



HAL
open science

Algebraic data structures and approximations for efficient geometric modeling

Guillaume Moroz

► **To cite this version:**

Guillaume Moroz. Algebraic data structures and approximations for efficient geometric modeling. Computer Science [cs]. Université de Lorraine, 2024. tel-04910802

HAL Id: tel-04910802

<https://hal.science/tel-04910802v1>

Submitted on 24 Jan 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Algebraic data structures and approximations for efficient geometric modeling

HABILITATION THESIS

Université de Lorraine
École doctorale IAEM
département informatique

GUILLAUME MOROZ

Defended on December 12, 2024

Committee

Reading and defense

Jean Cardinal
Mioara Joldes
Bruno Salvy

Université libre de Bruxelles
Laboratoire d'analyse et d'architecture des systèmes (LAAS) de Toulouse
Laboratoire de l'Informatique du Parallélisme (LIP) de Lyon

Defense

Stef Graillat
Marine Minier
Paul Zimmermann

Sorbonne Université
Université de Lorraine
Centre Inria de l'Université de Lorraine

Contents

1	Introduction	3
1.1	System of equations	3
1.1.1	Dimension reduction	3
1.1.2	Reduction of redundant computations with multipoint evaluation	5
1.1.3	Reduction of precision with certified approximations	6
1.2	Robotic and geometric applications	7
1.3	List of Masters supervised and theses co-supervised	8
1.4	Participation in thesis juries	9
1.5	Activities of valorization and transfer	10
2	Reduction of dimension	12
2.1	Introduction	12
2.2	Symbolic elimination theory	13
2.2.1	Application to mechanism design	13
2.2.2	Application to curve topology	16
2.3	Continuation	17
2.3.1	Path tracking for curve topology	17
3	Reduction of redundant computations	25
3.1	Fast multipoint evaluation and related algorithms	25
3.2	Application to computing the distance between terrains	29
3.3	Application to isosurface triangulation with the marching cube and subdivision algorithms	30
3.3.1	Grid with Chebyshev nodes	31
3.3.2	Subdivision with sparse tensors datastructure	32
3.3.3	Results and impact	37
4	Reduction of precision	38
4.1	Preliminary tools	38
4.1.1	Fast algorithms with finite absolute precision	38
4.1.2	Condition number	42
4.1.3	Newton polygon	43
4.2	Application to approximate evaluation and root finding	43

4.3	Results and impact	46
5	Research project	48
5.1	Short term: Algebraic structures for geometric problems	48
5.1.1	L_1 -distance between two terrains	48
5.1.2	Rigid embeddings	49
5.2	Medium term: Projection of analytic varieties	50
5.2.1	Robotic applications	51
5.2.2	Applications in control theory	51
5.3	Long term: Piecewise polynomial and rational functions	52
5.3.1	Piecewise polynomial approximations	52
5.3.2	Efficient algorithms for piecewise functions	53
5.4	Major fundamental problem: fast multipoint evaluation in the Real RAM model	53

Chapter 1

Introduction

My research activity focuses mainly on solving geometric problems from various applications, using algorithmic tools from computer algebra and algorithmic geometry. I'm particularly interested in the real algebraic structures underlying geometric problems. The particularity of my work is that although my training began in the field of computer algebra, I have also studied problems in the fields of robotics, geometry and, more recently, control theory and certified numerical computation. I have published my results in the leading conferences in each of these fields. This general knowledge has also enabled me to make unexpected contributions to fundamental algorithmic tools such as polynomial evaluation, a result which has been published in the prestigious FOCS (Foundation of Computer Science) conference. My main results, their areas of application, and the big ideas behind them are summarized in Table 1.1.

1.1 System of equations

To solve different geometric problems modeled by systems of equations, I've used various algebraic techniques that can be classified into three broad categories: dimension reduction, reduction of redundant operations, and reduction of unnecessary precision.

1.1.1 Dimension reduction

Given a system S of k equations in n variables, the solutions generically form a variety V of dimension $n - k$. To describe this variety, we can reduce the search dimension. Two general approaches are worth mentioning: *elimination of variables*, which consists in reducing the dimension of the ambient space, and *continuation* methods, which consists in reducing the dimension of the search space.

Applications		General underlying ideas		
		Dimension reduction	Multipoint evaluation	Certified approximation
Robotics	Mechanism Design [75, 94, 104]	✓		
Rigid geometric structures	Embeddings of rigid graphs [44]	✓		
	Distance between triangulations [102]		✓	
	Rigid voxels transformations [115]	✓		
	Repulsive point distribution [37]		✓	✓
Control theory	Stabilizer design [18]	✓		✓
Curve drawing	Curve topology (a) [66]	✓		
	Curve topology (b) [68, 85]	✓		✓
	Fast curve drawing [56]		✓	✓
Fundamental tools				
Solving systems of bivariate equations [20]		✓		
Resultant [105]		✓		✓
Evaluating and solving univariate equations [64, 98]			✓	✓

Table 1.1: Main advances on geometric problems and the fundamental tools.

Variable elimination

The principle here is to reduce the ambient dimension by creating an equivalent system of equations and inequalities with fewer variables. This is possible as long as the equations under consideration are polynomial equations. Algebraic tools for eliminating variables in this case include *Gröbner bases* [36, Chapter 3 §6] or *resultants* [36, Chapter 3 §1]. I developed software based on these variable elimination techniques to help robotics researchers design mechanisms. Subsequently, together with Éric Schost (University of Waterloo), we proposed an algorithm that significantly accelerates the computation of the resultant. In the modular case where powers greater than a given integer k are neglected, we designed a quasi-linear algorithm in the size of the input to compute the resultant [105].

Continuation

The principle of continuation is an approach that consists of moving on the variety V solution of the system of equations S , starting from a solution point on V . This approach makes it possible to explore a space of lower dimension than the ambient space. One of its advantages is that it can also be used for non-algebraic equations involving trigonometric or exponential functions, for example. Although this approach was originally intended to describe smooth varieties, I have used it to describe singular curves [68, 85]. Up to our knowledge, the algorithms we have developed with my colleagues and students are the first to rigorously describe analytic curves with singularities.

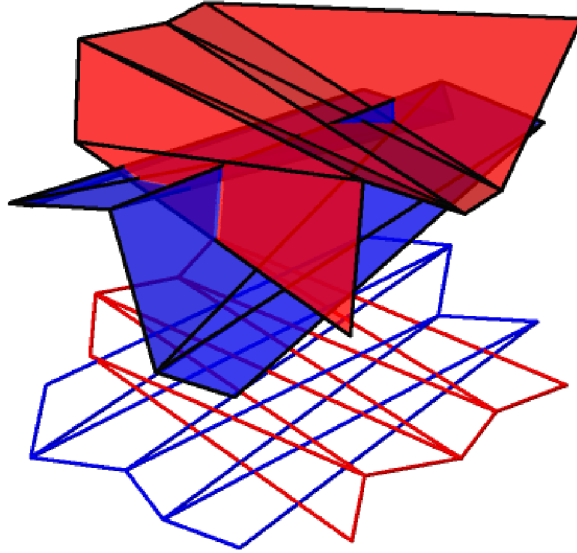


Figure 1.1: Calculating the distance between two triangulations.

Bivariate systems

Given a system of two equations $f(x, y) = g(x, y) = 0$ with f and g polynomials of degree d , a fundamental question is the computational time required to isolate the solutions of this system. Together with Yacine Bouzidi, Sylvain Lazard, Marc Pouget, Fabrice Rouillier and Michael Sagraloff (Hochschule Landshut), we designed the algorithm with the best state-of-the-art complexity [20], applying variable elimination techniques on a carefully chosen variation of the system of equations given as input. The works related to this result, not detailed in this document, have been published in a conference and a journal [19, 20].

1.1.2 Reduction of redundant computations with multipoint evaluation

Another problem underlying certain geometric problems is the evaluation of a polynomial of degree d at d points. For univariate polynomials, this *multipoint evaluation* can be performed with an almost linear number of arithmetic operations in d [48, Chapter 10]. In particular, the use of this tool enabled Boris Aronov (New York University) and I to calculate the distance between two triangulations in quasi-linear time in triangulation size [101, 102] (Figure 1.1), whereas the best known bound for this problem was quadratic.

Curve drawing is another important problem where the use of multipoint evaluation leads to faster computations. On this subject, I notably co-supervised with Marc Pouget the PhD student Nuwan Herath Mudiyansele [106], and

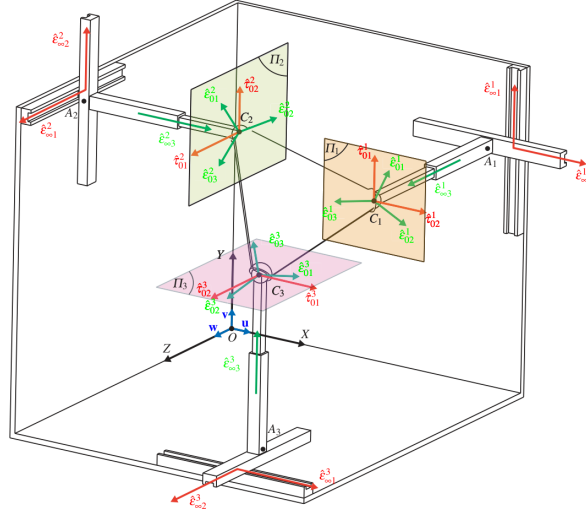


Figure 1.2: Robot 3-PPPS, with 3 legs, where each leg is a chain of prismatic and spherical joints. The underlined letters show which joints are actuated by motors.

published several results [56, 99] detailed in Section 3.3.

1.1.3 Reduction of precision with certified approximations

Finally, in many geometric application problems, a margin of error is tolerated in the result. In such cases, this margin of error can be exploited to design a faster algorithm. For example, with Laurent Decreasefond (Telecom Paris), we have accelerated the generation of a set of points in the plane with a repulsive law between them, which can be used to simulate a network of telephone antennas. This idea of exploiting error margins to speed up calculations also underlies the results on multipoint evaluation.

From a numerical point of view, while the fast multipoint evaluation algorithm can be performed with an almost linear number of arithmetic operations, each operation requires manipulating numbers with $O(d)$ decimal places, which in practice makes it a quadratic algorithm. In a recent body of work, I proposed a numerical multipoint evaluation algorithm that is quasi-linear in the number of machine operations, and efficient in practice for large degrees. This work required a careful reduction of the required precision, while keeping the algorithm numerically stable. The results have been published in FOCS and ISSAC conferences [64, 98].

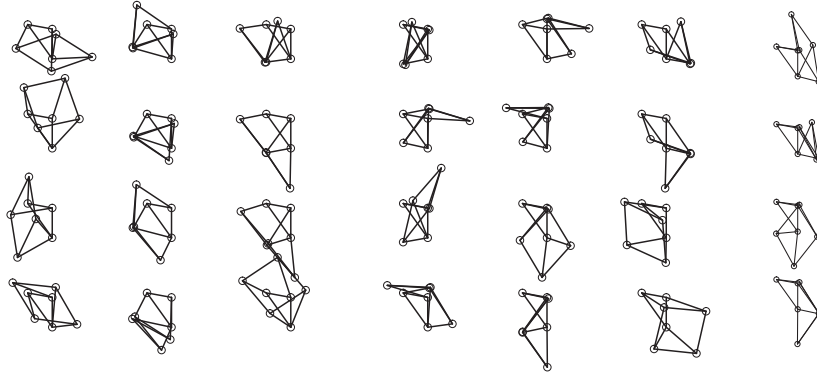


Figure 1.3: 28 embeddings for an 11-bar mechanism. The 28 others are obtained by symmetry.

1.2 Robotic and geometric applications

The main applications I've worked on can be grouped into three categories. More recently, I also investigated applications in control theory, that are detailed in Section 5.2.2 of my scientific project.

Robotics. Many of the applications I've worked on come from the design of robotic mechanisms, as illustrated in Figure 1.2 for example [25, 27–33, 75–78, 94, 103, 104]. This work was structured around the development of the Siropa software [30].

Algebraic and analytical curve drawing. Plotting curves was another problem that kept me busy [56, 66, 68, 85, 99]. My aim was to draw singular curves correctly and efficiently. In particular, I was interested in two cases: curves defined as smooth curve projections, and the drawing of implicit curves using certified numerical algorithms.

Rigid geometric structures. Geometric problems with rigid structures form another category for which I have had significant results [37, 44, 101, 102, 115]. For example, given a mechanism with bars of given lengths, how many embeddings does it admit in the plane [44]? With Ioannis Emiris (University of Athens), we showed that a rigid 11-bar mechanism can have at most 56 embeddings, and we exhibited an example of a mechanism that had exactly 56 embeddings. Another example in algorithmic geometry is the efficient calculation of distance between triangulated terrains. The algebraic structures underlying these problems have given us ideas for tackling new problems, which I detail in Section 5.1 of my scientific project.

1.3 List of Masters supervised and theses co-supervised

Master's degree:

- 2022. Léo Kulinski, MPRI (Master Parisien de Recherche en Informatique) internship, co-supervised with Marc Pouget, "Meshing Singular Surfaces".
This internship explored the use of the Dual Contouring method, to trace singular surfaces with better rendering than current software.
- 2021. Mohamed Sidi Ali Cherif, Master 2 internship at ENS Lyon, co-supervised with Marc Pouget. "Numeric and Topology Certified Meshing of Singular Surfaces."
This internship generalized a subdivision algorithm for plotting smooth curves to the case of singular curves in structured cases.
- 2014. Olive Chakraborty, Master's internship co-supervised with Marc Pouget. "Numerical algorithms for certified topological and geometrical description of singular curves".
This internship explored numerical subdivision methods for isolating curve singularities obtained by projections of smooth curves.
- 2013. Judit Recknagel, Master 1 internship, co-supervised with Marc Pouget. "Topology of planar singular curves resulting from two trivariate polynomials".
<https://inria.hal.science/hal-00927768>
The projection of a spatial curve in a plane is a curve generically containing singular points. This work studies the algebraic structure of this projection in order to certify the position of its singularities by means of numerical algorithms.
- 2013. Oswald Hounkonnou, Master's internship, co-supervised with Xavier Goaoc. "Study of a geometric conjecture using computer algebra systems".
This internship provided an opportunity to study a geometric conjecture in special cases, using computer tools derived from computer algebra.

PhD Theses :

- Nov. 2019 – June 2023. Nuwan Herath Mudiyansele [106], co-supervised at 50% with Marc Pouget. "Fast high-resolution drawing of algebraic curves and surfaces".
<https://www.theses.fr/2023LORR0099>
This PhD thesis shows how multipoint evaluation methods can be used to speed up the drawing of curves and surfaces that are solutions of polynomial equations.

- Oct. 2017 – May 2021. George Krait [83], co-supervised at 50%, with Marc Pouget and Sylvain Lazard. "Isolating the Singularities of the Plane Projection of Generic Space Curves and Applications in Robotics".
<https://www.theses.fr/2021LORR0092>
This PhD thesis presents an algorithm for certified plotting of analytic singular curves. In particular, this work deals with curves obtained by projecting smooth curves, which appear naturally in robotics and control theory applications.
- Dec. 2014 – Jan. 2020. Sény Diatta [40], co-supervised at 30% with Marie-Françoise Roy and Daouda Niang Diatta. "Topology of plane algebraic curves and projection of real analytic surfaces."
<https://hal.science/tel-03101708>
Sény Diatta did his PhD thesis partly in Ziguinchor, Senegal, and partly in Nancy, thanks to an Eiffel scholarship. His work with me focused on isolating the singularities of an analytic surface obtained as a projection of a smooth surface. On the part I supervised, the result of his thesis is a new symbolic-numerical algorithm for describing singular surfaces that appear in robotic applications.
- Oct. 2013 – July 2016. Ranjan Jha [74], co-supervised at 30% with Damien Chablat and Fabrice Rouillier. "Contributions to the performance analysis of parallel robots".
<https://theses.hal.science/tel-01524856/>
This PhD thesis provided an opportunity to collaborate with robotics researchers in Nantes. In particular, Ranjan used Siropa software to analyze new robotic mechanisms. Siropa is a robot design aid based on computer algebra tools that I began developing during my post-doctorate in Nantes in 2009.

1.4 Participation in thesis juries

- 2023. Reviewer. Catherine St-Pierre, supervised by Éric Schost. University of Waterloo.
- 2021. Reviewer. Nestor Djintelbe, supervised by Michel Coste, Seydou Moussa and Marie Salomon Sambou. Assane Seck University of Ziguinchor.
- 2019. Jury member. Romain Benoit, supervised by Nicolas Delanoue. University of Angers.
- 2018. Jury member. Guillaume Rance, supervised by Alban Quadrat. Paris-Saclay University.
- 2014. Reviewer. Esmaeil Mehrabi, supervised by Éric Schost. University of Western Ontario.

1.5 Activities of valorization and transfer

Dissemination to non-scientific audiences:

- 2022. *Loria summer festival*. "De l'astronomie grecque au calcul rapide".
https://ultv.univ-lorraine.fr/video/12300-manifestation-scm/?start=4189&is_iframe=true
- Since 2011. Design of subjects for the Nancy-Metz Olympiads.

Presentation of the research career to magister students:

- 2021. ENS Paris Saclay.
- 2020. ENS Paris Saclay.
- 2020. ENS Rennes.

Software transfer and development:

- The main software transfers I've contributed to are for the commercial computer algebra software Maple. The libraries included in Maple are listed below.

– *pwpoly*

url: <https://gitlab.inria.fr/gamble/hefroots>

APP: IDDN.FR.001.360018.000.S.P.2021.000.10000

number of lines: ≈ 40000

associated publication: [64]

This software was written by Rémi Imbach under my direction, based on a new algorithm that I designed. It is included in Maple 2024.

– *hefroots*

url: <https://gitlab.inria.fr/gamble/hefroots>

APP: IDDN.FR.001.360018.000.S.P.2021.000.10000

number of lines: ≈ 40000

associated publication: [98]

This software was written with Rémi Imbach, based on an initial prototype that I had developed in C. It is included in Maple 2023.

– *earoots*

url: <https://gitlab.inria.fr/gmoroz/earoots>

APP: IDDN.FR.001.190035.000.S.P.2022.000.10000

number of lines: ≈ 1500

This software written in C uses the Ehrlich-Aberth method to find complex roots of polynomials in double precision. The code was written to take advantage of auto-vectorization. This software is included in Maple 2023.

– *Rootfinding[Parametric]*
url: <https://www.maplesoft.com/support/help/Maple/view.aspx?path=RootFinding/Parametric>
number of lines: ≈ 3500
Developed with F. Rouillier and Maplesoft employees, this program allows for the certified resolution of parametrized polynomial systems. It has been included in Maple since 2008.

- I also developed open source software, notably.

– *voxelize*
url: <https://gitlab.inria.fr/gmoro/voxelize>
number of lines: ≈ 7000
associated publication: [99]
This software developed in C++ has allowed me to explore a new technique combining subdivision algorithms and multipoint evaluation. It serves as a support for my research and that of some colleagues.

– *Siropa Library*
url: <https://gitlab.inria.fr/gmoro/siropa>
documentation: <https://members.loria.fr/Guillaume.Moroz/Siropa/files/siropa-mpl.html>
APP: IDDN.FR.001.140015.000.S.P.2017.000.20600
number of lines: ≈ 10000
associated publications: [30, 75]
I developed this software during my postdoctoral position in Nantes in 2009 for researcher colleagues in robotics who continue to use it regularly.

