1080/10586458.2011.606184. MR2904909

[7] Lenore Blum, Felipe Cucker, Michael Shub, and Steve Smale, *Complexity and real computation*, Springer-Verlag, New York, 1998. With a foreword by Richard M. Karp. MR1479636 (99a:68070)

[8] David A. Cox, John Little, and Donal O'Shea, *Using algebraic geometry*, 2nd ed., Graduate Texts in Mathematics, vol. 185, Springer, New York, 2005. MR2122859 (2005i:13037)

[9] Barry H. Dayton, Tien-Yien Li, and Zhonggang Zeng, *Multiple zeros of nonlinear systems*, Math. Comp. **80** (2011), no. 276, 2143–2168, DOI 10.1090/S0025-5718-2011-02462-2. MR2813352 (2012h:65101)

[10] Barry H. Dayton and Zhonggang Zeng, *Computing the multiplicity structure in solving polynomial systems*, ISSAC'05, ACM, New York, 2005, pp. 116–123 (electronic), DOI 10.1145/1073884.1073902. MR2280537

[11] D. W. Decker, H. B. Keller, and C. T. Kelley, *Convergence rates for Newton's method at singular points*, SIAM J. Numer. Anal. **20** (1983), no. 2, 296–314, DOI 10.1137/0720020. MR694520 (84d:65041)

[12] D. W. Decker and C. T. Kelley, *Newton's method at singular points. I*, SIAM J. Numer. Anal. **17** (1980), no. 1, 66–70, DOI 10.1137/0717009. MR559463 (81k:65065a)

[13] D. W. Decker and C. T. Kelley, *Newton's method at singular points. II*, SIAM J. Numer. Anal. **17** (1980), no. 3, 465–471, DOI 10.1137/0717039. MR581492 (81k:65065b)

[14] Jean-Pierre Dedieu and Mike Shub, *On simple double zeros and badly conditioned zeros of analytic functions of n variables*, Math. Comp. **70** (2001), no. 233, 319–327, DOI 10.1090/S0025-5718-00-01194-7. MR1680867 (2001f:65033)

[15] Jacques Emsalem, *Géométrie des points épais*, Bull. Soc. Math. France **106** (1978), no. 4, 399–416 (French, with English summary). MR518046 (80j:14008)

[16] William Fulton, *Intersection theory*, 2nd ed., Ergebnisse der Mathematik und ihrer Grenzgebiete. 3. Folge. A Series of Modern Surveys in Mathematics [Results in Mathematics and Related Areas. 3rd Series. A Series of Modern Surveys in Mathematics], vol. 2, Springer-Verlag, Berlin, 1998. MR1644323 (99d:14003)

[17] M. Giusti, G. Lecerf, B. Salvy, and J.-C. Yakoubsohn, *On location and approximation of clusters of zeros: case of embedding dimension one*, Found. Comput. Math. **7** (2007), no. 1, 1–49, DOI 10.1007/s10208-004-0159-5. MR2283341 (2008e:65159)

[18] M. Giusti, G. Lecerf, B. Salvy, and J.-C. Yakoubsohn, *On location and approximation of clusters of zeros of analytic functions*, Found. Comput. Math. **5** (2005), no. 3, 257–311, DOI 10.1007/s10208-004-0144-z. MR2168678 (2006k:30012)

[19] G.-M. Greuel and G. Pfister, *Advances and improvements in the theory of standard bases and syzygies*, Arch. Math. (Basel) **66** (1996), no. 2, 163–176, DOI 10.1007/BF01273348. MR1367159 (96k:13038)

[20] A. O. Griewank, *Starlike domains of convergence for Newton's method at singularities*, Numer. Math. **35** (1980), no. 1, 95–111, DOI 10.1007/BF01396373. MR583659 (81j:65070)

[21] A. Griewank, *On solving nonlinear equations with simple singularities or nearly singular solutions*, SIAM Rev. **27** (1985), no. 4, 537–563, DOI 10.1137/1027141. MR812453 (87g:65071)

[22] Andreas Griewank and M. R. Osborne, *Newton's method for singular problems when the dimension of the null space is* $> nn1$, SIAM J. Numer. Anal. **18** (1981), no. 1, 145–149, DOI 10.1137/0718011. MR603436 (82c:65032)

[23] A. Griewank and M. R. Osborne, *Analysis of Newton's method at irregular singularities*, SIAM J. Numer. Anal. **20** (1983), no. 4, 747–773, DOI 10.1137/0720050. MR708455 (85a:65073)

[24] Gröbner, W. *Moderne Algebraische Geometrie*, vol. Bibliographisches Institut Mannheim. Springer, 1949.

[25] Joos Heintz, *Definability and fast quantifier elimination in algebraically closed fields*, Theoret. Comput. Sci. **24** (1983), no. 3, 239–277, DOI 10.1016/0304-3975(83)90002-6. MR716823 (85a:68062)

[26] A. S. Householder, *The numerical treatment of a single nonlinear equation*, McGraw-Hill Book Co., New York, 1970. International Series in Pure and Applied Mathematics. MR0388759 (52 #9593)

[27] Birkett Huber and Jan Verschelde, *Polyhedral end games for polynomial continuation*, Numer. Algorithms **18** (1998), no. 1, 91–108, DOI 10.1023/A:1019163811284. MR1659862 (99i:65057)

[28] Hidetsune Kobayashi, Hideo Suzuki, and Yoshihiko Sakai, *Numerical calculation of the multiplicity of a solution to algebraic equations*, Math. Comp. **67** (1998), no. 221, 257–270, DOI 10.1090/S0025-5718-98-00906-5. MR1434942 (98c:14047)

[29] G. Lecerf, *Quadratic Newton iteration for systems with multiplicity*, Found. Comput. Math. **2** (2002), no. 3, 247–293, DOI 10.1007/s102080010026. MR1907381 (2003f:65090)

[30] Anton Leykin, Jan Verschelde, and Ailing Zhao, *Newton's method with deflation for isolated singularities of polynomial systems*, Theoret. Comput. Sci. **359** (2006), no. 1-3, 111–122, DOI 10.1016/j.tcs.2006.02.018. MR2251604 (2007k:65083)

[31] Anton Leykin, Jan Verschelde, and Ailing Zhao, *Higher-order deflation for polynomial systems with isolated singular solutions*, Algorithms in algebraic geometry, IMA Vol. Math. Appl., vol. 146, Springer, New York, 2008, pp. 79–97, DOI 10.1007/978-0-387-75155-9_5. MR2397938 (2009f:65130)

[32] T. Y. Li, *Numerical solution of polynomial systems by homotopy continuation methods*, Handbook of numerical analysis, Vol. XI, Handb. Numer. Anal., XI, North-Holland, Amsterdam, 2003, pp. 209–304. MR2009773 (2004k:65089)

[33] Stanisław Lojasiewicz, *Introduction to complex analytic geometry*, Birkhäuser Verlag, Basel, 1991. Translated from the Polish by Maciej Klimek. MR1131081 (92g:32002)

[34] F. S. Macaulay, *The algebraic theory of modular systems*, Cambridge Mathematical Library, Cambridge University Press, Cambridge, 1994. Revised reprint of the 1916 original; With an introduction by Paul Roberts. MR1281612 (95i:13001)

[35] Angelos Mantzaflaris and Bernard Mourrain, *Deflation and certified isolation of singular zeros of polynomial systems*, ISSAC 2011—Proceedings of the 36th International Symposium on Symbolic and Algebraic Computation, ACM, New York, 2011, pp. 249–256, DOI 10.1145/1993886.1993925. MR2895219

[36] M. G. Marinari, H. M. Möller, and T. Mora, *On multiplicities in polynomial system solving*, Trans. Amer. Math. Soc. **348** (1996), no. 8, 3283–3321, DOI 10.1090/S0002-9947-96-01671-6. MR1360228 (96k:13039)

[37] John Milnor, *Singular points of complex hypersurfaces*, Annals of Mathematics Studies, No. 61, Princeton University Press, Princeton, N.J., 1968. MR0239612 (39 #969)

[38] Mora, T., Pfister, G., and Traverso, C. An introduction to the tangent cone algorithm. *Issues in non-linear geometry and robotics, CM Hoffman ed* (1992).

[39] Morgan, A. *Solving Polynominal Systems Using Continuation for Engineering and Scientific Problems*, vol. 57. Society for Industrial Mathematics, 2009.

[40] Shuichi Moritsugu and Kazuko Kuriyama, *On multiple zeros of systems of algebraic equations*, Proceedings of the 1999 International Symposium on Symbolic and Algebraic Computation (Vancouver, BC), ACM, New York, 1999, pp. 23–30 (electronic), DOI 10.1145/309831.309846. MR1802063 (2002b:65060)

[41] B. Mourrain, *Isolated points, duality and residues*, J. Pure Appl. Algebra **117/118** (1997), 469–493, DOI 10.1016/S0022-4049(97)00023-6. Algorithms for algebra (Eindhoven, 1996). MR1457851 (98g:14007)

[42] Neufeldt, V., Guralnik, D., et al. *Webster's new world college dictionary*. Macmillan New York, 1997.

[43] Takeo Ojika, *Modified deflation algorithm for the solution of singular problems. I. A system of nonlinear algebraic equations*, J. Math. Anal. Appl. **123** (1987), no. 1, 199–221, DOI 10.1016/0022-247X(87)90304-0. MR881541 (88f:65085)

[44] Takeo Ojika, Satoshi Watanabe, and Taketomo Mitsui, *Deflation algorithm for the multiple roots of a system of nonlinear equations*, J. Math. Anal. Appl. **96** (1983), no. 2, 463–479, DOI 10.1016/0022-247X(83)90055-0. MR719330 (85a:65083)

[45] A. M. Ostrowski, *Solution of equations and systems of equations*, Pure and Applied Mathematics, Vol. IX. Academic Press, New York-London, 1960. MR0127525 (23 #B571)

[46] P. J. Rabier and G. W. Reddien, *Characterization and computation of singular points with maximum rank deficiency*, SIAM J. Numer. Anal. **23** (1986), no. 5, 1040–1051, DOI 10.1137/0723072. MR859016 (87m:58024)

[47] L. B. Rall, *Convergence of the Newton process to multiple solutions*, Numer. Math. **9** (1966), 23–37. MR0210316 (35 #1209)

[48] G. W. Reddien, *On Newton's method for singular problems*, SIAM J. Numer. Anal. **15** (1978), no. 5, 993–996, DOI 10.1137/0715064. MR507559 (80b:65064)

[49] G. W. Reddien, *Newton's method and high order singularities*, Comput. Math. Appl. **5** (1979), no. 2, 79–86, DOI 10.1016/0898-1221(79)90061-0. MR539566 (81c:65026)

[50] Rouché, E. *Mémoire sur la série de Lagrange, par M. Eugène Rouché*. Imprimerie impériale, 1866.

[51] Schröder, E. Über unendlich viele Algorithmen zur Auflösung der Gleichungen. *Mathematische Annalen 2*, 2 (1870), 317–365.

[52] Michael Shub, *Complexity of Bezout's theorem. VI. Geodesics in the condition (number) metric*, Found. Comput. Math. **9** (2009), no. 2, 171–178, DOI 10.1007/s10208-007-9017-6. MR2496558 (2010f:65103)

[53] Michael Shub and Steve Smale, *Complexity of Bézout's theorem. I. Geometric aspects*, J. Amer. Math. Soc. **6** (1993), no. 2, 459–501, DOI 10.2307/2152805. MR1175980 (93k:65045)

[54] M. Shub and S. Smale, *Complexity of Bezout's theorem. V. Polynomial time*, Theoret. Comput. Sci. **133** (1994), no. 1, 141–164, DOI 10.1016/0304-3975(94)90122-8. Selected papers of the Workshop on Continuous Algorithms and Complexity (Barcelona, 1993). MR1294430 (96d:65091)

[55] Steve Smale, *The fundamental theorem of algebra and complexity theory*, Bull. Amer. Math. Soc. (N.S.) **4** (1981), no. 1, 1–36, DOI 10.1090/S0273-0979-1981-14858-8. MR590817 (83i:65044)

[56] Andrew J. Sommese and Charles W. Wampler II, *The numerical solution of systems of polynomials*, World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2005. Arising in engineering and science. MR2160078 (2007a:14065)

[57] Hans J. Stetter, *Numerical polynomial algebra*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2004. MR2048781 (2006a:65004)

[58] Bernd Sturmfels, *Solving systems of polynomial equations*, CBMS Regional Conference Series in Mathematics, vol. 97, Published for the Conference Board of the Mathematical Sciences, Washington, DC, 2002. MR1925796 (2003i:13037)

[59] Traub, J.F. *Iterative methods for the solution of equations*. Chelsea Publishing Company, 1982.

[60] Verschelde, J. Polynomial homotopy continuation with phcpack. *ACM Communications in Computer Algebra 44*, 3/4 (2011), 217–220.

[61] Alden H. Wright, *Finding all solutions to a system of polynomial equations*, Math. Comp. **44** (1985), no. 169, 125–133, DOI 10.2307/2007797. MR771035 (86i:12001)

[62] Jean-Claude Yakoubsohn, *Finding a cluster of zeros of univariate polynomials*, J. Complexity **16** (2000), no. 3, 603–638, DOI 10.1006/jcom.2000.0555. Complexity theory, real machines, and homotopy (Oxford, 1999). MR1787887 (2001j:65084)

[63] Jean-Claude Yakoubsohn, *Simultaneous computation of all the zero-clusters of a univariate polynomial*, Foundations of computational mathematics (Hong Kong, 2000), World Sci. Publ., River Edge, NJ, 2002, pp. 433–455. MR2021992 (2005a:65049)

[64] Zhonggang Zeng, *ApaTools: a software toolbox for approximate polynomial algebra*, Software for algebraic geometry, IMA Vol. Math. Appl., vol. 148, Springer, New York, 2008, pp. 149–167, DOI 10.1007/978-0-387-78133-4_9. MR2410720 (2009j:65382)

[65] Zhonggang Zeng, *The closedness subspace method for computing the multiplicity structure of a polynomial system*, Interactions of classical and numerical algebraic geometry, Contemp. Math., vol. 496, Amer. Math. Soc., Providence, RI, 2009, pp. 347–362, DOI 10.1090/conm/496/09733. MR2555964 (2010j:13056)

Laboratoire LIX, École Polytechnique, 91128 Palaiseau Cedex, France
E-mail address: Marc.Giusti@polytechnique.fr

Institut de Mathématiques de Toulouse, Université Paul Sabatier, 118 route de Narbonne, 31062 Toulouse Cedex 9, France
E-mail address: yak@mip.ups-tlse.fr

Contemporary Mathematics
Volume **604**, 2013
http://dx.doi.org/10.1090/conm/604/12071

On the intrinsic complexity of elimination problems in effective algebraic geometry

Joos Heintz, Bart Kuijpers, and Andrés Rojas Paredes

Dedicated to the memory of Jean-Pierre Dedieu

Abstract. The representation of polynomials by arithmetic circuits evaluating them is an alternative data structure which allowed considerable progress in polynomial equation solving in the last fifteen years. We present a circuit based computation model which captures the core of all known symbolic elimination algorithms that avoid unnecessary branchings in effective algebraic geometry and show the intrinsically exponential complexity character of elimination in this complexity model.

1. Introduction

Modern elimination theory starts with Kronecker's 1882 paper [**Kro82**] where the argumentation is essentially constructive, i.e., algorithmic. Questions of efficiency of algorithms become only indirectly and marginally addressed in this paper. However, later criticism of Kronecker's approach to algebraic geometry emphasized the algorithmic inefficiency of his argumentation ([**Mac16**], [**vdW50**]). In a series of more recent contributions, that started with [**CGH89**] and ended up with [**GHM98**], [**GHH97**], [**HMW01**] and [**GLS01**], it became apparent that this criticism is based on a too narrow interpretation of Kronecker's elimination method. In fact, these contributions are, implicitly or explicitly, based on this method, notwithstanding that they also contain other views and ideas coming from commutative algebra and algebraic complexity theory.

A turning point was achieved by the combination of a new, global view of Newton iteration with Kronecker's method ([**GHM98**], [**GHH97**]). The outcome was that elimination polynomials, although hard to represent by their coefficients, allow a reasonably efficient encoding by evaluation algorithms. This circumstance suggests to represent in elimination algorithms polynomials not by their coefficients but by arithmetic circuits (see [**HS81**], [**Kal88**] and [**FIK86**] for the first steps in this direction). This idea became fully realized by the "Kronecker" algorithm for the resolution of polynomial equation systems over algebraically closed fields. The algorithm was anticipated in [**GHMP95**], [**GHM98**], [**HMW01**], [**GLS01**] and implemented in a software package of identical name (see [**Lec**]).

Research partially supported by the following Argentinian, Belgian and Spanish grants: CONICET PIP 2461/01, UBACYT 20020100100945, PICT–2010–0525, FWO G.0344.05, MTM2010-16051.

This paper deals with lower complexity bounds mainly for elimination algorithms.

After some preparation of algebraic geometric tools in Section 2 we introduce in Section 3 the notion of a robust parameterized arithmetic circuit which represents in a suitable sense the branching–free evaluation of parameter dependent polynomials when divisions become replaced by suitable limits.

In Section 4 we exhibit an infinite family of parameter dependent elimination polynomials which require exponentially many operations for their evaluation by robust parameterized arithmetic circuits, whereas the circuit size of the corresponding elimination problems grows only polynomially.

In the past, many attempts to show the non–polynomial character of the elimination of just one existential quantifier block in the arithmetic circuit based elementary language over $\mathbb{C}$, employed the reduction to the claim that an appropriate candidate family of specific polynomials was hard to evaluate (this approach was introduced in [**HM93**] and became adapted to the BSS model in [**SS95**]). We give here the first example of such a family where hardness really can be proved (see also [**HKR13**]).

In Section 5 we present, along the lines of software engineering, a computational model containing a particular architectural feature, called procedure. This model constitutes a simplified, abstract version of that introduced in [**HKR13**]. It captures the core of all known elimination algorithms that avoid unnecessary branchings.

In particular, we exhibit in Section 6.1 an infinite family of arithmetic input circuits encoding efficiently certain elimination problems such that any procedure solving them requires exponential time. It turns out that the Kronecker algorithm is an optimal procedure. It follows that any arithmetic circuit based elimination method, designed by commonly accepted rules of software engineering, needs exponential time to solve these problems when unnecessary branchings are avoided.

2. Concepts and tools from algebraic geometry

In this section, we use freely standard notions and notations from commutative algebra and algebraic geometry. These can be found for example in [**Lan84**], [**ZS60**], [**Kun85**] and [**Sha94**]. In Sections 2.1 and 2.3, we introduce the notions and definitions which constitute our fundamental tool for the modelling of elimination problems and algorithms. Most of these notions and their definitions are taken from [**GHMS11**].

2.1. Basic notions and notations. For any $n \in \mathbb{N}$, we denote by $\mathbb{A}^n := \mathbb{A}^n(\mathbb{C})$ the n–dimensional affine space $\mathbb{C}^n$ equipped with its respective Zariski and Euclidean topologies over $\mathbb{C}$.

Let $X_1, \ldots, X_n$ be indeterminates over $\mathbb{C}$ and let $X := (X_1, \ldots, X_n)$. We denote by $\mathbb{C}[X]$ the ring of polynomials in the variables X with complex coefficients.

Let V be a closed affine subvariety of $\mathbb{A}^n$, i.e. the set of common zeroes of finitely many polynomials of $\mathbb{C}[X]$. As usual, we write $\dim V$ for the dimension of the variety V.

For $f_1, \ldots, f_s \in \mathbb{C}[V]$ we shall use the notation $\{f_1 = 0, \ldots, f_s = 0\}$ in order to denote the closed affine subvariety V of $\mathbb{A}^n$ defined by $f_1, \ldots, f_s$.

We denote by $\mathbb{C}[V] := \{\varphi : V \to \mathbb{C} \ ; \text{ there exists } f \in \mathbb{C}[X] \text{ with } \varphi(x) = f(x) \text{ for any } x \in V\}$ the coordinate ring of V. If V is irreducible, then $\mathbb{C}[V]$ is zero–divisor free and we denote by $\mathbb{C}(V)$ the field formed by the rational functions of V with maximal domain ($\mathbb{C}(V)$ is called the rational function field of V). Observe that $\mathbb{C}(V)$ is isomorphic to the fraction field of the integral domain $\mathbb{C}[V]$.

In the general situation where V is an arbitrary closed affine subvariety of $\mathbb{A}^n$, the notion of a rational function of V has also a precise meaning. The only point to underline is that the domain, say U, of a rational function of V has to be a maximal Zariski open and dense subset of V to which the given rational function can be extended. In particular, U has a nonempty intersection with any of the irreducible components of V.

As in the case where V is irreducible, we denote by $\mathbb{C}(V)$ the $\mathbb{C}$–algebra formed by the rational functions of V. In algebraic terms, $\mathbb{C}(V)$ is the total quotient ring of $\mathbb{C}[V]$ and is isomorphic to the direct product of the rational function fields of the irreducible components of V.

Let be given a partial map $\phi : V \dashrightarrow W$, where V and W are closed subvarieties of some affine spaces $\mathbb{A}^n$ and $\mathbb{A}^m$, and let $\phi_1, \ldots, \phi_m$ be the components of ϕ. The map ϕ is called a *morphism of affine varieties* or just a *polynomial map* if the complex valued functions $\phi_1, \ldots, \phi_m$ belong to $\mathbb{C}[V]$. Thus, in particular, ϕ is a total map.

We call ϕ a rational map of V to W, if the domain U of ϕ is a Zariski open and dense subset of V and $\phi_1, \ldots, \phi_m$ are the restrictions of suitable rational functions of V to U.

Observe that our notion of a rational map differs from the usual one in algebraic geometry, since we do not require that the domain U of ϕ is maximal. Hence, in the case $m := 1$, our concepts of rational function and rational map do not coincide (see also [**GHMS11**]).

2.2. Constructible sets and constructible maps. Let $\mathcal{M}$ be a subset of some affine space $\mathbb{A}^n$ and, for a given nonnegative integer m, let $\phi : \mathcal{M} \dashrightarrow \mathbb{A}^m$ be a partial map.

Definition 1 (Constructible set). *We call the set* $\mathcal{M}$ constructible *if* $\mathcal{M}$ *is definable by a Boolean combination of polynomial equations.*

A basic fact we shall use in the sequel is that if $\mathcal{M}$ is constructible, then its Zariski closure is equal to its Euclidean closure (see, e.g., [**Mum88**], Chapter I, §10, Corollary 1). In the same vein we have the following definition.

Definition 2 (Constructible map). *We call the partial map* ϕ constructible *if the graph of* ϕ *is constructible as a subset of the affine space* $\mathbb{A}^n \times \mathbb{A}^m$.

We say that ϕ is *polynomial* if ϕ is the restriction of a morphism of affine varieties $\mathbb{A}^n \to \mathbb{A}^m$ to the constructible subset $\mathcal{M}$ of $\mathbb{A}^n$ and hence a total map from $\mathcal{M}$ to $\mathbb{A}^m$. Furthermore, we call ϕ a *rational* map of $\mathcal{M}$ if the domain U of ϕ is contained in $\mathcal{M}$ and ϕ is the restriction to $\mathcal{M}$ of a rational map of the Zariski closure $\overline{\mathcal{M}}$ of $\mathcal{M}$. In this case U is a Zariski open and dense subset of $\mathcal{M}$.

Since the elementary, i.e., first–order theory of algebraically closed fields with constants in $\mathbb{C}$ admits quantifier elimination, constructibility means just elementary definability. In particular, ϕ is constructible implies that the domain and the image of ϕ are constructible subsets of $\mathbb{A}^n$ and $\mathbb{A}^m$, respectively.

REMARK 3. *A partial map $\phi : \mathcal{M} \dashrightarrow \mathbb{A}^m$ is constructible if and only if it is piecewise rational. If ϕ is a constructible total map there exists a Zariski open and dense subset U of $\mathcal{M}$ such that the restriction $\phi|_U$ of ϕ to U is a rational map of $\mathcal{M}$ (and of $\overline{\mathcal{M}}$).*

For details we refer to [**GHMS11**], Lemma 1.

2.3. Geometrically robust constructible maps. The main mathematical tool of this paper is the notion of geometrical robustness which we are going to introduce now.

Let $\mathcal{M}$ be a constructible subset of the affine space $\mathbb{A}^n$ and let $\phi : \mathcal{M} \to \mathbb{A}^m$ be a (total) constructible map with components $\phi_1, \ldots, \phi_m$.

We consider now the Zariski closure $\overline{\mathcal{M}}$ of the constructible subset $\mathcal{M}$ of $\mathbb{A}^n$. Observe that $\overline{\mathcal{M}}$ is a closed affine subvariety of $\mathbb{A}^n$ and that we may interpret $\mathbb{C}(\overline{\mathcal{M}})$ as a $\mathbb{C}[\overline{\mathcal{M}}]$–module (or $\mathbb{C}[\overline{\mathcal{M}}]$–algebra).

Fix now an arbitrary point x of $\overline{\mathcal{M}}$. By $\mathfrak{M}_x$ we denote the maximal ideal of coordinate functions of $\mathbb{C}[\overline{\mathcal{M}}]$ which vanish at the point x. By $\mathbb{C}[\overline{\mathcal{M}}]_{\mathfrak{M}_x}$ we denote the local $\mathbb{C}$–algebra of the variety $\overline{\mathcal{M}}$ at the point x, i.e., the localization of $\mathbb{C}[\overline{\mathcal{M}}]$ at the maximal ideal $\mathfrak{M}_x$. By $\mathbb{C}(\overline{\mathcal{M}})_{\mathfrak{M}_x}$ we denote the localization of the $\mathbb{C}[\overline{\mathcal{M}}]$–module $\mathbb{C}(\overline{\mathcal{M}})$ at $\mathfrak{M}_x$.

Following Remark 3, we may interpret $\phi_1, \ldots, \phi_m$ as rational functions of the affine variety $\overline{\mathcal{M}}$ and therefore as elements of the total fraction ring $\mathbb{C}(\overline{\mathcal{M}})$ of $\mathbb{C}[\overline{\mathcal{M}}]$. Thus $\mathbb{C}[\overline{\mathcal{M}}][\phi_1, \ldots, \phi_m]$ and $\mathbb{C}[\overline{\mathcal{M}}]_{\mathfrak{M}_x}[\phi_1, \ldots, \phi_m]$ are $\mathbb{C}$–subalgebras of $\mathbb{C}(\overline{\mathcal{M}})$ and $\mathbb{C}(\overline{\mathcal{M}})_{\mathfrak{M}_x}$ which contain $\mathbb{C}[\overline{\mathcal{M}}]$ and $\mathbb{C}[\overline{\mathcal{M}}]_{\mathfrak{M}_x}$, respectively.

The following result establishes for constructible maps a bridge between a topological and an algebraic notion. It will be fundamental in the context of this paper.

Theorem–Definition 4. *Let notations and assumptions be as before. We call the constructible map $\phi : \mathcal{M} \to \mathbb{A}^m$ geometrically robust if ϕ is continuous with respect to the Euclidean topologies of $\mathcal{M}$ and $\mathbb{A}^m$ or equivalently, if $\phi_1, \ldots, \phi_m$, interpreted as rational functions of the affine variety $\overline{\mathcal{M}}$, satisfy at any point $x \in \mathcal{M}$ the following two conditions:*

(i) *$\mathbb{C}[\overline{\mathcal{M}}]_{\mathfrak{M}_x}[\phi_1, \ldots, \phi_m]$ is a finite $\mathbb{C}[\overline{\mathcal{M}}]_{\mathfrak{M}_x}$–module.*
(ii) *$\mathbb{C}[\overline{\mathcal{M}}]_{\mathfrak{M}_x}[\phi_1, \ldots, \phi_m]$ is a local $\mathbb{C}[\overline{\mathcal{M}}]_{\mathfrak{M}_x}$–algebra whose maximal ideal is generated by $\mathfrak{M}_x$ and $\phi_1 - \phi_1(x), \ldots, \phi_m - \phi_m(x)$.*

For a proof of this result, which is based on Zariski's Main Theorem ([**Ive73**], §IV.2) we refer to [**HKR13**] (see also [**CGH03**] and [**GHMS11**]).

From the topological definition of a geometrically robust constructible map one deduces immediately the following statement.

COROLLARY 5. *If we restrict a geometrically robust constructible map to a constructible subset of its domain of definition we obtain again a geometrically robust map. Moreover the composition and the cartesian product of two geometrically robust constructible maps are geometrically robust. The geometrically robust constructible functions form a commutative $\mathbb{C}$–algebra which contains the polynomial functions.*

The origin of the concept of a geometrically robust map can be found, implicitly, in [**GH01**]. It was introduced explicitly for constructible maps with irreducible domains of definition in [**GHMS11**], where it is used to analyze the complexity character of multivariate Hermite–Lagrange interpolation.

For a constructible subset of an affine space we denote by $\mathbb{C}\langle\mathcal{M}\rangle$ the $\mathbb{C}$–algebra of all geometrically robust constructible functions defined on $\mathcal{M}$.

The constructible subsets of affine spaces together with the geometrically robust constructible maps between them form a category which we denote throughout this paper by $\mathcal{D}$.

3. Robust parameterized arithmetic circuits

We shall use freely standard concepts from algebraic complexity theory which can be found in [**BCS97**].

Let us fix natural numbers n and r, indeterminates $X_1, \ldots, X_n$ and a non–empty constructible subset $\mathcal{M}$ of $\mathbb{A}^r$. By $\pi_1, \ldots, \pi_r$ we denote the restrictions to $\mathcal{M}$ of the canonical projections $\mathbb{A}^r \to \mathbb{A}^1$.

A *(by $\mathcal{M}$) parameterized arithmetic circuit* β (with *basic parameters* $\pi_1, \ldots, \pi_r$ and *inputs* $X_1, \ldots, X_n$) is a labelled directed acyclic graph (labelled DAG) satisfying the following conditions:
each node of indegree zero is labelled by a scalar from $\mathbb{C}$, a basic parameter $\pi_1, \ldots, \pi_r$ or a input variable $X_1, \ldots, X_n$. Following the case, we shall refer to the *scalar, basic parameter* and (standard) *input* nodes of β. All other nodes of β have indegree two and are called *internal*. They are labelled by arithmetic operations (addition, subtraction, multiplication, division). A *parameter* node of β depends only on scalar and basic parameter nodes, but not on any input node of β (here "dependence" refers to the existence of a connecting path). A parameter node of outdegree zero or with an outgoing edge into a node that depends on an input is called *essential*. Moreover, at least one circuit node becomes labelled as output. Without loss of generality we may suppose that all nodes of outdegree zero are outputs of β.

We consider β as a syntactical object which we wish to equip with a certain semantics. In principle there exists a canonical evaluation procedure of β assigning to each node a rational function of $\mathcal{M} \times \mathbb{A}^n$ which, in case of a parameter node, may also be interpreted as a rational function of $\mathcal{M}$. In either situation we call such a rational function an *intermediate result* of β.

The evaluation procedure may fail if we divide at some node an intermediate result by another one which vanishes on a Zariski dense subset of a whole irreducible component of $\mathcal{M} \times \mathbb{A}^n$. If this happens, we call the labelled DAG β *inconsistent*, otherwise *consistent*.

If nothing else is said, we shall from now on assume that β is a consistent parameterized arithmetic circuit. The intermediate results associated with output nodes will be called *final results* of β.

We call an intermediate result associated with a parameter node a *parameter* of β and interpret it generally as a rational function of $\mathcal{M}$. If this node is essential, we call the corresponding parameter also *essential*. In the sequel we shall refer to the constructible set $\mathcal{M}$ as the *parameter domain* of β.

We consider β as a syntactic object which represents the final results of β, i.e., the rational functions of $\mathcal{M} \times \mathbb{A}^n$ assigned to its output nodes.

Now we suppose that the consistent parameterized arithmetic circuit β has been equipped with an additional structure, linked to the semantics of β. We assume that for each node ρ of β there is given a *total* constructible map $\mathcal{M} \times \mathbb{A}^n \to \mathbb{A}^1$ which extends the intermediate result associated with ρ. In this way, if β has K

nodes, we obtain a total constructible map $\Omega : \mathcal{M} \times \mathbb{A}^n \to \mathbb{A}^K$ which extends the rational map $\mathcal{M} \times \mathbb{A}^n \dashrightarrow \mathbb{A}^K$ given by the intermediate results of β.

DEFINITION 6 (Robust circuit). *Let notations and assumptions be as before. The pair (β, Ω) is called a robust parameterized arithmetic circuit if the constructible map Ω is geometrically robust.*

Observe that the above rational map $\mathcal{M} \times \mathbb{A}^n \dashrightarrow \mathbb{A}^K$ can be extended *to at most one* geometrically robust constructible map $\Omega : \mathcal{M} \times \mathbb{A}^n \to \mathbb{A}^K$. Therefore we shall apply from now on the term "robust" also to the consistent circuit β.

Robust parameterized arithmetic circuits may be pulled back as follows: Let $\mathcal{N}$ be a constructible subset of an affine space and let $\varphi : \mathcal{N} \to \mathcal{M}$ be a geometrically robust constructible map (i.e. a morphism of the category $\mathcal{D}$). Suppose that (β, Ω) is robust. Then Corollary 5 implies that the pullback $\Omega \circ (\varphi \times \mathrm{id}_{\mathbb{A}^n})$ is still a geometrically robust constructible map.

Hence (β, Ω) induces a by $\mathcal{N}$ parameterized arithmetical circuit $\varphi^*(\beta)$. Observe that $\varphi^*(\beta)$ may become inconsistent. If $\varphi^*(\beta)$ is consistent then $(\varphi^*(\beta), \Omega \circ (\varphi \times \mathrm{id}_{\mathbb{A}^n}))$ is robust. The nodes where the evaluation of $\varphi^*(\beta)$ fails correspond to divisions of zero by zero which may be replaced by so called approximative algorithms having unique limits (see [**HKR13**], Section 3.3.2). These limits are given by the map $\Omega \circ (\varphi \times \mathrm{id}_{\mathbb{A}^n})$. We call $(\varphi^*(\beta), \Omega \circ (\varphi \times \mathrm{id}_{\mathbb{A}^n}))$, or simply $\varphi^*(\beta)$, the *pullback* of (β, Ω) or β to $\mathcal{N}$.

We cannot exclude inconsistent parameterized arithmetic circuits from our considerations. However we may restrict our attention to such ones which are pullbacks of consistent robust parameterized arithmetic circuits. These inconsistent parameterized arithmetic circuits will also be called robust.

We say that the parameterized arithmetic circuit β is *totally division–free* if any division node of β corresponds to a division by a non–zero complex scalar.

We call β *essentially division–free* if only parameter nodes are labelled by divisions. Thus the property of β being totally division–free implies that β is essentially division–free, but not vice versa. Moreover, if β is totally division-free, the rational map given by the intermediate results of β is polynomial and therefore a geometrically robust constructible map. Thus, any by $\mathcal{M}$ parameterized, totally division–free circuit is in a natural way robust.

We observe the following elementary fact.

LEMMA 7. *Let notations and assumptions be as before and suppose that the parameterized arithmetic circuit β is robust. Then all intermediate results of β are polynomials in $X_1, \ldots, X_n$ over $\mathbb{C}\langle \mathcal{M} \rangle$.*

For a proof of Lemma 7 we refer to [**HKR13**], Section 3.1.

The statement of this lemma should not lead to confusions with the notion of an essentially division–free parameterized circuit. We say just that the intermediate results of β are polynomials in $X_1, \ldots, X_n$ and do not restrict the type of arithmetic operations contained in β (as we did defining the notion of an essentially division–free parameterized circuit).

To our parameterized arithmetic circuit β we may associate different complexity measures and models. In this paper we shall mainly be concerned with *sequential computing time*, measured by the *size* of β. Here we refer with "size" to the number of internal nodes of β which count for the given complexity measure. Our basic complexity measure is the *non–scalar* one (also called *Ostrowski measure*) over the

ground field $\mathbb{C}$. This means that we count, at unit costs, only essential multiplications and divisions (involving basic parameters or input variables in both arguments in the case of a multiplication and in the second argument in the case of a division), whereas $\mathbb{C}$–linear operations are free (see [**BCS97**] for details).

In [**HKR13**] we defined three operations on robust parameterized arithmetic circuits, namely the operations join which mimicks composition of circuit represented polynomial maps and reduction and broadcasting which embody rewriting of circuits by means of polynomial identities. In the present paper only reduction will be relevant. A circuit which at different nodes computes the same result may be simplified into a circuit which computes this result only once. The intermediate results of the new circuit are the same as those of the original one. This is the meaning reduction of circuits. For details we refer to [**HKR13**].

4. A family of hard elimination polynomials

As a major result of this paper we are now going to exhibit an infinite family of parameter dependent elimination polynomials which require exponential many operations for their evaluation by essentially division–free *robust* parameterized arithmetic circuits, whereas the circuit size of the corresponding input problems grows only polynomially. The proof of this result, which is absolutely new in his kind, is astonishly elementary and simple.

Let $T, U_1, \ldots, U_n$ and $X_1, \ldots, X_n$ be indeterminates and let $U := (U_1, \ldots, U_n)$ and $X := (X_1, \ldots, X_n)$. Consider for given $n \in \mathbb{N}$ the polynomial $H^{(n)} := \sum_{1 \le i \le n} 2^{i-1}\ X_i + T \prod_{1 \le i \le n}(1 + (U_i - 1)X_i)$. Observe that $H^{(n)}$ can be evaluated using $n-1$ non–scalar multiplications involving $X_1, \ldots, X_n$.

The set $\mathcal{O} := \{\sum_{1 \le i \le n} 2^{i-1} X_i + t \prod_{1 \le i \le n}(1 + (u_i - 1)X_i); (t, u_1, \ldots, u_n) \in \mathbb{A}^{n+1}\}$ is contained in a finite–dimensional $\mathbb{C}$–linear subspace of $\mathbb{C}[X]$ and therefore $\mathcal{O}$ and its closure $\overline{\mathcal{O}}$ are constructible sets.

From [**GHMS11**], Section 3.3.3 we deduce the following facts:
there exist $K := 16n^2 + 2$ integer points $\xi_1, \ldots, \xi_K \in \mathbb{Z}^n$ of bit length at most $4n$ such that for any two polynomials $f, g \in \overline{\mathcal{O}}$ the equalities $f(\xi_k) = g(\xi_k), 1 \le k \le K$, imply $f = g$. Thus the polynomial map $\Xi : \overline{\mathcal{O}} \to \mathbb{A}^K$ defined for $f \in \overline{\mathcal{O}}$ by $\Xi(f) := (f(\xi_1), \ldots, f(\xi_K))$ is injective. Moreover $\mathcal{M} := \Xi(\mathcal{O})$ is an irreducible constructible subset of $\mathbb{A}^K$ and we have $\overline{\mathcal{M}} = \Xi(\overline{\mathcal{O}})$. Finally, the constructible map $\phi := \Xi^{-1}$, which maps $\mathcal{M}$ onto $\mathcal{O}$ and $\overline{\mathcal{M}}$ onto $\overline{\mathcal{O}}$, is a restriction of a geometrically robust map and therefore by Corollary 5 itself geometrically robust.

For $\epsilon \in \{0,1\}^n$ we denote by ϕ_ϵ the map $\overline{\mathcal{M}} \to \mathbb{A}^1$ which assigns to each point $v \in \overline{\mathcal{M}}$ the value $\phi(v)(\epsilon)$. From Corollary 5 we conclude that ϕ_ϵ is a geometrically robust constructible function which belongs to the function field $\mathbb{C}(\overline{\mathcal{M}})$ of the irreducible algebraic variety $\overline{\mathcal{M}}$.

Observe that for $t \in \mathbb{A}^1$ and $u \in \mathbb{A}^n$ the identities $\phi_\epsilon(\Xi(H^{(n)}(t, u, X))) = \phi(\Xi(H^{(n)}\ (t, u, X)))(\epsilon) = ((\Xi^{-1} \circ \Xi)(H^{(n)}(t, u, X)))(\epsilon) = H^{(n)}(t, u, \epsilon)$ hold.

Let $P^{(n)} := \prod_{\epsilon \in \{0,1\}^n}(Y - \phi_\epsilon)$. Then $P^{(n)}$ is a geometrically robust constructible function which maps $\overline{\mathcal{M}} \times \mathbb{A}^1$ (and hence $\mathcal{M} \times \mathbb{A}^1$) into $\mathbb{A}^1$.

Consider now the polynomial $F^{(n)} := \prod_{\epsilon \in \{0,1\}^n}(Y - H^{(n)}(T, U, \epsilon)) = \prod_{0 \le j \le 2^n - 1}$ $(Y - (j + T \prod_{1 \le i \le n} U_i^{[j]_i}))$, where $[j]_i$ denotes the i–th digit of the binary representation of the integer j, $0 \le j \le 2^n - 1$, $1 \le i \le n$. We have for $t \in \mathbb{A}^1$ and $u \in \mathbb{A}^n$ the identities

$$P^{(n)}(\Xi(H^{(n)}(t,u,X)),Y) = \prod_{\epsilon\in\{0,1\}^n}(Y-\phi_\epsilon(\Xi(H^{(n)}(t,u,X)))) = \prod_{\epsilon\in\{0,1\}^n}(Y-H^{(n)}(t,u,\epsilon)) = F^{(n)}(t,u,Y) \tag{1}$$

Let $S_1,\ldots,S_K$ be new indeterminates and observe that the existential first order formula of the elementary theory of $\mathbb{C}$, namely

$$(\exists X_1)\ldots(\exists X_n)(\exists T)(\exists U_1)\ldots(\exists U_n)(X_1^2-X_1=0\wedge\cdots\wedge X_n^2-X_n=0\wedge \bigwedge_{1\le j\le K} S_j = H^{(n)}(T,U,\xi_j)\wedge Y = H^{(n)}(T,U,X)) \tag{2}$$

describes the constructible subset $\{(s,y)\in\mathbb{A}^{K+1}; s\in\mathcal{M}, y\in\mathbb{A}^1, P^{(n)}(s,y)=0\}$ of $\mathbb{A}^{K+1}$. Moreover, $P^{(n)}$ is the greatest common divisor in $\mathbb{C}(\overline{\mathcal{M}})[Y]$ of all polynomials of $\mathbb{C}[\overline{\mathcal{M}}][Y]$ which vanish identically on the constructible subset of $\mathbb{A}^{K+1}$ defined by the formula (2). Hence $P^{(n)}\in\mathbb{C}(\overline{\mathcal{M}})[Y]$ is a (parameterized) *elimination polynomial*.

Observe that the polynomials contained in the formula (2) can be represented by a totally division–free arithmetic circuit of size $O(n^3)$. Therefore, the formula (2) is also of size $O(n^3)$.

THEOREM 8. *Let notations and assumptions be as before and let γ be an essentially division–free, robust parameterized arithmetic circuit with domain of definition $\mathcal{M}$ such that γ evaluates the elimination polynomial $P^{(n)}$. Then γ has size at least $\Omega(2^n)$.*

PROOF. We fix the natural number n. Let us write $H := H^{(n)} = \sum_{1\le i\le n} 2^{i-1} X_i + \prod_{1\le i\le n} T(U_i-1)X_i$ as a polynomial in the main indeterminates $X_1,\ldots,X_n$ with coefficients $\theta_{\kappa_1,\ldots,\kappa_n}\in\mathbb{C}[T,U]$, $\kappa_1,\ldots,\kappa_n\in\{0,1\}$, namely

$$H = \sum_{\kappa_1,\ldots,\kappa_n\in\{0,1\}} \theta_{\kappa_1,\ldots,\kappa_n} X_1^{\kappa_1},\ldots,X_n^{\kappa_n}.$$

Observe that for $\kappa_1,\ldots,\kappa_n\in\{0,1\}$ the polynomial $\theta_{\kappa_1,\ldots,\kappa_n}(0,U)\in\mathbb{C}[U]$ is of degree at most zero, i.e., a constant complex number, independent of $U_1,\ldots,U_n$.

Let $\theta := (\theta_{\kappa_1,\ldots,\kappa_n})_{\kappa_1,\ldots,\kappa_n\in\{0,1\}}$ and observe that the vector $\theta(0,U)$ is a fixed point of the affine space $\mathbb{A}^{2^n}$. We denote by $\mathfrak{M}$ the vanishing ideal of the $\mathbb{C}$–algebra $\mathbb{C}[\theta]$ at this point. We interpret θ as a geometrically robust constructible map $\mathbb{A}^{n+1}\to\mathbb{A}^{2^n}$ with (constructible) image $\mathcal{T}$.

Let us write $F := F^{(n)} = \prod_{0\le j\le 2^n-1}(Y-(j+T\prod_{1\le i\le n} U_i^{[j]_i}))$ as a polynomial in the main indeterminate Y with coefficients $\varphi_\kappa\in\mathbb{C}[T,U]$, $1\le\kappa\le 2^n$, namely $F = Y^{2^n}+\varphi_1 Y^{2^n-1}+\cdots+\varphi_{2^n}$.

Observe that for $1\le\kappa\le 2^n$ the polynomial $\varphi_\kappa(0,U)\in\mathbb{C}[U]$ is of degree at most zero. Let $\lambda_\kappa := \varphi_\kappa(0,U)$, $\lambda := (\lambda_\kappa)_{1\le\kappa\le 2^n}$ and $\varphi := (\varphi_\kappa)_{1\le\kappa\le 2^n}$. Then λ is a fixed point of the affine space $\mathbb{A}^{2^n}$.

Let $\nu:\mathbb{A}^{n+1}\to\mathbb{A}^K$ be the polynomial map defined for $t\in\mathbb{A}^1$ and $u\in\mathbb{A}^n$ by $\nu(t,u) := \Xi(H(t,u,X)) = (H(t,u,\xi_1),\ldots,H(t,u,\xi_K))$. Observe that there exists a geometrically robust constructible map $\sigma:\mathcal{T}\to\mathbb{A}^K$ such that $\sigma\circ\theta=\nu$ holds. Since by assumption the parameterized arithmetic circuit γ is essentially division–free and robust, there exists a geometrically robust constructible map ψ defined on

$\mathcal{M}$ such that the entries of ψ constitute the essential parameters of the circuit γ. Moreover, for m being the number of components of ψ, there exists a vector ω of m–variate polynomials over $\mathbb{C}$ such that the entries of $\omega(\psi) = \omega \circ \psi$ become the coefficients of the elimination polynomial $P := P^{(n)} = \prod_{\epsilon \in \{0,1\}^n}(Y - \phi_\epsilon)$.

One sees easily that there exists a totally division–free ordinary arithmetic circuit γ' which evaluates the polynomials $H(T, U, \xi_1), \ldots, H(T, U, \xi_K)$.

The join $\gamma * \gamma'$ of γ' with γ at the basic parameter nodes of γ is an essentially division–free robust parameterized circuit with domain of definition $\mathbb{A}^{n+1}$ which by (1) evaluates the polynomial $F(T, U, Y) := F^{(n)}(T, U, Y)$. The entries of the vector $\tilde{\nu} := \psi \circ \nu$ constitute the essential parameters of the circuit $\gamma * \gamma'$ and the entries of $\omega \circ \tilde{\nu} = \omega \circ \psi \circ \nu$ become by (1) the coefficients of the polynomial $F(T, U, Y)$ with respect to Y. So we have $\varphi = \omega \circ \tilde{\nu}$.

Taking into account $\tilde{\nu} = \psi \circ \nu = \psi \circ \sigma \circ \theta$, Theorem–Definition 4 (i) and [**GHMS11**], Corollary 12 we conclude that the entries of $\tilde{\nu}$ are polynomials of $\mathbb{C}[T, U]$ which are integral over the local $\mathbb{C}$–subalgebra $\mathbb{C}[\theta]_{\mathfrak{M}}$ of $\mathbb{C}(T, U)$.

Let $\mu \in \mathbb{C}[T, U]$ be such an entry. Then there exists an integer s and polynomials $a_0, a_1, \ldots, a_s \in \mathbb{C}[\theta]$ with $a_0 \notin \mathfrak{M}$ such that the algebraic dependence relation

$$a_0\mu^s + a_1\mu^{s-1} + \cdots + a_s = 0 \tag{3}$$

is satisfied in $\mathbb{C}[T, U]$. From (3) we deduce the algebraic dependence relation

$$a_0(0, U)\mu(0, U)^s + a_1(0, U)\mu(0, U)^{s-1} + \cdots + a_s(0, U) = 0 \tag{4}$$

in $\mathbb{C}[U]$.

Since the polynomials $a_0, a_1, \ldots, a_s$ belong to $\mathbb{C}[\theta]$ and $\theta(0, U)$ is a fixed point of $\mathbb{A}^{2^n}$, we conclude that $\alpha_0 := a_0(0, U), \alpha_1 := a_1(0, U), \ldots, \alpha_s := a_s(0, U)$ are complex numbers. Moreover, $a_0 \notin \mathfrak{M}$ implies $\alpha_0 \neq 0$.

Thus (4) may be rewritten into the algebraic dependence relation

$$\alpha_0\mu(0, U)^s + \alpha_1\mu(0, U)^{s-1} + \cdots + \alpha_s = 0 \tag{5}$$

in $\mathbb{C}[U]$ with $\alpha_0 \neq 0$.

This implies that the polynomial $\mu(0, U)$ of $\mathbb{C}[U]$ is of degree at most zero. Therefore $w := \tilde{\nu}(0, U)$ is a fixed point of the affine space $\mathbb{A}^m$.

Recall that $\lambda = (\lambda_\kappa)_{1 \leq \kappa \leq 2^n}$ with $\lambda_\kappa := \varphi_\kappa(0, U)$, $1 \leq \kappa \leq 2^n$, is a fixed point of the affine space $\mathbb{A}^{2^n}$.

From [**CGH03**], Lemma 6 we deduce that for $1 \leq \kappa \leq 2^n$ the coefficient φ_κ of F is an element of $\mathbb{C}[T, U]$ of the form

$$\varphi_\kappa = \lambda_\kappa + TL_\kappa + \text{ terms of higher degree in } T \tag{6}$$

where $L_1, \ldots, L_{2^n} \in \mathbb{C}[U]$ are $\mathbb{C}$–linearly independent.

Consider now an arbitrary point $u \in \mathbb{A}^n$ and let $\epsilon_u : \mathbb{A}^1 \to \mathbb{A}^m$ and $\delta_u : \mathbb{A}^1 \to \mathbb{A}^{2^n}$ be the polynomial maps defined for $t \in \mathbb{A}^1$ by $\epsilon_u(t) := \tilde{\nu}(t, u)$ and $\delta_u(t) := \varphi(t, u)$. Then we have $\epsilon_u(0) = \tilde{\nu}(0, u) = w$ and $\delta_u(0) = \varphi(0, u) = \lambda$, independently of u. Moreover, from $\varphi = \omega \circ \tilde{\nu}$ we deduce $\delta_u = \omega \circ \epsilon_u$.

Thus (6) implies

$$(L_1(u), \ldots, L_{2^n}(u)) = \frac{\partial \varphi}{\partial t}(0, u) = \delta_u'(0) = (D\omega)_w(\epsilon_u'(0)), \tag{7}$$

where $(D\omega)_w$ denotes the (first) derivative of the m–variate polynomial map ω at the point $w \in \mathbb{A}^m$ and $\delta_u'(0)$ and $\epsilon_u'(0)$ are the derivatives of the parameterized curves δ_u and ϵ_u at the point $0 \in \mathbb{A}^1$. We rewrite now (7) in matrix form, replacing

$(D\omega)_w$ by the corresponding transposed Jacobi matrix $M \in \mathbb{A}^{m\times 2^n}$ and $\delta'_u(0)$ and $\epsilon'_u(0)$ by the corresponding points of $\mathbb{A}^{2^n}$ and $\mathbb{A}^m$, respectively.

Then (7) takes the form

$$(L_1(u), \ldots, L_{2^n}(u)) = \epsilon'_u(0)M, \tag{8}$$

where the complex $(m \times 2^n)$–matrix M is independent of u.

Since the polynomials $L_1, \ldots, L_{2^n} \in \mathbb{C}[U]$ are $\mathbb{C}$–linearly independent, we may choose points $u_1, \ldots, u_{2^n} \in \mathbb{A}^n$ such that the complex $(2^n \times 2^n)$–matrix

$$N := (L_\kappa(u_l))_{1\leq l,\kappa\leq 2^n}$$

has rank 2^n.

Let K be the complex $(2^n \times m)$–matrix whose rows are $\epsilon'_{u_1}(0), \ldots, \epsilon'_{u_{2^n}}(0)$.

Then (8) implies the matrix identity

$$N = K \cdot M.$$

Since N has rank 2^n, the rank of the complex $(m \times 2^n)$–matrix M is at least 2^n. This implies

$$m \geq 2^n. \tag{9}$$

Therefore the circuit γ contains $m \geq 2^n$ essential parameters.

Let L be the number of multiplications that are executed by the parameterized arithmetic circuit γ and that involve at least one factor depending on Y ($\mathbb{C}$–linear operations and multiplications between parameters are free). Then, after a well–known standard rearrangement [**PS73**] of γ, we may suppose without loss of generality, that there exists a constant $c > 0$ (independent of the input circuit γ) such that $L \geq cm$ holds.

From the estimation (9) we deduce now that the circuit γ performs at least $\Omega(2^n)$ multiplications. Therefore the size of γ is at least $\Omega(2^n)$. This finishes the proof of the theorem. □

Theorem 8 is essentially contained in the arguments of the proof of [**GH01**], Theorem 5 and [**CGH03**], Theorem 4.

Observe that a quantifier–free description of $\mathcal{M}$ by means of circuit represented polynomials, together with an essentially division–free, robust parameterized arithmetic circuit γ with domain of definition $\mathcal{M}$, which evaluates the elimination polynomial $P^{(n)}$ captures the intuitive meaning of an algorithmic solution of the elimination problem described by the formula (2), when we restrict our attention to solutions of this kind and minimize the number of equations and branchings. In particular the circuit γ can be evaluated for any input point (s, y) with $s \in \mathcal{M}$ and $y \in \mathbb{C}$ and the intermediate results of γ are polynomials of $\mathbb{C}(\overline{\mathcal{M}})[Y]$ whose coefficients are geometrically robust constructible functions defined on $\mathcal{M}$.

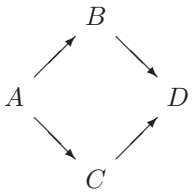
With respect to the indeterminate Y, the coefficients of the polynomial $P^{(n)} \in \mathbb{C}(\overline{\mathcal{M}})[Y]$ are geometrically robust constructible functions of the parameter domain $\mathcal{M}$. In order to consider $P^{(n)}$ as an elimination polynomial as we did, the reader might expect that the coefficients of $P^{(n)}$ should belong, for any point $s \in \mathcal{M}$, to the local ring of $\overline{\mathcal{M}}$ at s. This would be true if the algebraic variety $\overline{\mathcal{M}}$ would be normal at any $s \in \mathcal{M}$ (see [**GHMS11**], Corollary 12). From [**CGH03**], Corollary 3 we deduce that the variety $\overline{\mathcal{M}}$ is definitely not normal. This leads us to the question

how elimination polynomials should look like when the closure of the parameter domain is not normal.

In order to elucidate this question we shall consider the following general situation. It turns out that the requirement that the coefficients of elimination polynomials should be geometrically robust constructible functions is quite natural.

Let $\varphi : V \to W$ be a finite surjective morphism of irreducible affine varieties V and W over $\mathbb{C}$ such that there exists a coordinate function $y \in \mathbb{C}[V]$ with $\mathbb{C}[V] = \mathbb{C}[W][y]$. Let $d := [\mathbb{C}(V) : \mathbb{C}(W)]$ be the degree of φ and suppose that for any point $w \in W$ the cardinality of the fiber $\varphi^{-1}(w)$ is exactly d. Finally, let Y be a new indeterminate and $F := Y^d + \varphi_{d-1}Y^{d-1} + \cdots + \varphi_0 \in \mathbb{C}(W)[Y]$ the minimal polynomial of y. Observe that coefficients of F, namely $\varphi_0, \ldots, \varphi_{d-1} \in \mathbb{C}(W)$, are integral over $\mathbb{C}[W]$. We are now going to discuss a condition under which F may be considered as an elimination polynomial. This condition will imply that $\varphi_0, \ldots, \varphi_{d-1}$ are geometrically robust constructible functions.

We shall use the following abbreviations: $A := \mathbb{C}[W]$, $B := A[\varphi_0, \ldots, \varphi_{d-1}]$, $C := A[y]$, $D := B[y]$. We have the following commutative diagram of integral $\mathbb{C}$–algebra extensions:



Observe that D is isomorphic to $B[Y]/B[Y] \cdot F$ and in particular a free B–module of rank d.

PROPOSITION 9. *Suppose that for any maximal ideal $\mathfrak{m}$ of A the canonical $\mathbb{C}$–algebra homomorphism $A/\mathfrak{m} \to C/\mathfrak{m}C$ is unramified ([**Ive73**], Chapter I) and that $\mathfrak{m}$ is contained in at most d maximal ideals of D (thus, intuitively, F is an elimination polynomial). Then $\varphi_0, \ldots, \varphi_{d-1}$ are geometrically robust constructible functions of W.*

PROOF. Let $\mathfrak{m}$ be an arbitrary maximal ideal of A. Since $A \to B$ is an integral ring extension, we deduce from Theorem–Definition 4 that it suffices to show that there exists a single maximal ideal $\mathfrak{n}$ of B which contains $\mathfrak{m}$. Our assumptions yield a commutative diagram

$$\begin{array}{ccc} A/\mathfrak{m} & \longrightarrow & C/\mathfrak{m}C \\ \| & & \| \\ \mathbb{C} & \longrightarrow & \mathbb{C}^d \end{array}$$

Taking into account $C = A[y]$ we conclude that there exists a monic polynomial $G \in A[Y]$ of degree d with discriminant $\rho \in A$ such that $C/\mathfrak{m}C$ is isomorphic to $A[Y]$ divided by the ideal generated $\mathfrak{m}$ and G and such that ρ does not belong to $\mathfrak{m}$.

Let $\mathfrak{n}$ be an arbitrary maximal ideal of B which contains $\mathfrak{m}$ and let $\overline{F}$ and $\overline{G}$ be the images of F and G in $B/\mathfrak{n}[Y]$. Then we have $D/\mathfrak{n}D \cong B/\mathfrak{n}[Y]/B/\mathfrak{n}[Y] \cdot \overline{F}$ and therefore $\overline{F}$ divides $\overline{G}$ in $B/\mathfrak{n}[Y]$. From $d = \deg \overline{F} = \deg \overline{G}$ and the fact that F and G are monic we deduce $\overline{F} = \overline{G}$. Since the discriminant ρ of G does not belong to $\mathfrak{m}$ we have $\rho \notin \mathfrak{n}$ and therefore the polynomial $\overline{F}$ is separable. Thus we obtain a commutative diagram

$$\begin{array}{ccc} B/\mathfrak{n} & \longrightarrow & D/\mathfrak{n}D \\ \| & & \| \\ \mathbb{C} & \longrightarrow & \mathbb{C}^d \end{array}$$

and in particular the canonical $\mathbb{C}$–algebra homomorphism $B/\mathfrak{n} \to D/\mathfrak{n}D$ is unramified. Hence the number of maximal ideals of D which contain $\mathfrak{n}$ is exactly d. By assumption there are at most d maximal ideals of D containing $\mathfrak{m}$. Therefore any such ideal must contain $\mathfrak{n}$. Since $B \to D$ is an integral ring extension, we conclude that $\mathfrak{n}$ is the unique maximal ideal of B which contains $\mathfrak{m}$. □

5. A computation model with robust parameterized arithmetic circuits

This section is devoted to a deeper understanding of the assumptions which lead to Theorem 8. It will become clear why all known elimination methods in effective algebraic geometry which avoid unnecessary branchings are exponential. To this end we introduce a computation model which will be comprehensive enough to capture the hard core of all known circuit based elimination algorithms and, mutatis mutandis, also of all other (linear algebra and truncated rewriting) elimination procedures (see [**Mor03**], [**Mor05**], and the references cited therein, and for truncated rewriting methods especially [**DFGS91**]). However, this has to be understood with some caution. We do not claim that all elimination algorithms become *completely* captured by this model. For example, deformation based elimination procedures may contain ingredients which will escape from our modelling. Our computation model will constitute a simplified version of that of [**HKR13**].

The elimination problem and polynomial of Section 4 were somewhat artificial. We shall show that the conclusions of Theorem 8 are still valid for much more natural elimination problems and polynomials if we restrict the notion of algorithm to the computation model we are going to introduce in this section.