- My other software are available at <https://gitlab.inria.fr/gmoro>

Chapter 2

Reduction of dimension

2.1 Introduction

Once a geometric problem is modeled by an algebraic system of polynomial equations, it is now possible to use algebraic structures to solve it efficiently. Consider the following system S of polynomial equations:

$$S \left\{ \begin{array}{l} f_1(x_1, \dots, x_n) = 0 \\ \vdots \\ f_k(x_1, \dots, x_n) = 0 \end{array} \right.$$

Assume that the solutions of this system form a curve in dimension n , denoted by V . One of the main ideas to solve this system more efficiently is to reduce its dimension. There are two main approaches to reduce the dimension that I investigated while solving geometric problems. The first one consists in eliminating variables to compute the projection of the curve on a plane using algebraic tools to eliminate $n - 2$ variables. The second approach is a continuation approach, and it is based on tracking paths from one or several initial points.

Elimination

Assume that V is the curve solution of a system of polynomial equations. Eliminating $n - 2$ variables from a system S consists in computing a new system S' of polynomial equations in the 2 remaining variables, such that the solutions of S are solutions of S' . Moreover, we require that the solution set of S' is the Zariski closure of the projection on the plane spanned by the 2 variables. The Zariski closure of a set E is the smallest set F containing E such that F is the solution of a system of polynomial equations. As we will see in Section 2.2, eliminating variables can be done with several classical algebraic tools such as Gröbner bases or resultants.

Continuation

Path tracking. For a system defining a curve, another approach to reduce the dimension of the search space consists in using path tracking. The idea is to start from a point and to use a continuation algorithm to compute the next point on the curve. When the curve is smooth, this approach can also compute an enclosing tube around the curve. Moreover, this approach can be generalized for solution set V of arbitrary dimension [14–16].

Numerical optimization. Some geometrical problems can be reformulated as optimization problems. In this case, the goal is to find the maximum value of a function in n variables. A naive approach would consist in exploring the n -dimensional space to find this value. Numerical optimization techniques are a way to reduce the dimension, by starting with one or several points, and then computing the next points toward the maximal value. This can also be seen as a continuation approach to reduce the dimension of the search space. In particular, we used those derivative-free optimization techniques to find the maximal number of real embeddings of a rigid graph [44].

2.2 Symbolic elimination theory

From a geometric point of view, eliminating variables in a system S of polynomial equations can be viewed as projecting the solutions of S . One of the main challenges is to handle the singularities that appear in the projection. Moreover, as shown in Figures 2.1 and 2.2, the solutions of the system obtained after eliminating variables can sometimes add spurious singularities that are not in the geometric projection.

2.2.1 Application to mechanism design

A fundamental problem in the design of mechanisms is the calculation of their workspace, i.e., the set of poses that the manipulator’s end-effector can reach. Various techniques can be used to calculate this, based on geometric, discretization and algebraic methods. One of the advantages of the algebraic approach is that we can use elimination methods to efficiently reduce the dimension. This led to the creation of the Siropa software library [30], which combines algebraic and mechanism design tools to help robotics researchers design mechanisms using efficient algebraic tools. The main modules of the library are illustrated in Figure 2.3. The library is written in Maple. In particular, it uses Gröbner bases from Maple for variable elimination.

Workspace and cusp points for robotic mechanisms

I have developed a new method for efficiently and automatically calculating the cusp points of a parallel robot [94, 103, 104]. Following this result, and to facilitate the dissemination of computational methods in robotics, I developed

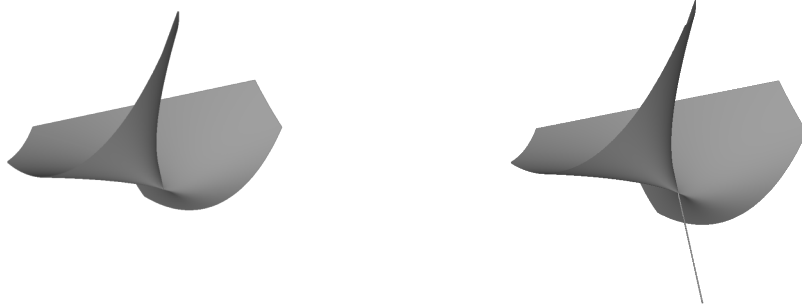


Figure 2.1: Projection of the Whitney Umbrella from \mathbb{R}^4 to \mathbb{R}^3 on the left, and its Zariski closure on the right.

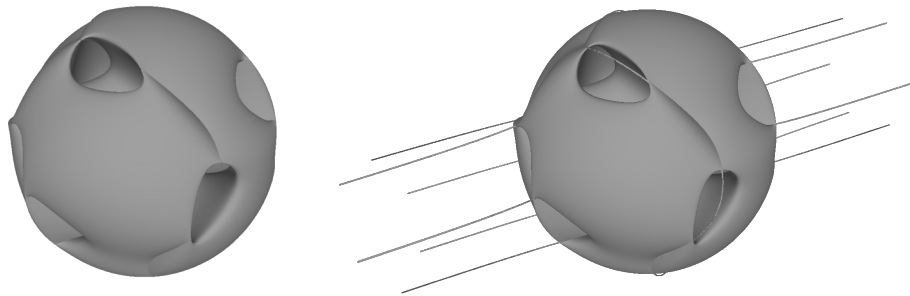


Figure 2.2: Projection of an algebraic surface from \mathbb{R}^4 to \mathbb{R}^3 on the left, and its Zariski closure on the right.

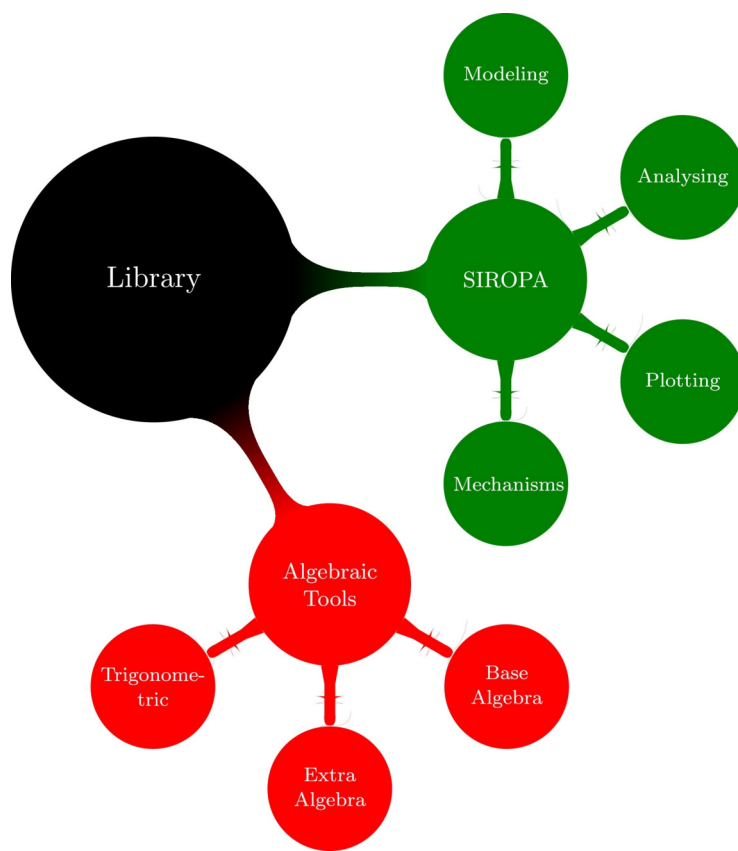


Figure 2.3: The Siropa library, its two sub-modules (Siropa and AlgebraicTools) and their main sub-modules

the Siropa library in the Maple language, and wrote its documentation. The user interface and certain algorithmic choices were discussed with Damien Chablat, Philippe Wenger and Fabrice Rouillier. We then used this library to help us analyze various mechanisms. I also co-supervised Ranjan Jha’s thesis on the analysis of parallel robots.

One of the first difficulties was the interdisciplinary aspect of the project. In order to create this library, I had to acquire skills in mechanism design to combine them with my skills in symbolic computation. In particular, fundamental notions such as ”singularity” do not have the same meaning for an algebraic variety as for a robot. This is the only library that makes algorithms derived from computer algebra available to robotics researchers.

A second difficulty and originality of the Siropa software was the development of algorithms for automatic modeling of mechanisms, so that a robotics researcher could use the library without being an expert in computer algebra.

The work on cusp points enabled me to calculate the cusp points of the 3-RPR mechanism, followed by other parallel mechanisms [94, 103, 104]. This library also enabled my co-authors to analyze the workspace associated with various parallel robots [25, 27–33, 75–78, 94, 103, 104].

Siropa library

Documentation for the Siropa library is available online [30], and a more detailed description of its use has been published in a robotics journal [75]. Siropa has also been registered with the APP. This library has been used directly by robotics researchers, first by my co-authors Damien Chablat and Philippe Wenger (Laboratoire des Sciences du Numérique de Nantes), and Ranjan Jha (Central Scientific Instruments Organisation in India). The library is now being used by other researchers without my involvement, such as Guillaume Michel (doctor at Nantes University Hospital) [97].

2.2.2 Application to curve topology

Consider the case of a robot with 2 degrees of freedom. The problem of computing the boundary of its workspace or jointspace can be reformulated as describing the apparent contour of a surface. In the case of a surface of \mathbb{R}^3 , its apparent contour is the projection on the horizontal plane of the points of the surface where the tangent plane contains a vertical line (see Figure 2.5).

One of the main issues with this curve is that even if the surface is smooth, it contains self intersections and other kinds of singular points. From transversality theory, it is known that nodes and cusp points can appear in the apparent contours of generic surfaces [39, §4.7].

More specifically, let $f(x, y, z) = 0$ be a polynomial equation defining a smooth surface in $3D$. The apparent contour is the projection on the (x, y) -plane of the curve defined by $f(x, y, z) = 0$ and $\frac{\partial f}{\partial z}(x, y, z) = 0$. Using tools from elimination theory, such as the resultant for example, we can compute a polynomial $g(x, y)$ that vanishes on the apparent contour. Computing the

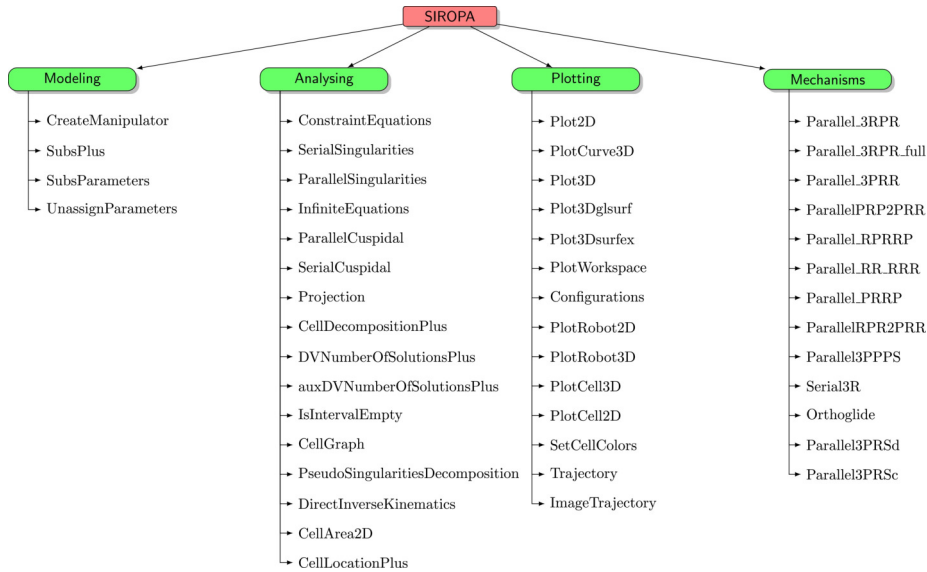


Figure 2.4: Main modules of the Siropa library.

topology of the apparent contour requires taking into account the singular points of the projection.

To handle this problem, we worked on a dedicated symbolic elimination algorithm that also computes the equations defining the singular points of the projected curve using subresultant theory. One of the main advantage of our new system of equations defining the singularities is that it is suitable for numerical solvers. This allows us to design an algorithm optimized for describing the apparent contour of a generic surface. In particular, our approach was faster than state-of-the-art approaches that didn't take into account the specific algebraic structure of the singularities induced by the subresultant theory [66, 118].

2.3 Continuation

2.3.1 Path tracking for curve topology

One of the drawbacks of our approach to compute the singularities of an apparent contour with subresultant theory is that it is restricted to surfaces in \mathbb{R}^3 . The subresultant theory is well understood for the elimination of one variable. However, its generalization to more than one variable is not straightforward. Thus, it is not easy to generalize the subresultant approach to a higher number of variables.

Instead, we used path tracking to reduce the dimension for the problem of describing the topology of the apparent contour. The tracking allowed us to

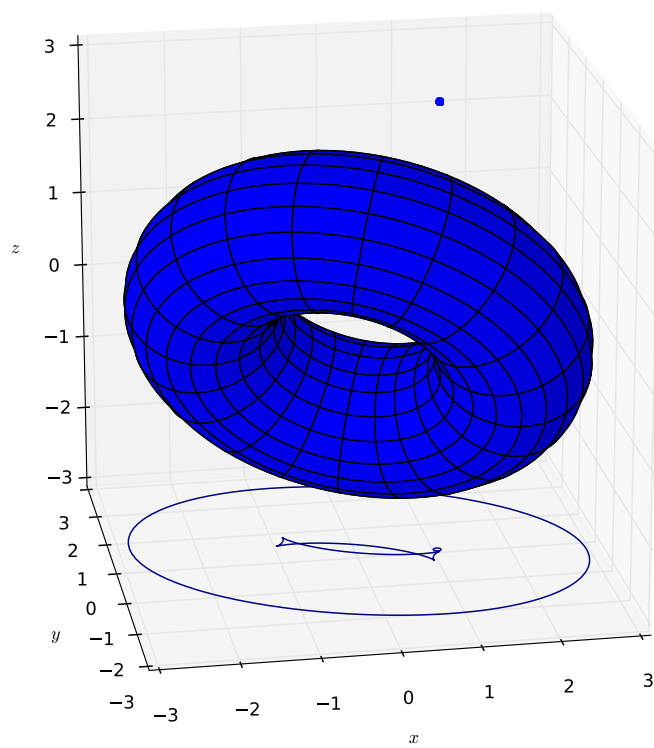


Figure 2.5: Apparent contour of a torus.

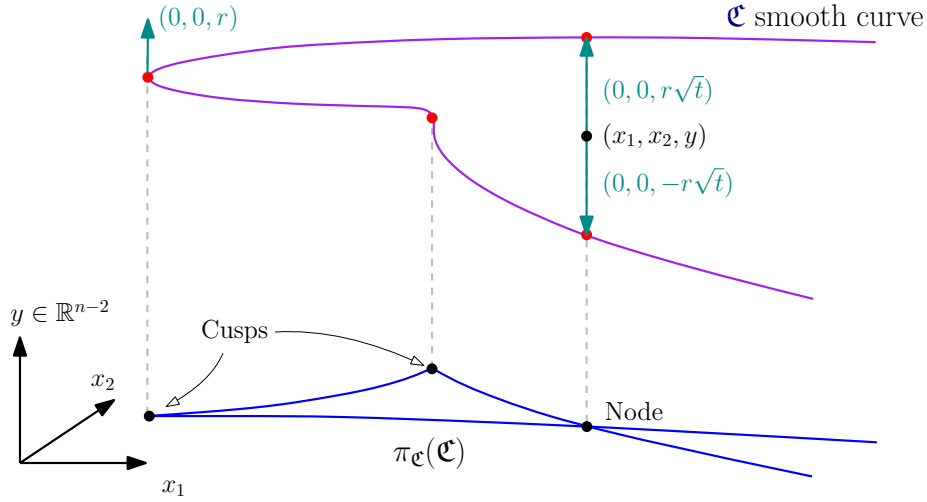


Figure 2.6: Projection of a smooth curve, and its singularities.

enclose the apparent contour. For the singularities, we designed a specific system of equations, so-called *ball system*, that allowed us to compute the singularities of the apparent contour without computing the equation defining it in the plane.

Effective implicit function theorem

When a curve defined by equations is smooth, the implicit function theorem guarantees that it is possible to locally parametrize the curve with one variable. That is, there exists a box containing the curve such that the curve can be parametrized by a variable in the box. Snyder [126] introduced an effective criterion based on interval arithmetic to decide if the curve is parametrizable in a given box. Given a box B defined by a product of intervals, and the function map $F(x_1, \dots, x_n) = (f_1, \dots, f_{n-r})$, let $\square J_{k_1, \dots, k_r}(B)$ be the Jacobian matrix of size $k \times k$ where the entry (i, j) is the interval $\frac{\partial f_i}{\partial x_{\ell_j}}(B)$ for $\ell_j \notin \{k_1, \dots, k_r\}$.

Theorem 1 (Interval implicit function theorem [126]). *Given a system S of equations of the form $f_1 = \dots = f_{n-r} = 0$, if*

$$0 \notin \det(\square J_{k_1, \dots, k_r}(B)),$$

then the solution set of S is parametrizable by x_{k_1}, \dots, x_{k_r} in B .

Combined with a path tracking algorithm, this allows us to enclose a curve in a union of boxes, as illustrated in Figure 2.7.

Singularities of the projection

Unlike previous state-of-the-art results, our approach computes the singularity of the apparent contour of a surface, without computing the bivariate equa-

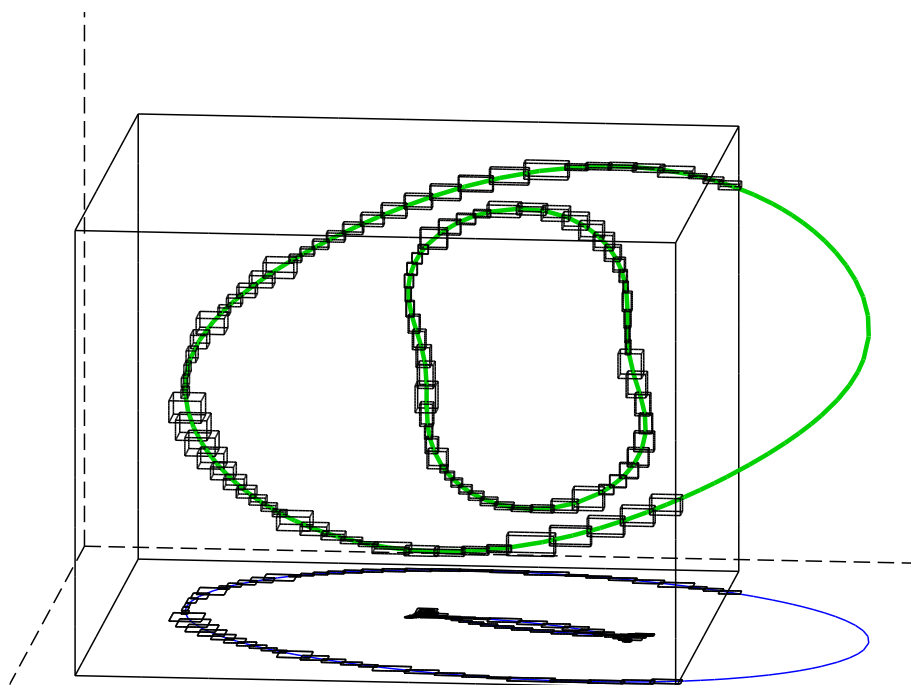


Figure 2.7: Boxes enclosing a smooth curve, each of them satisfying the interval implicit function criterion.

tion defining it. By avoiding the computation of the bivariate equation defining the apparent contour, our approach is both more efficient in practice and more adapted to surfaces embedded in higher dimensions. Moreover, it is not restricted to polynomial system of equations, and our approach can compute the singularities of apparent contour of surfaces defined by more general implicit smooth functions such as cosine and sine (see Figure 2.9). One of the difficulties is to design a system of equations vanishing on the singularities of the projected curve, without computing an explicit formula for the projection.