In the sequel we shall use freely basic notions of category theory (see [**Mitchell**]).

Let $X_1, \ldots, X_n, \ldots$ be indeterminates over $\mathbb{C}$.

Throughout this paper we shall consider the following contravariant functor $\mathcal{O}$ which maps the category $\mathcal{D}$ of constructible sets of Section 3 into the category of commutative $\mathbb{C}$–algebras. The functor $\mathcal{O}$ associates with a constructible subset $\mathcal{M}$ of an affine space the $\mathbb{C}$–algebra

$$\mathcal{O}(\mathcal{M}) := \{(H_n)_{n\geq 0}; H_n \in \mathbb{C}\,\langle \mathcal{M}\rangle\,[X_1, \ldots, X_n], \#\{n; H_n \neq 0\} < \infty\}$$

and with a geometrically robust constructible map $\varphi : \mathcal{N} \to \mathcal{M}$ the canonical $\mathbb{C}$–algebra homomorphism $\mathcal{O}(\varphi) : \mathcal{O}(\mathcal{M}) \to \mathcal{O}(\mathcal{N})$ induced by the pullback by φ of the polynomials with coefficients in $\mathbb{C}\,\langle \mathcal{M}\rangle$.

Let $\mathcal{M}$ be a constructible subset of an affine space, x a point of $\mathcal{M}$ and $H = (H_n)_{n\geq 0}$ with $H_n \in \mathbb{C}\,\langle \mathcal{M}\rangle\,[X_1, \ldots, X_n]$ an element of $\mathcal{O}(\mathcal{M})$. Then for any $n \geq 0$ the coefficients of H_n belong to $\mathbb{C}\,\langle \mathcal{M}\rangle$ and may therefore be evaluated in x. Hence we obtain from H_n a polynomial of $\mathbb{C}[X_1, \ldots, X_n]$ which we denote by $H_n(x)$. Let

$H(x) := (H_n(x))_{n\geq 0}$ and observe that for the inclusion map φ of the constructible subset $\{x\}$ of $\mathcal{M}$ the following holds:
$\mathcal{O}(\varphi) : \mathcal{O}(\mathcal{M}) \to \mathcal{O}(\{x\})$ assigns $H(x)$ to H.

Note also that for two points x and y of two affine spaces the canonical map $\varphi : \{x\} \to \{y\}$ is geometrically robust and $\mathcal{O}(\varphi) : \mathcal{O}(\{y\}) \to \mathcal{O}(\{x\})$ is the identity map.

A *specification* of a computational problem with polynomials is given by two (contravariant) subfunctors $\mathcal{G}$ and $\mathcal{F}$ of $\mathcal{O}$ which map the category $\mathcal{D}$ in the category of sets and by a natural transformation $\mathcal{S} : \mathcal{G} \to \mathcal{F}$.

Thus for any constructible subset $\mathcal{M}$ of an affine space and for any geometrically robust constructible map $\varphi : \mathcal{N} \to \mathcal{M}$ the objects $\mathcal{G}(\mathcal{M})$ and $\mathcal{F}(\mathcal{M})$ are subsets of $\mathcal{O}(\mathcal{M})$ and the diagrams

$$\begin{array}{ccc} \mathcal{G}(\mathcal{M}) & \hookrightarrow & \mathcal{O}(\mathcal{M}) \\ \downarrow{\scriptstyle \mathcal{G}(\varphi)} & & \downarrow{\scriptstyle \mathcal{O}(\varphi)} \\ \mathcal{G}(\mathcal{N}) & \hookrightarrow & \mathcal{O}(\mathcal{N}), \end{array} \qquad \begin{array}{ccc} \mathcal{F}(\mathcal{M}) & \hookrightarrow & \mathcal{O}(\mathcal{M}) \\ \downarrow{\scriptstyle \mathcal{F}(\varphi)} & & \downarrow{\scriptstyle \mathcal{O}(\varphi)} \\ \mathcal{F}(\mathcal{N}) & \hookrightarrow & \mathcal{O}(\mathcal{N}) \end{array}$$

and

$$\begin{array}{ccc} \mathcal{G}(\mathcal{M}) & \xrightarrow{\mathcal{S}(\mathcal{M})} & \mathcal{F}(\mathcal{M}) \\ \downarrow{\scriptstyle \mathcal{G}(\varphi)} & & \downarrow{\scriptstyle \mathcal{F}(\varphi)} \\ \mathcal{G}(\mathcal{N}) & \xrightarrow{\mathcal{S}(\mathcal{N})} & \mathcal{F}(\mathcal{N}) \end{array}$$

commute.

With these notations let be given a specification $\mathcal{S} : \mathcal{G} \to \mathcal{F}$. Then $\mathcal{S}$ is *isoparametric* in the following sense (compare [**HKR13**]):

LEMMA 10. *Let $\mathcal{M}$ be a constructible subset of an affine space and let be given $G \in \mathcal{G}(\mathcal{M})$, $F \in \mathcal{F}(\mathcal{M})$ with $\mathcal{S}(\mathcal{M})(G) = F$ and two points $x, y \in \mathcal{M}$. Then $G(x) = G(y)$ implies $F(x) = F(y)$.*

PROOF. Consider the canonical map $\varphi : \{x\} \to \{y\}$ which is geometrically robust and constructible. Recall that $\mathcal{O}(\varphi) : \mathcal{O}(\{y\}) \to \mathcal{O}(\{x\})$ is the identity map. Therefore the same is true for $\mathcal{G}(\varphi) : \mathcal{G}(\{y\}) \to \mathcal{G}(\{x\})$ and $\mathcal{F}(\varphi) : \mathcal{F}(\{y\}) \to \mathcal{F}(\{x\})$. Observing that $\mathcal{S}(\mathcal{M})(G) = F$ implies $\mathcal{S}(\{x\})(G(x)) = F(x)$ and $\mathcal{S}(\{y\})(G(y)) = F(y)$, we deduce from $G(x) = G(y)$ and the commutative diagram

$$\begin{array}{ccc} \mathcal{G}(\{y\}) & \xrightarrow{\mathcal{S}(\{y\})} & \mathcal{F}(\{y\}) \\ \downarrow{\scriptstyle \mathcal{G}(\varphi)} & & \downarrow{\scriptstyle \mathcal{F}(\varphi)} \\ \mathcal{G}(\{x\}) & \xrightarrow{\mathcal{S}(\{x\})} & \mathcal{F}(\{x\}) \end{array}$$

that $F(x) = F(y)$ holds. □

Since the elementary theory of algebraically closed fields of characteristic zero admits quantifier elimination we deduce from Lemma 10 the following statement.

COROLLARY 11. *Let notations and assumptions be as in Lemma* 10. *Let θ be the coefficient vector of the non–zero polynomials contained in G. Then there exists $m \in \mathbb{N}$ and a constructible map $\sigma_G : \theta(\mathcal{M}) \to \mathbb{A}^m$ such that $\sigma_G \circ \theta$ is the coefficient vector of the non–zero polynomials contained in F.*

DEFINITION 12. *We call the specification $\mathcal{S} : \mathcal{G} \to \mathcal{F}$ continuous if for any constructible subset $\mathcal{M}$ of an affine space and any $G \in \mathcal{F}(\mathcal{M})$ the constructible map σ_G is geometrically robust (i.e. continuous with respect to the Euclidean topology).*

Observe that compositions of continuous specifications are again continuous specifications.

We are now ready to define the notion of an algorithm which implements a given continuous specification.

DEFINITION 13. *Let notations be as before and let $\mathcal{S} : \mathcal{G} \to \mathcal{F}$ be a continuous specification. An algorithm $\mathcal{A}$ which implements $\mathcal{S}$ is a partial mapping between consistent robust and essentially division–free parameterized arithmetic circuits over the same parameter domain which assigns to each (for $\mathcal{A}$ admissible) input circuit β an output circuit $\mathcal{A}(\beta)$ such that the following conditions are satisfied:*

- *for any constructible subset $\mathcal{M}$ of an affine space and any for $\mathcal{A}$ admissible robust and essentially division–free, by $\mathcal{M}$ parameterized arithmetic circuit β which computes the non–zero polynomials contained in an element G of $\mathcal{G}(\mathcal{M})$, the circuit $\mathcal{A}(\beta)$ computes the non-zero polynomials contained in $S(\mathcal{M})(G)$.*
- *the parameters of the circuit $\mathcal{A}(\beta)$ are obtained by composing the vector of essential parameters of β with a suitable geometrically robust constructible map.*

The idea behind this notion of algorithm is to avoid branchings by replacing them by suitable divisions which become evaluated using limits. In this sense we speak about *branching–parsimonious* algorithms. This concept is consistent with that of an (output isoparametric) algorithm introduced in [**HKR13**].

DEFINITION 14. *Let notations be as before and let $\mathcal{A}$ be an algorithm that implements a continuous specification $\mathcal{S} : \mathcal{G} \to \mathcal{F}$. We call $\mathcal{A}$ a procedure if for any (for $\mathcal{A}$ admissible) input circuit β the parameters of the output circuit $\mathcal{A}(\beta)$ are obtained by composing the vector of coefficients of the final results of β with a suitable geometrically robust constructible map.*

Our notions of specified algorithm and procedure constitute our computation model. For motivations and a concrete realization of these concepts we refer to [**HKR13**].

We finish this section with some comments motivating the notions of (continuous) specification, algorithm and procedure and three examples.

A specification $\mathcal{S} : \mathcal{G} \to \mathcal{F}$ as above encodes a computation problem with polynomials. Let $\mathcal{M}$ be a constructible subset of an affine space, $G \in \mathcal{G}(\mathcal{M})$ and $F \in \mathcal{F}(\mathcal{M})$ with $F = \mathcal{S}(\mathcal{M})(G)$. Then G and F represent (finite length) vectors

of input and output polynomials and $\mathcal{S}(\mathcal{M})$ the underlying computational problem which transforms the input G into the output F.

We try now to explain why we define specifications as natural transformations between suitable functors. For this purpose let us consider the simple case that $\mathcal{M}$ and $\mathcal{N}$ are constructible subsets of the same affine space, $\mathcal{N}$ a subset of $\mathcal{M}$ and $\varphi : \mathcal{N} \to \mathcal{M}$ the inclusion map of $\mathcal{N}$ into $\mathcal{M}$. Clearly, φ is geometrically robust and constructible. Then we wish to be able to specialize the computational problem represented by $\mathcal{S}(\mathcal{M}) : \mathcal{G}(\mathcal{M}) \to \mathcal{F}(\mathcal{M})$ to the constructible subset $\mathcal{N}$ of $\mathcal{M}$. Such a specialization yields a new computational problem $\mathcal{S}(\mathcal{N}) : \mathcal{G}(\mathcal{N}) \to \mathcal{F}(\mathcal{N})$ with commutative diagram

$$\begin{array}{ccc} \mathcal{G}(\mathcal{M}) & \xrightarrow{\mathcal{S}(\mathcal{M})} & \mathcal{F}(\mathcal{M}) \\ \Big\downarrow{\scriptstyle \mathcal{G}(\varphi)} & & \Big\downarrow{\scriptstyle \mathcal{F}(\varphi)} \\ \mathcal{G}(\mathcal{N}) & \xrightarrow{\mathcal{S}(\mathcal{N})} & \mathcal{F}(\mathcal{N}). \end{array}$$

The requirement that $\mathcal{S}$ should be a natural transformation between functors is a straight forward generalization of this reasoning.

The property of continuity of specifications embodies a necessary condition which has to be satisfied if we look for branching–parsimonious algorithms implementing the given specification. We refer to [**HKR13**] for a discussion of the relationship between branching–parsimoniousness and continuity.

We exhibit now three examples of our concepts of continuous specification, algorithm and procedure.

Example 1. For any constructible subset $\mathcal{M}$ of an affine space let $\mathcal{G}(\mathcal{M})$ be the set of all elements of $\mathcal{O}(\mathcal{M})$ with at most one entry different from zero. Furthermore, let $\mathcal{F} := \mathcal{O}$. For $\mathcal{M}$ as above let $\mathcal{S}(\mathcal{M}) : \mathcal{G}(\mathcal{M}) \to \mathcal{F}(\mathcal{M})$ be the map which assigns to any $G \in \mathcal{G}(\mathcal{M})$, with $G_n \in \mathbb{C}\langle\mathcal{M}\rangle[X_1, \ldots, X_n]$ being the unique entry of G possibly different from zero, the element of $\mathcal{F}(\mathcal{M})$ composed by zeroes and the partial derivatives $\frac{\partial G_n}{\partial X_1}, \ldots, \frac{\partial G_n}{\partial X_n}$. Obviously $\mathcal{G}$ and $\mathcal{F}$ may be interpreted as (contravariant) subfunctors of $\mathcal{O}$ and $\mathcal{S}$ as a natural transformation between them. Observe that $\mathcal{S}$ is a continuous specification.

We have two algorithms $\mathcal{A}$ and $\mathcal{B}$ which implement $\mathcal{S}$ and are defined as follows: Let $\mathcal{M}$ be a constructible subset of an affine space and β a robust and essentially division–free, by $\mathcal{M}$ parameterized arithmetic circuit with a single output node which evaluates the unique possibly non–zero polynomial occurring in an element G of $\mathcal{G}(\mathcal{M})$.

Then $\mathcal{A}$ transforms the circuit β into a circuit $\mathcal{A}(\beta)$ by means of the forward mode and $\mathcal{B}$ transforms β into $\mathcal{B}(\beta)$ by means of the reverse mode of automatic differentiation (see [**GW08**] for the notions of forward and reverse mode). The algorithms $\mathcal{A}$ and $\mathcal{B}$ use recursion on the internal structure of the input circuit and they do not constitute procedures. They are efficient in the sense that the size of $\mathcal{A}(\beta)$ is linear in n times the size of β and the size of $\mathcal{B}(\beta)$ is linear in the size of β.

Example 2. For any constructible subset $\mathcal{M}$ of an affine space let $\mathcal{G}(\mathcal{M})$ be the set of all elements $(G_n)_{n\in\mathbb{N}}$ of $\mathcal{O}(\mathcal{M})$ where at most G_1 is different from zero. Furthermore, let $\mathcal{F} := \mathcal{G}$ and let $\mathcal{S}(\mathcal{M}) : \mathcal{G}(\mathcal{M}) \to \mathcal{F}(\mathcal{M})$ be the map which assigns

to any $G \in \mathcal{G}(\mathcal{M})$ the element $F = (F_n)_{n\geq 0}$ of $\mathcal{F}(\mathcal{M})$ where F_1 is the primitive integral of G_1 satisfying the condition $F_1(0) = 0$ and where the other entries of F are zero.

Again $\mathcal{G}$ and $\mathcal{F}$ may be interpreted as subfunctors of $\mathcal{O}$ and $\mathcal{S}$ as a natural transformation between them. Moreover $\mathcal{S}$ is a continuous specification.

The most obvious algorithm $\mathcal{A}$ which implements $\mathcal{S}$ can be described as follows. Let $\mathcal{M}$ be a constructible subset of an affine space and β a robust and essentially division–free by $\mathcal{M}$ parameterized arithmetic circuit with a single output node which evaluates the polynomial G_1 occurring in an element $G = (G_n)_{n\geq 0}$ of $\mathcal{G}(\mathcal{M})$. Then $\mathcal{A}$ computes the coefficients of the polynomial G_1 and produces finally a robust and essentially division–free parameterized arithmetic circuit $\mathcal{A}(\beta)$ which evaluates the primitive integral of G_1 applying term–by–term integration to the coefficient representation of the polynomial G_1.

The algorithm $\mathcal{A}$ is clearly a procedure. However it may be very inefficient if β is a small circuit which evaluates a polynomial of high degree. Therefore one may imagine alternative algorithms which do not have this drawback and which make a clever use of integration by substitution and by parts.

It is not likely that a general purpose integration algorithm of this type exists. For example the polynomial $X_1^d + \cdots + X_1 + 1$, $d \in \mathbb{N}$, may be evaluated using $O(\log d)$ arithmetic operations whereas the complexity status of its primitive integral, namely $\frac{1}{d+1}X_1^{d+1} + \cdots + \frac{1}{2}X_1^2 + X_1$, is unknown. There exists a conjecture that this latter polynomial is hard to evaluate.

Let $\mathcal{M}$ be a constructible subset of an affine space, $(z_k)_{k\in\mathbb{N}}$ a (not necessarily convergent) sequence of points of $\mathcal{M}$ and β a robust and essentially division–free by $\mathcal{M}$ parameterized arithmetic circuit with a single output node such that β evaluates a polynomial $P \in \mathbb{C}\langle\mathcal{M}\rangle[X_1]$. Suppose that the sequence of polynomials $(P(z_k, X_1))_{k\in\mathbb{N}}$ converges to a polynomial $Q \in \mathbb{C}[X_1]$ such that there exists a point $z \in \mathcal{M}$ with $Q = P(z, X_1)$. Then the primitive integrals of $P(z_k, X_1) \in \mathbb{C}[X_1]$, $k \in \mathbb{N}$, converge to the primitive integral of Q. Let be given a procedure $\mathcal{A}$ that implements the continuous specification $\mathcal{S}$. Then the parameters of the circuit $\mathcal{A}(\beta)$ constitute a geometrically robust constructible map ν with domain of definition $\mathcal{M}$. Since $\mathcal{A}$ is a procedure, ν depends constructibly and continuously on the coefficients of P. Therefore the sequence $(\nu(z_k))_{k\in\mathbb{N}}$ converges to a point, say ζ. We may interpret $\mathcal{A}(\beta)$ as a composition of ν with a robust and essentially division–free parameterized arithmetic circuit whose parameter domain is the image of ν. If we specialize the parameters of this circuit into ζ we obtain an ordinary arithmetic circuit in $\mathbb{C}[X_1]$ which evaluates the primitive integral of Q.

Therefore procedures may be used to compute primitive integrals by limits.

We are now going to show that any procedure for the computation of primitive integrals becomes intrinsically inefficient.

THEOREM 15. *Let $\mathcal{A}$ be a procedure for the computation of primitive integrals as above and let $d \in \mathbb{N}$. Then there exists a constructible subset $\mathcal{M}$ of an affine space and a polynomial $P_d \in \mathbb{C}\langle\mathcal{M}\rangle[X_1]$ of degree d such that P can be evaluated by a robust and essentially division–free by $\mathcal{M}$ parameterized arithmetic circuit β_d of size $O(\log d)$ and the size of $\mathcal{A}(\beta_d)$ is at least $\Omega(d)$.*

PROOF. As in Example 1 let $\mathcal{S}_1 : \mathcal{F} \to \mathcal{G}$ the specification corresponding to derivation and let $\mathcal{A}_1$ be the algorithm implementing $\mathcal{S}_1$ which is defined by the

forward mode of automatic differentiation. Thus $\mathcal{S}_1 \circ \mathcal{S} : \mathcal{G} \to \mathcal{G}$ is the identity specification and the composition $\mathcal{B}$ of $\mathcal{A}_1$ and $\mathcal{A}$ represents a procedure which implements this specification.

Let $\mathcal{M} := \mathbb{A}^1$, $d \in \mathbb{N}$ and $P_d := (T^{d+1} - 1) \sum_{0 \leq k \leq d} T^k X_1^k$.

We interpret P_d as a polynomial of $\mathbb{C}\langle \mathcal{M} \rangle [X_1]$. Observe that P_d can be evaluated by an ordinary division–free arithmetic circuit β_d of size $O(\log d)$. The size of $\mathcal{B}(\beta) = \mathcal{A}_1(\mathcal{A}(\beta))$ is at most three times the size of $\mathcal{A}(\beta)$. On the other hand we may interpret $\mathcal{B}(\beta)$ as the composition of the vector of parameters ν of $\mathcal{A}(\beta)$ with a robust and essentially division–free parameterized arithmetic circuit γ whose parameter domain is the image of ν. Let m be the vector length of ν. Since $\mathcal{A}$ is a procedure there exists a geometrically robust constructible map which, composed with the coefficient vector of P_d, yields ν. Mimicking now the proof of [**GHMS11**], Proposition 22 we see that $m \geq d + 1$ holds. This implies that the size of γ and hence that of $\mathcal{B}(\beta)$ and $\mathcal{A}(\beta)$ is at least of order $\Omega(d)$. □

6. Applications to elimination theory

In this section we apply our conceptual tools to the discussion of the complexity of some basic problems in computational elimination theory.

Example 3. Let $\mathcal{G}$ be the subfunctor of $\mathcal{O}(\mathcal{M})$ which associates at each constructible subset $\mathcal{M}$ of an affine space the set of all elements of $\mathcal{O}(\mathcal{M})$ whose entries are all zero except for an $(n+1)$–tuple of polynomials $G^{(1)}, \ldots, G^{(n)}, H \in \mathbb{C}\langle \mathcal{M} \rangle [X_1, \ldots, X_n]$ such that for any point $z \in \mathcal{M}$ the ideal of $\mathbb{C}[X_1, \ldots, X_n]$ generated by $G^{(1)}(z, X_1, \ldots, X_n)$, $\ldots, G^{(n)}(z, X_1, \ldots, X_n)$ is radical and of dimension zero. Moreover we require that the zero–dimensional algebraic variety defined by $G^{(1)}(z, X_1, \ldots, X_n), \ldots, G^{(n)}(z, X_1, \ldots, X_n)$ has for any $z \in \mathcal{M}$ the same number of points.

Furthermore, let $\mathcal{F} := \mathcal{O}$. For $\mathcal{M}$ as above let $\mathcal{S}(\mathcal{M}) : \mathcal{G}(\mathcal{M}) \to \mathcal{F}(\mathcal{M})$ be the map which assigns to any $G \in \mathcal{G}(\mathcal{M})$ the element $F = (F_n)_{n \geq 0}$ of $\mathcal{F}(\mathcal{M})$ where all entries except F_{n+1} are zero and where F_{n+1} belongs to $\mathbb{C}\langle \mathcal{M} \rangle [X_{n+1}]$ and satisfies for any point $z \in \mathcal{M}$ the following condition:

$$F_{n+1}(z, X_{n+1}) = \prod_{\substack{\xi \in \{G^{(1)}(z, X_1, \ldots, X_n) = 0, \ldots, \\ G^{(n)}(z, X_1, \ldots, X_n) = 0\}}} (X_{n+1} - H(z, \xi)).$$

Observe that the Implicit Function Theorem implies that there really exists such a polynomial $F_{n+1} \in \mathbb{C}\langle \mathcal{M} \rangle [X_{n+1}]$ and that $\mathcal{S}$ is a continuous specification of an elimination task.

Typical branching–parsimonious elimination methods implement restrictions of the continuous specification $\mathcal{S}$ to subfunctors of $\mathcal{G}$. They are all procedures.

Let $\mathcal{A}$ be a procedure which implements $\mathcal{S}$. We are now going to discuss a mathematical property of $\mathcal{A}$ which is, in terms of software engineering, a quality attribute of $\mathcal{A}$. To this end, let $\mathcal{M}$ be a constructible subset of an affine space, G an element of $\mathcal{G}(\mathcal{M})$ given by polynomials $G^{(1)}, \ldots, G^{(n)}$ and H of $\mathbb{C}\langle \mathcal{M} \rangle [X_1, \ldots, X_n]$ and let β be a robust and essentially division–free parameterized arithmetic circuit with parameter domain $\mathcal{M}$ which computes the polynomials $G^{(1)}, \ldots, G^{(n)}$ and H. Then $\mathcal{S}(\mathcal{M})(G)$ is given by a single polynomial $F_{n+1} \in \mathbb{C}\langle \mathcal{M} \rangle [X_{n+1}]$ as above and $\mathcal{A}(\beta)$ is a robust and essentially division–free arithmetic circuit with parameter

domain $\mathcal{M}$ which computes F_{n+1}. The parameters of the circuit $\mathcal{A}(\beta)$ form the entries of a geometrically robust constructible map ψ with domain of definition $\mathcal{M}$.

Let us consider a (not necessarily convergent) sequence $(z_k)_{k\in\mathbb{N}}$ of points $z_k \in \mathcal{M}$ such that $(G^{(1)}(z_k, X_1, \ldots, X_n), \ldots, G^{(n)}(z_k, X_1, \ldots, X_n), H(z_k, X_1, \ldots, X_n))_{k\in\mathbb{N}}$ converges for some $z \in \mathcal{M}$ to $(G^{(1)}(z, X_1, \ldots, X_n), \ldots, G^{(n)}(z, X_1, \ldots, X_n), H(z, X_1, \ldots, X_n))$. Then the assumption that $\mathcal{A}$ is a procedure implies that $(\psi(z_k))_{k\in\mathbb{N}}$ converges to $\psi(z)$.

This means that $\mathcal{A}$ transforms any approximative computation (in the sense of [**Ald84**] and [**Lic90**] §A) which for some $z \in \mathcal{M}$ represents the polynomials $G^{(1)}(z, X_1, \ldots, X_n), \ldots, G^{(n)}(z, X_1, \ldots, X_n)$ and $H(z, X_1, \ldots, X_n)$ into an ordinary arithmetic circuit computing the polynomial $F_{n+1}(z, X_{n+1})$. This constitutes a natural quality attribute of the procedure $\mathcal{A}$. This quality attribute represents also a fundamental ingredient of deformation based elimination methods and is essential for the proof of Theorem 16 below.

We are now going to explain, in terms of software engineering, in which sense all known branching–parsimonious elimination methods which implement the specification $\mathcal{S}$ are procedures. First they are all designed by means of specification languages which express only mathematical relations between polynomials and algebraic varieties as abstract data types and encapsulate their representations. Let Σ be an expression in such a language denoting a branching–parsimonious descriptive program which allows the derivation of an algorithm $\mathcal{A}$ that implements the specification $\mathcal{S}$. Using the terminology of [**LG01**] we are now going to analyze how Σ acts on the polynomials $G^{(1)}, \ldots, G^{(n)}$ and H whose circuit representations remain encapsulated. In the given situation Σ produces a branching–free non–recursive program Π of the specification language which is composed by observers and constructors and, applied to $G^{(1)}, \ldots, G^{(n)}$ and H, computes the polynomial F_{n+1}. In a first stage the observers become applied to $G^{(1)}, \ldots, G^{(n)}$ and H and precomputed intermediate results of Π which are geometrically robust constructible functions with domain of definition $\mathcal{M}$. Moreover these functions can be obtained by composing the coefficient vector of $G^{(1)}, \ldots, G^{(n)}$ and H with suitable geometrically robust constructible maps. The outputs of the observers are of the same kind. They may be combined with constructors which do not involve the indeterminate Y.

All these results become processed in a second stage in order to compute without branchings the polynomial F_{n+1} by means of the constructors contained in Π. The first and the second stage of this interpretation of the descriptive program Σ yield now a procedure which implements the continuous specification $\mathcal{S}$. In this sense, Theorem 16 below implies that any arithmetic circuit based elimination method, designed by commonly accepted rules of software engineering, needs exponential time to solve the computational task given by the specification $\mathcal{S}$ when we require that outputs are represented by robust parameterized arithmetic circuits.

Here a word of caution is at order: the method used to prove Theorem 16 indicates only that the design of branching–parsimonious algorithms by means of specifications like above leads to an exponential complexity blow up. The question remains open whether there exist more efficient alternative algorithms which do not encapsulate the representation of polynomials. The previous Theorem 8 contains a condition of branching–parsimoniousness which cannot be satisfied by such algorithms (if they exist).

6.1. A hard elimination problem. Let $n \in \mathbb{N}$ and $S_1, \ldots, S_n, T, U_1, \ldots, U_n$ and $X_1, \ldots, X_n$ be indeterminates. Let $U := (U_1, \ldots, U_n)$, $S := (S_1, \ldots, S_n)$, $X := (X_1, \ldots, X_n)$ and $G^{(1)} := X_1^2 - X_1 - S_1, \ldots, G^{(n)} := X_n^2 - X_n - S_n$, $H := \sum_{1 \leq i \leq n} 2^{i-1} X_i + T \prod_{1 \leq i \leq n} (1 + (U_i - 1)X_i)$.

Observe that the polynomials $G^{(1)}, \ldots, G^{(n)}$ form a reduced regular sequence in $\mathbb{C}[S, T, U, X]$ and that they define a subvariety V of the affine space $\mathbb{A}^n \times \mathbb{A}^1 \times \mathbb{A}^n \times \mathbb{A}^n$ which is isomorphic to $\mathbb{A}^n \times \mathbb{A}^1 \times \mathbb{A}^n$ and hence irreducible and of dimension $2n+1$. Moreover, the morphism $V \to \mathbb{A}^n \times \mathbb{A}^1 \times \mathbb{A}^n$ which associates to any point $(s, t, u, x) \in V$ the point (s, t, u), is finite and generically unramified. Therefore the morphism $\pi : V \to \mathbb{A}^n \times \mathbb{A}^1 \times \mathbb{A}^n \times \mathbb{A}^1$ which associates to any $(s, t, u, x) \in V$ the point $(s, t, u, H(t, u, x)) \in \mathbb{A}^n \times \mathbb{A}^1 \times \mathbb{A}^n \times \mathbb{A}^1$ is finite and its image $\pi(V)$ is a hypersurface of $\mathbb{A}^n \times \mathbb{A}^1 \times \mathbb{A}^n \times \mathbb{A}^1$ with irreducible minimal equation $P \in \mathbb{C}[S, T, U, Y]$.

Thus, P is an irreducible elimination polynomial of degree 2^n. Therefore any equation of $\mathbb{C}[S, T, U, Y]$ which defines $\pi(V)$ in $\mathbb{A}^n \times \mathbb{A}^1 \times \mathbb{A}^n \times \mathbb{A}^1$ is up to a scalar factor a power of P.

The equations $G^{(1)} = 0, \ldots, G^{(n)} = 0$ and the polynomial H represent a so called *flat family of zero–dimensional elimination problems* with associated elimination polynomial P (see [**HKR13**], Section 4.1 for the notion of a flat family of zero–dimensional elimination problems).

We consider again the continuous specification $\mathcal{S} : \mathcal{G} \to \mathcal{F}$ of Example 3. Let $\mathcal{M} := \{(s_1, \ldots, s_n); s_1 \neq -\frac{1}{4}, \ldots, s_n \neq -\frac{1}{4}\} \times \mathbb{A}^1 \times \mathbb{A}^n$. We interpret $G^{(1)}, \ldots, G^{(n)}, H$ and P as polynomials with coefficients in $\mathbb{C}\langle \mathcal{M} \rangle$. Ones sees easily that there exists an element G of $\mathcal{G}(\mathcal{M})$ whose entries are all zero except $n+1$ of them which are the polynomials $G^{(1)}, \ldots, G^{(n)}$ and H.

The entries of $F := \mathcal{S}(\mathcal{M})(G)$ are all zero except one which is the polynomial P.

Let $\mathcal{A}$ be a procedure of our computation model which implements the continuous specification $\mathcal{S}$. Then we have the following complexity result.

THEOREM 16. *There exist an ordinary division–free arithmetic circuit β of size $O(n)$ over $\mathbb{C}$ with inputs $S_1, \ldots, S_n$, T, $U_1, \ldots, U_n$, $X_1, \ldots, X_n$ and final results $G^{(1)}, \ldots, G^{(n)}, H$. The robust and essentially division–free, parameterized arithmetic circuit $\gamma := \mathcal{A}(\beta)$ depends on the basic parameters $S_1, \ldots, S_n$, T, $U_1, \ldots, U_n$ and the input X_{n+1} and its single final result is the polynomial P. The circuit γ has size at least $\Omega(2^n)$.*

The proof of Theorem 16 is similar as that of Theorem 8. Moreover, Theorem 16 implies that the Kronecker algorithm is an asymptotically optimal procedure. For details we refer the reader to [**HKR13**], Section 4, where also other examples of elimination problems are exhibited which are hard for procedures of our computation model.

Branching–parsimoniousness is a substantial ingredient of the proof of Theorem 16. If we allow arbitrary branchings, the number of arithmetic operations necessary to solve certain elimination problems may become polynomial in the size of the input (see [**GriHK12**]).

References

[Ald84] A. Alder. *Grenzrang und Grenzkomplexität aus algebraischer und topologischer Sicht.* PhD thesis, Universität Zürich, Philosophische Fakultät II, 1984.

[BCS97] Peter Bürgisser, Michael Clausen, and M. Amin Shokrollahi, *Algebraic complexity theory*, Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], vol. 315, Springer-Verlag, Berlin, 1997. With the collaboration of Thomas Lickteig. MR1440179 (99c:68002)

[CGH89] Leandro Caniglia, André Galligo, and Joos Heintz, *Some new effectivity bounds in computational geometry*, Applied algebra, algebraic algorithms and error-correcting codes (Rome, 1988), Lecture Notes in Comput. Sci., vol. 357, Springer, Berlin, 1989, pp. 131–151, DOI 10.1007/3-540-51083-4_54. MR1008498 (90j:13001)

[CGH03] D. Castro, M. Giusti, J. Heintz, G. Matera, and L. M. Pardo, *The hardness of polynomial equation solving*, Found. Comput. Math. **3** (2003), no. 4, 347–420, DOI 10.1007/s10208-002-0065-7. MR2009683 (2004k:68056)

[DFGS91] Alicia Dickenstein, Noaï Fitchas, Marc Giusti, and Carmen Sessa, *The membership problem for unmixed polynomial ideals is solvable in single exponential time*, Discrete Appl. Math. **33** (1991), no. 1-3, 73–94, DOI 10.1016/0166-218X(91)90109-A. Applied algebra, algebraic algorithms, and error-correcting codes (Toulouse, 1989). MR1137741 (92m:13025)

[FIK86] T. Freeman, G. Imirzian, E. Kaltofen. A system for manipulating polynomials given by straight-line programs. In *Proceedings of the fifth ACM Symposium on Symbolic and Algebraic Computation*, SYMSAC '86, 169–175. ACM, 1986.

[GW08] Andreas Griewank and Andrea Walther, *Evaluating derivatives*, 2nd ed., Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2008. Principles and techniques of algorithmic differentiation. MR2454953 (2011a:65053)

[GH01] Marc Giusti and Joos Heintz, *Kronecker's smart, little black boxes*, Foundations of computational mathematics (Oxford, 1999), London Math. Soc. Lecture Note Ser., vol. 284, Cambridge Univ. Press, Cambridge, 2001, pp. 69–104. MR1836615 (2002e:65075)

[GHH97] M. Giusti, J. Heintz, K. Hägele, J. E. Morais, L. M. Pardo, and J. L. Montaña, *Lower bounds for Diophantine approximations*, J. Pure Appl. Algebra **117/118** (1997), 277–317, DOI 10.1016/S0022-4049(97)00015-7. Algorithms for algebra (Eindhoven, 1996). MR1457843 (99d:68106)

[GHM98] M. Giusti, J. Heintz, J. E. Morais, J. Morgenstern, and L. M. Pardo, *Straight-line programs in geometric elimination theory*, J. Pure Appl. Algebra **124** (1998), no. 1-3, 101–146, DOI 10.1016/S0022-4049(96)00099-0. MR1600277 (99d:68128)

[GHMP95] M. Giusti, J. Heintz, J. E. Morais, and L. M. Pardo, *When polynomial equation systems can be "solved" fast?*, (Paris, 1995), Lecture Notes in Comput. Sci., vol. 948, Springer, Berlin, 1995, pp. 205–231, DOI 10.1007/3-540-60114-7_16. MR1448166 (98a:68106)

[GHMP97] Marc Giusti, Joos Heintz, Jose Enrique Morais, and Luis Miguel Pardo, *Le rôle des structures de données dans les problèmes d'élimination*, C. R. Acad. Sci. Paris Sér. I Math. **325** (1997), no. 11, 1223–1228, DOI 10.1016/S0764-4442(97)83558-6 (French, with English and French summaries). MR1490129 (98j:68068)

[GHMS11] Nardo Giménez, Joos Heintz, Guillermo Matera, and Pablo Solernó, *Lower complexity bounds for interpolation algorithms*, J. Complexity **27** (2011), no. 2, 151–187, DOI 10.1016/j.jco.2010.10.003. MR2776490 (2012b:41006)

[GLS01] Marc Giusti, Grégoire Lecerf, and Bruno Salvy, *A Gröbner free alternative for polynomial system solving*, J. Complexity **17** (2001), no. 1, 154–211, DOI 10.1006/jcom.2000.0571. MR1817612 (2002b:68123)

[GriHK12] R. Grimson, J. Heintz, and B. Kuijpers, *Evaluating geometric queries using few arithmetic operations*, Appl. Algebra Engrg. Comm. Comput. **23** (2012), no. 3-4, 179–193, DOI 10.1007/s00200-012-0172-x. MR3000508

[HKR13] Joos Heintz, Bart Kuijpers, and Andrés Rojas Paredes, *Software Engineering and complexity in effective algebraic geometry*, J. Complexity **29** (2013), no. 1, 92–138, DOI 10.1016/j.jco.2012.04.005. MR2997853

[HM93] Joos Heintz and Jacques Morgenstern, *On the intrinsic complexity of elimination theory*, J. Complexity **9** (1993), no. 4, 471–498, DOI 10.1006/jcom.1993.1031. MR1250549 (94k:12012)

[HMW01] Joos Heintz, Guillermo Matera, and Ariel Waissbein, *On the time-space complexity of geometric elimination procedures*, Appl. Algebra Engrg. Comm. Comput. **11** (2001), no. 4, 239–296, DOI 10.1007/s002000000046. MR1818975 (2002c:68108)

[HS81] Joos Heintz and Malte Sieveking, *Absolute primality of polynomials is decidable in random polynomial time in the number of variables*, Automata, languages and programming (Akko, 1981), Lecture Notes in Comput. Sci., vol. 115, Springer, Berlin, 1981, pp. 16–28. MR635127 (83m:12004)

[Ive73] Birger Iversen, *Generic local structure of the morphisms in commutative algebra*, Lecture Notes in Mathematics, Vol. 310, Springer-Verlag, Berlin, 1973. MR0360575 (50 #13023)

[Kal88] Erich Kaltofen, *Greatest common divisors of polynomials given by straight-line programs*, J. Assoc. Comput. Mach. **35** (1988), no. 1, 231–264, DOI 10.1145/42267.45069. MR926181 (89e:12002)

[Kro82] L. Kronecker. Grundzüge einer arithmetischen Theorie der algebraischen Grössen (Fundamentals of an arithmetic theory of algebraic quantities). *J. Reine Angew. Math.*, 92 1–122, 1882.

[Kun85] Ernst Kunz, *Introduction to commutative algebra and algebraic geometry*, Birkhäuser Boston Inc., Boston, MA, 1985. Translated from the German by Michael Ackerman; With a preface by David Mumford. MR789602 (86e:14001)

[Lan84] Serge Lang, *Algebra*, 2nd ed., Addison-Wesley Publishing Company Advanced Book Program, Reading, MA, 1984. MR783636 (86j:00003)

[Lec] G. Lecerf. Kronecker: a Magma package for polynomial system solving. `http://lecerf.perso.math.cnrs.fr/software/kronecker/index.html`.

[Lic90] T. M. Lickteig. On semialgebraic decision complexity. Habilitationsschrift, Universität Tübingen TR-90-052, Int. Comp. Sc. Inst., Berkeley, 1990.

[LG01] B. Liskov, J. Guttag. *Program development in Java: Specification, and Object–Oriented Design*. Third Edition, Addison-Wesley, 2001.

[Mac16] F. S. Macaulay, *The algebraic theory of modular systems*, Cambridge Mathematical Library, Cambridge University Press, Cambridge, 1994. Revised reprint of the 1916 original; With an introduction by Paul Roberts. MR1281612 (95i:13001)

[Mitchell] Barry Mitchell, *Theory of categories*, Pure and Applied Mathematics, Vol. XVII, Academic Press, New York, 1965. MR0202787 (34 #2647)

[Mor03] Teo Mora, *Solving polynomial equation systems. I*, Encyclopedia of Mathematics and its Applications, vol. 88, Cambridge University Press, Cambridge, 2003. The Kronecker-Duval philosophy. MR1966700 (2004d:12001)

[Mor05] Teo Mora, *Solving polynomial equation systems. II*, Encyclopedia of Mathematics and its Applications, vol. 99, Cambridge University Press, Cambridge, 2005. Macaulay's paradigm and Gröbner technology. MR2164357 (2006k:13059)

[Mum88] David Mumford, *The red book of varieties and schemes*, Lecture Notes in Mathematics, vol. 1358, Springer-Verlag, Berlin, 1988. MR971985 (89k:14001)

[PS73] Michael S. Paterson and Larry J. Stockmeyer, *On the number of nonscalar multiplications necessary to evaluate polynomials*, SIAM J. Comput. **2** (1973), 60–66. MR0314238 (47 #2790)

[Sha94] Igor R. Shafarevich, *Basic algebraic geometry. 1*, 2nd ed., Springer-Verlag, Berlin, 1994. Varieties in projective space; Translated from the 1988 Russian edition and with notes by Miles Reid. MR1328833 (95m:14001)

[SS95] Michael Shub and Steve Smale, *On the intractability of Hilbert's Nullstellensatz and an algebraic version of "*NP $\neq$ P*?"*, Duke Math. J. **81** (1995), no. 1, 47–54 (1996), DOI 10.1215/S0012-7094-95-08105-8. A celebration of John F. Nash, Jr. MR1381969 (97h:03067)

[vdW50] B. L. van der Waerden. *Modern Algebra II*. Ungar, New York, 1950.

[ZS60] O. Zariski, P. Samuel. *Commutative algebra II*, 39. Springer, New York, 1960.

Departamento de Computación, Universidad de Buenos Aires and CONICET, Ciudad Universitaria, Pab.I, 1428 Buenos Aires, Argentina, and Departamento de Matemáticas, Estadística y Computación, Facultad de Ciencias, Universidad de Cantabria, Avda. de los Castros, s/n, E-39005 Santander, Spain
E-mail address: joos@dc.uba.ar

Database and Theoretical Computer Science Research Group, Hasselt University, Agoralaan, Gebouw D, 3590 Diepenbeek, Belgium.
E-mail address: bart.kuijpers@uhasselt.be

Departamento de Computación, Universidad de Buenos Aires, Ciudad Universitaria, Pab.I, 1428 Buenos Aires, Argentina
E-mail address: arojas@dc.uba.ar

Contemporary Mathematics
Volume **604**, 2013
http://dx.doi.org/10.1090/conm/604/12072

Newton iteration, conditioning and zero counting

Gregorio Malajovich

Abstract. Those lectures revolve around the following problem: given a system of n real polynomials in n variables, count the number of real roots. The first lecture is a course on Newton iteration and alpha-theory. The second describes an inclusion-exclusion algorithm for real polynomials, developed by Felipe Cucker, Teresa Krick, Mario Wschebor and myself. The third lecture introduces tools for complexity analysis of numerical algorithms, and uses those tools to analyze our root-counting algorithm.

1. Introduction

Mathematicians' obsession with counting led to many interesting and far-fetched problems. These lectures are structured around a seemingly innocent counting problem:

Problem 1.1 (Real root counting). Given a system $\mathbf{f} = (f_1, \ldots, f_n)$ of real polynomial equations in n variables, count the number of real solutions.

You can also find here a crash-course in Newton iteration. We will state and analyze a Newton iteration based 'inclusion-exclusion' algorithm to count (and find) roots of real polynomials.

That algorithm was investigated in a sequence of three papers by Felipe Cucker, Teresa Krick, Mario Wschebor and myself (**2008**, **2009**, **2012**). Good numerical properties are proved in the first paper. For instance, the algorithm is tolerant to controlled rounding error. Instead of covering such technicalities, I will present a simplified version and focus on the main ideas.

The interest of Problem 1.1 lies in the fact that it is **complete** for the complexity class $\#\mathbf{P}_{\mathbb{R}}$ over the **BSS** (Blum-Shub-Smale) computation model over $\mathbb{R}$. See **Blum et al. (1998**) for the BSS model of computation. The class $\#\mathbf{P}_{\mathbb{R}}$ was defined by Meer (**2000**) as the class of all functions $f : \mathbb{R}^\infty \to \{0,1\}^\infty \cup \{\infty\}$ such that there exists a BSS machine M working in polynomial time and a polynomial

2010 *Mathematics Subject Classification.* Primary 65H10, Secondary 65H20.

Lecture notes for the Santaló summer school on Recent Advances in Real Complexity and Computation, held at the Palacio de la Magdalena, Santander, and sponsored by the Universidad Internacional Menéndez Pelayo and the Universidad de Cantábria.

The author was partially supported by CNPq and CAPES (Brazil) and by the MathAmSud grant *complexity*.

q satisfying

$$f(\mathbf{y}) = \#\{\mathbf{z} \in \mathbb{R}^{q(\text{size}(\mathbf{y})} : M(\mathbf{y}, \mathbf{z}) \text{ is an accepting computation.}\}$$

We refer to **Bürgisser and Cucker (2006)** for the proof of completeness and to **Cucker et al. (2008)** for references on the subject of counting zeros.

Counting real polynomial roots in $\mathbb{R}^n$ can be reduced to counting polynomial roots in $\mathbb{S}^{n+1}$. Given a degree d polynomial $f(x_1, \ldots, x_n)$, its homogenization is $f^{\text{homo}}(x_0, \ldots, x_n) = x_0^d f(x_1/x_0, \ldots, x_n/x_0)$.

Exercise 1.1 (Beware of infinity[1]). Find an homogeneous polynomial $g = g(\mathbf{y}, u)$ of degree 2 in $n + 2$ variables such that

$$\#\{\mathbf{x} \in \mathbb{R}^n : f_1(\mathbf{x}) = \cdots = f_n(\mathbf{x}) = 0\} + 1 = \\ = \frac{1}{2}\#\{(\mathbf{y}, u) \in \mathbb{S}^{n+1} : f_1^{\text{homo}}(\mathbf{y}) = \cdots = f_n^{\text{homo}}(\mathbf{y}) = g(\mathbf{y}, u) = 0\}.$$

Because of the exercise above, replacing n by $n - 1$, Problem 1.1 reduces to:

Problem 1.2 (Real root counting on S^n). Given a system $\mathbf{f} = (f_1, \ldots, f_n)$ of real homogeneous polynomial equations in $n + 1$ variables, count the number of solutions in S^n.

These lectures are organized as follows. We start by a review of **alpha-theory**. This theory originated with a couple of theorems proved by Steve Smale (**2006**) and improved subsequently by several authors. It allows to guarantee (quantitatively) from the available data that Newton iterations will converge quadratically to the solution of a system of equations.

Then I will speak about the inclusion-exclusion algorithm. It uses crucially several results of alpha-theory.

The complexity of the inclusion-exclusion algorithm depends upon a condition number. By endowing the input space with a probability distribution, one can speak of the expected value of the condition number and of the expected running time. The final section is a review of the complexity analysis performed in **Cucker et al. (2009)** and **Cucker et al. (2012)**.

A warning: these lectures are informal. The model of computation is **cloud computing**. This means that we will allow for exponentially many parallel processors (essentially, BSS machines) at no additional cost. Moreover, we will be informal in the sense that we will assume that square roots and operator norms can be computed exactly in finite time. While this does not happen in the BSS model, those can be approximated and all our algorithms can be rewritten as rigorous BSS algorithms at the cost of a harder complexity analysis (**Cucker et al., 2008**).

Exercise 1.2. What would happen if you could design a true polynomial time algorithm to solve Problem 1.2?

Acknowledgments. I would like to thank Teresa Krick, Felipe Cucker, Mike Shub and an anonymous referee for pointing out some mistakes in previous versions.

[1]**Hint for exercise 1.1:** Try $g(y, u) = y_0^2 + y_1^2 + \cdots + y_n^2 - y_0 u$.

Contents

Part 1. Newton Iteration and Alpha theory

2. Outline

Let $\mathbf{f}$ be a mapping between Banach spaces. **Newton Iteration** is defined by

$$N(\mathbf{f}, \mathbf{x}) = \mathbf{x} - D\mathbf{f}(\mathbf{x})^{-1}\mathbf{f}(\mathbf{x})$$

wherever $D\mathbf{f}(\mathbf{x})^{-1}$ exists and is bounded. Its only possible fixed points are those satisfying $\mathbf{f}(\mathbf{x}) = 0$. When $\mathbf{f}(\mathbf{x}) = 0$ and $D\mathbf{f}(\mathbf{x})$ is invertible, we say that $\mathbf{x}$ is a **nondegenerate zero** of $\mathbf{f}$.

It is well-known that Newton iteration is quadratically convergent in a neighborhood of a nondegenerate zero ζ. Indeed, $N(\mathbf{f}, \mathbf{x}) - \zeta = \frac{1}{2}D^2\mathbf{f}(\zeta)(\mathbf{x} - \zeta)^2 + \cdots$.

There are two main approaches to quantify how fast is quadratic convergence. One of them, pioneered by **Kantorovich (1996)** assumes that the mapping $\mathbf{f}$ has a bounded second derivative, and that this bound is known.

The other approach, developed by Smale (**1985**, **2006**) and described here, assumes that the mapping $\mathbf{f}$ is analytic. Then we will be able to estimate a neighborhood of quadratic convergence around a given zero (Theorem 4.2) or to certify an 'approximate root' (Theorem 5.3) from data that depends only on the value and derivatives of $\mathbf{f}$ at one point.

A more general exposition on this subject may be found in **Dedieu (1997b)**, covering also overdetermined and undetermined polynomial systems.

3. The gamma invariant

Throughout this chapter, $\mathbb{E}$ and $\mathbb{F}$ are Banach spaces, $\mathcal{D} \subseteq \mathbb{E}$ is open and $\mathbf{f} : \mathbb{E} \to \mathbb{F}$ is analytic.

This means that if $\mathbf{x}_0 \in \mathbb{E}$ is in the domain of $\mathbb{E}$, then there is $\rho > 0$ with the property that the series

$$\mathbf{f}(\mathbf{x}_0) + \mathbf{D}f(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0) + \frac{1}{2}\mathbf{D}^2 f(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0, \mathbf{x} - \mathbf{x}_0) + \cdots \tag{1}$$

converges uniformly for $\|\mathbf{x} - \mathbf{x}_0\| < \rho$, and its limit is equal to $\mathbf{f}(\mathbf{x})$ (For more details about analytic functions between Banach spaces, see Nachbin (**1964**, **1969**)).

In order to abbreviate notations, we will write (1) as

$$\mathbf{f}(\mathbf{x}_0) + \mathbf{D}f(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0) + \sum_{k \geq 2} \frac{1}{k!}\mathbf{D}^k f(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0)^k$$

where the exponent k means that $\mathbf{x} - \mathbf{x}_0$ appears k times as an argument to the preceding multi-linear operator.

The maximum of such ρ will be called the **radius of convergence**. (It is ∞ when the series (1) is globally convergent). This terminology comes from univariate complex analysis. When $\mathbf{E} = \mathbb{C}$, the series will converge for all $\mathbf{x} \in B(\mathbf{x}_0, \rho)$ and diverge for all $\mathbf{x} \notin \overline{B(\mathbf{x}_0, \rho)}$. This is no longer true in several complex variables, or Banach spaces (Exercise 4.1).

The norm of a k-linear operator in Banach Spaces (such as the k-th derivative) is the **operator norm**, for instance

$$\|D^k \mathbf{f}(\mathbf{x}_0)\|_{\mathbb{E} \to \mathbb{F}} = \sup_{\|\mathbf{u}_1\|_{\mathbb{E}} = \cdots = \|\mathbf{u}_k\|_{\mathbb{E}} = 1} \|D^k \mathbf{f}(\mathbf{x}_0)(\mathbf{u}_1, \ldots, \mathbf{u}_k)\|_{\mathbb{F}}.$$

As long as there is no ambiguity, we drop the subscripts of the norm.

Definition 3.1 (Smale's γ invariant). Let $\mathbf{f} : \mathcal{D} \subseteq \mathbb{E} \to \mathbb{F}$ be an analytic mapping between Banach spaces, and $\mathbf{x}_0 \in \mathcal{D}$. When $D\mathbf{f}(\mathbf{x}_0)$ is invertible, define

$$\gamma(\mathbf{f}, \mathbf{x}_0) = \sup_{k \geq 2} \left(\frac{\|D\mathbf{f}(\mathbf{x}_0)^{-1} D^k \mathbf{f}(\mathbf{x}_0)\|}{k!} \right)^{\frac{1}{k-1}}.$$

Otherwise, set $\gamma(\mathbf{f}, \mathbf{x}_0) = \infty$.

In the one variable setting, this can be compared to the radius of convergence ρ of $\mathbf{f}'(\mathbf{x})/\mathbf{f}'(\mathbf{x}_0)$, that satisfies

$$\rho^{-1} = \limsup_{k \geq 2} \left(\frac{\|\mathbf{f}'(\mathbf{x}_0)^{-1} \mathbf{f}^{(k)}(\mathbf{x}_0)\|}{k!} \right)^{\frac{1}{k-1}}.$$

More generally,

Proposition 3.2. *Let $\mathbf{f} : \mathcal{D} \subseteq \mathbb{E} \to \mathbb{F}$ be a C^∞ map between Banach spaces, and $\mathbf{x}_0 \in \mathcal{D}$. Then f is analytic in x_0 if and only if, $\gamma(f, x_0)$ is finite. The series*

$$\mathbf{f}(\mathbf{x}_0) + \mathbf{D}f(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0) + \sum_{k \geq 2} \frac{1}{k!}\mathbf{D}^k f(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0)^k \tag{2}$$

is uniformly convergent for $\mathbf{x} \in B(\mathbf{x}_0, \rho)$ for any $\rho < 1/\gamma(\mathbf{f}, \mathbf{x}_0))$.

PROOF OF THE **if** IN PROP.3.2. The series

$$\mathbf{D}f(\mathbf{x}_0)^{-1}\mathbf{f}(\mathbf{x}_0) + (\mathbf{x}-\mathbf{x}_0) + \sum_{k\geq 2}\frac{1}{k!}\mathbf{D}f(\mathbf{x}_0)^{-1}\mathbf{D}^k f(\mathbf{x}_0)(\mathbf{x}-\mathbf{x}_0)^k$$

is uniformly convergent in $B(\mathbf{x}_0,\rho)$ where

$$\begin{aligned}\rho^{-1} &< \limsup_{k\geq 2}\left(\frac{\|D\mathbf{f}(\mathbf{x}_0)^{-1}D^k\mathbf{f}(\mathbf{x}_0)\|}{k!}\right)^{\frac{1}{k}}\\ &\leq \limsup_{k\geq 2}\gamma(\mathbf{f},\mathbf{x}_0)^{\frac{k-1}{k}}\\ &= \lim_{k\to\infty}\gamma(\mathbf{f},\mathbf{x}_0)^{\frac{k-1}{k}}\\ &= \gamma(\mathbf{f},\mathbf{x}_0)\end{aligned}$$

□

Before proving the **only if** part of Proposition 3.2, we need to relate the norm of a multi-linear map to the norm of the corresponding polynomial.

LEMMA 3.3. *Let $k\geq 2$. Let $\mathbf{T}:\mathbb{E}^k\to\mathbb{F}$ be k-linear and symmetric. Let $\mathbf{S}:\mathbb{E}\to\mathbb{F}$, $\mathbf{S}(\mathbf{x})=T(\mathbf{x},\mathbf{x},\ldots,\mathbf{x})$ be the corresponding polynomial. Then,*

$$\|\mathbf{T}\|\leq e^{k-1}\sup_{\|\mathbf{x}\|\leq 1}\|\mathbf{S}(\mathbf{x})\|$$

PROOF. The polarization formula for (real or complex) tensors is

$$\mathbf{T}(\mathbf{x}_1,\cdots,\mathbf{x}_k)=\frac{1}{2^k k!}\sum_{\substack{\epsilon_j=\pm 1\\ j=1,\ldots,k}}\epsilon_1\cdots\epsilon_k\mathbf{S}\left(\sum_{l=1}^k\epsilon_l\mathbf{x}_l\right)$$

It is easily derived by expanding the expression inside parentheses. There will be $2^k k!$ terms of the form

$$\epsilon_1\cdots\epsilon_k T(\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_k)$$

or its permutations. All other terms miss at least one variable (say $\mathbf{x}_j$). They cancel by summing for $\epsilon_j=\pm 1$.

It follows that when $\|\mathbf{x}\|\leq 1$,

$$\begin{aligned}\mathbf{T}(\mathbf{x}_1,\cdots,\mathbf{x}_k) &\leq \frac{1}{k!}\max_{\substack{\epsilon_j=\pm 1\\ j=1,\ldots,k}}\left\|\mathbf{S}\left(\sum_{l=1}^k\epsilon_l\mathbf{x}_l\right)\right\|\\ &\leq \frac{k^k}{k!}\sup_{\|\mathbf{x}\|\leq 1}\|\mathbf{S}(\mathbf{x})\|\end{aligned}$$

The Lemma follows from using Stirling's formula,

$$k!\geq\sqrt{2\pi k}k^k e^{-k}e^{1/(12k+1)}.$$

We obtain:

$$\|\mathbf{T}\|\leq\left(\frac{1}{\sqrt{2\pi k}}e^{-\frac{1}{12k+1}}\right)e^k\sup_{\|\mathbf{x}\|\leq 1}\|\mathbf{S}(\mathbf{x})\|.$$

Then we use the fact that $k\geq 2$, hence $\sqrt{2\pi k}\geq e$. □

PROOF OF PROP.3.2, **only if** PART. Assume that the series (2) converges uniformly for $\|\mathbf{x} - \mathbf{x}_0\| < \rho$. Without loss of generality assume that $\mathbb{E} = \mathbb{F}$ and $D\mathbf{f}(\mathbf{x}_0) = I$.

We claim that

$$\limsup_{k\geq 2} \sup_{\|\mathbf{u}\|=1} \|\frac{1}{k!} D^k \mathbf{f}(\mathbf{x}_0)\mathbf{u}^k\|^{1/k} \leq \rho^{-1}.$$

Indeed, assume that there is $\delta > 0$ and infinitely many pairs $(k, \mathbf{u})$ with $\|\mathbf{u}_i\| = 1$ and

$$\|\frac{1}{k!} D^k \mathbf{f}(\mathbf{x}_0)\mathbf{u}^k\|^{1/k} > \rho^{-1}(1+\delta).$$

In that case,

$$\|\frac{1}{k!} D^k \mathbf{f}(\mathbf{x}_0) \left(\frac{\rho}{\sqrt{1+\delta}}\mathbf{u}\right)^k\| > \left(\sqrt{1+\delta}\right)^k$$

infinitely many times, and hence (2) does not converge uniformly on $B(\mathbf{x}_0, \rho)$.

Now, we can apply Lemma 3.3 to obtain:

$$\begin{aligned}
\limsup_{k\geq 2} \|\frac{1}{k!} D^k \mathbf{f}(\mathbf{x}_0)\|^{1/(k-1)} &\leq e \limsup_{k\geq 2} \sup_{\|\mathbf{u}\|=1} \|\frac{1}{k!} D^k \mathbf{f}(\mathbf{x}_0)\mathbf{u}^k\|^{\frac{1}{k-1}} \\
&\leq e \lim_{k\to\infty} \rho^{-(1+1/(k-1))} \\
&= e\rho^{-1}
\end{aligned}$$

and therefore $\|\frac{1}{k!} D^k f(x_0)\|^{1/(k-1)}$ is bounded. □

EXERCISE 3.1. Show the polarization formula for an Hermitian product:

$$\langle \mathbf{u}, \mathbf{v} \rangle = \frac{1}{4} \sum_{\epsilon^4=1} \epsilon \|\mathbf{u} + \epsilon \mathbf{v}\|^2$$

Explain why this is different from the one in Lemma 3.3.

EXERCISE 3.2. If one drops the uniform convergence hypothesis in the definition of analytic functions, what happens to Proposition 3.2?

4. The γ-Theorems

The following concept provides a good abstraction of quadratic convergence.

DEFINITION 4.1 (Approximate zero of the first kind). Let $\mathbf{f} : \mathcal{D} \subseteq \mathbf{E} \to \mathbf{F}$ be as above, with $\mathbf{f}(\zeta) = 0$. An **approximate zero of the first kind** associated to ζ is a point $\mathbf{x}_0 \in \mathcal{D}$, such that

(1) The sequence $(\mathbf{x})_i$ defined inductively by $\mathbf{x}_{i+1} = N(\mathbf{f}, \mathbf{x}_i)$ is well-defined (each $\mathbf{x}_i$ belongs to the domain of $\mathbf{f}$ and $D\mathbf{f}(\mathbf{x}_i)$ is invertible and bounded).
(2)

$$\|\mathbf{x}_i - \zeta\| \leq 2^{-2^i+1} \|\mathbf{x}_0 - \zeta\|.$$

The existence of approximate zeros of the first kind is not obvious, and requires a theorem.