More specifically, we consider a curve V defined by a system S of $n - 1$ equalities of the form $f_i(x_1, \dots, x_n) = 0$ where the f_i are smooth functions. We require that V satisfies some assumptions that we detail below. Those assumptions have a geometric interpretation and are illustrated in Figure 2.8. Note that they are satisfied generically [85, §7]. In the following, we denote by J_S the Jacobian matrix associated to the functions f_i , and π is the projection map from V to the dimension-2 space generated by x_1 and x_2 .

- \mathcal{A}_1 – For all $q \in V$, $\text{rank}(J_S(q)) = n - 1$. In particular, V is a smooth curve.
- \mathcal{A}_2 – The set of points of V with a vertical tangent is discrete and does not intersect the boundary of B .
- \mathcal{A}_3 – For all points $p \in \pi(V)$, the pre-image of p under π consists of at most two points in \overline{B} counted with multiplicities.
- \mathcal{A}_4 – The set of points of V where π is not injective is discrete and does not intersect the boundary of B .
- \mathcal{A}_5 – The singular points of $\pi(V)$ are only nodes.

The system we use for computing the singularities is constructed using the following operators that send an analytic function to an analytic function.

Definition 1. Let x_1, x_2, t be variables in \mathbb{R} and y, r in \mathbb{R}^{n-2} . For a smooth function $f : \overline{B} \subset \mathbb{R}^n \rightarrow \mathbb{R}$, we define the functions $S \cdot f$ and $D \cdot f$ from \mathbb{R}^{2n-1} to \mathbb{R} .

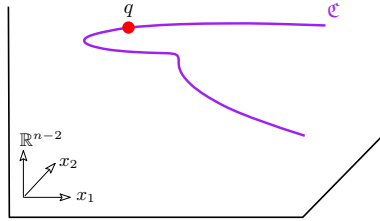
$$S \cdot f(x_1, x_2, y, r, t) = \begin{cases} \frac{1}{2}[f(x_1, x_2, y + r\sqrt{t}) + f(x_1, x_2, y - r\sqrt{t})], & \text{for } t > 0 \\ f(x_1, x_2, y), & \text{for } t = 0 \end{cases}$$

and

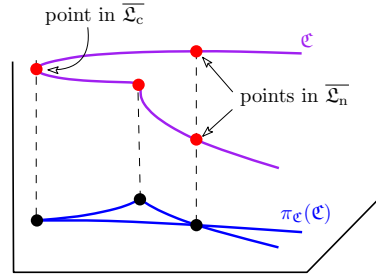
$$D \cdot f(x_1, x_2, y, r, t) = \begin{cases} \frac{1}{2\sqrt{t}}[f(x_1, x_2, y + r\sqrt{t}) - f(x_1, x_2, y - r\sqrt{t})], & \text{for } t > 0 \\ \nabla f(x_1, x_2, y) \cdot (0, 0, r), & \text{for } t = 0. \end{cases}$$

This allows us to define a system of equations that vanishes on the singularities of the projection of the considered curve. Moreover, under our assumptions, all the solutions of this system are regular.

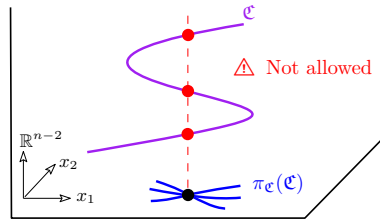
Theorem 2 ([85]). Consider $P = (P_1, \dots, P_{n-1}) \in C^\infty(\mathbb{R}^n, \mathbb{R}^{n-1})$ that satisfies Assumptions $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$ and \mathcal{A}_4 . Then, $X = (x_1, x_2, y, r, t) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}^{n-2} \times$



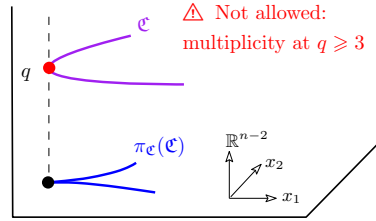
(a) Assumption \mathcal{A}_1 : $\bar{\mathcal{C}}$ is a smooth; the rank of the Jacobian of P at q is full.



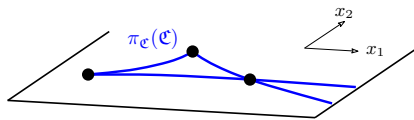
(b) Assumptions \mathcal{A}_2 and \mathcal{A}_4 : the sets $\bar{\Sigma}_c$ and $\bar{\Sigma}_n$ are finite and do not intersect the boundary of B .



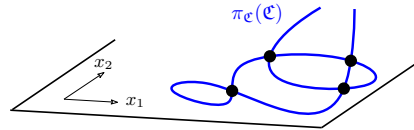
(c) Assumption \mathcal{A}_3 : No three points (counted with multiplicity) have the same projection.



(d) Assumption \mathcal{A}_3 : No three points (counted with multiplicity) have the same projection.

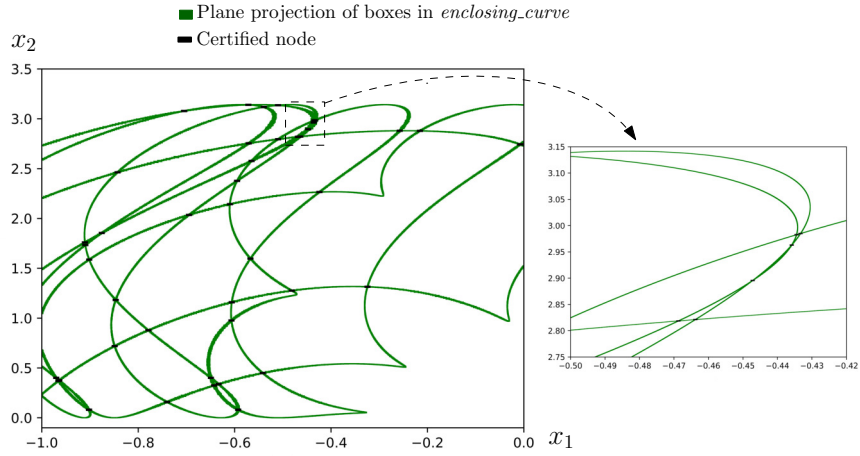


(e) Assumption \mathcal{A}_5^- : points in $\pi_{\mathcal{C}}(\mathcal{C})$ are either smooth, nodes or ordinary cusps.



(f) Assumption \mathcal{A}_5 : points in $\pi_{\mathcal{C}}(\mathcal{C})$ are only smooth or nodes.

Figure 2.8: Illustration of the assumptions.



$$\begin{cases} 0 &= x_1 + 2 \sin(x_1) - \cos(x_4) - (3 \cos(x_3) - \cos(2.8571x_3)) \\ 0 &= x_2 + 0.2 \cos(x_2) + (3 \sin(x_3) - \sin(2.8571x_3)) + \sin(x_4) \\ 0 &= x_4^2 - \sin(x_2) \end{cases}$$

Figure 2.9: Projection of a smooth curve defined by 3 trigonometric equations in 4 variables.

$\mathbb{R}^{n-2} \times \mathbb{R}$ is a solution of the Ball system

$$\text{Ball}(P) = \begin{cases} S \cdot P_1(X) = \dots = S \cdot P_{n-1}(X) = 0 \\ D \cdot P_1(X) = \dots = D \cdot P_{n-1}(X) = 0 \\ \|r\|^2 - 1 = 0 \end{cases} \quad (2.1)$$

if and only if (x_1, x_2) is a singular point of the projection of the curve. Moreover, if Assumption \mathcal{A}_5 is satisfied, then X is a regular solution of $\text{Ball}(P)$.

This theorem allowed us to use numerical methods to compute the singularities of curves defined with trigonometric equations, as shown in Figure 2.9.

Results and impact

In the case of analytic curves, our approach is the only one to describe their projections (Figure 2.9), and for algebraic curves, our approach allows us to gain an order of magnitude in computation time compared to variable elimination methods. This has allowed me to get closer to the community of researchers working notably on contractors in interval arithmetic [26, 100].

In the long term, the idea is to generalize our approach to the case of surfaces. A first step in this direction was made during Sény Diatta's thesis [40, 41], and other researchers like Mathijs Wintraecken are also very interested in working

on these problems, which I discuss in more detail in my research program in Section [5.2](#)

Our work has been published in conferences and journals bridging the communities of computer algebra [67] and interval arithmetic [68, 85]. I also co-supervised George Krait's thesis, who developed software for plotting singular curves distributed on GitHub [84].

Chapter 3

Reduction of redundant computations

Some algebraic computations involve redundant operations, notably when some intermediate values are algebraically dependent. In particular, the evaluation of a polynomial of degree d at one point can be done with a linear number of arithmetic operations, e.g. using the Horner algorithm. If we evaluate a polynomial of degree d at d points using the Horner algorithm at each point, it will require a quadratic number of arithmetic operations. As we will see in this chapter, evaluating a polynomial at multiple points is a fundamental tool in some geometric algorithms, such as the computation of the distance between two 3D triangulations (Section 3.2). First we briefly recall in Section 3.1 the main ideas for evaluating a polynomial of degree d at d points with a quasi-linear number of arithmetic operations. Then in Section 3.2 we show how this result can be combined with other state-of-the-art results in computational geometry to obtain a new and quasi-optimal algorithm for computing the distance between two 3D triangulated irregular networks. Finally, in Section 3.3, we show how multipoint evaluation approaches can also be used to speed up classical approaches based on marching cube or subdivision algorithms to enclose the zero set of polynomial equations.

3.1 Fast multipoint evaluation and related algorithms

Real RAM model. The real Random Access Machine model is a model that is commonly used in computational geometry. In this model, variables can store arbitrary real numbers, and algebraic operations $(+, -, \times, \div)$ can be performed in constant time, and it is possible to check the sign of a number. However, it is not possible to compute its integer part in constant time, since it would allow solving in polynomial time any problem in PSPACE [122]. This model

was introduced by Shamos in the late 70s [116, 124]. It is similar to the Blum-Shub-Smale machine, that is more commonly used in computer algebra [13]. Remark that in divide-and-conquer algorithms for example, the integer part function is commonly used to get integer indices, which is not directly available in the classical BSS or real RAM machines. A more recent careful formalisation extends the real RAM model to support such indices manipulations [45].

Fast multiplication using Karatsuba algorithm. One standard operation in computer algebra is the multiplication of univariate polynomials. More specifically, let $f(x) = f_0 + f_1x$ and $g(x) = g_0 + g_1x$, where x is a symbolic variable. A naive algorithm to compute the multiplication

$$f(x)g(x) = f_0g_0 + (f_1g_0 + f_0g_1)x + f_1g_1x^2$$

requires 4 multiplications. On the other hand, using the distributivity of the multiplication, we can remark that

$$f(1)g(1) - f_0g_0 - f_1g_1 = f_1g_0 + f_0g_1.$$

Thus, the coefficients of $f(x)g(x)$ can be computed with 3 multiplications instead of 4. Similarly, a naive algorithm to multiply two polynomials of degree $2^k - 1$ requires 4^k multiplications, which is quadratic in the degree. However, by splitting the polynomials in two parts of the form $f(x) = f_0(x) + x^{2^{k-1}}f_1(x)$ and $g(x) = g_0(x) + x^{2^{k-1}}g_1(x)$, we can use a recursive algorithm that computes $f(x)g(x)$ using only 3^k multiplications. This was the first subquadratic algorithm to multiply two polynomials, found in the early sixties by Karatsuba [79]. This shows also that in computer algebra naive algorithms sometimes involve redundant computations that can be avoided.

Fast multiplication using Fast Fourier Transform. In Karatsuba's algorithm, the multiplication of two polynomials has been sped-up by evaluating f and g on 0 and 1. This approach can be generalized by evaluating f and g on the roots of unity. More precisely, if f and g are polynomials of degree $d - 1$, we evaluate each of them on the roots of unity $w_k = e^{i2\pi/2d}$. Then we compute the $f(w_k)g(w_k)$, and we use interpolation to recover the $2d$ coefficients of the polynomial $f(x)g(x)$. Evaluating a polynomial on the points w_k is equivalent to the computation of a $2d$ -points Discrete Fourier Transform. A $2d$ -points DFT transforms a sequence of $2d$ complex numbers p_0, \dots, p_{2d-1} into a sequence of $2d$ complex numbers q_0, \dots, q_{2d-1} satisfying:

$$q_\ell = \sum_{k=0}^{d-1} p_k w_{d-\ell}^k$$

Using a divide and conquer approach, a DFT can be computed with a quasi-linear number of arithmetic operations [35, 50]. A quasi-linear algorithm to compute a DFT is called a Fast Fourier Transform or FFT. Moreover, interpolating a polynomial of degree $2d - 1$ from its values on the points w_k is also

equivalent to computing a $2d$ -points DFT. This leads to the following fast algorithm to multiply two algorithms using a quasi-linear number of operations: first evaluate the $f(w_k)$ and the $g(w_k)$ using FFT, then compute the values $f(w_k)g(w_k)$, and finally recover the product polynomial by interpolation using FFT again.

Fast multipoint evaluation for univariate polynomials in the real RAM model. In the real RAM model, you don't have access to roots of unity. However, using techniques based on field extension, you can still design a fast multiplication algorithm, that uses only a quasi-linear number of arithmetic operations [48, Theorem 8.23]. Then, once we have a quasi-linear algorithm to multiply two polynomials, it is possible to compute the quotient and the remainder of the division of two polynomials with a quasi-linear number of arithmetic operations [48, Theorem 9.6].

Using fast division algorithms, it is possible to design a divide-and-conquer algorithm to evaluate a polynomial f of degree $d - 1$ on d points z_0, \dots, z_{d-1} . First, for two given integers $\ell \leq u$ between 0 and $d - 1$, let $r_{\ell,u}$ be the remainder of the division of f by the polynomial $p_{\ell,u} := (x - z_\ell) \cdots (x - z_u)$. Then, we can remark that for all $\ell \leq k \leq u$ such that z_k is a root of $p_{\ell,u}$:

$$r_{\ell,\ell} = f(z_\ell) \tag{3.1}$$

$$r_{\ell,u}(z_k) = f(z_k). \tag{3.2}$$

Using these two properties, we can design a divide-and-conquer algorithm to compute $r_{k,k}$ for all $1 \leq k \leq d$ as follows. Let $m = \lfloor d/2 \rfloor$ and compute $r_{0,m-1}$ and $r_{m,d-1}$ with a quasi-linear number of arithmetic operations. Then, letting $p = \lfloor m/2 \rfloor$ and $q = \lfloor (d+m)/2 \rfloor$, we can compute $r_{0,p-1}, r_{p,m-1}, r_{m,q-1}, r_{q,d-1}$ with a quasi-linear number of arithmetic operations.

Fast multipoint evaluation for multivariate polynomials in the real RAM model. To evaluate a polynomial on a general set of points, the case of a univariate polynomial is well understood [46]. For multivariate polynomials, there are fewer results that are efficient in practice when the points are not arranged as a grid.

For multipoint evaluation in the real RAM model, the first subquadratic algorithm for multivariate polynomials is from [108]. In this case, Nüsken and Ziegler show that it is possible to evaluate a bivariate polynomial of degree d in each variable on d^2 points using only $\tilde{O}(d^{\omega(1,1,2)/2+1+\varepsilon})$ arithmetic operations for any $\varepsilon > 0$. Here, $\omega(a, b, c)$ is the exponent such that the multiplication of an $n^a \times n^b$ matrix by an $n^b \times n^c$ matrix can be done in $O(n^{\omega(a,b,c)+\varepsilon})$ arithmetic operations. Moreover, $w(1, 1, 2) = w(1, 2, 1)$ [63] and as of 2024, $\omega(1, 2, 1) < 3.250385$ [128]. In particular, it is possible to evaluate a bivariate polynomial on d^2 points in $O(d^{2.626})$ operations, which is the best current bound, up to our knowledge. More generally, this bound can be extended to evaluate on d^n points a polynomial in n variables, of degree d in each variable, using

$O(d^{(n-1)\omega(1,2,1)/2+1+\varepsilon})$ operations. For $n = 3$ or $n \geq 7$, with the current bounds on matrix multiplications, a recent result improves this bound and shows that it is possible to evaluate a polynomial on those points in $\tilde{O}(d^{1+\frac{(\omega-1)(n-2)}{(n-2)\omega+1}})$, where $\omega = \omega(1, 1, 1) < 2.371552$ [128].

Unfortunately, none of these approaches allows to evaluate a multivariate polynomial on multiple points with a quasi-linear number of operations.

Fast multipoint evaluation for bivariate polynomials over other fields.

When the underlying field is not the field of the real numbers, the elements are not necessarily ordered. In this case, handling degenerate cases where points share the same first coordinate becomes the bottleneck in the algorithm of Nüsken and Ziegler [108]. A recent result shows that for any field, even those where elements are not ordered, it is possible to evaluate one bivariate polynomial of degree d on d^2 points in $\tilde{O}(d^{(4\omega+2)/(\omega+1)})$ operations, which is in $O(d^{2.796})$ [61]. For evaluating several bivariate polynomials on a fixed set P of points, another recent interesting approach consists in precomputing a data structure associated to P . After this precomputation, the evaluation can be done with a quasi-linear number of arithmetic operations [59, 60]. This led to a recent preprint where the authors show that this precomputation can be done in $\tilde{O}(d^{\omega+1})$ operations, which is approximately $\tilde{O}(d^{3.372})$ [61]. This result is also extended to polynomials in more than 2 variables.

In the case of rings of the form $\mathbb{Z}/r\mathbb{Z}$, a breakthrough result showed that it is possible to evaluate a bivariate polynomial of degree d on d^2 points in $O(d^{2+\varepsilon} \log^{1+o(1)} r)$ operations [81]. This result is also extended to the case of more variables, when the number of variables is small compared to the degree. This constraint was recently removed [8, 9].

The case of approximate multipoint evaluation of multivariate polynomials was also recently investigated, with respect to an absolute error on the evaluation, leading also to a quasi-linear algorithm in the number of points [51].

Fast double sum of rational functions. Using the previous state-of-the-art algorithms, we can design a fast algorithm to compute the double sum of rational fractions. More precisely, let k be a positive integer, and let A and B be two sets of \mathbb{R}^k . Moreover, let f be a polynomial of constant degree d in each of its $2k$ variables, and g be a bivariate polynomial of constant degree d in each variable. Assume that A and B contains at most n points. In the real RAM model, the sum

$$S = \sum_{(a_1, \dots, a_k) \in A} \sum_{(b_1, \dots, b_k) \in B} \frac{f(a_1, \dots, a_k, b_1, \dots, b_k)}{g(a_1, b_1)}$$

can be computed in $\tilde{O}(d^{2k}n)$ arithmetic operations.

This can be achieved by using the symbolic variables x_1, \dots, x_k , and com-

putting the rational function:

$$\frac{N(x_1, \dots, x_k)}{D(x_1)} = \sum_{(b_1, \dots, b_k) \in B} \frac{f(x_1, \dots, x_k, b_1, \dots, b_k)}{g(x_1, b_1)}$$

where N and D are polynomials of degree in $O(nd)$ in x_1 and d in the other variables. Summing the symbolic fractions with a divide-and-conquer approach, and using a fast multiplication algorithm, D can be computed in $\tilde{O}(nd)$ and N in $\tilde{O}(d^{2k}n)$ arithmetic operations. Then using a fast multipoint evaluation algorithm, we can evaluate N and D on all the points of A in $\tilde{O}(d^k n)$ arithmetic operations, which allows us to compute S within the same complexity.

3.2 Application to computing the distance between terrains

The use of fast multipoint evaluation had an unexpected geometrical application. More precisely, this algebraic tool allowed us to defy a seemingly obvious lower bound to compute the distance between two terrains.

A *terrain* is the graph of a bivariate function over some planar domain, say, a square. It is sometimes used to model actual geographic terrains, such as elevation in a mountainous locale, but can also be applied to storing any two-dimensional data set, such as precipitation or snow cover data. A common way of interpolating and representing discrete two-dimensional data is a *triangulated irregular network*: the values of a bivariate function are given at discrete points in, say, the unit square. The square is triangulated, using data points as vertices. The function is then linearly interpolated over each triangle. This produces a piecewise-linear approximation of the “real” (and unknown) function.

A fundamental problem is that of comparing two terrains over the same domain, say, the unit square M , but given over two unrelated triangulations, denoted by A and B [1, 2]. If the two terrains are the graphs of the two bivariate functions $f, g : M \rightarrow \mathbb{R}$, one can naturally define a distance between them as

$$\|f - g\|_2 = \left(\iint_M (f(x, y) - g(x, y))^2 dx dy \right)^{1/2}.$$

In the case of triangulated irregular network, the function f and g are piecewise-linear functions defined over the triangulations A and B respectively (see Figure 1.1). The direct approach to compute the distance consists in computing the integral over the polygonal domains obtained by intersecting each triangle from A with each triangle from B . The main issue is that in the worst case, if each triangulation has n triangles, this approach requires computing $\Omega(n^2)$ polygonal domains.

However, using state-of-the-art multipoint algorithms from computer algebra, we showed that it is possible to compute the distance between two terrains with a number of operations quasi-linear in the number of input triangles.

Theorem 3 ([102]). *Given piecewise-linear functions f and g defined over different triangulations of the same domain M , with at most n triangles each, $\|f - g\|_2$ can be computed using $\mathcal{O}(n \log^4 n \log \log n)$ arithmetic operations.*

The proof of this result is based on three key ideas. The first one is to remark that the integral over the overlay of the two triangulations A and B can be reduced to a sum of rational functions over pairs of edges, plus some additional terms computable in linear time.

Then, the second idea is to use the bipartite clique decomposition to arrange the pairs of edges that intersect in complete grids. More precisely, if I is the set of pairs of edges from A and B that intersect, then, it is possible to compute in quasi-linear time the subsets A_i of edges in A and the subsets B_i of edges in B such that $I = \bigcup_i A_i \times B_i$. Moreover, the sum of the sizes of all the A_i and B_i is quasi-linear in the number of triangles of A and B .

Finally, the L_2 -distance can be written as a sum of double sums of the form:

$$\sum_i \sum_{a \in A_i} \sum_{b \in B_i} \frac{f_i(a, b)}{g_i(a, b)}$$

where the f_i and g_i are polynomials of constant degree. Using fast polynomial multiplication and multipoint evaluation algorithms presented in Section 3.1, we can evaluate those sums in a quasi-linear number of arithmetic operations.

Results and impact

The theoretical complexity we obtained is quasi-optimal, whereas before our result, the best known complexity was quadratic in the size of the input. Moreover, the underlying ideas of this work have been used in several works in algorithmic geometry [4, 23, 24].

Furthermore, following these results, I acquired a reputation as an expert in algebraic structures for geometric problems, which led, for example, to Yukiko Kenmochi contacting me to use algebraic methods on a discrete geometry problem [115]. This also allowed me to start a new collaboration with other researchers in algorithmic geometry such as Pankaj Agarwal (Duke University), detailed in Section 5.1.1 of my research program.

The results on distances between piecewise linear functions were published in the ACM Transactions on Algorithms journal, a journal read in the community of algorithmic geometry and combinatorial geometry on discrete data structures.

3.3 Application to isosurface triangulation with the marching cube and subdivision algorithms

The marching cube algorithm is a common algorithm to triangulate a surface defined by an equation of the form $f(x, y, z) = 0$. The first step of this algorithm consists in evaluating a polynomial on a grid of points. Then, using the signs

of the computed evaluations, a table indicates which triangle should be added to the triangulation.

We introduced two new ideas from computer algebra and numerical analysis to speed up the marching-cube algorithm.

3.3.1 Grid with Chebyshev nodes

The first idea is to take advantage of fast-multiplication algorithms to speed up the evaluation of the input functions on the grid. However, state-of-the-art algorithms recalled in Section 3.1 are numerically unstable [87] if we approximate the numbers with floating point numbers with fixed precision, as it is usually done in numerical computations. To overcome this limitation, the second idea is to replace the grid used in the marching cube algorithm with a grid based on Chebyshev nodes, where we can use fast multipoint evaluation algorithms that are numerically stable. Interpolating functions on Chebyshev nodes is a standard approach in numerical analysis [127], and `Chebfun` is a library based on this idea that is available for `Matlab` [42].

If we restrict our multipoint evaluation on a set of Chebyshev nodes, we can then use the Discrete Cosine Transform (DCT), in the same way as the Discrete Fourier Transform (DFT) can be used for multipoint evaluation on the roots of unity in the complex field [48, §8.2]. In this case, the bit complexity is linear in d , even with a precision lower than d [121, §3]. This special multipoint evaluation inherits the numerical stability of the DCT, and we extend the error analysis already known for the DFT [21].

Chebyshev nodes, Chebyshev basis and IDCT

Chebyshev polynomials can be defined as the unique sequence of polynomials $(T_n)_{n \in \mathbb{N}}$ satisfying

$$T_n(\cos \theta) = \cos(n\theta). \quad (3.3)$$

For $N \in \mathbb{N}$, the Chebyshev nodes $(c_k)_{k \in \llbracket 0, N-1 \rrbracket}$ are the roots of T_N :

$$c_k = \cos\left(\frac{2k+1}{2N}\pi\right) \text{ for } k \in \llbracket 0, N-1 \rrbracket. \quad (3.4)$$

The Chebyshev polynomials T_k for $k \leq d$ form a basis of the polynomials of degree at most d . The relation between the monomial and the Chebyshev bases is given by [95, section 2.3.1]:

$$x^k = 2^{1-k} \sum_{i=0}^{\lfloor k/2 \rfloor} \binom{k}{i} T_{k-2i}(x), \quad (3.5)$$

where the prime symbol denotes that the term $T_0(x)$, if there is one, is to be halved.

Lemma 1. For a polynomial of degree d , the change-of-basis matrix from the monomial basis to the Chebyshev basis is $B = (B_{i,j}) \in \mathbb{R}^{(d+1) \times (d+1)}$ where

$$B_{i,j} = \begin{cases} 2^{-j} \binom{j}{\frac{j}{2}} & \text{if } i = 0 \text{ and } j \text{ is even,} \\ 2^{1-j} \binom{j}{\frac{j-i}{2}} & \text{if } \exists k, i = j - 2k \text{ and } i \leq j, \\ 0 & \text{otherwise.} \end{cases} \quad (3.6)$$

Proof. The entry $B_{i,j}$ is the coefficient of T_i in the expansion of x^j in the Chebyshev basis. The formula for this entry follows from Equation (3.5). \square

Performing a change of basis is a multiplication of a $(d+1) \times (d+1)$ -matrix by a $(d+1)$ -vector. Thus, given a polynomial of degree d in the monomial basis, computing its coefficients in the Chebyshev basis can be done in $O(d^2 p \log(p))$ bit-operations in precision- p arithmetic. Remark that it is also possible to do the change of basis in $O(d \text{ polylog}(d))$ arithmetic operations [17, 113], however those methods are based on Taylor shift operations that require a number of bit operations quadratic in d [49].

Definition 2. Given $d, N \in \mathbb{N}^*$ with $d+1 \leq N$ and $(X_i)_{i \in \llbracket 0, d \rrbracket}$ a sequence of real numbers, the Inverse Discrete Cosine Transform $\text{IDCT}((X_i)_{i \in \llbracket 0, d \rrbracket}, N)$ is the sequence $(x_k)_{k \in \llbracket 0, N-1 \rrbracket}$ defined by

$$\forall k \in \llbracket 0, N-1 \rrbracket, x_k = \frac{1}{N} \left(\frac{1}{2} X_0 + \sum_{i=1}^d X_i \cos \left(\frac{i(2k+1)}{2N} \pi \right) \right). \quad (3.7)$$

When input and output sizes match, it is omitted: $\text{IDCT}((X_i)_{i \in \llbracket 0, d \rrbracket})$ denotes $\text{IDCT}((X_i)_{i \in \llbracket 0, d \rrbracket}, d+1)$.

Main result and experiments

For the fast multipoint evaluation, the polynomial is written in the Chebyshev basis and then the IDCT is applied on these coefficients.

Figure 3.1 experimentally verifies the relevance of using Chebyshev grids. It displays the computation times of the certified partial evaluation of our polynomials using fast discrete cosine transform, with a complexity of $O(d^3 + dN \log_2(N))$, and using Horner's method, with a complexity of $O(d^2 N)$. For the range of values we are interested in, specifically $10 \leq d \leq 100$ and $500 < N$ the discrete cosine transform is always faster. This justifies the use of the Chebyshev polynomials despite the preliminary costly change of basis.

3.3.2 Subdivision with sparse tensors datastructure

The second idea is to speed up evaluation costs in subdivision algorithms. A classical approach to reduce the number of evaluations is the use of a subdivision algorithm with a criterion that ensures that a large part of the input domain does not contain any zero of the input function. Usually, such criteria are

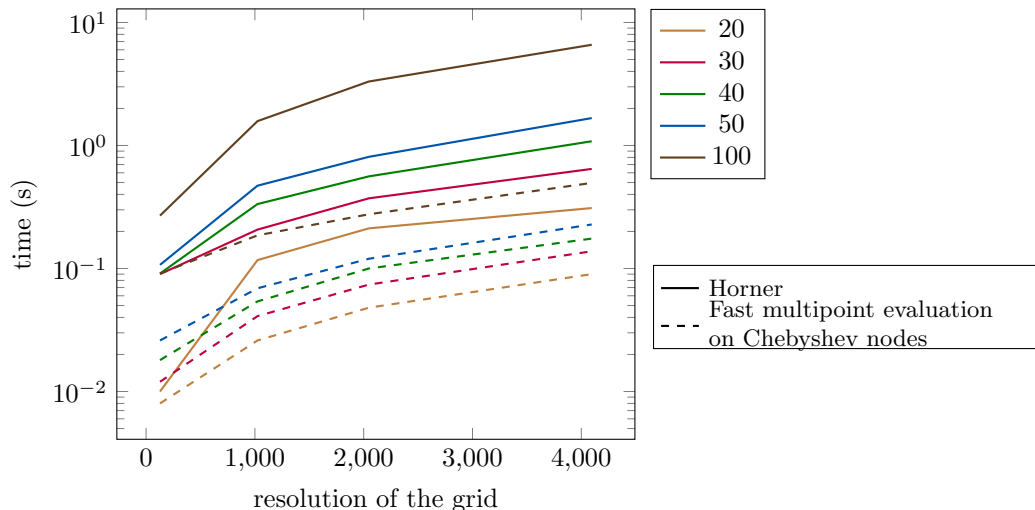


Figure 3.1: Computation times of the IDCT and the interval Horner evaluation for the partial evaluation step. The color in the legend indicates the degree of Kac polynomials. Dotted lines correspond to the IDCT and solid lines to the interval Horner scheme.

based on interval arithmetic ([73, 80, 88, 107, 114, 126] among others). Interval arithmetic is the generalization of standard arithmetic operations to the case where numbers are replaced by intervals. If $[a, b]$ and $[c, d]$ are two intervals, the result of $[a, b] + [c, d]$ is the interval $[a + c, b + d]$. If f is a polynomial in k variables and B is a product of k intervals, we denote by $\square f(B)$ the product of intervals returned when f is evaluated on B using interval arithmetic. The main property of interval arithmetic is that the interval $\square f(B)$ satisfies:

$$\square f(B) \supset \{f(x) \mid x \in B\}.$$

Subdivision algorithms roughly consist in evaluating f on boxes created along a subdivision tree, and discarding them if f doesn't vanish in them. If the input function is a high degree polynomial, one of the bottlenecks of those algorithms is the time required to evaluate f . I proposed a new approach that amortizes the evaluation cost over the boxes created in a subdivision algorithm [99]. It combines on the one hand partial evaluations of the input polynomial with interval arithmetic, and on the other hand sparse tensors [34, 125] to store the boxes created during the subdivision algorithm. This approach was implemented in the software `voxelize`, and the source code is available on gitlab¹. Experimental results show that this software can enclose the zero set of polynomial systems that were not reachable with state-of-the-art software. For example, Figure 3.3 shows the output boxes of the software `voxelize` enclosing

¹<https://gitlab.inria.fr/gmoro/voxelize>

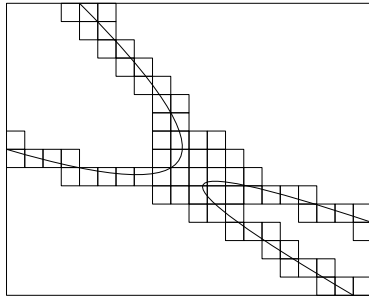


Figure 3.2: Boxes on the same level of the subdivision tree.

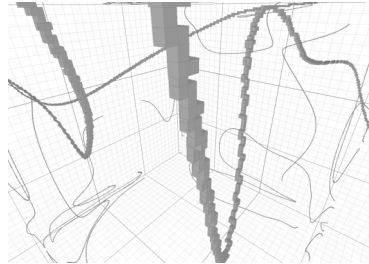


Figure 3.3: Enclosing of a curve defined by 2 trivariate polynomials of degree 100.

an algebraic curve defined by two polynomial equations of degree 100, where the coefficients are randomly drawn from a normal law centered at zero.

Amortized Evaluation on a Grid of Boxes

The first idea to reduce the evaluation redundancies is to use partial evaluation. Assume that $f(x_1, x_2)$ is a bivariate polynomial of degree d . Moreover, let $(I_i)_{0 \leq i < n}$ and $(J_j)_{0 \leq j < n}$ be two sequences of real intervals. Using the Horner scheme, evaluating f on a box requires $O(d^2)$ arithmetic operations, and evaluating f on all the boxes $I_i \times J_j$ for $0 \leq i, j < n$ requires $O(d^2 n^2)$ arithmetic operations. By reorganizing the operations using partial evaluations, the number of arithmetic operations can be reduced to $O(dn(d+n))$. This idea is well known and was used for example to speed up the multiplication of polynomials [111]. It is also currently implemented in the well-spread library NumPy to evaluate polynomials in 2 and 3 variables [54].

More precisely the operations are reordered as follows. For a given I_i , the partial evaluation of f in I_i results in a univariate polynomial f_i of degree d . This step requires $O(d^2)$ arithmetic operations. Then evaluating f_i on n intervals requires $O(dn)$ arithmetic operations. Finally, repeating these operations for all the n intervals I_i , this allows us to evaluate f on all the boxes of the grid with a total number of arithmetic operations in $O(dn(d+n))$. More generally, for higher dimensions, this leads to the following result.

Property 1 ([111]). *Let f be a polynomial in k variables and of degree at most $d-1$ in each variable. Let X_1, \dots, X_k be k sets of n real intervals each. Then it is possible to evaluate f on all the boxes of $X_1 \times \dots \times X_k$ in $O(kdn \max(n, d)^{k-1})$ arithmetic operations.*

In the case where $n > d$, this approach results in a significant speedup since the amortized number of arithmetic operations to evaluate f on each box of the grid is $O(kd)$ instead of $O(d^k)$.

	0	1	2	3	4	5
0	5	1				
1	7	3				
2						
3	8			4	9	

Figure 3.4: Numbers stored in a sparse matrix.

Amortized Evaluation on a Sparse Subset of a Grid

For the simple subdivision algorithm mentioned at the beginning of the introduction, if the boxes created are never discarded, then each level of the subdivision tree forms a dense grid of boxes. In this case, the partial evaluation approach shown in the previous section can be applied directly to reduce the total number of arithmetic operations required to evaluate f on each box with interval methods. In the general case though, many boxes are discarded, and the boxes appearing in a given level of the subdivision tree form a subset of a grid, as shown in Figure 3.2. The boxes created in the subdivision algorithm can be handled in different orders. Using a breadth-first walk on the subdivision, the boxes on the same level are a subset of a grid. In this case, we need to evaluate a polynomial on a sparse subset of a grid.

Boxes in a sparse subset of a grid can be gathered and stored as a sparse tensor in the Compressed Sparse Fiber (CSF) format [34, 125]. The CSF is a generalization of the Compressed Row Format used to store the entries of a sparse matrix. Performing the partial evaluation approach on a set of boxes in a CSF format leads to Theorem 4.

For a subset of a 2D grid, the data structure is a labeled tree that stores the positions of the non-empty rows in the children of the root node, and then in each row, the position of the non-empty entries are stored in the children of the corresponding node (Figure 3.5). In higher dimension k , this idea is applied recursively.

As an example, using the compressed sparse data structure to store the sparse matrix given in Figure 3.4, we get the tree shown in Figure 3.5, and its support is $\{(0, 0), (0, 1), (1, 0), (1, 1), (3, 0), (3, 3), (3, 4)\}$.

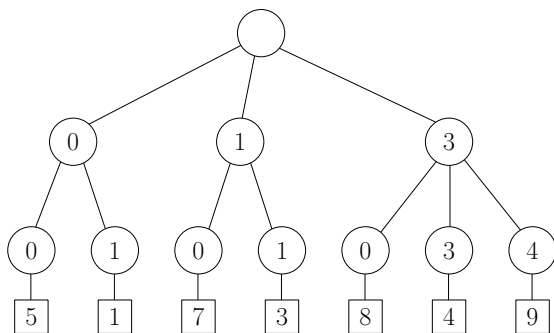


Figure 3.5: Compressed Sparse Fiber associated to the sparse matrix.

Main Result

For a set E , we denote by $|E|$ its number of elements. Then, we define the notations for the size of the projection of a subset E of a grid. In particular, the size of the projection is smaller when the elements of E are aligned within the grid.

Definition 3. Given a finite set $E \subset \mathbb{N}^k$, and an integer i between 1 and k , we denote by $N_i(E)$ (resp. $\tilde{N}_i(E)$) the number of elements in the projection E on the first (resp. last) i coordinates, counting repeated projections only once.

Even though this definition holds for a set of integer tuples, it can be naturally extended for multivariate polynomials. Indeed, for each monomial, we can associate its vector of exponents. If f is a polynomial in k variables, for a given integer i , we can define $N_i(f)$ (resp. $\tilde{N}_i(f)$) as the size of the projections of the set of vectors of exponents of f to their first (resp. last) i coordinates.