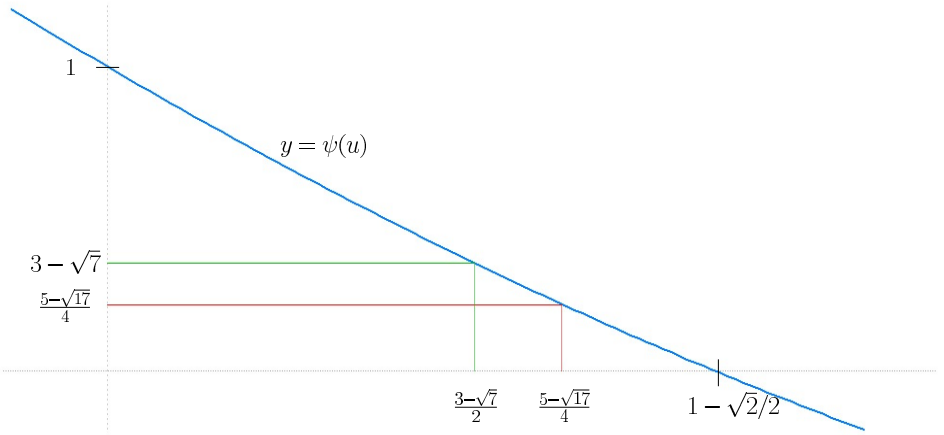


FIGURE 1. $y = \psi(u)$

THEOREM 4.2 (Smale). *Let $\mathbf{f} : \mathcal{D} \subseteq \mathbb{E} \to \mathbb{F}$ be an analytic map between Banach spaces. Let ζ be a nondegenerate zero of $\mathbf{f}$. Assume that*

$$B = B\left(\zeta, \frac{3-\sqrt{7}}{2\gamma(\mathbf{f},\zeta)}\right) \subseteq \mathcal{D}.$$

Every $\mathbf{x}_0 \in B$ is an approximate zero of the first kind associated to ζ. The constant $(3-\sqrt{7})/2$ is the smallest with that property.

Before going further, we remind the reader of the following fact.

LEMMA 4.3. *Let $d \geq 1$ be integer, and let $|t| < 1$. Then,*

$$\frac{1}{(1-t)^d} = \sum_{k\geq 0} \binom{k+d-1}{d-1} t^k.$$

PROOF. Differentiate $d-1$ times the two sides of the expression $1/(1-t) = 1 + t + t^2 + \cdots$, and then divide both sides by $d-1!$ □

LEMMA 4.4. *The function $\psi(u) = 1 - 4u + 2u^2$ is decreasing and non-negative in $[0, 1-\sqrt{2}/2]$, and satisfies:*

$$\frac{u}{\psi(u)} < 1 \qquad \text{for } u \in [0, (5-\sqrt{17})/4) \tag{3}$$

$$\frac{u}{\psi(u)} \leq \frac{1}{2} \qquad \text{for } u \in [0, (3-\sqrt{7})/2]\ . \tag{4}$$

The proof of Lemma 4.4 is left to the reader (but see Figure 1).

Another useful result is:

LEMMA 4.5. *Let A be a $n \times n$ matrix. Assume $\|A - I\|_2 < 1$. Then A has full rank and, for all y,*

$$\frac{\|y\|}{1 + \|A-I\|_2} \leq \|A^{-1}y\|_2 \leq \frac{\|y\|}{1 - \|A-I\|_2}.$$

PROOF. By hypothesis, $\|Ax\| > 0$ for all $x \neq 0$ so that A has full rank. Let $y = Ax$. By triangular inequality,

$$\|Ax\| \geq \|x\| - \|(A-I)x\| \geq (1 - \|(A-I)\|_2)\|x\|.$$

Also by triangular inequality,

$$\|Ax\| \leq \|x\| + \|(A-I)x\| \leq (1 + \|(A-I)\|_2)\|x\|.$$

□

The following Lemma will be needed:

LEMMA 4.6. *Assume that* $u = \|\mathbf{x} - \mathbf{y}\|\gamma(\mathbf{f}, \mathbf{x}) < 1 - \sqrt{2}/2$. *Then,*

$$\|D\mathbf{f}(\mathbf{y})^{-1} D\mathbf{f}(\mathbf{x})\| \leq \frac{(1-u)^2}{\psi(u)}.$$

PROOF. Expanding $\mathbf{y} \mapsto D\mathbf{f}(\mathbf{x})^{-1} D\mathbf{f}(\mathbf{y})$ around $\mathbf{x}$, we obtain:

$$D\mathbf{f}(\mathbf{x})^{-1} D\mathbf{f}(\mathbf{y}) = I + \sum_{k \geq 2} \frac{1}{k-1!} D\mathbf{f}(\mathbf{x})^{-1} D^k \mathbf{f}(\mathbf{x})(\mathbf{y} - \mathbf{x})^{k-1}.$$

Rearranging terms and taking norms, Lemma 4.3 yields

$$\|D\mathbf{f}(\mathbf{x})^{-1} D\mathbf{f}(\mathbf{y}) - I\| \leq \frac{1}{(1 - \gamma\|\mathbf{y} - \mathbf{x}\|)^2} - 1.$$

By Lemma 4.5 we deduce that $D\mathbf{f}(\mathbf{x})^{-1} D\mathbf{f}(\mathbf{y})$ is invertible, and

$$\|D\mathbf{f}(\mathbf{y})^{-1} D\mathbf{f}(\mathbf{x})\| \leq \frac{1}{1 - \|D\mathbf{f}(\mathbf{x})^{-1} D\mathbf{f}(\mathbf{y}) - I\|} = \frac{(1-u)^2}{\psi(u)}. \tag{5}$$

□

Here is the method for proving Theorem 4.2 and similar ones: first we study the convergence of Newton iteration applied to a 'universal' function. In this case, set

$$h_\gamma(t) = t - \gamma t^2 - \gamma^2 t^3 - \cdots = t - \frac{\gamma t^2}{1 - \gamma t}.$$

(See figure 2).

The function h_γ has a zero at $t = 0$, and $\gamma(h_\gamma, 0) = \gamma$. Then, we compare the convergence of Newton iteration applied to an arbitrary function to the convergence when applied to the universal function.

LEMMA 4.7. *Assume that* $0 \leq u_0 = \gamma t_0 < \frac{5-\sqrt{17}}{4}$. *Then the sequences*

$$t_{i+1} = N(h_\gamma, t_i) \text{ and } u_{i+1} = \frac{u_i^2}{\psi(u_i)}$$

are well-defined for all i, $\lim_{i\to\infty} t_i = 0$, *and*

$$\frac{|t_i|}{|t_0|} = \frac{u_i}{u_0} \leq \left(\frac{u_0}{\psi(u_0)}\right)^{2^i - 1}.$$

Moreover,

$$\frac{|t_i|}{|t_0|} \leq 2^{-2^i+1}$$

for all i *if and only if* $u_0 \leq \frac{3-\sqrt{7}}{2}$.

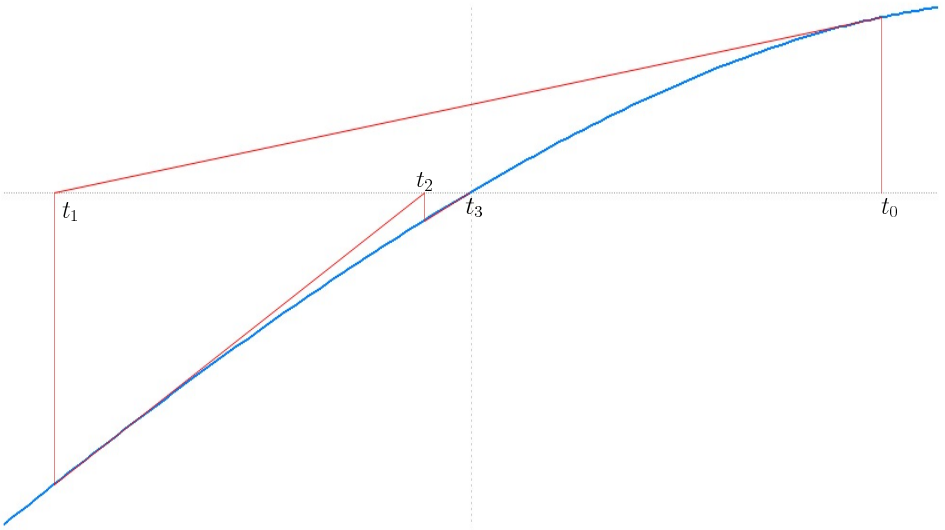


FIGURE 2. $y = h_\gamma(t)$

PROOF. We just compute

$$\begin{aligned} h'_\gamma(t) &= \frac{\psi(\gamma t)}{(1-\gamma t)^2} \\ th'_\gamma(t) - h_\gamma(t) &= -\frac{\gamma t^2}{(1-\gamma t)^2} \\ N(h_\gamma, t) &= -\frac{\gamma t^2}{\psi(\gamma t)}. \end{aligned}$$

When $u_0 < \frac{5-\sqrt{17}}{4}$, (3) implies that the sequence u_i is decreasing, and by induction

$$u_i = \gamma |t_i|.$$

Moreover,

$$\frac{u_{i+1}}{u_0} = \left(\frac{u_i}{u_0}\right)^2 \frac{u_0}{\psi(u_i)} \leq \left(\frac{u_i}{u_0}\right)^2 \frac{u_0}{\psi(u_0)} < \left(\frac{u_i}{u_0}\right)^2.$$

By induction,

$$\frac{u_i}{u_0} \leq \left(\frac{u_0}{\psi(u_0)}\right)^{2^i - 1}.$$

This also implies that $\lim t_i = 0$.

When furthermore $u_0 \leq (3-\sqrt{7})/2$, $u_0/\psi(u_0) \leq 1/2$ by (4) hence $u_i/u_0 \leq 2^{-2^i+1}$. For the converse, if $u_0 > (3-\sqrt{7})/2$, then

$$\frac{|t_1|}{|t_0|} = \frac{u_0}{\psi(u_0)} > \frac{1}{2}.$$

□

Before proceeding to the proof of Theorem 4.2, a remark is in order.

Both Newton iteration and γ are invariant with respect to translation and to linear changes of coordinates: let $\mathbf{g}(\mathbf{x}) = A\mathbf{f}(\mathbf{x} - \zeta)$, where A is a continuous and invertible linear operator from $\mathbb{F}$ to $\mathbb{E}$. Then

$$N(\mathbf{g}, \mathbf{x} + \zeta) = N(\mathbf{f}, \mathbf{x}) + \zeta \text{ and } \gamma(\mathbf{g}, \mathbf{x} + \zeta) = \gamma(\mathbf{f}, \mathbf{x}).$$

Also, distances in $\mathbb{E}$ are invariant under translation.

Proof of Th.4.2. Assume without loss of generality that $\zeta = 0$ and $D\mathbf{f}(\zeta) = I$. Set $\gamma = \gamma(\mathbf{f}, \mathbf{x})$, $u_0 = \|\mathbf{x}_0\|\gamma$, and let h_γ and the sequence (u_i) be as in Lemma 4.7.

We will bound

$$\|N(\mathbf{f}, \mathbf{x})\| = \left\|\mathbf{x} - D\mathbf{f}(\mathbf{x})^{-1}\mathbf{f}(\mathbf{x})\right\| \leq \|D\mathbf{f}(\mathbf{x})^{-1}\|\|\mathbf{f}(\mathbf{x}) - D\mathbf{f}(\mathbf{x})\mathbf{x}\|. \tag{6}$$

The Taylor expansions of $\mathbf{f}$ and $D\mathbf{f}$ around 0 are respectively:

$$\mathbf{f}(\mathbf{x}) = \mathbf{x} + \sum_{k\geq 2} \frac{1}{k!} D^k\mathbf{f}(0)\mathbf{x}^k$$

and

$$D\mathbf{f}(\mathbf{x}) = I + \sum_{k\geq 2} \frac{1}{k-1!} D^k\mathbf{f}(0)\mathbf{x}^{k-1}. \tag{7}$$

Combining the two equations, above, we obtain:

$$\mathbf{f}(\mathbf{x}) - D\mathbf{f}(\mathbf{x})\mathbf{x} = \sum_{k\geq 2} \frac{k-1}{k!} D^k\mathbf{f}(0)\mathbf{x}^k.$$

Using Lemma 4.3 with $d = 2$, the rightmost term in (6) is bounded above by

$$\|\mathbf{f}(\mathbf{x}) - D\mathbf{f}(\mathbf{x})\mathbf{x}\| \leq \sum_{k\geq 2} (k-1)\gamma^{k-1}\|\mathbf{x}\|^k = \frac{\gamma\|\mathbf{x}\|^2}{(1-\gamma\|\mathbf{x}\|)^2}. \tag{8}$$

Combining Lemma 4.6 and (8) in (6), we deduce that

$$\|N(\mathbf{f}, \mathbf{x})\| \leq \frac{\gamma\|\mathbf{x}\|^2}{\psi(\gamma\|\mathbf{x}\|)}.$$

By induction, $u_i \leq \gamma\|\mathbf{x}_i\|$. When $u_0 \leq (3 - \sqrt{7})/2$, we obtain as in Lemma 4.7 that

$$\frac{\|\mathbf{x}_i\|}{\|\mathbf{x}_0\|} \leq \frac{u_i}{u_0} \leq 2^{-2^i+1}.$$

We have seen in Lemma 4.7 that the bound above fails for $i = 1$ when $u_0 > (3 - \sqrt{7})/2$. □

Notice that in the proof above,

$$\lim_{i\to\infty} \frac{u_0}{\psi(u_i)} = u_0.$$

Therefore, convergence is actually faster than predicted by the definition of approximate zero. We proved actually a sharper result:

	1/32	1/16	1/10	1/8	$\frac{3-\sqrt{7}}{2}$
1	4.810	3.599	2.632	2.870	1.000
2	14.614	11.169	8.491	6.997	3.900
3	34.229	26.339	20.302	16.988	10.229
4	73.458	56.679	43.926	36.977	22.954
5	151.917	117.358	91.175	76.954	48.406

TABLE 1. Values of $-log_2(u_i/u_0)$ in function of u_0 and i.

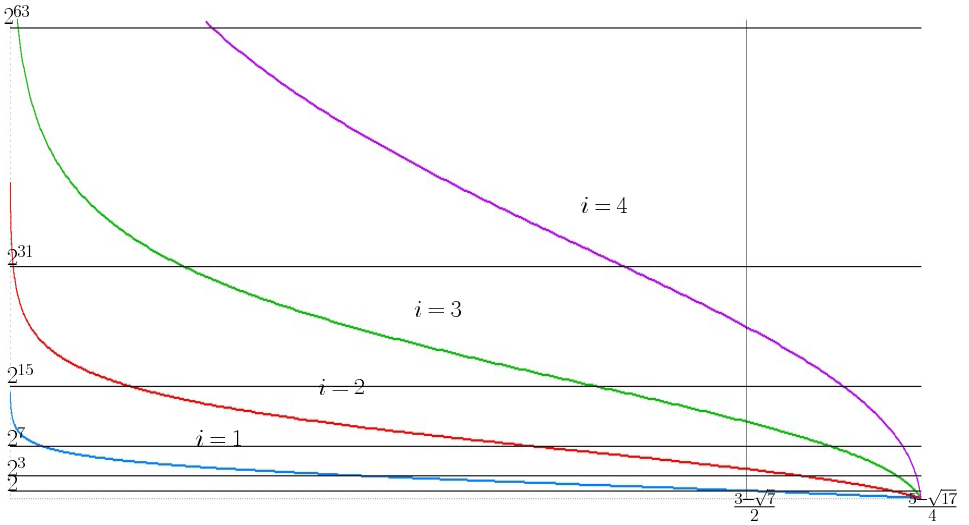


FIGURE 3. Values of $log_2(u_i/u_0)$ in function of u_0 for $i = 1, \ldots, 4$.

THEOREM 4.8. *Let $\mathbf{f} : \mathcal{D} \subseteq \mathbb{E} \to \mathbb{F}$ be an analytic map between Banach spaces. Let ζ be a nondegenerate zero of $\mathbf{f}$. Let $u_0 < (5-\sqrt{17})/4$.*

Assume that

$$B = B\left(\zeta, \frac{u_0}{\gamma(\mathbf{f}, \zeta)}\right) \subseteq \mathcal{D}.$$

If $\mathbf{x}_0 \in B$, then the sequences

$$\mathbf{x}_{i+1} = N(\mathbf{f}, \mathbf{x}_i) \text{ and } u_{i+1} = \frac{u_i^2}{\psi(u_i)}$$

are well-defined for all i, and

$$\frac{\|\mathbf{x}_i - \zeta\|}{\|\mathbf{x}_0 - \zeta\|} \leq \frac{u_i}{u_0} \leq \left(\frac{u_0}{\psi(u_0)}\right)^{-2^i+1}.$$

Table 1 and Figure 3 show how fast u_i/u_0 decreases in terms of u_0 and i.

To conclude this section, we need to address an important issue for numerical computations. Whenever dealing with digital computers, it is convenient to perform calculations in floating point format. This means that each real number is stored as a **mantissa** (an integer, typically no more than 2^{24} or 2^{53}) times an exponent. (The

IEEE-754 standard for computer arithmetic (**IEEE, 2008**) is taught at elementary numerical analysis courses, see for instance **Higham (2002**, Ch.2)).

By using floating point numbers, a huge gain of speed is obtained with regard to exact representation of, say, algebraic numbers. However, computations are inexact (by a typical factor of 2^{-24} or 2^{-53}). Therefore, we need to consider **inexact** Newton iteration. An obvious modification of the proof of Theorem 4.2 gives us the following statement:

THEOREM 4.9. *Let $\mathbf{f} : \mathcal{D} \subseteq \mathbb{E} \to \mathbb{F}$ be an analytic map between Banach spaces. Let ζ be a nondegenerate zero of $\mathbf{f}$. Let*

$$0 \leq 2\delta \leq u_0 \leq 2 - \frac{\sqrt{14}}{2} \simeq 0.129\cdots$$

Assume that

(1)
$$B = B\left(\zeta, \frac{u_0}{\gamma(\mathbf{f},\zeta)}\right) \subseteq \mathcal{D}.$$

(2) *$\mathbf{x}_0 \in B$, and the sequence $\mathbf{x}_i$ satisfies*
$$\|\mathbf{x}_{i+1} - N(\mathbf{f},\mathbf{x}_i)\|\gamma(\mathbf{f},\zeta) \leq \delta$$

(3) *The sequence u_i is defined inductively by*
$$u_{i+1} = \frac{u_i^2}{\psi(u_i)} + \delta.$$

Then the sequences u_i and $\mathbf{x}_i$ are well-defined for all i, $\mathbf{x}_i \in \mathcal{D}$, and

$$\frac{\|\mathbf{x}_i - \zeta\|}{\|\mathbf{x}_0 - \zeta\|} \leq \frac{u_i}{u_0} \leq \max\left(2^{-2^i+1}, 2\frac{\delta}{u_0}\right).$$

PROOF. By hypothesis,
$$\frac{u_0}{\psi(u_0)} + \frac{\delta}{u_0} < 1$$
so the sequence u_i is decreasing and positive. For short, let $q = \frac{u_0}{\psi(u_0)} \leq 1/4$. By induction,
$$\frac{u_{i+1}}{u_0} \leq \frac{u_0}{\psi(u_i)}\left(\frac{u_i}{u_0}\right)^2 + \frac{\delta}{u_0} \leq \frac{1}{4}\left(\frac{u_i}{u_0}\right)^2 + \frac{\delta}{u_0}.$$

Assume that $u_i/u_0 \leq 2^{-2^i+1}$. In that case,
$$\frac{u_{i+1}}{u_0} \leq 2^{-2^{i+1}} + \frac{\delta}{u_0} \leq \max\left(2^{-2^{i+1}+1}, 2\frac{\delta}{u_0}\right).$$

Assume now that $2^{-2^i+1}, u_i/u_0 \leq 2\delta/u_0$. In that case,
$$\frac{u_{i+1}}{u_0} \leq \frac{\delta}{u_0}\left(\frac{\delta}{4u_0} + 1\right) \leq \frac{2\delta}{u_0} = \max\left(2^{-2^{i+1}+1}, 2\frac{\delta}{u_0}\right).$$

From now on we use the assumptions, notations and estimates of the proof of Theorem 4.2. Combining (5) and (8) in (6), we obtain again that
$$\|N(\mathbf{f},\mathbf{x})\| \leq \frac{\gamma\|\mathbf{x}\|^2}{\psi(\gamma\|\mathbf{x}\|)}.$$

This time, this means that

$$\|\mathbf{x}_{i+1}\|\gamma \leq \delta + \|N(\mathbf{f},\mathbf{x})\|\gamma \leq \delta + \frac{\gamma^2\|\mathbf{x}\|^2}{\psi(\gamma\|\mathbf{x}\|)}.$$

By induction that $\|\mathbf{x}_i - \zeta\|\gamma(\mathbf{f},\zeta) < u_i$ and we are done. □

EXERCISE 4.1. Consider the following series, defined in $\mathbb{C}^2$:

$$g(x) = \sum_{i=0}^{\infty} x_1^i x_2^i.$$

Compute its radius of convergence. What is its domain of absolute convergence ?

EXERCISE 4.2. The objective of this exercise is to produce a non-optimal algorithm to approximate $\sqrt{y}$. In order to do that, consider the mapping $f(x) = x^2 - y$.

(1) Compute $\gamma(f,x)$.
(2) Show that for $1 \leq y \leq 4$, $x_0 = 1/2 + y/2$ is an approximate zero of the first kind for x, associated to $\sqrt{y}$.
(3) Write down an algorithm to approximate $\sqrt{y}$ up to relative accuracy 2^{-63}.

EXERCISE 4.3. Let $\mathbf{f}$ be an analytic map between Banach spaces, and assume that ζ is a nondegenerate zero of $\mathbf{f}$.

(1) Write down the Taylor series of $D\mathbf{f}(\zeta)^{-1}(\mathbf{f}(\mathbf{x}) - \mathbf{f}(\zeta))$.
(2) Show that if $\mathbf{f}(\mathbf{x}) = 0$, then

$$\gamma(\mathbf{f},\zeta)\|\mathbf{x} - \zeta\| \geq 1/2.$$

This shows that two nondegenerate zeros cannot be at a distance less than $1/2\gamma(\mathbf{f},\zeta)$. Results of this type appeared in **Dedieu (1997a)**, but some of them were known before **Malajovich (1993**, Th.16).

5. Estimates from data at a point

Theorem 4.2 guarantees quadratic convergence in a neighborhood of a known zero ζ. In practical situations, ζ is not known. A major result in alpha-theory is the criterion to detect an approximate zero with just local information. We need to slightly modify the definition.

DEFINITION 5.1 (Approximate zero of the second kind). Let $\mathbf{f} : \mathcal{D} \subseteq \mathbb{E} \to \mathbb{F}$ be as above. An **approximate zero of the second kind** associated to $\zeta \in \mathcal{D}$, $\mathbf{f}(\zeta) = 0$, is a point $\mathbf{x}_0 \in \mathcal{D}$, such that

(1) The sequence $(\mathbf{x})_i$ defined inductively by $\mathbf{x}_{i+1} = N(\mathbf{f},\mathbf{x}_i)$ is well-defined (each $\mathbf{x}_i$ belongs to the domain of $\mathbf{f}$ and $D\mathbf{f}(\mathbf{x}_i)$ is invertible and bounded).
(2)
$$\|\mathbf{x}_{i+1} - \mathbf{x}_i\| \leq 2^{-2^i+1}\|\mathbf{x}_1 - \mathbf{x}_0\|.$$
(3) $\lim_{i\to\infty} \mathbf{x}_i = \zeta$.

For detecting approximate zeros of the second kind, we need:

DEFINITION 5.2 (Smale's β and α invariants).

$$\beta(\mathbf{f},\mathbf{x}) = \|D\mathbf{f}(\mathbf{x})^{-1}\mathbf{f}(\mathbf{x})\| \text{ and } \alpha(\mathbf{f},\mathbf{x}) = \beta(\mathbf{f},\mathbf{x})\gamma(\mathbf{f},\mathbf{x}).$$

The β invariant can be interpreted as the size of the Newton step $N(\mathbf{f},\mathbf{x}) - \mathbf{x}$.

THEOREM 5.3 (Smale). *Let $\mathbf{f} : \mathcal{D} \subseteq \mathbb{E} \to \mathbb{F}$ be an analytic map between Banach spaces. Let*

$$\alpha \leq \alpha_0 = \frac{13 - 3\sqrt{17}}{4}.$$

Define

$$r_0 = \frac{1 + \alpha - \sqrt{1 - 6\alpha + \alpha^2}}{4\alpha} \text{ and } r_1 = \frac{1 - 3\alpha - \sqrt{1 - 6\alpha + \alpha^2}}{4\alpha}.$$

Let $\mathbf{x}_0 \in \mathcal{D}$ be such that $\alpha(\mathbf{f}, \mathbf{x}_0) \leq \alpha$ and assume furthermore that $B(\mathbf{x}_0, r_0\beta(\mathbf{f}, \mathbf{x}_0)) \subseteq \mathcal{D}$. Then,

1. *$\mathbf{x}_0$ is an approximate zero of the second kind, associated to some zero $\zeta \in \mathcal{D}$ of $\mathbf{f}$.*
2. *Moreover, $\|\mathbf{x}_0 - \zeta\| \leq r_0\beta(\mathbf{f}, \mathbf{x}_0)$.*
3. *Let $\mathbf{x}_1 = N(\mathbf{f}, \mathbf{x}_0)$. Then $\|\mathbf{x}_1 - \zeta\| \leq r_1\beta(\mathbf{f}, \mathbf{x}_0)$.*

The constant α_0 is the largest possible with those properties.

This theorem appeared in **Dedieu (2006)**. The value for α_0 was found by Wang Xinghua **Wang (1993)**. Numerically,

$$\alpha_0 = 0.157, 670, 780, 786, 754, 587, 633, 942, 608, 019 \cdots$$

Other useful numerical bounds, under the hypotheses of the theorem, are:

$$r_0 \leq 1.390, 388, 203 \cdots \text{ and } r_1 \leq 0.390, 388, 203 \cdots .$$

The proof of Theorem 5.3 follows from the same method as the one for Theorem 4.2. We first define the 'worst' real function with respect to Newton iteration. Let us fix $\beta, \gamma > 0$. Define

$$h_{\beta\gamma}(t) = \beta - t + \frac{\gamma t^2}{1 - \gamma t} = \beta - t + \gamma t^2 + \gamma^2 t^3 + \cdots .$$

We assume for the time being that $\alpha = \beta\gamma < 3 - 2\sqrt{2} = 0.1715 \cdots$. This guarantees that $h_{\beta\gamma}$ has two distinct zeros $\zeta_1 = \frac{1+\alpha-\sqrt{\Delta}}{4\gamma}$ and $\zeta_2 = \frac{1+\alpha+\sqrt{\Delta}}{4\gamma}$ with of course $\Delta = (1 + \alpha)^2 - 8\alpha$. An useful expression is the product formula

$$h_{\beta\gamma}(x) = 2\frac{(x - \zeta_1)(x - \zeta_2)}{\gamma^{-1} - x}. \tag{9}$$

From (9), $h_{\beta\gamma}$ has also a pole at γ^{-1}. We have always $0 < \zeta_1 < \zeta_2 < \gamma^{-1}$.

The function $h_{\beta\gamma}$ is, among the functions with $h'(0) = -1$ and $\beta(h, 0) \leq \beta$ and $\gamma(h, 0) \leq \gamma$, the one that has the first zero ζ_1 furthest away from the origin.

PROPOSITION 5.4. *Let $\beta, \gamma > 0$, with $\alpha = \beta\gamma \leq 3 - 2\sqrt{2}$. let $h_{\beta\gamma}$ be as above. Define recursively $t_0 = 0$ and $t_{i+1} = N(h_{\beta\gamma}, t_i)$. then*

$$t_i = \zeta_1 \frac{1 - q^{2^i - 1}}{1 - \eta q^{2^i - 1}}, \tag{10}$$

with

$$\eta = \frac{\zeta_1}{\zeta_2} = \frac{1 + \alpha - \sqrt{\Delta}}{1 + \alpha + \sqrt{\Delta}} \text{ and } q = \frac{\zeta_1 - \gamma\zeta_1\zeta_2}{\zeta_2 - \gamma\zeta_1\zeta_2} = \frac{1 - \alpha - \sqrt{\Delta}}{1 - \alpha + \sqrt{\Delta}}.$$

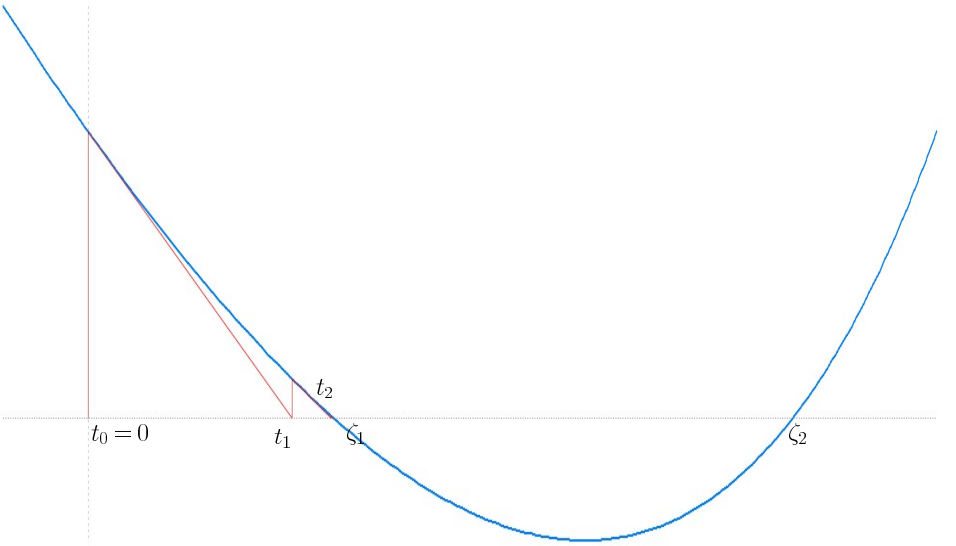


FIGURE 4. $y = h_{\beta\gamma}(t)$.