For S a set of points or boxes that is a subset of a grid, we can also extend the definition of N_i by simply indexing the elements of S by their integer positions in the grid. Letting S_{ind} be the set of integer indices of the boxes of S , we can define $N_i(S)$ by $N_i(S_{ind})$.

We can now state our main theorem to evaluate a multivariate polynomial on a set of boxes that is a sparse subset of a grid of boxes G that is the Cartesian product of k sets of intervals $X_1 \times \cdots \times X_k$.

Theorem 4 ([99]). Let f be a polynomial in k variables, and S be a subset of boxes of G . It is possible to evaluate f on all the boxes of S in $O(\sum_{i=0}^{k-1} \tilde{N}_{k-i}(f)N_{i+1}(S))$ arithmetic operations.

When the set of boxes enclose a variety of dimension j , the projection of S on the first j coordinates is often a dense grid. In this case, we have the following corollary.

Corollary 1 ([99]). For $1 \leq j \leq k - 1$, assume that the projection of S on the first j coordinates is:

i. a dense grid, denoted by $X_1 \times \cdots \times X_j$

ii. $|X_i| > d$ for all $1 \leq i \leq j$.

Then we can evaluate each box of S in $O(j(d+1)^{k-j+1})$ arithmetic operations on average, instead of $O((d+1)^k)$ operations.

3.3.3 Results and impact

The new approaches I have developed have, on one hand, improved the asymptotic complexity for plotting algebraic curves in the plane [56]. To plot a curve of degree d on a grid of size $N \times N$, the naive marching cube algorithm uses $O(d^2N^2)$ arithmetic operations, whereas in well-conditioned cases, our method uses a number of operations that is quasi-linear in dN .

Moreover, for algebraic curves embedded in 3-dimensional spaces, the software `voxelize` is the only one capable of plotting algebraic curves defined by polynomials of degree 100 (Figure 3.3).

Part of this work has been published, notably in the symbolic computation conference ISSAC [56], and more recent work on the subject has been accepted to the conference Computer Algebra in Scientific Computing 2024 [99]. This more recent work also allows for rapidly reducing the areas of space containing semi-algebraic varieties, which could help solve certain open problems related to the rigid embedding of certain varieties (Axis 5.1.2 of the Research Program).

Chapter 4

Reduction of precision

Finally, while working with geometric model, another route for optimization is to approximate the results. In numerical algorithms, it is common to use classical algorithms, and to perform arithmetic operations with limited precision. In this section, we will go further. Taking into account that the output will be approximated, we will change the classical algorithms to reduce computations while introducing approximation errors that are within a required order of magnitude. More precisely, the literature covers well the cases where the polynomials have a degree lower than the required precision. However, there are few complexity results for the cases where the required precision is lower than the degree. This case appear for example if we want to sample points roughly uniformly in a disk, while avoiding clusters of points that could appear if the points are sampled independently. This can be done by computing the roots of a polynomial of high degree ([43],[62, chapter 2] and references therein). After introducing several state-of-the-art results on fast algorithms to handle univariate polynomials, this chapter will present new results on fast multipoint evaluation and root finding for polynomials with a degree larger than the required precision.

4.1 Preliminary tools

Notations. In this chapter, for a polynomial in one variable f , we let:

$$\|f\|_1 = |f_0| + \cdots + |f_d|$$
$$\tilde{f}(|z|) = |f_0| + \cdots + |f_d||z|^d.$$

Moreover, for a complex number γ and a positive real ρ , we denote by $D(\gamma, \rho)$ the closed disk of center γ and radius ρ .

4.1.1 Fast algorithms with finite absolute precision

In this section, the numbers are stored with fixed-point representation. More precisely, a number with a magnitude less than 2^τ and m bits after the binary

points will be represented with $O(m + \tau)$ bits. We say that the value m is the absolute precision of this number.

Fast elementary operations

We start with classical results on elementary operations, notably on the multiplication and the composition of polynomials.

Proposition 1 ([121, Theorem 2.2], [119, Proposition 3.2]). *Let f and g be two univariate polynomials of degree d with $\|f\|_1$ and $\|g\|_1$ less than 2^τ , and an integer $m \geq \log(d + 1)$. It is possible to compute a polynomial h such that $\|h - fg\|_1 \leq 2^{-m}$ in $\tilde{O}(d(m + \tau))$ bit operations.*

Using the Fast Fourier transform algorithm, we can also evaluate in a quasi-linear time a polynomial on the roots of unity. Note that in this case, even if the required precision m is smaller than d , the algorithm is still quasi-linear in dm .

Proposition 2 ([121, §3], [119, Proposition 3.3]). *Let f be a polynomial of degree d , and $\|f\|_1 \leq 2^\tau$ with $\tau \geq 1$, and an integer $m \geq \log(d + 1)$. It is possible to compute complex numbers y_0, \dots, y_{d-1} such that $\sum_{k=0}^{d-1} |y_k - f(e^{i2\pi k/d})| \leq 2^{-m}$ in $\tilde{O}(d(m + \tau))$ bit operations.*

Finally, another classical result that we will use is the fast composition of polynomials.

Proposition 3 ([119, Theorem 2.2]). *Let f and g be two polynomials of degree d with $\|f\|_1 \leq 2^\tau$ and $\|g\|_1 \leq 2^\nu$ where $\tau \geq 1$ and $\nu \geq 1$. Let m be a positive integer. It is possible to compute a polynomial h of degree $d - 1$ such that $\|h(X) - f(g(X)) \bmod X^d\|_1 \leq 2^{-m}$ in $\tilde{O}(d(m + \tau + d\nu))$ bit operations.*

Remark 1. *Ritzmann [119, Theorem 2.2] used the same bound for $\|f\|_1$ and $\|g\|_1$. Our proposition is a direct consequence of Ritzmann's theorem if we multiply f in the input by $2^{\nu-\tau}$, and the result by $2^{\tau-\nu}$. This reduction can be done in $\tilde{O}(d(\tau + \nu + m))$ bit operations.*

Fast approximate multipoint evaluation

When we want to evaluate a polynomial on multiple points, it is possible to amortize the number of bit operations when the precision required m is larger than the degree d [58, 82]. In a recent work [58, §3.2], van der Hoeven showed that even when m is less than d , it can be done in $\tilde{O}(d^{3/2}m^{3/2})$. However, this bound is not optimal when m is greater than d . For the case $m > d$, we recall here another state-of-the-art bound on the bit complexity for fast multipoint evaluation.

Proposition 4 ([58, Lemma 11], [82, Theorem 9]). *Let f be a polynomial of degree d , with $\|f\|_1 \leq 2^\tau$, with $\tau \geq 1$, and let $x_1, \dots, x_d \in \mathbb{C}$ be complex points with absolute values bounded by 1. Then, computing y_k such that $|y_k - f(x_k)| \leq 2^{-m}$ for all k is possible in $\tilde{O}(d(m + \tau + d))$ bit operations.*

Even though the bit complexity is quadratic in d , this approach is near optimal when m is greater than d , since its complexity matches the size of the output in this case. Notably our approach relies on this result since we reduce the problem of evaluating a polynomial of degree d to the problem of evaluating several polynomials of degree m with a precision greater than m .

Fast approximate factorization

Another important result on univariate polynomials is a bound on the bit complexity to approximate all its roots. In the complex plane, approximating the roots is equivalent to compute an approximate factorization. We recall the state-of-the-art bound on the bit complexity for this problem.

Proposition 5 ([112, Theorem 2.1.1]). *Let f be a polynomial of degree d with leading coefficient c_d and all its roots ζ_k in the unit disk, and $m \geq d \log d$ a fixed real number. It is possible to compute complex numbers z_1, \dots, z_d such that $\|f(X) - c_d \prod_{k=1}^d (X - z_k)\|_1 \leq 2^{-m} \|f\|_1$ in $\tilde{O}(dm)$ bit operations.*

Remark 2. *The theorem also holds with the same complexity for a polynomial h that has all its roots in the disk centered at the origin and of radius $c2^{m/d}$ for a constant $c \geq 1$.*

Proof of the remark. Let $f(Y) = h(c2^{m/d}Y)$. Then f has all its roots in the unit disk and $\|f\|_1 \leq c^d 2^{m/d} \|h\|_1$. Computing the approximate factorization g of f such that $\|f - g\|_1 \leq 2^{-2m-d \log_2 c} \|f\|_1$ can be done in $\tilde{O}(dm)$ since $m \geq d \log d$. Then with the change of variable $Y = X2^{-m/d}/c$, we have $\|h(X) - \tilde{f}(X2^{-m/d}/c)\|_1 \leq 2^{-m} \|h\|_1$. \square

This theorem does not directly give a bound on the distances between the roots and their approximations. Schönhage shows [123, §19] that in the worst case $|\zeta_k - z_k| < 4 \cdot 2^{-m/d}$. This bound can be improved for well-conditioned roots. In our case, we will also need a bound on the distance between some pairs of roots of two polynomials that have different degrees. We will use Kantorovich theory for the bounds in these cases (Section 4.1.1).

Note that Remark 2 requires a bound on the modulus of the roots of the polynomial. For that, we will use the classical Fujiwara bound.

Proposition 6 ([47]). *Let $f = \sum_{k=0}^d f_k X^k$ be a polynomial of degree d . Then the moduli of the roots of f are lower or equal to $2 \max_{1 \leq k \leq d} \sqrt[k]{\left| \frac{f_{d-k}}{f_d} \right|}$.*

Applying this bound on the polynomial returned by a hyperbolic approximation, slightly perturbed, will allow us to verify that the assumptions of Remark 2 are satisfied.

Certification of the roots

Several approaches in the literature guarantee that a neighborhood of a point contains a unique root of a given polynomial. We may cite notably Kantorovich

criterion [38, §3.2], Smale's alpha theorem [38, §3.3], Newton interval method [107, Theorem 5.1.7], Pellet's test [96], Pellet's test combined with Graeffe iteration [6, 72], Cauchy's integral theorem [69, 70], and others [120], ... A first crude bound from the literature to guarantee that a disk centered at a point x contains a root of a polynomial f is the following.

Proposition 7 ([55, Theorem 6.4e], [10, Theorem 9]). *Let f be a polynomial of degree d , and x a complex point. Let $r_k = \sqrt[k]{k! \binom{d}{k} \left| \frac{f(x)}{f^{(k)}(x)} \right|}$. Then, for all $1 \leq k \leq d$, the disk $D(x, r_k)$ contains a root of f . In particular, for $k = 1$, the disk $D(x, d|f(x)/f'(x)|)$ contains a root of f .*

With some additional conditions, Kantorovich criterion provides a smaller radius to guarantee that a disk contains a root f .

Proposition 8 ([38, Theorem 88]). *Given a function f of class C^2 , and a point x such that $f'(x) \neq 0$, let $\beta = |f(x)/f'(x)|$, and $K = \sup_{|y-x| \leq 2\beta} |f''(y)/f'(x)|$. If $2\beta K \leq 1$, then f has a root in the disk $D(x, 2|f(x)/f'(x)|)$.*

The previous propositions are useful to find a disk that contains a root of f , but they don't guarantee that the disk contains a unique root of f . In order to guarantee the unicity of the root in a disk, we use the lower bound from Kantorovich theory on the size of the basin of attraction of the roots of f . More precisely, for each root ζ , we bound the size of a disk containing ζ where the Newton method always converges toward ζ .

Proposition 9 ([38, Theorem 85]). *Given a function f of class C^2 , and a root ζ such that $f'(\zeta) \neq 0$. If there exists $r > 0$ such that $2rK \leq 1$, with $K = \sup_{|x-\zeta| \leq r} |f''(x)/f'(\zeta)|$, then ζ is the unique root of f in the disk $D(\zeta, r)$. Moreover, for any $x_0 \in D(\zeta, r)$, the Newton sequence defined by $x_{n+1} = x_n - f(x_n)/f'(x_n)$ converges toward ζ .*

Remark 3. *If f is a polynomial of degree d , and $s \geq \sup_{x \in D(0,1)} |f''(x)|$, then a consequence of Proposition 9 is that the set of disks $D(\zeta, 1/(2s\kappa_1(f)))$, for all roots ζ of f in the unit disk, are pairwise distinct, where κ_1 is an absolute condition number defined in Section 4.1.2.*

For any complex point x , we also prove the following lemma to have a criterion guaranteeing that a ball around x is included in a basin of attraction of a root of f .

Lemma 2. *Given a function of class C^2 and a point $x \in \mathbb{C}$ such that $f(x) \neq 0$. Let $r > 2|f(x)/f'(x)|$ and $K > |f''(y)/f'(x)|$ for all $y \in D(x, 4r)$. If $5rK \leq 1$ then, f has a unique root ζ in $D(x, r)$, and for all $x_0 \in D(x, r)$, the Newton sequence starting from x_0 converges to ζ .*

Proof. According to Proposition 8, f has a root ζ in $D(x, r)$. Then, $|f'(\zeta)| \geq |f'(x)| - |x - \zeta| \sup_{y \in D(x, r)} |f''(y)|$. This implies that $f'(\zeta) \geq |f'(x)|(1 - rK) \geq 4|f'(x)|/5$. Letting $\tilde{K} = 5K/4$, we have that $\tilde{K} \geq |f''(y)/f'(\zeta)|$ for all $y \in$

$D(x, 4r) \supset D(\zeta, 2r)$. Remark that $4r\tilde{K} \leq 1$, such that using Proposition 9, this implies that for all $x_0 \in D(\zeta, 2r) \supset D(x, r)$, the Newton sequence starting from x_0 converges toward ζ . □

4.1.2 Condition number

Absolute condition number

Our root isolation algorithm has a bit complexity that depends on the condition number of the input polynomial. The absolute condition number is a measure of the displacement of its roots with respect to the displacement of its coefficients. More precisely, for a polynomial f with a vector of coefficients $c \in \mathbb{C}^{d+1}$, and a simple root ζ of f such that $f'(\zeta) \neq 0$, there exists a neighborhood $U \subset \mathbb{C}^{d+1}$ of c , a neighborhood $V \subset \mathbb{C}$ of ζ and a differentiable function $\psi : U \rightarrow V$ that maps $c \in U$ to the unique zero in V of the corresponding polynomial. Letting $D\psi(\zeta)$ be the gradient of ψ at ζ , the condition number of ψ at ζ is the induced norm $\|D\psi(\zeta)\|_2 = \max_{\|\delta\|_2=1} |D\psi(\zeta) \cdot \delta| = \left(\sum_{k=0}^d |\zeta|^{2k}\right)^{1/2}$. If we consider the induced 1-norm instead, we have $\|D\psi(\zeta)\|_1 = \max_{\|\delta\|_1=1} |D\psi(\zeta) \cdot \delta| = \max_{k=0}^d (|\zeta|^k) = \max(1, |\zeta|^d)$.

Definition 4. [22, §14.1.1] *The standard local absolute condition number of a polynomial f of degree d at a root ζ is $\kappa_2(f, \zeta) = \frac{1}{|f'(\zeta)|} \left(\sum_{k=0}^d |\zeta|^{2k}\right)^{1/2}$. Considering all the roots, we define the standard absolute condition number of f as $\kappa_2(f) = \max_{f(\zeta)=0} \kappa_2(f, \zeta)$.*

Remark 4. *The standard absolute condition number is obtained by considering the 2-norm. If we consider the induced 1-norm instead, we get the condition number $\kappa_1(f, \zeta) = \frac{\max(1, |\zeta|^d)}{|f'(\zeta)|}$, such that $\kappa_1(f) \leq \kappa_2(f) \leq \sqrt{d}\kappa_1(f)$.*

As shown in Remark 4, the bound depending on the logarithm of the condition number for the 1-norm and the 2-norm will be the same up to a factor logarithmic in d . In the following, we will focus on the condition number κ_1 induced by the 1-norm.

For square-free polynomials with integer coefficients, the condition number is finite. The following proposition bounds the condition number for square-free polynomials with integer coefficients.

Proposition 10 ([91, last inequality]). *Given a square-free polynomial f of degree d with integer coefficients, let τ be a real such that $\|f\|_1 \leq 2^\tau$. Then $\log(\kappa_1(f))$ is in $O(d\tau + d \log d)$.*

Relative condition number

In floating-point representation, the errors are relative. Let f_ε be the polynomial obtained by adding a relative error bounded by ε on the coefficients of f . This amounts to multiply all the coefficients f_j by $(1 + \varepsilon_j)$ with $|\varepsilon_j| \leq \varepsilon$.

In the *evaluation problem*, for any complex point z we have

$$|f(z) - f_\varepsilon(z)| \leq \varepsilon \tilde{f}(|z|)$$

and this bound is tight. In Theorem 5 we focus on the problem of finding the approximate value of $f(z)$ with the same error bound.

For the *root-finding problem*, we can also define the relative condition number [53],[57, §1.6] associated to a relative perturbation of the coefficients of f . More precisely, if ζ is a non-zero root of f such that $f'(\zeta) \neq 0$, and $\check{\zeta}$ is the closest root of f_ε , the relative condition number of ζ denoted by $\text{cond}(f, \zeta)$ is defined as $\lim_{\varepsilon \rightarrow 0} |\zeta - \check{\zeta}| / (|\zeta| \varepsilon)$. Using the Taylor expansion $f_\varepsilon(\zeta) = (\zeta - \check{\zeta})f'_\varepsilon(\zeta) + O(|\zeta - \check{\zeta}|^2)$ and $|f(\zeta) - f_\varepsilon(\zeta)| \leq \varepsilon \tilde{f}(|\zeta|)$, this leads to:

$$\text{cond}(f, \zeta) = \frac{\tilde{f}(|\zeta|)}{|\zeta| |f'(\zeta)|}.$$

This means that the number of bits of precision required to compute the first significant bit of ζ is in $O(\log(\text{cond}(f, \zeta)))$. In the case where $\zeta = 0$, we let $\text{cond}(f, \zeta) = 0$, and if ζ is a non-zero multiple root, then we let $\text{cond}(f, \zeta) = \infty$ by convention. As in the previous section, we define the *relative condition number* of f as $\kappa = \max_{\zeta | f(\zeta)=0} \text{cond}(f, \zeta)$. One of our main results focus on the problem of finding the roots of f with a number of bit operations quasi-linear in κ and quasi-linear in d .

4.1.3 Newton polygon

Given a non-zero complex coefficient f_j of the polynomial f , we can associate it with the value $-\log_2 |f_j|$. Then for each index $0 \leq j \leq d$ such that $f_j \neq 0$, let P_j be the point $(j, -\log_2 |f_j|)$. The *Newton polygon* of f is the lower convex hull of the set of points P_j , as illustrated on Figure 4.1. It has been used to give a rough first estimation of the modules of the roots of a polynomial [10, 11, 52, 71, 92, 109, 110, 123]. The main idea is that for each edge of the Newton polygon, we can associate its slope s . Then $\exp(s)$ can be used as a rough estimation of the modulus of a root of f . Combined with the Ehrlich-Aberth method, this lead to state-of-the-art timings for finding the complex roots of a polynomial [10, 12]. It is an essential tool of our method to compute a partition of the complex plane for a piecewise approximation of the input polynomial.

4.2 Application to approximate evaluation and root finding

We can now state our main results, with some reasonable assumptions on the precision such as $m > \log d$ for a simpler presentation, where d is the degree of the input polynomial and m the relative precision of its coefficients. In this section, the numbers are stored with a floating-point precision. More precisely,

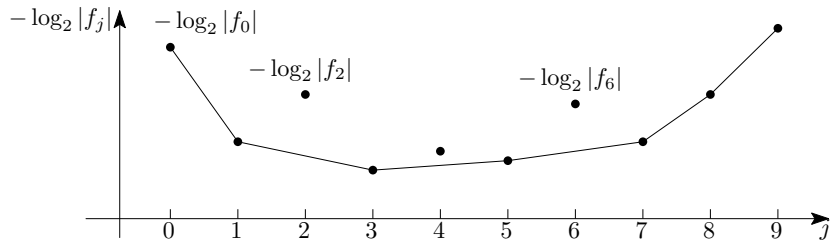


Figure 4.1: The Newton polygon of the polynomial f , using the valuation $-\log_2 |f_j|$ for the j -th coefficient.

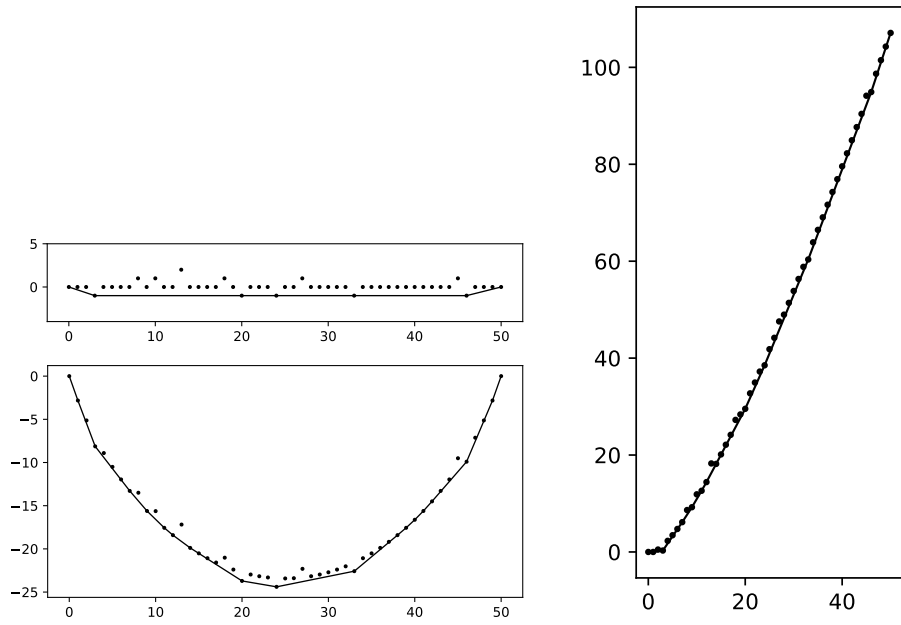


Figure 4.2: Newton polygon of a hyperbolic (top left), elliptic (bottom left), and flat (right) polynomial of degree 50.

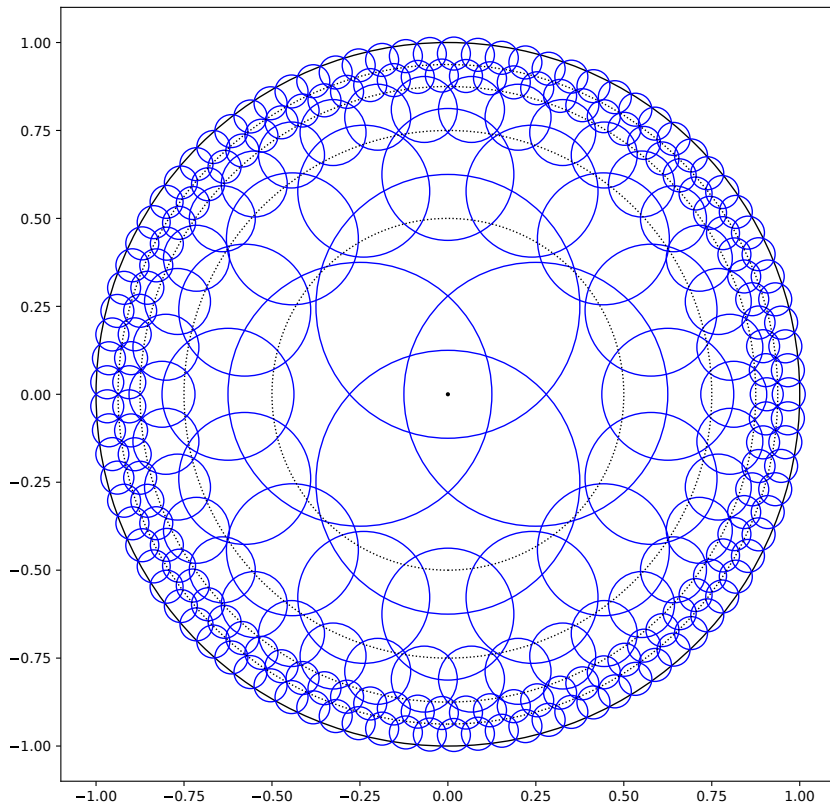


Figure 4.3: Hyperbolic covering: the union of the blue disks contains the unit disk.

a number of magnitude less than 2^τ with a relative precision m will be stored with a mantissa containing its first m bits, and with an exponent part, a relative integer stored on $O(\log \tau)$ bits.

The first theorem shows that we can evaluate a polynomial of degree d on d points with a complexity quasi-linear in d times the size of a floating-point representation of the coefficients. In particular, if the magnitudes of the coefficients are bounded by 2^τ , this means that our algorithm will be quasi-linear in $d(m + \log \tau)$.

The main idea is to use piecewise polynomial approximations. A first approach was aimed at bounding the absolute error bound, using fixed-point approximations [98]. This led to the introduction of a fixed partition of the complex plane, so-called *hyperbolic covering* (Figure 4.3).

Then, in order to obtain a bound on the relative error bound, the main idea is to adapt the partition of the complex plane with the Newton polygon of the input polynomial (Figure 4.4). Then on each cell of the piecewise polynomial approximation, we have computed a small degree polynomial, and we can use

the state-of-the-art algorithms from the previous sections to evaluate them or find their roots efficiently. This leads to the following theorems [64, 65].

Theorem 5 (Evaluation [65]). *Let f be a polynomial of degree d with complex floating-point coefficients of magnitudes less than 2^τ and let $m > \log d$ be a positive integer. It is possible to compute in $\tilde{O}(d(m + \log \tau))$ bit operations a piecewise polynomial approximation f_{pw} such that*

$$|f(z) - f_{pw}(z)| \leq 2^{-m} \tilde{f}(z)$$

for all $z \in \mathbb{C}$. Moreover, given d complex points z_1, \dots, z_d with $|z_j| \leq 2^\tau$, it is possible to evaluate:

- i. $f_{pw}(z_1)$ in $\tilde{O}(m(m + \log \tau))$ bit operations,
- ii. $f_{pw}(z_1), \dots, f_{pw}(z_d)$ in $\tilde{O}(d(m + \log \tau))$ bit operations.

Remark 5. *In the case i., if $d < m$, the evaluation of $f_{pw}(z_1)$ can be done in $\tilde{O}(d(m + \log \tau))$ bit operations, by choosing $f_{pw} = f$ and using the Horner algorithm to evaluate f .*

The second theorem allows us to find the roots of polynomials with a number of bit operations quasi-linear in $\log \tau$, and to handle polynomials with large coefficients such as resultant polynomials of two bivariate polynomials.

Theorem 6 (Root finding [65]). *Let f be a polynomial of degree d with complex floating-point coefficients of magnitudes less than 2^τ and let m be a positive integer. It is possible to compute the m most significant bits of the roots ζ of f such that $\text{cond}(f, \zeta) \leq 2^m$ in $\tilde{O}(d(m + \log \tau))$ bit operations.*

Moreover, if $\kappa = \max_{\zeta | f(\zeta)=0} \text{cond}(f, \zeta)$, it is possible to compute the m most significant bits of all the roots of f in $\tilde{O}(d(m + \log \tau + \log \kappa))$ bit operations.

Our approach consists in computing a piecewise approximation in two steps. A first approximation is computed on a partition of the complex plane in concentric rings. Then we refine our piecewise approximation by partitioning each ring uniformly in angular sectors.

4.3 Results and impact

This fundamental problem has been the subject of numerous studies, and before my result, there was no algorithm capable of numerically evaluating a polynomial of degree d at d points with a sub-quadratic number of machine operations in d . For the problem of root isolation, up to our knowledge, it is also the first time an algorithm is quasi-linear in the logarithm of the numerical conditioning. This result is the culmination of resolving many theoretical obstacles. The first challenge was to design the data structure for a specific family of polynomials with random coefficients [51]. Then, the second challenge was to generalize this approach to other families of random polynomials. The third difficulty consisted

of generalizing previous approaches to handle all types of polynomials, using a fundamental tool called the *Newton polygon*. Finally, for the general case, proving that the data structure based on the Newton polygon could be calculated in quasi-linear time was also a significant challenge. The proof of this complexity result was achieved using an original geometric method [64].

The new approach I presented has been well received by the scientific community and has earned me invitations to several conferences (RTCA 2023, RAIM 2023) and team seminars, as well as an article on the Inria blog. Jürgen Gerhard from Maplesoft has also shown interest and funded several months of engineering to ensure the integration of the computation library into the Maple software. Moreover, the algorithms I developed allow the generation of points in the plane following a repulsive random law in quasi-linear time. This could have applications particularly for learning algorithms [86] or physical simulations [5, 90].

This work has resulted in several publications [64, 65, 98], and two software packages have been developed. The first, `hefroots` is currently integrated into Maple 2023 and has been registered with the APP. It relies on the first approach, which was only effective for a specific family of polynomials. The second software, `pwpoly`, has been integrated into Maple 2024.

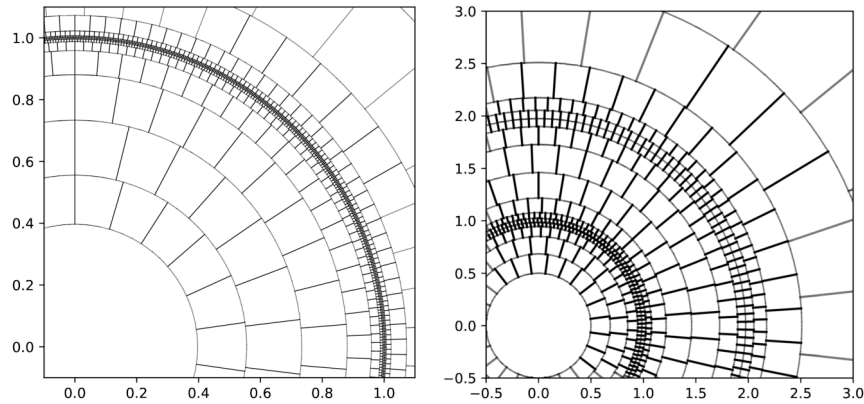


Figure 4.4: Partitions of \mathbb{C} for piecewise polynomial approximations adapted to a hyperbolic polynomial on the left, and adapted to the polynomial $(z^{200} - 1)(z^{200} - 2^{200})$ on the right.

Chapter 5

Research project

In many fields such as robotics, control theory, or mathematics, researchers and engineers are interested in the geometric properties of real solutions S of a system of polynomial or analytic equations. Is S empty? What are the extremal points of S ? What are the approximate volumes of the cells of S ? Are two points in the same connected component of S ? Are they in the same connected component of the complement of S ? If S is a surface, what does its 3D plot look like?

In many cases, these systems of equations possess hidden structures that make it easier to answer these questions. A classic example is the case where equations can be factored, thus reducing a difficult problem to several easy problems. My research project revolves around using algebraic structures to solve open problems in geometry and to accelerate the resolution of engineering problems manipulating real varieties.

5.1 Short term: Algebraic structures for geometric problems

Initially, some problems I have studied have highlighted algebraic structures rarely used in solving certain geometric problems. In particular, multipoint evaluation seems to be applicable for calculating other types of distances between terrains. Moreover, the study of assembly modes of mechanisms with rigid bars has a particular algebraic structure that can be used to study several other geometric problems of rigid embeddings.

5.1.1 L_1 -distance between two terrains

In an initial work with Boris Aronov, we showed that the L_2 -distance between two terrains of size n could be computed in a quasi-linear number of operations in n , whereas previous state-of-the-art algorithms used a quadratic number of operations [102]. However, if we consider the L_1 -distance, new combinatoric issues

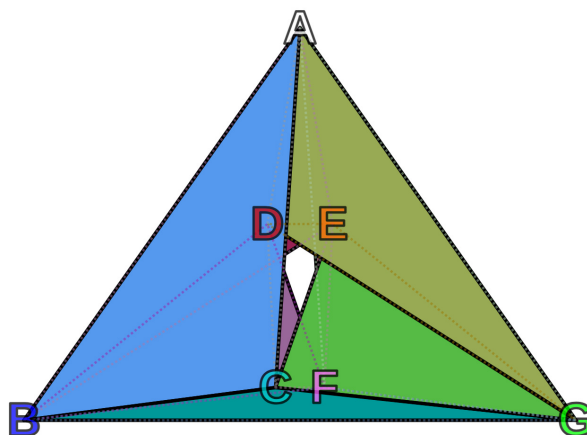


Figure 5.1: The Császár torus is a triangulation of a torus with only seven vertices. However, the sum of the angles around each vertex is not 2π . Thus, it does not have the same metric as the flat torus.

related to the introduction of absolute values makes the problem significantly more difficult. In this case, using the same approach based on fast single-variable multipoint evaluation, we couldn't obtain a sub-quadratic bound. On the other hand, following discussions with Boris Aronov and Pankaj K. Agarwal, we now have good hopes of obtaining a sub-quadratic bound for this problem by combining recent work on locating real points in an arrangement of semi-algebraic sets [3] with work on multipoint evaluation of polynomials in 2 variables [108].

5.1.2 Rigid embeddings

The construction of mechanisms in the plane with prescribed bar lengths is a classic problem in robotics that can be seen as a problem of rigid graph embedding. More generally in geometry, several conjectures concern the rigid embedding of mathematical objects. For example, is it possible to construct in 3D with seven vertices the triangulation of a torus (Figure 5.1) that has the same metric as the flat torus [117]? Or, given the four face areas of a tetrahedron, its volume, and the radius of its circumscribed sphere, how many embeddings does it admit in \mathbb{R}^3 [129]? A fundamental tool for modeling these problems is the Cayley-Menger matrix, which encodes the distances between points. However, the resulting systems of equations possess a specific algebraic structure related to Sylvester's determinant identity that has never been exploited to attempt to solve these problems. By combining this structure with other techniques such as amortized subdivision presented in 3.3.2, I have good hope of making progress on these problems.

More specifically, in a recent article [93], Ileana Streinu and Malić Goran were interested in calculating the elimination of variables in the Cayley-Menger

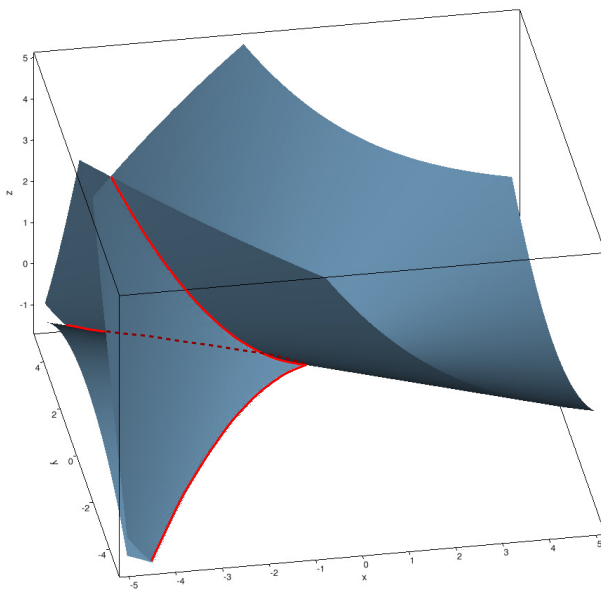


Figure 5.2: Singular surface with a dovetail, obtained by projection of a smooth surface.

ideal associated with a rigid graph embedded in the plane. The authors show that with a well-chosen sequence of resultant calculations, they significantly improve the efficiency of the calculation compared to a basic Gröbner basis calculation. However, by using algebraic structures specific to Cayley-Menger ideals, some preliminary experiments I have conducted show that it is possible to significantly accelerate the resolution of their problem.

5.2 Medium term: Projection of analytic varieties

I also plan to work on the projection of smooth surfaces and on the apparent contour of projected smooth volumes. Many software packages offer numerical approximations for these problems, which are often of poor quality as soon as the projected surface contains singularities such as the cross-cap or the swallowtail (Figure 5.2). In the case of curves, we have exhibited hidden structures related to transversality theory, which allowed us to plot the projection of singular non-algebraic curves [85]. The next step is to exploit this structure to correctly triangulate singular surfaces resulting from the projection of analytic surfaces. One of the major difficulties of this work comes from the increase in dimension. For curves, the singularities were points, and the neighborhood of a singular point was entirely described by the number of curves connecting to it. For a surface, singularities can be points or curves, and the neighborhood of a cross-

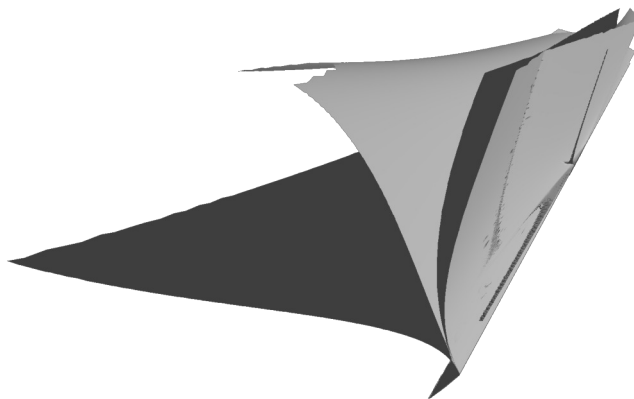


Figure 5.3: Singular surface from a control theory problem, with 5 manifolds appearing to meet on a curve.

cap or a swallowtail is much more difficult to calculate rigorously and efficiently.

5.2.1 Robotic applications

An analytic variety is the set of solutions of a system of analytic equations, for example trigonometric or exponential. This type of equation appears naturally in robotics, where mechanisms are described by geometric constraints in space involving translation and rotation equations. These equations then define a variety whose projection describes the boundaries of a robot's workspace, for example. Moreover, for generic robotic mechanisms, transversality theory tells us that the number of types of singularities appearing in the projection is finite. By coupling this singularity table with smooth surface continuation algorithms, it seems possible to develop numerical algorithms that are both fast and certified to rigorously describe the singularities appearing in these generic mechanisms. This would allow expanding the family of mechanisms for which we could rigorously calculate the boundaries of the workspace.

5.2.2 Applications in control theory

In control theory, a fundamental problem is the design of a stabilizing controller. More precisely, given an unstable dynamic system, in the case where this system is stabilizable, the problem consists of designing a signal that allows stabilizing the system. This problem can be reduced to finding in a polynomial ideal I a polynomial s that does not vanish in the complex unit polydisk [18, 89]. In particular, an open problem is to describe the set of stabilizing controllers of degree 0, i.e., those for which s is a linear combination of the generators of I . The calculation of the boundary of this space can typically be modeled as the projection of an algebraic variety.