PROOF. By differentiating (9), one obtains

$$h'_{\beta\gamma}(t) = h_{\beta\gamma}(t)\left(\frac{1}{t-\zeta_1} + \frac{1}{t-\zeta_2} + \frac{1}{\gamma^{-1}-t}\right)$$

and hence the Newton operator is

$$N(h_{\beta\gamma}, t) = t - \frac{1}{\frac{1}{t-\zeta_1} + \frac{1}{t-\zeta_2} + \frac{1}{\gamma^{-1}-t}}.$$

A tedious calculation shows that $N(h_{\beta\gamma}, t)$ is a rational function of degree 2. Hence, it is defined by 5 coefficients, or by 5 values.

In order to solve the recurrence for t_i, we change coordinates using a fractional linear transformation. As the Newton operator will have two attracting fixed points (ζ_1 and ζ_2), we will map those points to 0 and ∞ respectively. For convenience, we will map $t_0 = 0$ into $y_0 = 1$. Therefore, we set

$$S(t) = \frac{\zeta_2 t - \zeta_1\zeta_2}{\zeta_1 t - \zeta_1\zeta_2} \quad \text{and} \quad S^{-1}(y) = \frac{-\zeta_1\zeta_2 y + \zeta_1\zeta_2}{-\zeta_1 y + \zeta_2}$$

Let us look at the sequence $y_i = S(t_i)$. By construction $y_0 = 1$, and subsequent values are given by the recurrence

$$y_{i+1} = S(N(h_{\beta\gamma}, S^{-1}(y_i))).$$

It is an exercise to check that

$$y_{i+1} = q y_i^2, \tag{11}$$

Therefore we have $y_i = q^{2^i - 1}$, and equation (10) holds. □

PROPOSITION 5.5. *Under the conditions of Proposition* 5.4*, 0 is an approximate zero of the second kind for $h_{\beta\gamma}$ if and only if*

$$\alpha = \beta\gamma \leq \frac{13 - 3\sqrt{17}}{4}.$$

PROOF. Using the closed form for t_i, we get:

$$\begin{aligned} t_{i+1} - t_i &= \frac{1 - q^{2^{i+1}-1}}{1 - \eta q^{2^{i+1}-1}} - \frac{1 - q^{2^i-1}}{1 - \eta q^{2^i-1}} \\ &= q^{2^i-1} \frac{(1-\eta)(1-q^{2^i})}{(1 - \eta q^{2^{i+1}-1})(1 - \eta q^{2^i-1})} \end{aligned}$$

In the particular case $i = 0$,

$$t_1 - t_0 = \frac{1-q}{1-\eta q} = \beta$$

Hence

$$\frac{t_{i+1} - t_i}{\beta} = C_i q^{2^i-1}$$

with

$$C_i = \frac{(1-\eta)(1-\eta q)(1-q^{2^i})}{(1-q)(1-\eta q^{2^{i+1}-1})(1-\eta q^{2^i-1})}.$$

Thus, $C_0 = 1$. The reader shall verify in Exercise 5.1 that C_i is a non-increasing sequence. Its limit is non-zero.

From the above, it is clear that 0 is an approximate zero of the second kind if and only if $q \leq 1/2$. Now, if we clear denominators and rearrange terms in $(1 + \alpha - \sqrt{\Delta})/(1 + \alpha + \sqrt{\Delta}) = 1/2$, we obtain the second degree polynomial

$$2\alpha^2 - 13\alpha + 2 = 0.$$

This has solutions $(13 \pm \sqrt{17})/2$. When $0 \leq \alpha \leq \alpha_0 = (13 - \sqrt{17})/2$, the polynomial values are positive and hence $q \leq 1/2$. □

PROOF OF TH.5.3. Let $\beta = \beta(\mathbf{f}, \mathbf{x}_0)$ and $\gamma = \gamma(\mathbf{f}, \mathbf{x}_0)$. Let $h_{\beta\gamma}$ and the sequence t_i be as in Proposition 5.4. By construction, $\|\mathbf{x}_1 - \mathbf{x}_0\| = \beta = t_1 - t_0$. We use the following notations:

$$\beta_i = \beta(\mathbf{f}, \mathbf{x}_i) \text{ and } \gamma_i = \gamma(\mathbf{f}, \mathbf{x}_i).$$

Those will be compared to

$$\hat{\beta}_i = \beta(h_{\beta\gamma}, t_i)) \text{ and } \hat{\gamma}_i = \gamma(h_{\beta\gamma}, t_i)).$$

Induction hypothesis: $\beta_i \leq \hat{\beta}_i$ and for all $l \geq 2$,

$$\|D\mathbf{f}(\mathbf{x}_i)^{-1} D^l \mathbf{f}(\mathbf{x}_i)\| \leq -\frac{h^{(l)}_{\beta\gamma}(t_i)}{h'_{\beta\gamma}(t_i)}.$$

The initial case when $i = 0$ holds by construction. So let us assume that the hypothesis holds for i. We will estimate

$$\beta_{i+1} \leq \|D\mathbf{f}(\mathbf{x}_{i+1})^{-1} D\mathbf{f}(\mathbf{x}_i)\| \|D\mathbf{f}(\mathbf{x}_i)^{-1} \mathbf{f}(\mathbf{x}_{i+1})\| \tag{12}$$

and

$$\gamma_{i+1} \leq \|D\mathbf{f}(\mathbf{x}_{i+1})^{-1} D\mathbf{f}(\mathbf{x}_i)\| \frac{\|D\mathbf{f}(\mathbf{x}_i)^{-1} D^k \mathbf{f}(\mathbf{x}_{i+1})\|}{k!}. \tag{13}$$

By construction, $\mathbf{f}(\mathbf{x}_i) + D\mathbf{f}(\mathbf{x}_i)(\mathbf{x}_{i+1} - \mathbf{x}_i) = 0$. The Taylor expansion of $\mathbf{f}$ at $\mathbf{x}_i$ is therefore

$$D\mathbf{f}(\mathbf{x}_i)^{-1}\mathbf{f}(\mathbf{x}_{i+1}) = \sum_{k\geq 2} \frac{D\mathbf{f}(\mathbf{x}_i)^{-1}D^k\mathbf{f}(\mathbf{x}_i)(\mathbf{x}_{i+1} - \mathbf{x}_i)^k}{k!}$$

Passing to norms,

$$\|D\mathbf{f}(\mathbf{x}_i)^{-1}\mathbf{f}(\mathbf{x}_{i+1})\| \leq \frac{\beta_i^2\gamma_i}{1-\gamma_i}$$

The same argument shows that

$$-\frac{h_{\beta\gamma}(t_{i+1})}{h'_{\beta\gamma}(t_i)} = \frac{\beta(h_{\beta\gamma}, t_i)^2\gamma(h_{\beta\gamma}, t_i)}{1-\gamma(h_{\beta\gamma}, t_i)}$$

From Lemma 4.6,

$$\|D\mathbf{f}(\mathbf{x}_{i+1})^{-1}D\mathbf{f}(\mathbf{x}_i)\| \leq \frac{(1-\beta_i\gamma_i)^2}{\psi(\beta_i\gamma_i)}.$$

Also, computing directly,

$$\frac{h'_{\beta\gamma}(t_{i+1})}{h'_{\beta\gamma}(t_i)} = \frac{(1-\hat{\beta}\hat{\gamma})^2}{\psi(\hat{\beta}\hat{\gamma})}. \tag{14}$$

We established that

$$\beta_{i+1} \leq \frac{\beta_i^2\gamma_i(1-\beta_i\gamma_i)}{\psi(\beta_i\gamma_i)} \leq \frac{\hat{\beta}_i^2\hat{\gamma}_i(1-\hat{\beta}_i\hat{\gamma}_i)}{\psi(\hat{\beta}_i\hat{\gamma}_i)} = \hat{\beta}_{i+1}.$$

Now the second part of the induction hypothesis:

$$D\mathbf{f}(\mathbf{x}_i)^{-1}D^l\mathbf{f}(\mathbf{x}_{i+1}) = \sum_{k\geq 0}\frac{1}{k!}\frac{D\mathbf{f}(\mathbf{x}_i)^{-1}D^{k+l}\mathbf{f}(\mathbf{x}_i)(\mathbf{x}_{i+1} - \mathbf{x}_i)^k}{k+l}$$

Passing to norms and invoking the induction hypothesis,

$$\|D\mathbf{f}(\mathbf{x}_i)^{-1}D^l\mathbf{f}(\mathbf{x}_{i+1})\| \leq \sum_{k\geq 0} -\frac{h_{\beta\gamma}^{(k+l)}(t_i)\hat{\beta}_i^k}{k!h'_{\beta\gamma}(t_i)}$$

and then using Lemma 4.6 and (14),

$$\|D\mathbf{f}(\mathbf{x}_{i+1})^{-1}D^l\mathbf{f}(\mathbf{x}_{i+1})\| \leq \frac{(1-\hat{\beta}_i\hat{\gamma}_i)^2}{\psi(\hat{\beta}_i\hat{\gamma}_i)}\sum_{k\geq 0} -\frac{h_{\beta\gamma}^{(k+l)}(t_i)\hat{\beta}_i^k}{k!h'_{\beta\gamma}(t_i)}.$$

A direct computation similar to (14) shows that

$$-\frac{h_{\beta\gamma}^{(k+l)}(t_{i+1})}{k!h'_{\beta\gamma}(t_{i+1})} = \frac{(1-\hat{\beta}_i\hat{\gamma}_i)^2}{\psi(\hat{\beta}_i\hat{\gamma}_i)}\sum_{k\geq 0} -\frac{h_{\beta\gamma}^{(k+l)}(t_i)\hat{\beta}_i^k}{k!h'_{\beta\gamma}(t_i)}.$$

and since the right-hand-terms of the last two equations are equal, the second part of the induction hypothesis proceeds. Dividing by $l!$, taking $l-1$-th roots and maximizing over all l, we deduce that $\gamma_i \leq \hat{\gamma}_i$.

Proposition 5.5 then implies that $\mathbf{x}_0$ is an approximate zero.

The second and third statement follow respectively from

$$\|\mathbf{x}_0 - \zeta\| \leq \beta_0 + \beta_1 + \cdots = \zeta_1$$

	1/32	1/16	1/10	1/8	$\frac{13-3\sqrt{17}}{4}$
1	4.854	3.683	2.744	2.189	1.357
2	14.472	10.865	7.945	6.227	3.767
3	33.700	25.195	18.220	14.41	7.874
4	72.157	53.854	38.767	29.648	15.881
5	149.71	111.173	79.861	60.864	31.881
6	302.899	225.811	162.49	123.295	63.881

TABLE 2. Values of $-log_2(\|\mathbf{x}_i - \zeta\|/\beta)$ in function of α and i.

and

$$\|\mathbf{x}_1 - \zeta\| \leq \beta_1 + \beta_2 + \cdots = \zeta_1 - \beta.$$

□

The same issues as in Theorem 4.2 arise. First of all, we actually proved a sharper statement. Namely,

THEOREM 5.6. *Let $\mathbf{f} : \mathcal{D} \subseteq \mathbb{E} \to \mathbb{F}$ be an analytic map between Banach spaces. Let*

$$\alpha \leq 3 - 2\sqrt{2}.$$

Define

$$r = \frac{1 + \alpha - \sqrt{1 - 6\alpha + \alpha^2}}{4\alpha}.$$

Let $\mathbf{x}_0 \in \mathcal{D}$ be such that $\alpha(\mathbf{f}, \mathbf{x}_0) \leq \alpha$ and assume furthermore that $B(\mathbf{x}_0, r\beta(\mathbf{f}, \mathbf{x}_0)) \subseteq \mathcal{D}$. Then, the sequence $\mathbf{x}_{i+1} = N(\mathbf{f}, \mathbf{x}_i)$ is well defined, and there is a zero $\zeta \in \mathcal{D}$ of $\mathbf{f}$ such that

$$\|\mathbf{x}_i - \zeta\| \leq q^{2^i - 1} \frac{1 - \eta}{1 - \eta q^{2^i - 1}} r\beta(\mathbf{f}, \mathbf{x}_0).$$

for η and q as in Proposition 5.4.

Table 2 and Figure 5 show how fast $\|\mathbf{x}_i - \zeta\|/\beta$ decreases in terms of α and i.

The final issue is robustness. There is no obvious modification of the proof of Theorem 5.3 to provide a nice statement, so we will rely on Theorem 4.9 indeed.

THEOREM 5.7. *Let $\mathbf{f} : \mathcal{D} \subseteq \mathbb{E} \to \mathbb{F}$ be an analytic map between Banach spaces. Let δ, α and $u0$ satisfy*

$$0 \leq 2\delta < u_0 = \frac{r\alpha}{(1 - r\alpha)\psi(r\alpha)} < 2 - \frac{\sqrt{14}}{2}$$

with $r = \frac{1+\alpha-\sqrt{1-6\alpha+\alpha^2}}{4\alpha}$. Assume that

(1)
$$B = B\left(\mathbf{x}_0, 2r\beta(\mathbf{f}, \mathbf{x}_0)\right) \subseteq \mathcal{D}.$$

(2) *$\mathbf{x}_0 \in B$, and the sequence $\mathbf{x}_i$ satisfies*
$$\|\mathbf{x}_{i+1} - N(\mathbf{f}, \mathbf{x}_i)\| \frac{r\beta(f, x_0)}{(1 - r\alpha)\psi(r\alpha)} \leq \delta$$

(3) *The sequence u_i is defined inductively by*
$$u_{i+1} = \frac{u_i^2}{\psi(u_i)} + \delta.$$

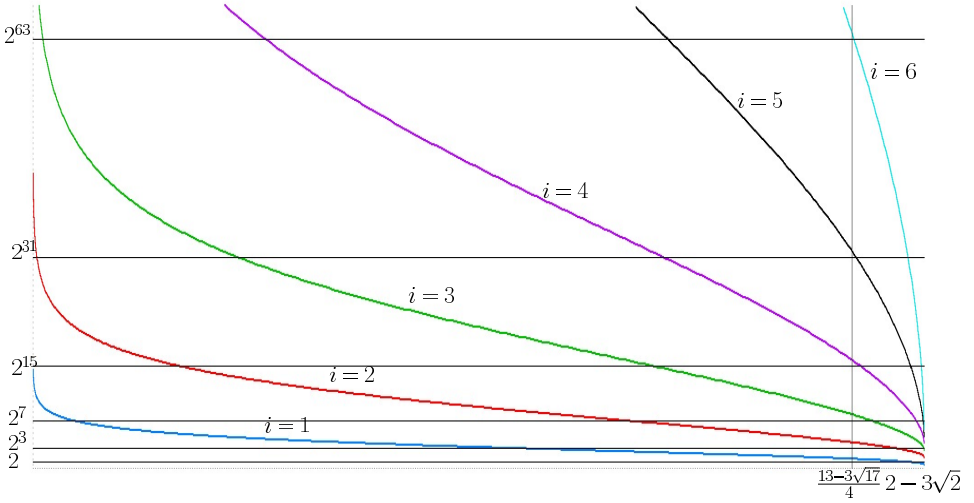


FIGURE 5. Values of $-log_2(\|\mathbf{x}_i - \zeta\|/\beta)$ in function of α for $i = 1$ to 6.

Then the sequences u_i and $\mathbf{x}_i$ are well-defined for all i, $\mathbf{x}_i \in \mathcal{D}$, and

$$\frac{\|\mathbf{x}_i - \zeta\|}{\|\mathbf{x}_1 - \mathbf{x}_0\|} \leq \frac{ru_i}{u_0} \leq r \max\left(2^{-2^i+1}, 2\frac{\delta}{u_0}\right).$$

Numerically, $\alpha_0 = 0.074,290\cdots$ satisfies the hypothesis of the Theorem. A version of this theorem (not as sharp, and another metric) appeared as Theorem 2 in **Malajovich (1994)**.

The following Lemma will be useful:

LEMMA 5.8. *Assume that $u = \gamma(\mathbf{f}, \mathbf{x})\|\mathbf{x} - \mathbf{y}\| \leq 1 - \sqrt{2}/2$. Then,*

$$\gamma(\mathbf{f}, \mathbf{y}) \leq \frac{\gamma(\mathbf{f}, \mathbf{x})}{(1-u)\psi(u)}.$$

PROOF. In order to estimate the higher derivatives, we expand:

$$\frac{1}{l!}D\mathbf{f}(\mathbf{x})^{-1}D^l\mathbf{f}(\mathbf{y}) = \sum_{k\geq 0}\binom{k+l}{l}\frac{D\mathbf{f}(\mathbf{x})^{-1}D^{k+l}\mathbf{f}(\mathbf{x})(\mathbf{y}-\mathbf{x})^k}{k+l}$$

and by Lemma 4.3 for $d = l + 1$,

$$\frac{1}{l!}\|D\mathbf{f}(\mathbf{x})^{-1}D^l\mathbf{f}(\mathbf{y})\| \leq \frac{\gamma(\mathbf{f}, \mathbf{x})^{l-1}}{(1-u)^{l+1}}.$$

Combining with Lemma 4.6,

$$\frac{1}{l!}\|D\mathbf{f}(\mathbf{y})^{-1}D^l\mathbf{f}(\mathbf{y})\| \leq \frac{\gamma(\mathbf{f}, \mathbf{x})^{l-1}}{(1-u)^{l-1}\psi(u)}.$$

Taking the $l-1$-th power,

$$\gamma(\mathbf{f}, \mathbf{y}) \leq \frac{\gamma(\mathbf{f}, \mathbf{x})}{(1-u)\psi(u)}.$$

□

PROOF OF THEOREM 5.7. We have necessarily $\alpha < 3 - 2\sqrt{2}$ or r is undefined. Then (Theorem 5.6) there is a zero ζ of $\mathbf{f}$ with $\|\mathbf{x}_0 - \zeta\| \leq r\beta(f, x_0)$. Then, Lemma 5.8 implies that $\|\mathbf{x}_0 - \zeta\|\gamma(\mathbf{f}, \zeta) \leq u_0$. Now apply Theorem 4.9.

□

EXERCISE 5.1. The objective of this exercise is to show that C_i is non-increasing.

(1) Show the following trivial lemma: **If** $0 \leq s < a \leq b$**, then** $\frac{a-s}{b-s} \leq \frac{a}{b}$.
(2) Deduce that $q \leq \eta$.
(3) Prove that $C_{i+1}/C_i \leq 1$.

EXERCISE 5.2. Show that

$$\zeta_1\gamma(\zeta_1) = \frac{1+\alpha-\sqrt{\Delta}}{3-\alpha+\sqrt{\Delta}} \frac{1}{\psi\left(\frac{1+\alpha-\sqrt{\Delta}}{4}\right)}.$$

Part 2. Inclusion and exclusion

6. Eckart-Young theorem

The following classical theorem in linear algebra is known as the **singular value decomposition** (**svd** for short).

THEOREM 6.1. *Let $A : \mathbb{R}^n \mapsto \mathbb{R}^m$ (resp. $\mathbb{C}^n \to \mathbb{C}^m$) be linear. Then, there are $\sigma_1 \geq \cdots \geq \sigma_r > 0$, $r \leq m, n$, such that*

$$A = U\Sigma V^*$$

with $U \in O(m)$ (resp. $U(m)$), $V \in O(n)$ (resp. $U(n)$) and $\Sigma_{ij} = \sigma_i$ for $i = j \leq r$ and 0 otherwise.

It is due to Sylvester (real $n \times n$ matrices) and to Eckart and Young (**1939**) in the general case, now exercise 6.1 below.

Σ is a $m \times n$ matrix. It is possible to rewrite this in an 'economical formulation with Σ an $r \times r$ matrix, U and V orthogonal (resp. unitary) $m \times r$ and $n \times r$ matrices. The numbers $\sigma_1, \ldots, \sigma_r$ are called **singular values** of A. They may be computed by extracting the positive square root of the non-zero eigenvalues of A^*A or AA^*, whatever matrix is smaller. The operator and Frobenius norm of A may be written in terms of the σ_i's:

$$\|A\|_2 = \sigma_1 \qquad \|A\|_F = \sqrt{\sigma_1^2 + \cdots + \sigma_r^2}.$$

The discussion and the results above hold when A is a linear operator between finite dimensional inner product spaces. It suffices to choose an orthonormal basis, and apply Theorem 6.1 to the corresponding matrix.

When $m = n = r$, $\|A^{-1}\|_2 = \sigma_n^{-1}$. In this case, the **condition number** of A for linear solving is defined as

$$\kappa(A) = \|A\|_*\|A^{-1}\|_{**}.$$

The choice of norms is arbitrary, as long as operator and vector norms are consistent. Two canonical choices are

$$\kappa_2(A) = \|A\|_2\|A^{-1}\|_2 \text{ and } \kappa_D(A) = \|A\|_F\|A^{-1}\|_2.$$

The second choice was suggested by Demmel (**1988**). Using that definition he obtained bounds on the probability that a matrix is poorly conditioned. The exact probability distribution for the most usual probability measures in matrix space was computed in **Edelman (1992**).

Assume that $A(t)\mathbf{x}(t) \equiv \mathbf{b}(t)$ is a family of problems and solutions depending smoothly on a parameter t. Differentiating implicitly,

$$\dot{A}\mathbf{x} + A\dot{\mathbf{x}} = \dot{\mathbf{b}}$$

which amounts to

$$\dot{\mathbf{x}} = A^{-1}\dot{\mathbf{b}} - A^{-1}\dot{A}\mathbf{x}.$$

Passing to norms and to relative errors, we quickly obtain

$$\frac{\|\dot{\mathbf{x}}\|}{\|\mathbf{x}\|} \leq \kappa_D(A)\left(\frac{\|\dot{A}\|_F}{\|A\|_F} + \frac{\|\dot{\mathbf{b}}\|}{\|\mathbf{b}\|}\right).$$

This bounds the relative error in the solution $\mathbf{x}$ in terms of the relative error in the coefficients. The usual paradigm in numerical linear algebra dates from **Turing (1948**) and **Wilkinson (1963**). After the rounding-off during computation, we obtain the exact solution of a perturbed system. Bounds for the perturbation or **backward error** are found through line by line analysis of the algorithm. The output error or **forward error** is bounded by the backward error, times the condition number.

Condition numbers provide therefore an important metric invariant for numerical analysis problems. A geometric interpretation in the case of linear equation solving is:

THEOREM 6.2. *Let A be a nondegenerate square matrix.*

$$\|A^{-1}\|_2 = \min_{\det(A+B)=0} \|B\|_F$$

In particular, this implies that

$$\kappa_D(A)^{-1} = \min_{\det(A+B)=0} \frac{\|B\|_F}{\|A\|_F}$$

A pervading principle in the subject is: **the inverse of the condition number is related to the distance to the ill-posed problems**.

It is possible to define the condition number for a full-rank non- square matrix by

$$\kappa_D(A) = \|A\|_F \ \sigma_{\min(m,n)}(A)^{-1}.$$

THEOREM 6.3. (**Eckart and Young, 1936**) *Let A be an $m \times n$ matrix of rank r. Then,*

$$\sigma_r(A)^{-1} = \min_{\sigma_r(A+B)=0} \|B\|_F.$$

In particular, if $r = \min(m, n)$,

$$\kappa_D(A)^{-1} = \min_{\sigma_r(A+B)=0} \frac{\|B\|_F}{\|A\|_F}.$$

EXERCISE 6.1. Prove Theorem 6.1. Hint: let u, v, σ such that $Av = \sigma u$ with σ maximal, $\|u\| = 1$, $\|v\| = 1$. What can you say about $A_{|v^\perp}$?

EXERCISE 6.2. Prove Theorem 6.3.

EXERCISE 6.3. Assume furthermore that $m < n$. Show that the same interpretation for the condition number still holds, namely the norm of the perturbation of **some** solution is bounded by the condition number, times the perturbation of the input.

7. The space of homogeneous polynomial systems

We will denote by $\mathcal{H}_d^{\mathbb{R}}$ the space of polynomials of degree d in $n+1$ variables. This space can be associated to the space of symmetric d-linear forms. For instance, when $d = 2$, the polynomial

$$f(x_0, x_1) = f_0 x_0^2 + f_1 x_0 x_1 + f_2 x_1^2 = \begin{bmatrix} x_0 & x_1 \end{bmatrix} \begin{bmatrix} f_0 & f_1/2 \\ f_1/2 & f_0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix}$$

can be associated to a symmetric bilinear form and can be represented by a matrix. In general, a homogeneous polynomial can be represented by a symmetric tensor

$$f(\mathbf{x}) = \sum_{|\mathbf{a}|=d} f_{\mathbf{a}} x_0^{a_0} \cdots x_n^{a_n} = \sum_{0 \le i_1, \ldots, i_d \le n} T_{i_1 i_2 \ldots i_d} x_{i_1} x_{i_2} \cdots x_{i_d}$$

where

$$f_{\mathbf{a}} = \sum_{\substack{1 \le i_1, \ldots, i_n \le n \\ \mathrm{e}_{i_1} + \mathrm{e}_{i_2} + \cdots \mathrm{e}_{i_d} = \mathbf{a}}} T_{i_1 i_2 \ldots i_d}$$

and e_i denotes the i-th vector of the canonical basis of $\mathbb{R}^n$.

The canonical inner product for tensors is given by

$$\langle S, T \rangle = \sum_{0 \le i_1, \ldots, i_d \le n} S_{i_1 i_2 \ldots i_d} T_{i_1 i_2 \ldots i_d}$$

Writing polynomials f and g as symmetric tensors, we obtain Weyl's inner product in the space of polynomials:

$$\langle f, g \rangle = \sum_{|\mathbf{a}|=d} \frac{f_{\mathbf{a}} g_{\mathbf{a}}}{\binom{d}{\mathbf{a}}}.$$

where $\binom{d}{\mathbf{a}} = \frac{d!}{a_0! a_1! \cdots a_n!}$ is the coefficient of $(x_0 + \cdots + x_n)^d$ in x^a.

LEMMA 7.1. *Let Q be an orthogonal $n \times n$ matrix, that is $Q^T Q = I$. Then,*

$$\langle f \circ Q, g \circ Q \rangle = \langle f, g \rangle$$

EXERCISE 7.1. Prove Lemma 7.1

We say that the above inner product is **invariant under orthogonal action**. We will always assume this inner-product for $\mathcal{H}_d^{\mathbb{R}}$.

It is also important to notice that $\mathcal{H}_d^{\mathbb{R}}$ is that it is a **reproducing kernel space**. Let

$$K_d(\mathbf{x}, \mathbf{y}) = \langle \mathbf{x}, \mathbf{y} \rangle^d.$$

Then

$$f(\mathbf{y}) = \langle f(\cdot), K_d(\cdot, \mathbf{y}) \rangle,$$
$$Df(\mathbf{y})\mathbf{u} = \langle f(\cdot), D_{\mathbf{y}} K_d(\cdot, \mathbf{y})\mathbf{u} \rangle,$$

etc...

8. The condition number

Now, let's denote by $\mathcal{H}_{\mathbf{d}}^{\mathbb{R}}$ the space of systems of homogeneous polynomials of degree $\mathbf{d} = (d_1, \ldots, d_n)$. The **condition number** measures how the solution of an equation depends upon the coefficients.

Therefore, assume that both a polynomial system $\mathbf{f} \in S(\mathcal{H}_{\mathbf{d}}^{\mathbb{R}})$ and a point $\mathbf{x} \in S(\mathbb{R}^{n+1})$ depend upon a parameter t. Say,

$$\mathbf{f}_t(\mathbf{x}_t) \equiv 0.$$

Differentiating, one gets

$$D\mathbf{f}_t(\mathbf{x}_t)\dot{\mathbf{x}}_t = -\dot{\mathbf{f}}_t(\mathbf{x}_t)$$

so

$$\|\dot{\mathbf{x}}_t\| \leq \|D\mathbf{f}_t(\mathbf{x}_t)^{-1}_{|\mathbf{x}_t^\perp}\| \|\dot{\mathbf{f}}_t(\mathbf{x}_t)\|. \tag{15}$$

The **normalized condition number** is defined for $\mathbf{f} \in \mathcal{H}_{\mathbf{d}}^{\mathbb{R}}$ and $\mathbf{x} \in \mathbb{R}^{n+1}$ as

$$\mu(\mathbf{f}, \mathbf{x}) = \|\mathbf{f}\| \left\| \left(\begin{bmatrix} d_1^{-1/2}\|\mathbf{x}\|^{-d_1+1} & & \\ & \ddots & \\ & & d_n^{-1/2}\|\mathbf{x}\|^{-d_n+1} \end{bmatrix} D\mathbf{f}(\mathbf{x})_{|\mathbf{x}^\perp} \right)^{-1} \right\|.$$

In the special case $\mathbf{f} \in S(\mathcal{H}_{\mathbf{d}}^{\mathbb{R}})$ and $\mathbf{x} \in S(\mathbb{R}^{n+1})$,

$$\mu(\mathbf{f}, \mathbf{x}) = \left\| \left(\begin{bmatrix} d_1^{-1/2} & & \\ & \ddots & \\ & & d_n^{-1/2} \end{bmatrix} D\mathbf{f}(\mathbf{x})_{|\mathbf{x}^\perp} \right)^{-1} \right\|.$$

Proposition 8.1.

(1) *If $\mathbf{f}_t$ and $\mathbf{x}_t$ are paths in $S(\mathcal{H}_{\mathbf{d}}^{\mathbb{R}})$ and $S(\mathbb{R}^{n+1})$ respectively, and $\mathbf{f}_t(\mathbf{x}_t) \equiv 0$ then*

$$\|\dot{\mathbf{x}}_t\| \leq \mu(\mathbf{f}_t, \mathbf{x}_t)\|\dot{\mathbf{f}}_\mathbf{t}\|.$$

(2) *Let $\mathbf{x} \in S(\mathbb{R}^{n+1})$ be fixed. Then the mapping*

$$\begin{array}{rccl} \pi: & \mathcal{H}_{\mathbf{d}}^{\mathbb{R}} & \to & L(\mathbf{x}^\perp, \mathbb{R}^n), \\ & \mathbf{f} & \mapsto & \begin{bmatrix} d_1^{-1/2} & & & \\ & d_2^{-1/2} & & \\ & & \ddots & \\ & & & d_n^{-1/2} \end{bmatrix} D\mathbf{f}(\mathbf{x})_{|\mathbf{x}^\perp} \end{array}$$

restricts to an isometry $\pi_{|(\ker\pi)^\perp} : (\ker\pi)^\perp \to L(\mathbf{x}^\perp, \mathbb{R}^n)$.