Thus, the tools developed for robotic applications can be applied to solve open problems in control theory. However, the ideals appearing here have a different structure from those appearing in robotics. The first step will be to study, using transversality theory, the types of singularities that can appear in varieties arising from control theory problems (Figure 5.3). Then, the last step would be to design a certified numerical algorithm to calculate them efficiently.

5.3 Long term: Piecewise polynomial and rational functions

A longer-term objective is to develop tools to efficiently manipulate piecewise polynomial or rational functions of several variables. These functions appear naturally in approximation theory, and from a certain point of view, can appear as underlying certain data structures in machine learning. However, there are currently few tools to manipulate them efficiently.

5.3.1 Piecewise polynomial approximations

Using the Newton polygon, we saw that it was possible to reduce a univariate polynomial of large degree to several polynomials of small degrees in quasi-linear time [64]. A related question is that of approximating the characteristic polynomial of a matrix with the same type of piecewise polynomial functions. Ideally, for an $n \times n$ matrix, the objective is to calculate such a representation with a number of operations quasi-linear in n^2 . One of the main challenges is to calculate this representation without calculating the characteristic polynomial. In this case, we don't have direct access to the Newton polygon, which requires using other ideas to calculate the adapted partition of the complex plane. A first approach is to study matrices with identically distributed Gaussian random entries.

Moreover, we can also be interested in the piecewise polynomial approximation of a polynomial function in several variables. For the case of two or more variables, recent work proposes an approach reducing the problem to the case of finite fields [51]. However, an approach based on the Newton polygon seems more efficient in practice a priori. Tropical geometry seems to be a suitable tool for generalizing the univariate case to the multivariate case. By combining tropical geometry tools with the new tools developed for the univariate case, I hope to obtain an efficient algorithm allowing to replace a multivariate polynomial of large degree with several polynomials of small degrees. This could prove very useful in addition to other more classic approaches. In the context of solving polynomial systems, for example, methods based on variable elimination increase the degree of polynomials exponentially. We can imagine combining phases of variable elimination with phases of degree reduction to make existing algorithms more efficient.

5.3.2 Efficient algorithms for piecewise functions

One of the main difficulties in manipulating piecewise functions is efficiently managing the constraints partitioning the domain of definition of the function. The simple problem of fast evaluation of such a function thus requires combining fast evaluation algorithms in computer algebra with point location algorithms in an arrangement of semi-algebraic sets in computational geometry. In particular, a fundamental problem is to design a data structure and algorithms to allow more efficient manipulation of this type of function.

While in computer algebra and computational geometry, piecewise functions are defined by explicitly partitioning the domain of definition, there are other ways to represent piecewise functions using the max function, for example. In particular, in the field of machine learning, the multilayer perceptron can be used to represent piecewise linear functions with the ReLU (Rectified Linear Unit) function, which introduces the max operator. Just as there are fast multipoint evaluation algorithms for polynomial functions, it would be interesting to see if it is possible to design fast multipoint evaluation algorithms for functions defined with a combination of algebraic operators and max operators.

5.4 Major fundamental problem: fast multipoint evaluation in the Real RAM model

Multipoint evaluation algorithms are at the heart of my research. The Real RAM model is a model classically used in algorithmic geometry, where the intermediate variables are real numbers whose integer or decimal parts cannot be calculated. In this model, evaluating a univariate polynomial of degree d in d points can be done in quasi-linear time in d . However, for a polynomial f of degree d in n variables with n a constant greater than or equal to 2, there is no algorithm quasi-linear in d^n to evaluate f at d^n points.

Yet, in other calculation models, when we allow ourselves an approximate evaluation, for example, or when the evaluation is in finite fields, recent results have shown that quasi-linear algorithms exist [7–9, 51, 81]. A fundamental idea behind all these approaches is the use of a hash function on the coordinates of the points in order to reduce the problem to the case of points placed on a grid, which is easier to solve. Unfortunately, this idea doesn't work in the Real RAM model. Using binary-splitting approaches, other works show that matrix products can be used to improve complexity in the bivariate case [61, 108]. However, we're still far from obtaining a quasi-linear upper bound, or even a non-trivial lower bound for this problem.

Bibliography

- [1] P. K. Agarwal, B. Aronov, M. van Kreveld, M. Löffler, and R. Silveira. “Computing Correlation Between Piecewise-linear Functions”. In: *SIAM J. Computing* 42.5 (2013), pp. 1867–1887.
- [2] P. K. Agarwal, B. Aronov, M. van Kreveld, M. Löffler, and R. Silveira. “Computing Similarity Between Piecewise-linear Functions”. In: *Proc. 26th Ann. Sympos. Comput. Geometry*. 2010, pp. 375–383.
- [3] Pankaj K. Agarwal, Boris Aronov, Esther Ezra, Matthew J. Katz, and Micha Sharir. *Intersection Queries for Flat Semi-Algebraic Objects in Three Dimensions and Related Problems*. 2023. arXiv: [2203.10241](https://arxiv.org/abs/2203.10241) [cs.CG].
- [4] Boris Aronov and Matthew J. Katz. “Batched Point Location in SINR Diagrams via Algebraic Tools”. In: *ACM Trans. Algorithms* 14.4 (2018). URL: <https://doi.org/10.1145/3209678>.
- [5] Bardenet, Rémi, Lavancier, Frédéric, Mary, Xavier, and Vasseur, Aurélien. “On a few statistical applications of determinantal point processes”. In: *ESAIM: Procs* 60 (2017), pp. 180–202. URL: <https://doi.org/10.1051/proc/201760180>.
- [6] Ruben Becker, Michael Sagraloff, Vikram Sharma, and Chee Yap. “A near-optimal subdivision algorithm for complex root isolation based on the Pellet test and Newton iteration”. In: *Journal of Symbolic Computation* 86 (2018), pp. 51–96. URL: <http://www.sciencedirect.com/science/article/pii/S0747717117300378>.
- [7] V. Bhargava, S. Ghosh, Z. Guo, M. Kumar, and C. Umans. “Fast Multivariate Multipoint Evaluation Over All Finite Fields”. In: *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*. Los Alamitos, CA, USA: IEEE Computer Society, 2022, pp. 221–232. URL: <https://doi.ieeecomputersociety.org/10.1109/FOCS54457.2022.00028>.
- [8] Vishwas Bhargava, Sumanta Ghosh, Zeyu Guo, Mrinal Kumar, and Chris Umans. “Fast Multivariate Multipoint Evaluation over All Finite Fields”. In: *J. ACM* 71.3 (June 2024). URL: <https://doi.org/10.1145/3652025>.

- [9] Vishwas Bhargava, Sumanta Ghosh, Mrinal Kumar, and Chandra Kanta Mohapatra. “Fast, algebraic multivariate multipoint evaluation in small characteristic and applications”. English. In: *Proceedings of the 54th annual ACM SIGACT symposium on theory of computing, STOC '22, Rome, Italy June 20–24, 2022*. New York, NY: Association for Computing Machinery (ACM), 2022, pp. 403–415.
- [10] Dario A. Bini and Giuseppe Fiorentino. “Design, analysis, and implementation of a multiprecision polynomial rootfinder”. In: *Numerical Algorithms* 23.2 (2000), pp. 127–173. URL: <https://doi.org/10.1023/A:1019199917103>.
- [11] Dario A Bini and Leonardo Robol. “Solving secular and polynomial equations: A multiprecision algorithm”. In: *J. of Computational and Applied Mathematics* 272 (2014), pp. 276–292.
- [12] Dario A. Bini and Leonardo Robol. “Solving secular and polynomial equations: A multiprecision algorithm”. In: *Journal of Computational and Applied Mathematics* 272 (2014), pp. 276–292. URL: <https://www.sciencedirect.com/science/article/pii/S037704271300232X>.
- [13] 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”. English. In: *Bull. Am. Math. Soc., New Ser.* 21.1 (1989), pp. 1–46.
- [14] J-D. Boissonnat, S. Kachanovich, and M. Wintraecken. “Tracing Isomanifolds in d in Time Polynomial in d using Coxeter–Freudenthal–Kuhn Triangulations”. In: *SIAM Journal on Computing* (2023). URL: hal.science/hal-04083489.
- [15] Jean-Daniel Boissonnat, Siargey Kachanovich, and Mathijs Wintraecken. “Tracing Isomanifolds in \mathbb{R}^d in Time Polynomial in d Using Coxeter–Freudenthal–Kuhn Triangulations”. In: *37th International Symposium on Computational Geometry (SoCG 2021)*. Ed. by Kevin Buchin and Éric Colin de Verdière. Vol. 189. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021, 17:1–17:16. URL: <https://drops-dev.dagstuhl.de/entities/document/10.4230/LIPIcs.SoCG.2021.17>.
- [16] Jean-Daniel Boissonnat, Siargey Kachanovich, and Mathijs Wintraecken. “Triangulating submanifolds: An elementary and quantified version of Whitney’s method”. In: *Discrete & Computational Geometry* 66.1 (2021), pp. 386–434.
- [17] Alin Bostan, Bruno Salvy, and Éric Schost. “Power Series Composition and Change of Basis”. In: *Proceedings of the Twenty-First International Symposium on Symbolic and Algebraic Computation*. ISSAC '08. Linz/Hagenberg, Austria: Association for Computing Machinery, 2008, 269–276. URL: <https://doi.org/10.1145/1390768.1390806>.

- [18] Yacine Bouzidi, Thomas Cluzeau, Guillaume Moroz, and Alban Quadrat. “Computing effectively stabilizing controllers for a class of nD systems”. In: *The 20th World Congress of the International Federation of Automatic Control*. Vol. 50. 1. Toulouse, France, July 2017, pp. 1847–1852. URL: <https://hal.science/hal-01667161>.
- [19] Yacine Bouzidi, Sylvain Lazard, Guillaume Moroz, Marc Pouget, and Fabrice Rouillier. “Improved algorithm for computing separating linear forms for bivariate systems”. In: **ISSAC - 39th International Symposium on Symbolic and Algebraic Computation**. Kobe, Japan, July 2014. URL: <https://inria.hal.science/hal-00992634>.
- [20] Yacine Bouzidi, Sylvain Lazard, Guillaume Moroz, Marc Pouget, Fabrice Rouillier, and Michael Sagraloff. “Solving bivariate systems using Rational Univariate Representations”. In: **Journal of Complexity** 37 (2016), pp. 34–75. URL: <https://inria.hal.science/hal-01342211>.
- [21] Nicolas Brisebarre, Mioara Joldes, Jean-Michel Muller, Ana-Maria Nanes, and Joris Picot. “Error Analysis of Some Operations Involved in the Cooley-Tukey Fast Fourier Transform”. In: *ACM Trans. Math. Softw.* 46.2 (2020), 11:1–11:27. URL: <https://doi.org/10.1145/3368619>.
- [22] Peter Bürgisser and Felipe Cucker. *Condition: The Geometry of Numerical Algorithms*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. URL: <https://doi.org/10.1007/978-3-642-38896-5>.
- [23] Sergio Cabello. “Computing the Inverse Geodesic Length in Planar Graphs and Graphs of Bounded Treewidth”. In: *ACM Trans. Algorithms* 18.2 (2022). URL: <https://doi.org/10.1145/3501303>.
- [24] Sergio Cabello and Timothy M. Chan. “Computing Shapley Values in the Plane”. In: *Discrete & Computational Geometry* 67.3 (2022), pp. 843–881. URL: <https://doi.org/10.1007/s00454-021-00368-3>.
- [25] Stéphane Caro, Guillaume Moroz, Thibault Gayral, Damien Chablat, and Chao Chen. “Singularity Analysis of a Six-dof Parallel Manipulator using Grassmann-Cayley Algebra and Gröbner Bases”. In: *Brain, Body and Machine. Advances in Intelligent and Soft Computing*. Ed. by Angeles J., Boulet B., Clark J.J., Kövecses J., and Siddiqi K. Vol. 83. Nov. 2010, pp. 341–352. URL: <https://hal.science/hal-00545747>.
- [26] Gilles Chabert and Luc Jaulin. “Contractor programming”. English. In: *Artif. Intell.* 173.11 (2009), pp. 1079–1100.
- [27] Damien Chablat, Ranjan Jha, Fabrice Rouillier, and Guillaume Moroz. “Non-singular assembly mode changing trajectories in the workspace for the 3-RPS parallel robot”. In: *Advances in Robot Kinematics*. Ed. by Oussama Khatib Jadran Lenarčič. Springer, June 2014, pp. 149–159. URL: <https://hal.science/hal-00956325>.

- [28] Damien Chablat, Ranjan Jha, Fabrice Rouillier, and Guillaume Moroz. “Workspace and joint space analysis of the 3-RPS parallel robot”. In: *ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Vol. Volume 5A. Buffalo, United States, Aug. 2014, pp. 1–10. URL: <https://hal.science/hal-01006614>.
- [29] Damien Chablat, Guillaume Moroz, Vigen Arakelian, Sébastien Briot, and Philippe Wenger. “Solution regions in the parameter space of a 3-RRR decoupled robot for a prescribed workspace”. In: *Latest Advances in Robot Kinematics*. Ed. by Husty M. Lenarcic J. Springer, Dordrecht, Jan. 2012, pp. 357–364. URL: <https://hal.science/hal-00687005>.
- [30] Damien Chablat, Guillaume Moroz, Fabrice Rouillier, and Philippe Wenger. “Using Maple to analyse parallel robots”. In: *Maple in Mathematics Education and Research*. Ed. by Jürgen Gerhard and Ilias Kotsireas. Maple in Mathematics Education and Research. Springer, Cham, Feb. 2020, pp. 50–64. URL: <https://hal.science/hal-02289699>.
- [31] Damien Chablat, Guillaume Moroz, and Philippe Wenger. “Uniqueness domains and non singular assembly mode changing trajectories”. In: **International Conference on Robotics and Automation**. Shanghai, China: IEEE, May 2011, pp. 1–6. URL: <https://hal.science/hal-00583302>.
- [32] Damien Chablat, Erika Ottaviano, and Guillaume Moroz. “A comparative study of 4-cable planar manipulators based on cylindrical algebraic decomposition”. In: *Proceedings of the ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Washington, United States, Aug. 2011, pp. 1–10. URL: <https://hal.science/hal-00597924>.
- [33] Chao Chen, Thibault Gayral, Stéphane Caro, Damien Chablat, Guillaume Moroz, and Sajeeva Abeywardena. “A Six-Dof Epicyclic-Parallel Manipulator”. In: **Journal of Mechanisms and Robotics** 4.4 (Apr. 2012). URL: <https://hal.science/hal-00684803>.
- [34] Stephen Chou, Fredrik Kjolstad, and Saman Amarasinghe. “Format Abstraction for Sparse Tensor Algebra Compilers”. In: *Proc. ACM Program. Lang.* 2.OOPSLA (Oct. 2018), 123:1–123:30.
- [35] James W. Cooley and John W. Tukey. “An algorithm for the machine calculation of complex Fourier series”. English. In: *Math. Comput.* 19 (1965), pp. 297–301.
- [36] D. Cox, J. Little, and D. O’Shea. *Ideals, Varieties, and Algorithms*. 3rd. Undergraduate Texts in Mathematics. New York: Springer-Verlag, 2007.
- [37] Laurent Decreusefond and Guillaume Moroz. “Optimal transport between determinantal point processes and application to fast simulation”. In: *Modern Stochastics: Theory and Applications* 8.2 (2021), pp. 209–237. URL: <https://telecom-paris.hal.science/hal-02984323>.

- [38] Jean-Pierre Dedieu. *Points fixes, zéros et la méthode de Newton*. Mathématiques et Applications. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 75–110. URL: <https://doi.org/10.1007/3-540-37660-7>.
- [39] M. Demazure. *Bifurcations and catastrophes: geometry of solutions to nonlinear problems*. Springer, 2000.
- [40] Seny Diatta. “Topologie de courbes algébriques planes et projection de surfaces analytiques réelles.” Theses. Université Assane Seck de Ziguinchor (UASZ), Jan. 2020. URL: <https://hal.science/tel-03101708>.
- [41] Sény Diatta, Guillaume Moroz, and Marc Pouget. “Reliable Computation of the Singularities of the Projection in R^3 of a Generic Surface of R^4 ”. In: *MACIS 2019 - Mathematical Aspects of Computer and Information Sciences*. Gebze-Istanbul, Turkey, Nov. 2019. URL: <https://inria.hal.science/hal-02406758>.
- [42] T. A Driscoll, N. Hale, and L. N. Trefethen. *Chebfun Guide*. Pafnuty Publications, 2014. URL: <http://www.chebfun.org/docs/guide/>.
- [43] Alan Edelman and Eric Kostlan. “How many zeros of a random polynomial are real?” In: *Bulletin of the American Mathematical Society* 32.1 (1995), pp. 1–37.
- [44] Ioannis Z. Emiris and Guillaume Moroz. “The assembly modes of rigid 11-bar linkages”. In: *IFTToMM 2011 World Congress*. IFTToMM - Mexico, Universidad de Guanajuato. Guanajuato, Mexico, June 2011. URL: <https://inria.hal.science/inria-00530327>.
- [45] Jeff Erickson, Ivor van der Hoog, and Tillmann Miltzow. “Smoothing the gap between NP and ER”. English. In: *SIAM J. Comput.* 51.2 (2022), focs20–102–focs20–138.
- [46] Charles M. Fiduccia. “Polynomial Evaluation via the Division Algorithm the Fast Fourier Transform Revisited”. In: *Proceedings of the Fourth Annual ACM Symposium on Theory of Computing*. STOC ’72. Denver, Colorado, USA: Association for Computing Machinery, 1972, 88–93. URL: <https://doi.org/10.1145/800152.804900>.
- [47] Matsusaburô Fujiwara. “Über die obere Schranke des absoluten Betrages der Wurzeln einer algebraischen Gleichung”. In: *Tohoku Mathematical Journal, First Series* 10 (1916), pp. 167–171.
- [48] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. 3rd. Cambridge, U.K.: Cambridge Univ. Press, 2013.
- [49] Joachim von zur Gathen and Jürgen Gerhard. “Fast Algorithms for Taylor Shifts and Certain Difference Equations”. In: *Proceedings of the 1997 International Symposium on Symbolic and Algebraic Computation*. ISSAC ’97. Kihei, Maui, Hawaii, USA: Association for Computing Machinery, 1997, 40–47. URL: <https://doi.org/10.1145/258726.258745>.
- [50] C. F. Gauß. “Theoria interpolationis methodo nova tractata”. In: *Werke*. Vol. Band III. Göttingen: Königliche Gesellschaft der Wissenschaften, 1866, pp. 265–327.