(3) *Let $\mathbf{f} \in S(\mathcal{H}_{\mathbf{d}}^{\mathbb{R}})$ and $\mathbf{x} \in S(\mathbb{R}^{n+1})$. Then,*

$$\mu(\mathbf{f}, \mathbf{x}) = \frac{1}{\min_{g \in \mathcal{H}_{\mathbf{d}}^{\mathbb{R}}}\{\|\mathbf{f} - \mathbf{g}\| : D\mathbf{g}(\mathbf{x})_{|\mathbf{x}^\perp} \text{ singular}\}}.$$

(4) *If furthermore $\mathbf{f}(\mathbf{x}) = 0$,*

$$\mu(\mathbf{f}, \mathbf{x}) = \frac{1}{\min_{g \in \mathcal{H}_{\mathbf{d}}^{\mathbb{R}}}\{\|\mathbf{f} - \mathbf{g}\| : \mathbf{g}(\mathbf{x}) = 0 \text{ and } D\mathbf{g}(\mathbf{x})_{|\mathbf{x}^\perp} \text{ singular}\}}.$$

PROOF. Item 1 follows from (15). In order to prove item 2, let $\mathbf{x} \in S(\mathbb{R}^{n+1})$ be fixed and let $\mathbf{f} \in \mathcal{H}^{\mathbb{R}}_{\mathbf{d}}$. Assume that $\mathbf{y} \perp \mathbf{x}$. We can write $\mathbf{f}(\mathbf{x}+\mathbf{y})$ as

$$\mathbf{f}(\mathbf{x}+\mathbf{y}) = \mathbf{f}(\mathbf{x}) + D\mathbf{f}(\mathbf{x})_{|\mathbf{x}^\perp}\mathbf{y} + \frac{1}{2}D^2\mathbf{f}(\mathbf{x})_{|\mathbf{x}^\perp}(\mathbf{y}-\mathbf{x}, \mathbf{y}-\mathbf{x}) + \cdots$$

This suggests a decomposition of $\mathcal{H}^{\mathbb{R}}_{\mathbf{d}}$ into terms that are 'constant', 'linear' or 'higher order' at x.

$$\mathcal{H}^{\mathbb{R}}_{\mathbf{d}} = H_0 \oplus H_1 \oplus H_2 \oplus \cdots .$$

An orthonormal basis for H_1 would be

$$\left(\frac{1}{\sqrt{d}}\frac{\partial K_{d_i}(\cdot, \mathbf{x})}{\partial \mathbf{u}_j}\mathrm{e}_i\right)$$

where $(\mathbf{u}_1, \ldots, \mathbf{u}_n)$ is an orthonormal basis of $\mathbf{x}^\perp$ and $(\mathrm{e}_1, \ldots, \mathrm{e}_n)$ is the canonical basis of $\mathbb{R}^n$.

In this basis, the projection of $\mathbf{f}$ in H_1 is just

$$\begin{bmatrix} & \vdots & \\ \cdots & \left\langle \mathbf{f}_i, \frac{1}{\sqrt{d}}\frac{\partial K_{d_i}(\cdot,\mathbf{x})}{\partial \mathbf{u}_j} \right\rangle & \cdots \\ & \vdots & \end{bmatrix} = \begin{bmatrix} d_1^{-1/2} & & \\ & \cdots & \\ & & d_n^{-1/2} \end{bmatrix} D\mathbf{f}(\mathbf{x})_{|\mathbf{x}^\perp}.$$

Thus, the subspace H_1 of $\mathcal{H}^{\mathbb{R}}_{\mathbf{d}}$ is isomorphic to the space of $n \times n$ matrices. Moreover, $\pi : \mathcal{H}^{\mathbb{R}}_{\mathbf{d}} \to H_1$ is an orthogonal projection. Items 3 and 4 follow now easily from Theorem 6.3. □

EXERCISE 8.1. Deduce that for all $\mathbf{f} \in \mathcal{H}^{\mathbb{R}}_{\mathbf{d}}$, $0 \neq \mathbf{x} \in \mathbb{R}^{n+1}$, $\mu(\mathbf{f}, \mathbf{x}) \geq \sqrt{n}$.

We denote by $\rho(\mathbf{x}, \mathbf{y}) = \widehat{(\mathbf{x}0\mathbf{y})}$ the angular distance between $\mathbf{x} \in S^n$ and $\mathbf{y} \in S^n$. The following estimate is quite useful:

THEOREM 8.2. *Let* $\mathbf{f}, \mathbf{g} \in S(\mathcal{H}^{\mathbb{R}}_{\mathbf{d}})$ *and let* $\mathbf{x}, \mathbf{y} \in S(\mathbb{R}^{n+1})$. *Let*

$$u = (\max d_i)\mu(\mathbf{f}, \mathbf{x})\rho(\mathbf{x}, \mathbf{y}) \text{ and } v = \mu(\mathbf{f}, \mathbf{x})\|\mathbf{f} - \mathbf{g}\|.$$

Then,

$$\frac{1}{1+u+v}\mu(\mathbf{f}, \mathbf{x}) \leq \mu(\mathbf{g}, \mathbf{y}) \leq \frac{1}{1-u-v}\mu(\mathbf{f}, \mathbf{x}).$$

REMARK 8.3. Similar formulas were given by **Bürgisser and Cucker (2011)** and **Dedieu et al. (2013)**. The final form here appeared in **Malajovich (2011)** and generalizes to the sparse condition number.

PROOF. Let R be a rotation taking $\mathbf{y}$ to $\mathbf{x}$. Then, $\mu(\mathbf{g}, \mathbf{y}) = \mu(\mathbf{g} \circ R, \mathbf{x})$. Moreover, it is easy to check that $\|\mathbf{g} \circ R - \mathbf{g}\| \leq (\max d_i)\rho(\mathbf{x}, \mathbf{y})$. Thus,

$$\mu(\mathbf{f}, \mathbf{x})\|\mathbf{f} - \mathbf{g} \circ R\| \leq (u+v).$$

Now, notice that Proposition 8.1(3) implies:

$$\frac{1}{\mu(\mathbf{f}, \mathbf{x})} - \|\mathbf{f} - \mathbf{g} \circ R\|. \leq \frac{1}{\mu(\mathbf{g} \circ R, \mathbf{x})} \leq \frac{1}{\mu(\mathbf{f}, \mathbf{x})} + \|\mathbf{f} - \mathbf{g} \circ R\|.$$

The theorem follows by taking inverses. □

9. The inclusion theorem

Let $f \in \mathcal{H}^{\mathbb{R}}_{\mathbf{d}}$. For any $\mathbf{x} \in S^n$, we denote by $A_{\mathbf{x}}$ be the affine space $\mathbf{x} + \mathbf{x}^{\perp}$ and by $F_{\mathbf{x}} : A_{\mathbf{x}} \to \mathbb{R}^n$, $\mathbf{X} \mapsto \mathbf{f}(\mathbf{x}+\mathbf{X})$ the restriction of $\mathbf{f}$ to $A_{\mathbf{x}}$. Then $F_{\mathbf{x}}$ is an n-variate polynomial system of degree $\mathbf{d}$.

LEMMA 9.1. (**Shub and Smale, 1993**)

$$\gamma(\mathbf{F}_{\mathbf{x}}, 0) \leq \frac{(\max d_i)^{3/2}}{2} \|\mathbf{f}\| \mu(\mathbf{f}, \mathbf{x})$$

PROOF. For simplicity assume $\|\mathbf{f}\| = 1$. Let $k \geq 2$ and

$$\Delta = \begin{bmatrix} \sqrt{d_1} & & \\ & \ddots & \\ & & \sqrt{d_n} \end{bmatrix}.$$

$$\begin{aligned} \frac{1}{k!} \left\| D\mathbf{F}_{\mathbf{x}}(0)^{-1} D^k \mathbf{F}_{\mathbf{x}}(0) \right\| &= \frac{1}{k!} \left\| D\mathbf{f}(\mathbf{x})^{-1}_{|\mathbf{x}^{\perp}} D^k \mathbf{f}(\mathbf{x})_{|\mathbf{x}^{\perp}} \right\| \\ &\leq \frac{1}{k!} \left\| D\mathbf{f}(\mathbf{x})^{-1}_{|\mathbf{x}^{\perp}} \Delta \right\| \left\| \Delta^{-1} D^k \mathbf{f}(\mathbf{x})_{|\mathbf{x}^{\perp}} \right\| \\ &\leq \mu(\mathbf{f}, \mathbf{x}) \ \frac{1}{k!} \left\| \Delta^{-1} D^k \mathbf{f}(\mathbf{x})_{|\mathbf{x}^{\perp}} \right\| \end{aligned}$$

Now, notice that

$$\begin{aligned} |D^k \mathbf{f}_i(\mathbf{x})| = |\langle \mathbf{f}_i, D^k K_{d_i}(\cdot, \mathbf{x}) \rangle| \leq \\ \leq \|\mathbf{f}_i\| \sup_{\substack{\|\mathbf{u}_1\| = \cdots = \|\mathbf{u}_k\| = 1 \\ \mathbf{u}_1, \ldots, \mathbf{u}_k \perp \mathbf{x}}} \|D^k K_{d_i}(\cdot, \mathbf{x})(\mathbf{u}_1, \ldots, \mathbf{u}_k)\| \end{aligned}$$

where $K_{d_i}(\mathbf{y}, \mathbf{x}) = \langle \mathbf{y}, \mathbf{x} \rangle^{d_i}$ is the reproducing kernel of $\mathcal{H}^{\mathbb{R}}_{d_i}$. Differentiating K_{d_i} with respect to $\mathbf{y}$, one obtains:

$$\frac{1}{k!} D^k K_{d_i}(\mathbf{y}, \mathbf{x})(\mathbf{u}_1, \ldots, \mathbf{u}_k) = \binom{d_i}{k} \langle \mathbf{y}, \mathbf{x} \rangle^{d-k} \langle y, \mathbf{u}_1 \rangle \cdots \langle y, \mathbf{u}_k \rangle.$$

The norm of $\frac{1}{k!} D^k K_{d_i}(\mathbf{y}, \mathbf{x})(\mathbf{u}_1, \ldots, \mathbf{u}_k)$ (as a polynomial of $\mathbf{y}$) can be computed using the reproducing kernel property.

$$\begin{aligned} \left\| \frac{1}{k!} D^k K_{d_i}(\cdot, \mathbf{x})(\mathbf{u}_1, \ldots, \mathbf{u}_k) \right\|^2 = \\ &= \left\langle \frac{1}{k!} D^k K_{d_i}(\cdot, \mathbf{x})(\mathbf{u}_1, \ldots, \mathbf{u}_k), \frac{1}{k!} D^k K_{d_i}(\cdot, \mathbf{x})(\mathbf{u}_1, \ldots, \mathbf{u}_k) \right\rangle \\ &= \frac{1}{k!} \frac{\partial \mathbf{y}}{\partial \mathbf{u}_1} \cdots \frac{\partial \mathbf{y}}{\partial \mathbf{u}_k} \binom{d_i}{k} \langle \mathbf{y}, \mathbf{x} \rangle^{d-k} \langle \mathbf{y}, \mathbf{u}_1 \rangle \cdots \langle \mathbf{y}, \mathbf{u}_k \rangle \\ &= \frac{1}{k!} \binom{d_i}{k} \operatorname{Perm} \left[\langle \mathbf{u}_i, \mathbf{u}_j \rangle \right] \\ &\leq \binom{d_i}{k} \end{aligned}$$

It follows that

$$\frac{1}{k!} \left\| D\mathbf{F}_{\mathbf{x}}(0)^{-1} D^k \mathbf{F}_{\mathbf{x}}(0) \right\| \leq \mu(\mathbf{f}, \mathbf{x}) \max \frac{1}{\sqrt{d_i}} \binom{d_i}{k}.$$

Estimating $\binom{d_i}{k} \leq d_i^k 2^{-k}$ and using Exercise 8.1,

$$\gamma(\mathbf{F}_\mathbf{x}, 0) \leq \frac{d^{3/2}}{2}\mu(\mathbf{f}, \mathbf{x}).$$

□

Whenever the sequence $(\mathbf{X}_k)_{k\in\mathbb{N}}$ defined by $\mathbf{X}_0 = 0$, $\mathbf{X}_{k+1} = N(\mathbf{F}_\mathbf{x}, \mathbf{X}_k)$ converges, let $\mathbf{X}^* = \lim \mathbf{X}_k$ and define

$$\zeta_x = \frac{\mathbf{x} + \mathbf{X}^*}{\|\mathbf{x} + \mathbf{X}^*\|} \in S^{n+1}.$$

As in Theorem 5.3, define

$$r_0(\alpha) = \frac{1 + \alpha - \sqrt{1 - 6\alpha + \alpha^2}}{4\alpha}$$

Let α_* the smallest positive root of

$$\alpha_* = \alpha_0(1 - \alpha_* r_0(\alpha_*))^2.$$

Numerically, $\alpha_* > 0.116$. (This is better than (**Cucker et al., 2008**)). Let $B_\mathbf{x} = \{\mathbf{y} \in S^n : \rho(\mathbf{x}, \mathbf{y}) \leq r_\mathbf{x}\}$ with $r_\mathbf{x} = r_0(\alpha_*)\mu(\mathbf{f}, \mathbf{x})\|\mathbf{f}(\mathbf{x})\|$.

THEOREM 9.2. *Let $\mathbf{f} \in S(\mathcal{H}^\mathbb{R}_\mathbf{d})$ and $\mathbf{x} \in S^n$ be such that*

$$(\max d_i)^{3/2}\mu(\mathbf{f}, \mathbf{x})^2\|\mathbf{f}(\mathbf{x})\| \leq \alpha_*.$$

Then,

1. $\alpha(\mathbf{F}, 0) \leq \alpha_*$.
2. *0 is an approximate zero of the second kind of $\mathbf{F}_\mathbf{x}$, and in particular $\mathbf{f}(\zeta_x) = 0$.*
3. $\zeta_\mathbf{x} \in B_\mathbf{x}$.
4. *For any $\mathbf{z} \in B_\mathbf{x}$, $\zeta_\mathbf{z} = \zeta_\mathbf{x}$.*

PROOF. (1) By Lemma 9.1,

$$\alpha(\mathbf{F}_\mathbf{x}, 0) \leq (\max d_i)^{3/2}\mu(\mathbf{f}, \mathbf{x}) \left\|D\mathbf{f}(\mathbf{x})^{-1}_{\mathbf{x}^\perp}\mathbf{f}(\mathbf{x})\right\| \leq$$
$$\leq (\max d_i)^{3/2}\mu(\mathbf{f}, \mathbf{x})^2\|\mathbf{f}(\mathbf{x})\| \leq \alpha_*.$$

(2) Since $\alpha_* \leq \alpha$, we can apply Theorem 5.3 to $\mathbf{F}_\mathbf{x}$ and 0.

(3) Since 0 is a zero of the second kind for $\mathbf{F}_\mathbf{x}$,

$$\mathbf{F}_\mathbf{x}(\mathbf{X}^*) = \mathbf{f}(\|\mathbf{x} + \mathbf{X}^*\|\zeta_\mathbf{x}) = 0$$

and hence by homogeneity $\mathbf{f}(\zeta_\mathbf{x}) = 0$.

(4)

$$\rho(\mathbf{x}, \zeta_\mathbf{x}) \leq \tan\rho(\mathbf{x}, \zeta_\mathbf{x}) \leq r_0(\alpha_*)\beta(\mathbf{f}, \mathbf{x}) \leq r_0(\alpha_*)\mu(\mathbf{f}, \mathbf{x})\|\mathbf{f}(\mathbf{x})\|$$

(5) By Theorem 8.2,

$$\mu(\mathbf{f}, \mathbf{z}) \leq \frac{1}{1 - (\max d_i)\mu(\mathbf{f}, \mathbf{x})\rho(\mathbf{x}, \mathbf{z})}\mu(\mathbf{f}, \mathbf{x}) \leq \frac{1}{1 - \alpha^* r_0(\alpha_*)}\mu(\mathbf{f}, \mathbf{x})$$

and hence, as in item 1:

$$\alpha(\mathbf{F}_\mathbf{z}, 0) \leq \frac{1}{(1 - \alpha^* r_0(\alpha_*))^2}\alpha_* \leq \alpha_0.$$

□

This theorem appeared in **Cucker et al. (2008)**. For other inclusion/exclusion theorems based in alpha-theory, see **Giusti et al. (2007)**.

OPEN PROBLEM 9.3 (Mike Shub). Is it possible to improve α_* by replacing alpha-theory in Theorem 9.2 by the implicit gamma theorem of **Dedieu et al. (2003)**?

10. The exclusion lemma

LEMMA 10.1. *Let* $\mathbf{f} \in S(\mathcal{H}^{\mathbb{R}}_{\mathbf{d}})$ *and let* $\mathbf{x}, \mathbf{y} \in S^n$ *with* $\rho(\mathbf{x}, \mathbf{y}) \leq \sqrt{2}$. *Then,*

$$\|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y})\| \leq \sqrt{\max(d_i)}\rho(\mathbf{x}, \mathbf{y}).$$

In particular, let $\delta = \min(\|\mathbf{f}(\mathbf{x})\|/\sqrt{\max(d_i)}, \sqrt{2})$. *If* $\mathbf{f}(\mathbf{x}) \neq 0$, *then there is no zero of* $\mathbf{f}$ *in*

$$B(\mathbf{x}, \delta) = \{\mathbf{y} \in S^{n+1} : \rho(\mathbf{x}, \mathbf{y}) \leq \delta\}.$$

PROOF. First of all,

$$\begin{aligned} |f_i(x) - f_i(y)| &= |\langle f_i(\cdot), K_{d_i}(\cdot, \mathbf{x}) - K_{d_i}(\cdot, \mathbf{y})\rangle| \\ &\leq \|f_i\|\|K_{d_i}(\cdot, \mathbf{x}) - K_{d_i}(\cdot, \mathbf{y})\| \\ &\leq \|f_i\|\sqrt{K_{d_i}(\mathbf{x}, \mathbf{x}) + K_{d_i}(\mathbf{y}, \mathbf{y}) - 2K_{d_i}(\mathbf{x}, \mathbf{y})} \\ &= \|f_i\|\sqrt{2}\sqrt{1 - \cos(\theta)^d} \end{aligned}$$

with $\theta = \rho(x, y)$. Since $\theta \leq \pi < \sqrt{30}$, we have always

$$\cos(\theta) = 1 - \frac{1}{2}\theta^2 + \frac{1}{4!}\theta^4 - \frac{1}{6!}\theta^6 + \cdots > 1 - \frac{1}{2}\theta^2.$$

The reader will check that for $\epsilon < 1$, $(1 - \epsilon)^d > 1 - d\epsilon$. Therefore, using $\theta < 1/\sqrt{2}$,

$$|f_i(\mathbf{x}) - f_i(\mathbf{y})| \leq \|f_i\|\sqrt{d_i}\theta$$

and

$$\|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{y})\| \leq \sqrt{\max(d_i)}\theta.$$

□

Part 3. The algorithm and its complexity

11. Convexity and geometry Lemmas

DEFINITION 11.1. Let $\mathbf{y}_1, \ldots, \mathbf{y}_s \in S^n$ belong to the same hemisphere, that is $\langle \mathbf{y}_i, \mathbf{z}\rangle > 0$ for a fixed $\mathbf{z}$. The **spherical convex hull** of $\mathbf{y}_1, \ldots, \mathbf{y}_s$ is defined as

$$\mathrm{SCH}(\mathbf{y}_1, \ldots, \mathbf{y}_s) = \left\{ \frac{\lambda_1\mathbf{y}_1 + \cdots + \lambda_s\mathbf{y}_s}{\|\lambda_1\mathbf{y}_1 + \cdots + \lambda_s\mathbf{y}_s\|} : \lambda_1, \ldots, \lambda_s \geq 0 \right. \\ \left. \text{and } \lambda_1 + \cdots + \lambda_s = 1 \right\}.$$

This is the same as the intersection of the sphere with the cone $\{\lambda_1\mathbf{y}_1 + \cdots + \lambda_s\mathbf{y}_s : \lambda_1, \ldots, \lambda_s \geq 0\}$. We will need the following convexity Lemma from **Cucker et al. (2008)**:

LEMMA 11.2. *Let $\mathbf{y}_1, \ldots, \mathbf{y}_s \in S^n$ belong to the same hemisphere. Let $r_1, \ldots, r_s > 0$ and let $B(\mathbf{y}_i, r_i) = \{\mathbf{x} \in S^n : \rho(x, \mathbf{y}_i) < r_i\}$. If $\cap B(\mathbf{y}_i, r_i) \neq \emptyset$, then*

$$\mathrm{SCH}(\mathbf{y}_1, \ldots, \mathbf{y}_s) \subset \cup B(\mathbf{y}_i, r_i).$$

EXERCISE 11.1. Prove Lemma 11.2 above.

For the root counting algorithm, we will need to define a **mesh** on the sphere.

LEMMA 11.3. *For every $\eta = 2^{-t}$, we can construct a set $C(\eta) \subseteq S^n$ satisfying:*

1. *For all $\mathbf{z} \in S^n, \exists \mathbf{x} \in C(\eta)$ such that $\rho(\mathbf{z}, \mathbf{x}) \leq \eta\sqrt{n}/2$.*
2. *For all $\mathbf{x} \in S^n$, let $Y = \{\mathbf{y} \in C(\eta) : \rho(\mathbf{x}, \mathbf{y}) \leq \sqrt{n}\eta\}$. Then $\mathbf{x} \in \mathrm{SCH}(Y)$.*
3. *$\#C(\eta) \leq 2n(1 + 2^{t+1})^n$.*

PROOF. Just set

$$C(\eta) = \left\{ \frac{\mathbf{x}}{\|\mathbf{x}\|} : \mathbf{x} \in \mathbb{R}^{n+1}, x_i \eta^{-1} \in \mathbb{Z}, \|\mathbf{x}\|_\infty = 1 \right\}.$$

This corresponds to dividing $Q = \{\mathbf{x} : \|\mathbf{x}\|_\infty = 1\}$ into n-cubes of side $\tilde{\eta}$. The maximal distance in Q between a point $\mathbf{Z} \in Q$ and a point $\mathbf{X}$ in the mesh is half of the diagonal, or $\eta\sqrt{n}$. Then

$$\rho(\mathbf{Z}/\|\mathbf{Z}\|, \mathbf{X}/\|\mathbf{X}\|) < \eta\sqrt{n}.$$

Now, let Y' be the set of points $\mathbf{y} \in C(\eta)$ such that the distance along Q between $\mathbf{x}/\|\mathbf{x}\|_\infty$ and $\mathbf{y}/\|\mathbf{y}\|_\infty$ is at most η. Then clearly $\mathbf{x} \in \mathrm{SCH}(Y')$. Moreover, $Y' \subset Y$.

The last item is trivial. □

12. The counting algorithm

Given $\mathbf{f} \in S(\mathcal{H}_{\mathbf{d}}^{\mathbb{R}})$ and $\eta = 2^{-t}$, we construct a graph $\mathcal{G}_\eta = (\mathcal{V}_\eta, \mathcal{E}_\eta)$ as follows. Let

$$A(\mathbf{f}) = \{\mathbf{x} \in S^n : \max d_i^{3/2} \mu(\mathbf{f}, \mathbf{x})^2 \|\mathbf{f}(\mathbf{x})\| < \alpha_*\}$$

be the set of points satisfying the hypotheses of Theorem 9.2. The set of vertices of $\mathcal{G}_\eta$ is $\mathcal{V}_\eta = C(\eta) \cap A(\mathbf{f})$.

Recall that Let $B_{\mathbf{x}} = \{\mathbf{y} \in S^n : \rho(\mathbf{x}, \mathbf{y}) \leq r_{\mathbf{x}}\}$ with $r_{\mathbf{x}} = r_0(\alpha_*)\mu(\mathbf{f}, \mathbf{x})\|\mathbf{f}(\mathbf{x})\|$. The set of edges of $\mathcal{G}_\eta$ is $\mathcal{E}_\eta = \{(\mathbf{x}, \mathbf{y}) \in \mathcal{V}_\eta \times \mathcal{V}_\eta : B_{\mathbf{x}} \cap B_{\mathbf{y}} \neq \emptyset\}$. This graph is clearly constructible. Theorem 9.2 implies that for any edge $(\mathbf{x}, \mathbf{y}) \in \mathcal{E}_\eta$, $\zeta_{\mathbf{x}} = \zeta_{\mathbf{y}}$. More generally,

LEMMA 12.1. *The vertices of any connected component of $\mathcal{G}(\eta)$ are approximate zeros associated to the same zero of $\mathbf{f}$. Moreover, if $\mathbf{x}, \mathbf{y}$ belong to distinct connected components of $\mathcal{G}(\eta)$, then $\zeta_{\mathbf{x}} \neq \zeta_{\mathbf{y}}$.*

The algorithm is as follows:

Algorithm `RootCount`
Input: $\mathbf{f} \in S(\mathcal{H}_{\mathbf{d}}^{\mathbb{R}})$.
Output: $\#\zeta \in S^n : \mathbf{f}(\zeta) = 0$.

$\eta \leftarrow 2^{-\lceil \log_2(1/\sqrt{2n}) \rceil}$.

Repeat

$\eta \leftarrow \eta/2$.

Let $\mathcal{U}_1, \ldots, \mathcal{U}_r$ be the connected components of $\mathcal{G}_\eta$.

Until $\forall 1 \leq i < j \leq r, \forall \mathbf{x}$ vertex of $\mathcal{U}_i, \forall \mathbf{y}$ vertex of $\mathcal{U}_j$,

$$\rho(\mathbf{x}, \mathbf{y}) > 2\eta\sqrt{n}. \tag{16}$$

and $\forall \mathbf{x} \in C(\eta) \setminus A(\mathbf{f})$,

$$\|\mathbf{f}(\mathbf{x})\| > \eta\sqrt{n \max d_i}/2. \tag{17}$$

Return r.

THEOREM 12.2. *If the algorithm* `RootCount` *stops, then* r *is the correct number of roots of* $\mathbf{f}$ *in* S^n.

PROOF OF TH.12.2. Suppose the algorithm stopped at a certain value of η. As each connected component $\mathcal{U}_i$ determines a distinct and unique zero of $\mathbf{f}$, it remains to prove that there is no zero of $\mathbf{f}$ outside $\cup_{\mathbf{x} \in \mathcal{V}_\eta} B_{\mathbf{x}}$.

Therefore, assume by contradiction that there is $\zeta \in S^n$ with $\mathbf{f}(\zeta) = 0$ and $\zeta \notin B_{\mathbf{x}}$ for any $\mathbf{x} \in V_\eta$.

Let Y be the set of $\mathbf{y} \in C(\eta)$ with $\rho(\zeta, \mathbf{y}) \leq \eta\sqrt{n}$.

If there is $\mathbf{y} \in Y$ with $\mathbf{y} \notin A(\mathbf{f})$ let $\delta = \|\mathbf{f}(\mathbf{y})\|/\sqrt{\max d_i}$. Equation (17) guarantees that $\eta\sqrt{n}/2 < \delta$. By construction, $\eta\sqrt{n}/2 < \sqrt{2}$. Therefore, the exclusion lemma 10.1 guarantees that $\mathbf{f}(\zeta) \neq 0$, contradiction.

Therefore, we assume that $Y \subset A(\mathbf{f})$. Equation (16) guarantees that $Y \subset \mathcal{U}_k$ for a same connected component of $\mathcal{G}_\eta$. Therefore, $\cap_{\mathbf{y} \in Y} B_{\mathbf{y}} \ni \zeta$ is not empty.

By Lemma 11.3(2), $\mathbf{x} \in \mathrm{SCH}(Y)$. Lemma 11.2 says that

$$\mathrm{SCH}(Y) \subseteq \cup_{\mathbf{y} \in Y} B_{\mathbf{y}}$$

Thus, $\mathbf{x} \in B_{\mathbf{y}}$ for some $\mathbf{y}$, contradiction again.

□

A consequence of Th.12.2 is that **if the algorithm stops**, one can obtain an approximate zero of the second kind for each root of f by recovering one vertex for each connected component.

13. Complexity

We did not prove that algorithm `RootCount` stops. It actually stops **almost surely**, that is for input f outside a certain measure zero set.

Define

$$\kappa(\mathbf{f}, \mathbf{x}) = \frac{1}{\sqrt{\mu(\mathbf{f}, \mathbf{x})^{-2} + \|\mathbf{f}(\mathbf{x})\|^2}}$$

and notice that

$$\kappa(\mathbf{f}, \mathbf{x}) \leq \mu(\mathbf{f}, \mathbf{x}) \text{ and } \kappa(\mathbf{f}, \mathbf{x}) \leq \|\mathbf{f}(\mathbf{x})\|^{-1}.$$

Reciprocally,

$$\min(\mu(\mathbf{f}, \mathbf{x}), \|\mathbf{f}(\mathbf{x})\|^{-1}) \leq \sqrt{2}\kappa(\mathbf{f}, \mathbf{x}).$$

If $\mathbf{f}(\mathbf{x}) = 0$, then $\kappa(\mathbf{f}, \mathbf{x}) = \mu(\mathbf{f}, \mathbf{x})$.

DEFINITION 13.1. The **condition number** for for Problem 1.2 (counting real zeros on the sphere) is

$$\kappa(\mathbf{f}) = \max_{\mathbf{x} \in S^n} \kappa(\mathbf{f}, \mathbf{x}).$$

Assume that $\mathbf{f}$ has no degenerate root. Then the denominator is bounded away from zero, and $\kappa(\mathbf{f})$ is finite. We will prove later that the algorithm stops for $\kappa(\mathbf{f})$ finite. But before, we state and prove the **condition number theorem** to obtain some geometric intuition on $\kappa(\mathbf{f})$.

THEOREM 13.2. (**Cucker et al., 2009**) *Let* $\Sigma^{\mathbb{R}} = \{\mathbf{g} \in \mathcal{H}^{\mathbb{R}}_{\mathbf{d}} : \exists \zeta \in S^n : \mathbf{g}(\zeta) = 0$ *and* $\operatorname{rk}(D\mathbf{g}(\zeta)) < n\}$. *Let* $\mathbf{f} \in S(\mathcal{H}^{\mathbb{R}}_{\mathbf{d}})$, $\mathbf{f} \notin \Sigma^{\mathbb{R}}$. *Then,*

$$\kappa(\mathbf{f}) = \frac{1}{\min_{\mathbf{g}\in\Sigma^{\mathbb{R}}} \|\mathbf{f}-\mathbf{g}\|}.$$

In particular, $\kappa(\mathbf{f}) \geq 1$.

PROOF. It suffices to prove that

$$\kappa(\mathbf{f},\mathbf{x}) = \frac{1}{\min\limits_{\substack{\mathbf{g}\in\mathcal{H}^{\mathbb{R}}_{\mathbf{d}} \\ \mathbf{g}(\mathbf{x})=0 \\ \operatorname{rk}(D\mathbf{g}(\mathbf{x}))<n}} \|\mathbf{f}-\mathbf{g}\|}.$$

We proceed as in the proof of Prop.8.1. We decompose

$$\mathcal{H}^{\mathbb{R}}_{\mathbf{d}} = H_0 \oplus H_1 \oplus H_2 \oplus \cdots$$

where H_0 and H_1 correspond to the constant and linear terms of $\mathbf{y} \mapsto \mathbf{f}(\mathbf{x}+\mathbf{y})$. Let $\mathbf{u}_1, \ldots, \mathbf{u}_n$ be an orthonormal basis for $\mathbf{x}^\perp$.

An orthonormal basis for $H_0 \oplus H_1$ is

$$\left(K_{d_i}(\cdot,\mathbf{x}), \frac{1}{\sqrt{d}} \frac{\partial K_{d_i}(\cdot,\mathbf{x})}{\partial \mathbf{u}_j} \right).$$

The projection of $\mathbf{f}$ in $H_0 \oplus H_1$ is

$$\left[\langle \mathbf{f}(\cdot), K_{d_i}(\cdot,\mathbf{x})\rangle\right] \oplus \begin{bmatrix} & \vdots & \\ \cdots & \left\langle \mathbf{f}_i, \frac{1}{\sqrt{d}} \frac{\partial K_{d_i}(\cdot,\mathbf{x})}{\partial \mathbf{u}_j} \right\rangle & \cdots \\ & \vdots & \end{bmatrix} =$$

$$= \mathbf{f}(\mathbf{x}) \oplus \begin{bmatrix} d_1^{-1/2} & & \\ & d_2^{-1/2} & \\ & & d_n^{-1/2} \end{bmatrix} D\mathbf{f}(\mathbf{x})_{|\mathbf{x}^\perp}.$$

This is an orthogonal projection onto $\mathbb{R}^n \times \mathbb{R}^{n\times n}$.

Now,

$$\kappa(\mathbf{f},\mathbf{x})^{-2} = \|\mathbf{f}(\mathbf{x})\|^2 + \sigma_n \left(\begin{bmatrix} d_1^{-1/2} & & \\ & d_2^{-1/2} & \\ & & d_n^{-1/2} \end{bmatrix} D\mathbf{f}(\mathbf{x})_{|\mathbf{x}^\perp} \right).$$

Again, we apply Th.6.3. □

LEMMA 13.3. *Let* ζ_1, ζ_2 *be distinct roots of* $\mathbf{f}$ *in* S^n. *Then,*

$$\rho(\zeta_1,\zeta_2) \geq \frac{1}{\max d_i^{3/2} \kappa(\mathbf{f})}$$

PROOF.

$$\begin{aligned}\|\zeta_1 - \zeta_2\| &\geq \frac{1}{2\gamma(\mathbf{f}, \zeta_1)} && \text{by Ex.4.3}\\ &\geq \frac{1}{\max d_i^{3/2}\mu(\mathbf{f}, \zeta_1)} && \text{by Lem.9.1}\\ &\geq \frac{1}{\max d_i^{3/2}\kappa(\mathbf{f})} && \text{because } \mathbf{f}(\zeta_1) = 0.\end{aligned}$$

The Lemma follows. □

LEMMA 13.4. *Assume that*

$$\eta < \frac{1}{2\max d_i^{3/2}\sqrt{n}\kappa(\mathbf{f})}(1 - 2\alpha_* r_0(\alpha_*)).$$

Then (16) *holds.*

PROOF. Recall that $\mathbf{x}$ and $\mathbf{y}$ belong to $A_{\mathbf{f}}$, so that

$$\max d_i^{3/2}\mu(\mathbf{f}, \mathbf{x})^2\|\mathbf{f}(\mathbf{x})\| < \alpha_*$$

and the same for $\mathbf{y}$. In particular, the radius $r_{\mathbf{x}}$ of $B_{\mathbf{x}}$ satisfies

$$r_0(\alpha_*)\mu(\mathbf{f}, \mathbf{x})\|\mathbf{f}(\mathbf{x})\| < \frac{\alpha_* r_0(\alpha_*)}{\max d_i^{3/2}\mu(\mathbf{f}, \mathbf{x})} \leq \frac{\alpha_* r_0(\alpha_*)}{\max d_i^{3/2}\kappa(\mathbf{f}, \mathbf{x})}.$$

By Lemma 13.3 and the triangle inequality,

$$\begin{aligned}\rho(\mathbf{x}, \mathbf{y}) &\geq \rho(\zeta_{\mathbf{x}}, \zeta_{\mathbf{y}}) - r_0(\alpha_*)\mu(\mathbf{f}, \mathbf{x})\|\mathbf{f}(\mathbf{x})\| - r_0(\alpha_*)\mu(\mathbf{f}, \mathbf{y})\|\mathbf{f}(\mathbf{y})\|\\ &\geq \frac{1}{\max d_i^{3/2}\kappa(\mathbf{f})}(1 - 2\alpha_* r_0(\alpha_*)).\end{aligned}$$

□

LEMMA 13.5. *Let* $\mathbf{x} \notin A_f$. *Then,*

$$\|\mathbf{f}(\mathbf{x})\| \geq \frac{\alpha_*}{\kappa(\mathbf{f}, \mathbf{x})^2 \max d_i^{3/2}}.$$

PROOF. Let $\mathbf{x} \notin A_{\mathbf{f}}$, so that

$$\frac{\max d_i^{3/2}}{2}\mu(\mathbf{f}, \mathbf{x})^2\|\mathbf{f}(\mathbf{x})\| \geq \alpha_*.$$

Recall that

$$\min(\mu(\mathbf{f}, \mathbf{x}), \|\mathbf{f}(\mathbf{x})\|^{-1}) \leq \sqrt{2}\kappa(\mathbf{f}, \mathbf{x})$$

There are two possibilities. If $\mu(\mathbf{f}, \mathbf{x}) \leq \sqrt{2}\kappa(\mathbf{f}, \mathbf{x})$, then

$$\|\mathbf{f}(\mathbf{x})\| \geq \frac{\alpha_*}{\max d_i^{3/2}\kappa(\mathbf{f}, \mathbf{x})^2}.$$

Otherwise,

$$\|\mathbf{f}(\mathbf{x})\| \geq \frac{1}{\sqrt{2}\kappa(\mathbf{f}, \mathbf{x})} \geq \frac{\alpha_*}{\max d_i^{3/2}\kappa(\mathbf{f}, \mathbf{x})^2}.$$

□

Now we can state the 'cloud complexity' theorem.

THEOREM 13.6. *The algorithm* `RootCount` *will stop for*

$$\eta < \frac{1}{\max d_i^{3/2}\kappa(\mathbf{f})^2}\min\left(\alpha_*\ ,\ \frac{\kappa(\mathbf{f})}{2\sqrt{n}}(1-2\alpha_* r_0(\alpha_*))\right)$$

that is, after $O(\log\kappa(\mathbf{f})+\log\max d_i)$ *iterations. The total number of evaluations of* $\mathbf{f}$ *and* $D\mathbf{f}$ *is*

$$2n(1+4\max d_i^{3/2}\sqrt{n}\kappa(\mathbf{f})^2)^n.$$

That means that $2n(1+4\max d_i^{3/2}\sqrt{n}\kappa(\mathbf{f})^2)^n$ processors in parallel can compute the root count in time $O(\log\kappa(\mathbf{f})+\log\max d_i)$ times a polynomial in n for the linear algebra.

For people concerned with the overall computing cost, a price tag exponential in n is known as the **curse of dimensionality**. It usually plagues divide and conquer and Monte-Carlo algorithms.

But the situation $n=2$ is already interesting. How efficiently can we count zeros of a system of polynomials on the 2-sphere? As the parallel and sequential running time depends upon $\kappa(f)$, it is useful to known more about the condition number.

14. Probabilistic and smoothed analysis

One possibility is to pick the input system $\mathbf{f}$ at random, and treat $\kappa(\mathbf{f})$ as a random variable. For instance, let $\mathbf{f}\in\mathcal{H}_{\mathbf{d}}^{\mathbb{R}}$ be random with **Gaussian** probability distribution

$$\frac{1}{(2\pi)^{\dim\mathcal{H}_{\mathbf{d}}^{\mathbb{R}}/2}}e^{-\|f\|^2/2}\ \mathrm{d}\mathcal{H}_{\mathbf{d}}^{\mathbb{R}}.$$

The tail for the random variable $\kappa(\mathbf{f})$ and the expected value of $\log\kappa(\mathbf{f})$ can be bounded by

THEOREM 14.1 ((**Cucker, Krick, Malajovich, and Wschebor, 2012**)). *Let* $\mathbf{f}$ *be as above. Assume that* $n\geq 3$. *Then,*

(i) *For* $a>4\sqrt{2}\,(\max d_i)^2 n^{7/2}N^{1/2}$ *we have*

$$\mathrm{Prob}\big(\kappa(\mathbf{f})>a\big)\leq K_n\frac{\sqrt{2n}(1+\ln(a/\sqrt{2n}))^{1/2}}{a},$$

where $N=\dim\mathcal{H}_{\mathbf{d}}^{\mathbb{R}}$, $K_n:=8(\max d_i)^2\mathcal{D}^{1/2}N^{1/2}n^{5/2}+1$ *and* $\mathcal{D}=\prod d_i$.

(ii)

$$\mathbb{E}(\ln\kappa(\mathbf{f}))\leq\ln K_n+(\ln K_n)^{1/2}+(\ln K_n)^{-1/2}+\frac{1}{2}\ln(2n).$$

Notice as a consequence that the expected running time of `RootCount` is

$$\mathbb{E}(\ln\kappa(\mathbf{f}))\in\mathcal{O}(n\ln\max d_i).$$

This is cloud computing time, of course.

Average time analysis depends upon an arbitrary distribution. **Spielman and Teng (2004)** suggested looking instead at a small random perturbation for each given input. This is known as **smoothed analysis**.

For a given $\mathbf{f}\in S(\mathcal{H}_{\mathbf{d}}^{\mathbb{R}})$, we will consider the uniform distribution in the ball $B(\mathbf{f},\arcsin\sigma)\subset S(\mathcal{H}_{\mathbf{d}}^{\mathbb{R}})$ where σ is an arbitrary radius, and Riemannian metric on the sphere is assumed. The strange looking arcsine comes from the fact that $B(\mathbf{f},\arcsin\sigma)$ is the projection on the sphere of the ball $B(\mathbf{f},\sigma)\subset\mathcal{H}_{\mathbf{d}}^{\mathbb{R}}$. The reason

for looking at the uniform distribution for perturbations instead of Gaussian is the following result:

THEOREM 14.2. (**Bürgisser et al., 2008**) *Let $\Sigma \subset \mathbb{R}^N$ be contained in a projective hypersurface H of degree at most D and let $\kappa : \mathbb{S}^{N-1} \to [1, \infty]$ be given by*

$$\kappa(\mathbf{f}) = \frac{\|\mathbf{f}\|}{\min_{\mathbf{g}\in\Sigma} \|\mathbf{f} - \mathbf{g}\|}.$$

Then, for all $\sigma \in (0, 1]$,

$$\sup_{\mathbf{f}\in S^{N-1}} \mathrm{E}_{\mathbf{h}\in B(\mathbf{f},\arcsin\sigma)\subseteq S^{N-1}}(\ln \kappa(\mathbf{h})) \leq 2\ln(N-1) + 2\ln D - \ln\sigma + 5.5.$$

In the context of the root counting problem, the degree D of $\Sigma = \Sigma^{\mathbb{R}}$ is bounded by $n^2(\prod d_i)(\max d_i)$. Therefore,

COROLLARY 14.3. (**Cucker et al., 2009**)

$$\sup_{\mathbf{f}\in S(\mathcal{H}^{\mathbb{R}}_{\mathbf{d}})} \mathrm{E}_{\mathbf{h}\in B(\mathbf{f},\arcsin\sigma)\subseteq S(\mathcal{H}^{\mathbb{R}}_{\mathbf{d}})}(\ln \kappa(h)) \leq 2\ln(\dim(\mathcal{H}^{\mathbb{R}}_{\mathbf{d}})) + 4\ln(n) + 2\ln(\prod d_i) + \ln 1/\sigma + 6.$$

15. Conclusions

We sketched the average time analysis and a smoothed analysis of an algorithm for real root counting and, incidentally, root finding. The same algorithm can also decide if a given polynomial system admits a root.

Loosely speaking, deciding (resp. counting) roots of polynomial systems are NP-complete (resp. #P complete) problems. The formal NP-complete and #P-complete problems refer to **sparse** polynomial systems.

Our algorithm uses polynomial evaluations, so it can take advantage of the sparse structure. Moreover, the degree of the sparse discriminant is no more than the degree of the usual discriminant. In that sense Corollary 14.3 is still valid. The running time of the algorithm is polynomial in n and in the dimension of the input space. Again, this is a massively parallel algorithm so the number of processors is exponential in n.

References

Blum, L., F. Cucker, M. Shub, and S. Smale. 1998. *Complexity and real computation*, Springer-Verlag, New York. With a foreword by Richard M. Karp. MR1479636 (99a:68070)

Bürgisser, P. and F. Cucker. 2006. *Counting complexity classes for numeric computations. II. Algebraic and semialgebraic sets*, J. Complexity **22**, no. 2, 147–191, DOI 10.1016/j.jco.2005.11.001. MR2200367 (2007b:68059)

Bürgisser, P. and F. Cucker. 2011. *On a problem posed by Steve Smale*, Ann. of Math. (2) **174**, no. 3, 1785–1836, DOI 10.4007/annals.2011.174.3.8. MR2846491

Bürgisser, P., F. Cucker, and M. Lotz. 2008. *The probability that a slightly perturbed numerical analysis problem is difficult*, Math. Comp. **77**, no. 263, 1559–1583, DOI 10.1090/S0025-5718-08-02060-7. MR2398780 (2009a:65132)

Cucker, F., T. Krick, G. Malajovich, and M. Wschebor. 2008. *A numerical algorithm for zero counting. I. Complexity and accuracy*, J. Complexity **24**, no. 5-6, 582–605, DOI 10.1016/j.jco.2008.03.001. MR2467589 (2010d:68063)

Cucker, F., T. Krick, G. Malajovich, and M. Wschebor. 2009. *A numerical algorithm for zero counting. II. Distance to ill-posedness and smoothed analysis*, J. Fixed Point Theory Appl. **6**, no. 2, 285–294, DOI 10.1007/s11784-009-0127-4. MR2580979 (2011c:65317)

Cucker, F., T. Krick, G. Malajovich, and M. Wschebor. 2012. *A numerical algorithm for zero counting. III: Randomization and condition*, Adv. in Appl. Math. **48**, no. 1, 215–248, DOI 10.1016/j.aam.2011.07.001. MR2845516

Dedieu, J.-P. 1997a. *Estimations for the separation number of a polynomial system*, J. Symbolic Comput. **24**, no. 6, 683–693, DOI 10.1006/jsco.1997.0161. MR1487794 (99b:65065)

Dedieu, J.-P. 1997b. *Estimations for the separation number of a polynomial system*, J. Symbolic Comput. **24**, no. 6, 683–693, DOI 10.1006/jsco.1997.0161. MR1487794 (99b:65065)

Dedieu, J.-P., M.-H. Kim, M. Shub, and F. Tisseur. 2003. *Implicit gamma theorems. I. Pseudoroots and pseudospectra*, Found. Comput. Math. **3**, no. 1, 1–31, DOI 10.1007/s10208-001-0049-z. MR1951501 (2003k:65049)

Dedieu, J.-P., G. Malajovich, and M. Shub. 2013. *Adaptive step-size selection for homotopy methods to solve polynomial equations*, IMA J. Numer. Anal. **33**, no. 1, 1–29, DOI 10.1093/imanum/drs007. MR3020948

Demmel, J. W. 1988. *The probability that a numerical analysis problem is difficult*, Math. Comp. **50**, no. 182, 449–480, DOI 10.2307/2008617. MR929546 (89g:65062)

Edelman, A. 1992. *On the distribution of a scaled condition number*, Math. Comp. **58**, no. 197, 185–190, DOI 10.2307/2153027. MR1106966 (92g:15034)

Eckart, C. and G. Young. 1936. *The approximation of a matrix by another of lower rank*, Psychometrika **1**, no. 3, 211–218, DOI 10.1007/BF02288367.

Eckart, C. and G. Young. 1939. *A principal axis transformation for non-hermitian matrices*, Bull. Amer. Math. Soc. **45**, no. 2, 118–121, DOI 10.1090/S0002-9904-1939-06910-3. MR1563923

Giusti, M., G. Lecerf, B. Salvy, and J.-C. Yakoubsohn. 2007. *On location and approximation of clusters of zeros: case of embedding dimension one*, Found. Comput. Math. **7**, no. 1, 1–49, DOI 10.1007/s10208-004-0159-5. MR2283341 (2008e:65159)

Higham, N. J. 2002. *Accuracy and stability of numerical algorithms*, 2nd ed., Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA. MR1927606 (2003g:65064)

IEEE, Inc. 2008. *IEEE Standard for Floating Point Arithmetic IEEE Std 754-2008*, 3 Park Avenue, New York, NY 10016-5997, USA.

Kantorovich, L. V. 1996. *Selected works. Part II*, Classics of Soviet Mathematics, vol. 3, Gordon and Breach Publishers, Amsterdam. Applied functional analysis. Approximation methods and computers; Translated from the Russian by A. B. Sossinskii; Edited by S. S. Kutateladze and J. V. Romanovsky. MR1800892 (2002e:01067)

Malajovich, G. 1993. *On the complexity of path-following Newton algorithms for solving systems of polynomial equations with integer coefficients*, PhD Thesis, Department of Mathematics, University of California at Berkeley, `http://www.labma.ufrj.br/~gregorio/papers/thesis.pdf`.

Malajovich, G. 1994. *On generalized Newton algorithms: quadratic convergence, path-following and error analysis*, Theoret. Comput. Sci. **133**, no. 1, 65–84, DOI 10.1016/0304-3975(94)00065-4. Selected papers of the Workshop on Continuous Algorithms and Complexity (Barcelona, 1993). MR1294426 (95g:65073)

Malajovich, G. 2011. *Nonlinear equations*, Publicações Matemáticas do IMPA. [IMPA Mathematical Publications], Instituto Nacional de Matemática Pura e Aplicada (IMPA), Rio de Janeiro. With an appendix by Carlos Beltrán, Jean-Pierre Dedieu, Luis Miguel Pardo and Mike Shub; 28º Colóquio Brasileiro de Matemática. [28th Brazilian Mathematics Colloquium]. MR2798351 (2012j:65148)

Meer, K. 2000. *Counting problems over the reals*, Theoret. Comput. Sci. **242**, no. 1-2, 41–58, DOI 10.1016/S0304-3975(98)00190-X. MR1769145 (2002g:68041)

Nachbin, L. 1964. *Lectures on the theory of distributions*, Textos de Matemática, No. 15, Instituto de Física e Matemática, Universidade do Recife, Recife. MR0213868 (35 #4722)

Nachbin, L. 1969. *Topology on spaces of holomorphic mappings*, Ergebnisse der Mathematik und ihrer Grenzgebiete, Band 47, Springer-Verlag New York Inc., New York. MR0254579 (40 #7787)

Shub, M. and S. Smale. 1993. *Complexity of Bézout's theorem. I. Geometric aspects*, J. Amer. Math. Soc. **6**, no. 2, 459–501, DOI 10.2307/2152805. MR1175980 (93k:65045)

Smale, S. 1985. *On the efficiency of algorithms of analysis*, Bull. Amer. Math. Soc. (N.S.) **13**, no. 2, 87–121, DOI 10.1090/S0273-0979-1985-15391-1. MR799791 (86m:65061)

Dedieu, J.-P. 2006. *Points fixes, zéros et la méthode de Newton*, Mathématiques & Applications (Berlin) [Mathematics & Applications], vol. 54, Springer, Berlin (French). With a preface by Steve Smale. MR2510891

Spielman, D. A. and S.-H. Teng. 2004. *Smoothed analysis of algorithms: why the simplex algorithm usually takes polynomial time*, J. ACM **51**, no. 3, 385–463 (electronic), DOI 10.1145/990308.990310. MR2145860 (2006f:90029)

Turing, A. M. 1948. *Rounding-off errors in matrix processes*, Quart. J. Mech. Appl. Math. **1**, 287–308. MR0028100 (10,405c)

Wang, X. H. 1993. *Some results relevant to Smale's reports*, From Topology to Computation: Proceedings of the Smalefest (Berkeley, CA, 1990), Springer, New York, pp. 456–465. MR1246140

Wilkinson, J. H. 1963. *Rounding errors in algebraic processes*, Prentice-Hall Inc., Englewood Cliffs, N.J. MR0161456 (28 #4661)

Departamento de Matemática Aplicada, Instituto de Matemática, Universidade Federal do Rio de Janeiro. Caixa Postal 68530, Rio de Janeiro RJ 21941-909, Brasil

E-mail address: gregorio.malajovich@gmail.com

URL: www.labma.ufrj.br/~gregorio

Selected Published Titles in This Series

604 **José Luis Montaña and Luis M. Pardo, Editors,** Recent Advances in Real Complexity and Computation, 2013

598 **Eric Todd Quinto, Fulton Gonzalez, and Jens Gerlach Christensen, Editors,** Geometric Analysis and Integral Geometry, 2013

595 **James B. Serrin, Enzo L. Mitidieri, and Vicenţiu D. Rădulescu, Editors,** Recent Trends in Nonlinear Partial Differential Equations II, 2013

594 **James B. Serrin, Enzo L. Mitidieri, and Vicenţiu D. Rădulescu, Editors,** Recent Trends in Nonlinear Partial Differential Equations I, 2013

593 **Anton Dzhamay, Kenichi Maruno, and Virgil U. Pierce, Editors,** Algebraic and Geometric Aspects of Integrable Systems and Random Matrices, 2013

592 **Arkady Berenstein and Vladimir Retakh, Editors,** Noncommutative Birational Geometry, Representations and Combinatorics, 2013

591 **Mark L. Agranovsky, Matania Ben-Artzi, Greg Galloway, Lavi Karp, Vladimir Maz'ya, Simeon Reich, David Shoikhet, Gilbert Weinstein, and Lawrence Zalcman, Editors,** Complex Analysis and Dynamical Systems V, 2013

590 **Ursula Hamenstädt, Alan W. Reid, Rubí Rodríguez, Steffen Rohde, and Michael Wolf, Editors,** In the Tradition of Ahlfors-Bers, VI, 2013

589 **Erwan Brugallé, Mariá Angélica Cueto, Alicia Dickenstein, Eva-Maria Feichtner, and Ilia Itenberg, Editors,** Algebraic and Combinatorial Aspects of Tropical Geometry, 2013

588 **David A. Bader, Henning Meyerhenke, Peter Sanders, and Dorothea Wagner, Editors,** Graph Partitioning and Graph Clustering, 2013

587 **Wai Kiu Chan, Lenny Fukshansky, Rainer Schulze-Pillot, and Jeffrey D. Vaaler, Editors,** Diophantine Methods, Lattices, and Arithmetic Theory of Quadratic Forms, 2013

586 **Jichun Li, Hongtao Yang, and Eric Machorro, Editors,** Recent Advances in Scientific Computing and Applications, 2013

585 **Nicolás Andruskiewitsch, Juan Cuadra, and Blas Torrecillas, Editors,** Hopf Algebras and Tensor Categories, 2013

584 **Clara L. Aldana, Maxim Braverman, Bruno Iochum, and Carolina Neira Jiménez, Editors,** Analysis, Geometry and Quantum Field Theory, 2012

583 **Sam Evens, Michael Gekhtman, Brian C. Hall, Xiaobo Liu, and Claudia Polini, Editors,** Mathematical Aspects of Quantization, 2012

582 **Benjamin Fine, Delaram Kahrobaei, and Gerhard Rosenberger, Editors,** Computational and Combinatorial Group Theory and Cryptography, 2012

581 **Andrea R. Nahmod, Christopher D. Sogge, Xiaoyi Zhang, and Shijun Zheng, Editors,** Recent Advances in Harmonic Analysis and Partial Differential Equations, 2012

580 **Chris Athorne, Diane Maclagan, and Ian Strachan, Editors,** Tropical Geometry and Integrable Systems, 2012

579 **Michel Lavrauw, Gary L. Mullen, Svetla Nikova, Daniel Panario, and Leo Storme, Editors,** Theory and Applications of Finite Fields, 2012

578 **G. López Lagomasino,** Recent Advances in Orthogonal Polynomials, Special Functions, and Their Applications, 2012

577 **Habib Ammari, Yves Capdeboscq, and Hyeonbae Kang, Editors,** Multi-Scale and High-Contrast PDE, 2012

576 **Lutz Strüngmann, Manfred Droste, László Fuchs, and Katrin Tent, Editors,** Groups and Model Theory, 2012

575 **Yunping Jiang and Sudeb Mitra, Editors,** Quasiconformal Mappings, Riemann Surfaces, and Teichmüller Spaces, 2012

574 **Yves Aubry, Christophe Ritzenthaler, and Alexey Zykin, Editors,** Arithmetic, Geometry, Cryptography and Coding Theory, 2012

573 **Francis Bonahon, Robert L. Devaney, Frederick P. Gardiner, and Dragomir Šarić, Editors,** Conformal Dynamics and Hyperbolic Geometry, 2012

572 **Mika Seppälä and Emil Volcheck, Editors,** Computational Algebraic and Analytic Geometry, 2012

571 **José Ignacio Burgos Gil, Rob de Jeu, James D. Lewis, Juan Carlos Naranjo, Wayne Raskind, and Xavier Xarles, Editors,** Regulators, 2012

570 **Joaquín Pérez and José A. Gálvez, Editors,** Geometric Analysis, 2012

569 **Victor Goryunov, Kevin Houston, and Roberta Wik-Atique, Editors,** Real and Complex Singularities, 2012

568 **Simeon Reich and Alexander J. Zaslavski, Editors,** Optimization Theory and Related Topics, 2012

567 **Lewis Bowen, Rostislav Grigorchuk, and Yaroslav Vorobets, Editors,** Dynamical Systems and Group Actions, 2012

566 **Antonio Campillo, Gabriel Cardona, Alejandro Melle-Hernández, Wim Veys, and Wilson A. Zúñiga-Galindo, Editors,** Zeta Functions in Algebra and Geometry, 2012

565 **Susumu Ariki, Hiraku Nakajima, Yoshihisa Saito, Ken-ichi Shinoda, Toshiaki Shoji, and Toshiyuki Tanisaki, Editors,** Algebraic Groups and Quantum Groups, 2012

564 **Valery Alexeev, Angela Gibney, Elham Izadi, János Kollár, and Eduard Looijenga, Editors,** Compact Moduli Spaces and Vector Bundles, 2012

563 **Primitivo B. Acosta-Humánez, Federico Finkel, Niky Kamran, and Peter J. Olver, Editors,** Algebraic Aspects of Darboux Transformations, Quantum Integrable Systems and Supersymmetric Quantum Mechanics, 2012

562 **P. Ara, K. A. Brown, T. H. Lenagan, E. S. Letzter, J. T. Stafford, and J. J. Zhang, Editors,** New Trends in Noncommutative Algebra, 2012

561 **Óscar Blasco, José A. Bonet, José M. Calabuig, and David Jornet, Editors,** Topics in Complex Analysis and Operator Theory, 2012

560 **Weiping Li, Loretta Bartolini, Jesse Johnson, Feng Luo, Robert Myers, and J. Hyam Rubinstein, Editors,** Topology and Geometry in Dimension Three, 2011

559 **Guillaume Bal, David Finch, Peter Kuchment, John Schotland, Plamen Stefanov, and Gunther Uhlmann, Editors,** Tomography and Inverse Transport Theory, 2011

558 **Martin Grohe and Johann A. Makowsky, Editors,** Model Theoretic Methods in Finite Combinatorics, 2011

557 **Jeffrey Adams, Bong Lian, and Siddhartha Sahi, Editors,** Representation Theory and Mathematical Physics, 2011

556 **Leonid Gurvits, Philippe Pébay, J. Maurice Rojas, and David Thompson, Editors,** Randomization, Relaxation, and Complexity in Polynomial Equation Solving, 2011

555 **Alberto Corso and Claudia Polini, Editors,** Commutative Algebra and Its Connections to Geometry, 2011

554 **Mark Agranovsky, Matania Ben-Artzi, Greg Galloway, Lavi Karp, Simeon Reich, David Shoikhet, Gilbert Weinstein, and Lawrence Zalcman, Editors,** Complex Analysis and Dynamical Systems IV: Part 2. General Relativity, Geometry, and PDE, 2011

553 **Mark Agranovsky, Matania Ben-Artzi, Greg Galloway, Lavi Karp, Simeon Reich, David Shoikhet, Gilbert Weinstein, and Lawrence Zalcman, Editors,** Complex Analysis and Dynamical Systems IV: Part 1. Function Theory and Optimization, 2011

552 **Robert Sims and Daniel Ueltschi, Editors,** Entropy and the Quantum II, 2011

For a complete list of titles in this series, visit the
AMS Bookstore at **www.ams.org/bookstore/conmseries/**.

CONM 604

Real Complexity and Computation • Montaña et al., Editors

This volume is composed of six contributions derived from the lectures given during the UIMP-RSME Lluís Santaló Summer School on "Recent Advances in Real Complexity and Computation", held July 16–20, 2012, in Santander, Spain.

The goal of this Summer School was to present some of the recent advances on Smale's 17th Problem: "Can a zero of n complex polynomial equations in n unknowns be found approximately, on the average, in polynomial time with a uniform algorithm?"

These papers cover several aspects of this problem: from numerical to symbolic methods in polynomial equation solving, computational complexity aspects (both worse and average cases and both upper and lower complexity bounds) as well as aspects of the underlying geometry of the problem. Some of the contributions also deal with either real or multiple solutions solving.

American Mathematical Society
www.ams.org

Real Sociedad Matemática Española
www.rsme.es

ISBN 978-0-8218-9150-6

9 780821 891506

CONM/604

AMS/RSME