- [51] S. Ghosh, P. Harsha, S. Herdade, M. Kumar, and R. Saptharishi. “Fast Numerical Multivariate Multipoint Evaluation”. In: *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*. Los Alamitos, CA, USA: IEEE Computer Society, 2023, pp. 1426–1439. URL: <https://doi.ieeecomputersociety.org/10.1109/FOCS57990.2023.00088>.
- [52] Xavier Gourdon. *Algorithmique du theoreme fondamental de l'algebre*. Research Report RR-1852. INRIA, 1993. URL: <https://hal.inria.fr/inria-00074820>.
- [53] Stef Graillat. “Accurate simple zeros of polynomials in floating point arithmetic”. In: *Computers & Mathematics with Applications* 56.4 (2008), pp. 1114–1120.
- [54] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. “Array programming with NumPy”. In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. URL: <https://doi.org/10.1038/s41586-020-2649-2>.
- [55] Peter Henrici. *Applied and computational complex analysis, Vol. 1*. Wiley, New York, 1974.
- [56] Nuwan Herath Mudiyansele, Guillaume Moroz, and Marc Pouget. “Fast High-Resolution Drawing of Algebraic Curves”. In: **ISSAC 2022 - International Symposium on Symbolic and Algebraic Computation**. Villeneuve-d’Ascq France, France: ACM, July 2022, pp. 449–458. URL: <https://inria.hal.science/hal-03788409>.
- [57] Nicholas J Higham. *Accuracy and stability of numerical algorithms*. SIAM, 2002.
- [58] Joris van der Hoeven. *Fast composition of numeric power series*. Tech. rep. 2008-09. Orsay, France: Université Paris-Sud, 2008.
- [59] Joris van der Hoeven and Grégoire Lecerf. “Amortized multi-point evaluation of multivariate polynomials”. English. In: *J. Complexity* 74 (2023), p. 17.
- [60] Joris van der Hoeven and Grégoire Lecerf. “Fast amortized multi-point evaluation”. English. In: *J. Complexity* 67 (2021), p. 15.
- [61] Joris van der Hoeven and Grégoire Lecerf. “Faster multi-point evaluation over any field”. Nov. 2024. URL: <https://hal.science/hal-04774026>.
- [62] J. Ben Hough, Manjunath Krishnapur, Yuval Peres, and Bálint Virág. *Zeros of Gaussian analytic functions and determinantal point processes*. English. Vol. 51. Univ. Lect. Ser. Providence, RI: American Mathematical Society (AMS), 2009.

- [63] Xiaohan Huang and Victor Y. Pan. “Fast rectangular matrix multiplication and applications”. English. In: *J. Complexity* 14.2 (1998), pp. 257–299. URL: semanticsscholar.org/paper/ee64ab6a05667113e2f6808dceed08a187387db6.
- [64] Rémi Imbach and Guillaume Moroz. “Fast evaluation and root finding for polynomials with floating-point coefficients”. In: **ISSAC 2023**. Tromsø, Norway, July 2023. URL: <https://inria.hal.science/hal-03980098>.
- [65] Rémi Imbach and Guillaume Moroz. “Fast evaluation and root finding for polynomials with floating-point coefficients”. English. In: *J. Symb. Comput.* 127 (2025), p. 21.
- [66] Rémi Imbach, Guillaume Moroz, and Marc Pouget. “A certified numerical algorithm for the topology of resultant and discriminant curves”. In: *Journal of Symbolic Computation* 80, Part 2 (2016), pp. 285–306. URL: <https://inria.hal.science/hal-01402194>.
- [67] Rémi Imbach, Guillaume Moroz, and Marc Pouget. “Numeric and Certified Isolation of the Singularities of the Projection of a Smooth Space Curve”. In: *Proceedings of the 6th International Conferences on Mathematical Aspects of Computer and Information Sciences*. Berlin, Germany: Springer LNCS, Oct. 2015. URL: <https://inria.hal.science/hal-01239447>.
- [68] Rémi Imbach, Guillaume Moroz, and Marc Pouget. “Reliable Location with Respect to the Projection of a Smooth Space Curve”. In: *Reliable Computing* 26 (2018), pp. 13–55. URL: <https://hal.science/hal-01920444>.
- [69] Rémi Imbach and Victor Y. Pan. “New Practical Advances in Polynomial Root Clustering”. In: *Mathematical Aspects of Computer and Information Sciences*. Ed. by Daniel Slamanig, Elias Tsigaridas, and Zafeirakis Zafeirakopoulos. Cham: Springer International Publishing, 2020, pp. 122–137.
- [70] Rémi Imbach and Victor Y. Pan. “New Progress in Univariate Polynomial Root Finding”. In: *Proceedings of the 45th International Symposium on Symbolic and Algebraic Computation*. ISSAC '20. Kalamata, Greece: Association for Computing Machinery, 2020, 249–256. URL: <https://doi.org/10.1145/3373207.3404063>.
- [71] Rémi Imbach and Victor Y Pan. “Root radii and subdivision for polynomial root-finding”. In: *Computer Algebra in Scientific Computing: 23rd International Workshop, CASC 2021, Sochi, Russia, September 13–17, 2021, Proceedings 23*. Springer. 2021, pp. 136–156.
- [72] Rémi Imbach, Victor Y. Pan, and Chee Yap. “Implementation of a Near-Optimal Complex Root Clustering Algorithm”. In: *Mathematical Software – ICMS 2018*. Ed. by James H. Davenport, Manuel Kauers, George Labahn, and Josef Urban. Cham: Springer International Publishing, 2018, pp. 235–244.

- [73] Luc Jaulin, Michel Kieffer, Olivier Didrit, Eric Walter, Luc Jaulin, Michel Kieffer, Olivier Didrit, and Éric Walter. *Interval analysis*. Springer, 2001.
- [74] Ranjan Jha. “Contribution à l’analyse de performance des robots parallèles”. Theses. Ecole Centrale de Nantes (ECN), July 2016. URL: <https://hal.science/tel-01524856>.
- [75] Ranjan Jha, Damien Chablat, Luc Baron, Fabrice Rouillier, and Guillaume Moroz. “Workspace, Joint space and Singularities of a family of Delta-Like Robot”. In: **Mechanism and Machine Theory** 127 (Sept. 2018), pp. 73–95. URL: <https://hal.science/hal-01796066>.
- [76] Ranjan Jha, Damien Chablat, Fabrice Rouillier, and Guillaume Moroz. “An algebraic method to check the singularity-free paths for parallel robots”. In: *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. ASME. Boston, United States, Aug. 2015. URL: <https://hal.science/hal-01142989>.
- [77] Ranjan Jha, Damien Chablat, Fabrice Rouillier, and Guillaume Moroz. “Influence of the trajectory planning on the accuracy of the orthoglide 5-axis”. In: *ASME International Design Engineering Technical Conference and the Computer and Information in Engineering Conference (IDETC/CIE)*. Charlotte, NC, United States, Aug. 2016. URL: <https://hal.science/hal-01309190>.
- [78] Ranjan Jha, Damien Chablat, Fabrice Rouillier, and Guillaume Moroz. “Workspace and Singularity analysis of a Delta like family robot”. In: *4th IFTOMM International Symposium on Robotics and Mechatronics*. Poitiers, France, June 2015. URL: <https://hal.science/hal-01142465>.
- [79] Anatolii Alekseevich Karatsuba and Yu P Ofman. “Multiplication of many-digital numbers by automatic computers”. In: *Doklady Akademii Nauk*. Vol. 145. 2. Russian Academy of Sciences. 1962, pp. 293–294.
- [80] R. B. Kearfott. *Rigorous global search: continuous problems*. Nonconvex optimization and its applications. Dordrecht, Boston: Kluwer Academic Publishers, 1996. URL: <http://opac.inria.fr/record=b1092397>.
- [81] Kiran S. Kedlaya and Christopher Umans. “Fast polynomial factorization and modular composition”. English. In: *SIAM J. Comput.* 40.6 (2011), pp. 1767–1802.
- [82] Alexander Kobel and Michael Sagraloff. *Fast Approximate Polynomial Multipoint Evaluation and Applications*. 2016. arXiv: [1304.8069](https://arxiv.org/abs/1304.8069) [cs.NA].
- [83] George Krait. “Isolating the Singularities of the Plane Projection of Generic Space Curves and Applications in Robotics”. Theses. Université de Lorraine, May 2021. URL: <https://theses.hal.science/tel-03241794>.
- [84] George Krait. *Isolating_singularities*. 2020. URL: <https://gkrait.github.io/mysite/Proj1.html>.

- [85] George Krait, Sylvain Lazard, Guillaume Moroz, and Marc Pouget. “Certified numerical algorithm for isolating the singularities of the plane projection of generic smooth space curves”. In: **Journal of Computational and Applied Mathematics** 394 (Oct. 2021), p. 113553. URL: <https://inria.hal.science/hal-03161393>.
- [86] Alex Kulesza and Ben Taskar. “Determinantal Point Processes for Machine Learning”. In: *Foundations and Trends® in Machine Learning* 5.2–3 (2012), pp. 123–286. URL: <http://dx.doi.org/10.1561/22000000044>.
- [87] Sven Köhler and Martin Ziegler. “On the stability of fast polynomial arithmetic”. In: *Proceedings of the 8th Conference on Real Numbers and Computers*. RNC8. Santiago de Compostela, Spain, 2008, pp. 147–156. URL: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.154.7840>.
- [88] L. Lin and C. Yap. “Adaptive Isotopic Approximation of Nonsingular Curves: the Parameterizability and Nonlocal Isotopy Approach”. English. In: *Discrete & Computational Geometry* 45.4 (2011), pp. 760–795. URL: <http://dx.doi.org/10.1007/s00454-011-9345-9>.
- [89] Zhiping Lin. “Feedback Stabilizability of MIMO n-D Linear Systems”. In: *Multidimensional Systems and Signal Processing* 9.2 (1998), pp. 149–172. URL: <https://doi.org/10.1023/A:1008230204251>.
- [90] Odile Macchi. “The coincidence approach to stochastic point processes”. In: *Advances in Applied Probability* 7.1 (1975), 83–122. URL: <https://doi.org/10.2307/1425855>.
- [91] Kurt Mahler. “An inequality for the discriminant of a polynomial.” In: *Michigan Mathematical Journal* 11.3 (1964), pp. 257–262. URL: <https://doi.org/10.1307/mmj/1028999140>.
- [92] Gregorio Malajovich and Jorge P Zubelli. “On the geometry of Graeffe iteration”. In: *journal of complexity* 17.3 (2001), pp. 541–573.
- [93] Goran Malić and Ileana Streinu. “Computing Circuit Polynomials in the Algebraic Rigidity Matroid”. In: *SIAM Journal on Applied Algebra and Geometry* 7.2 (2023), pp. 345–385. URL: <https://doi.org/10.1137/21M1437986>.
- [94] Montserrat Manubens, Guillaume Moroz, Damien Chablat, Philippe Wenger, and Fabrice Rouillier. “Cusp Points in the Parameter Space of Degenerate 3-RPR Planar Parallel Manipulators”. In: **Journal of Mechanisms and Robotics** (2012), pp. 1–10. URL: <https://hal.science/hal-00690975>.
- [95] J.C. Mason and D.C. Handscomb. *Chebyshev Polynomials*. CRC Press, 2003.

- [96] Kurt Mehlhorn, Michael Sagraloff, and Pengming Wang. “From approximate factorization to root isolation with application to cylindrical algebraic decomposition”. In: *Journal of Symbolic Computation* 66 (2015), pp. 34–69. URL: <https://www.sciencedirect.com/science/article/pii/S0747717114000200>.
- [97] Guillaume Michel. “Étude d’un robot d’assistance pour la chirurgie endoscopique otologique et sinusienne”. Theses. École centrale de Nantes, July 2021. URL: <https://theses.hal.science/tel-03377490>.
- [98] Guillaume Moroz. “New data structure for univariate polynomial approximation and applications to root isolation, numerical multipoint evaluation, and other problems”. In: **FOCS 2021 - 62nd Annual IEEE Symposium on Foundations of Computer Science**. Denver, United States, Feb. 2022. URL: <https://hal.science/hal-03249123>.
- [99] Guillaume Moroz. “Sparse Tensors and Subdivision Methods for Finding the Zero Set of Polynomial Equations”. In: *Computer Algebra in Scientific Computing*. Ed. by F. Boulier, C. Mou, T. Sadykov, and E. Vorozhtsov. Rennes, France, Sept. 2024. URL: <https://hal.science/hal-04611464>.
- [100] Guillaume Moroz. “Sparse tensors and subdivision methods for finding the zero set of polynomials”. In: *Proceedings of SWIM 2023*. Zenodo, Oct. 2023. URL: <https://doi.org/10.5281/zenodo.8401730>.
- [101] Guillaume Moroz and Boris Aronov. “Computing the Distance between Piecewise-Linear Bivariate Functions”. In: **SODA - Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms - 2012**. Kyoto, Japan: SIAM, Jan. 2012. URL: <https://inria.hal.science/inria-00608255>.
- [102] Guillaume Moroz and Boris Aronov. “Computing the Distance between Piecewise-Linear Bivariate Functions”. In: **ACM Transactions on Algorithms** 12.1 (Feb. 2016), 3:1–3:13. URL: <https://hal.science/hal-01112394>.
- [103] Guillaume Moroz, Damien Chablat, Philippe Wenger, and Fabrice Rouillier. “Cusp points in the parameter space of RPR-2PRR parallel manipulators”. In: *3rd European Conference on Mechanism Science : EUCOMES 2010*. Ed. by D. Pisla, M. Ceccarelli, M. Husty, and B. Corves. Vol. 5. Mechanisms and Machine Science. Cluj-Napoca, Romania: Springer, Sept. 2010, pp. 29–37. URL: <https://hal.science/hal-00545445>.
- [104] Guillaume Moroz, Fabrice Rouillier, Damien Chablat, and Philippe Wenger. “On the determination of cusp points of 3-RPR parallel manipulators”. In: **Mechanism and Machine Theory** 45.11 (Nov. 2010), pp. 1555–1567. URL: <https://hal.science/hal-00545502>.

- [105] Guillaume Moroz and Éric Schost. “A Fast Algorithm for Computing the Truncated Resultant”. In: **ISSAC ’16**. Ed. by Markus Rosenkranz. Proceedings of the ACM on International Symposium on Symbolic and Algebraic Computation. Sergei A. Abramov and Eugene V. Zima. Waterloo, Canada: ACM, July 2016, pp. 341–348. URL: <https://hal.science/hal-01366386>.
- [106] Nuwan Herath Mudiyansele. “Fast high-resolution drawing of algebraic curves and surfaces”. Theses. Université de Lorraine, June 2023. URL: <https://theses.hal.science/tel-04176128>.
- [107] A. Neumaier. *Interval methods for systems of equations*. English. Cambridge University Press, 1990, xvi, 255 p. :
- [108] Michael Nüsken and Martin Ziegler. “Fast Multipoint Evaluation of Bivariate Polynomials”. In: *Algorithms – ESA 2004*. Ed. by Susanne Albers and Tomasz Radzik. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 544–555.
- [109] Alexandre Ostrowski. “Recherches sur la méthode de graeffe et les zéros des polynomes et des séries de laurent”. In: *Acta Mathematica* 72.1 (1940), pp. 157–257.
- [110] Victor Y Pan. “Approximating complex polynomial zeros: modified Weyl’s quadtree construction and improved Newton’s iteration”. In: *J. of Complexity* 16.1 (2000), pp. 213–264.
- [111] Victor Y. Pan. “Simple multivariate polynomial multiplication”. English. In: *J. Symb. Comput.* 18.3 (1994), pp. 183–186.
- [112] Victor Y. Pan. “Univariate Polynomials: Nearly Optimal Algorithms for Numerical Factorization and Root-finding”. In: *Journal of Symbolic Computation* 33.5 (2002), pp. 701–733. URL: <http://www.sciencedirect.com/science/article/pii/S0747717102905316>.
- [113] V.Y. Pan. “New fast algorithms for polynomial interpolation and evaluation on the Chebyshev node set”. In: *Computers & Mathematics with Applications* 35.3 (1998), pp. 125–129. URL: <https://www.sciencedirect.com/science/article/pii/S0898122197002836>.
- [114] Simon Plantinga and Gert Vegter. “Isotopic Approximation of Implicit Curves and Surfaces”. In: *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*. SGP ’04. Nice, France: ACM, 2004, pp. 245–254. URL: <http://doi.acm.org/10.1145/1057432.1057465>.
- [115] Kacper Pluta, Guillaume Moroz, Yukiko Kenmochi, and Pascal Romon. “Quadric Arrangement in Classifying Rigid Motions of a 3D Digital Image”. In: *The 18th International Workshop on Computer Algebra in Scientific Computing*. Vol. 9890. Lecture Notes in Computer Science. Bucharest, Romania: Springer, Sept. 2016, pp. 426–443. URL: <https://hal.science/hal-01334257>.

- [116] Franco P. Preparata and Michael Ian Shamos. *Computational geometry. An introduction*. English. Texts and Monographs in Computer Science. New York etc.: Springer-Verlag. XII, 390 p. DM 148.00 (1985). 1985.
- [117] Patricia Tanessi Quintanar Cortes. “Polyhedral embeddings of the Flat Square Torus”. Theses. Université de Lyon, Dec. 2019. URL: <https://theses.hal.science/tel-02613763>.
- [118] Judit Recknagel. “Topology of planar singular curves resultant of two trivariate polynomials”. MA thesis. Institute for Computer Science, Martin-Luther-University, Halle-Wittenberg, Aug. 2013. URL: <https://inria.hal.science/hal-00927768>.
- [119] Peter Ritzmann. “A fast numerical algorithm for the composition of power series with complex coefficients”. In: *Theoretical Computer Science* 44 (1986), pp. 1–16. URL: <https://www.sciencedirect.com/science/article/pii/0304397586901076>.
- [120] Siegfried M. Rump. “Ten methods to bound multiple roots of polynomials”. In: *Journal of Computational and Applied Mathematics* 156.2 (2003), pp. 403–432. URL: <http://www.sciencedirect.com/science/article/pii/S0377042703003819>.
- [121] Arnold Schönhage. “Asymptotically fast algorithms for the numerical multiplication and division of polynomials with complex coefficients”. In: *Computer Algebra*. Ed. by Jacques Calmet. Berlin, Heidelberg: Springer Berlin Heidelberg, 1982, pp. 3–15.
- [122] Arnold Schönhage. “On the power of random access machines”. In: *Automata, Languages and Programming*. Ed. by Hermann A. Maurer. Berlin, Heidelberg: Springer Berlin Heidelberg, 1979, pp. 520–529.
- [123] Arnold Schönhage. “The fundamental theorem of algebra in terms of computational complexity”. In: *Manuscript. Univ. of Tübingen, Germany* (1982).
- [124] Michael Ian Shamos. *Computational geometry*. Yale University, 1978.
- [125] Shaden Smith and George Karypis. “Tensor-matrix Products with a Compressed Sparse Tensor”. In: *Proceedings of the 5th Workshop on Irregular Applications: Architectures and Algorithms*. IA3 '15. Austin, Texas: ACM, 2015, 5:1–5:7.
- [126] John M. Snyder. “Interval Analysis for Computer Graphics”. In: *SIG-GRAPH Comput. Graph.* 26.2 (July 1992), pp. 121–130. URL: <http://doi.acm.org/10.1145/142920.134024>.
- [127] Lloyd N. Trefethen. *Approximation Theory and Approximation Practice, Extended Edition*. Philadelphia, PA, USA: SIAM-Society for Industrial and Applied Mathematics, 2019.

- [128] Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu, and Renfei Zhou. “New bounds for matrix multiplication: from alpha to omega”. English. In: *Proceedings of the 35th annual ACM-SIAM symposium on discrete algorithms, SODA 2024, Alexandria, Virginia, January 7–10, 2024*. Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM); New York, NY: Association for Computing Machinery (ACM), 2024, pp. 3792–3835.
- [129] Zhenbing Zeng, Lu Yang, and Lydia Dehbi. “On the Number of Congruent Classes of the Tetrahedra Determined by Given Volume, Circumradius and Face Areas”. In: *Proceedings of the 2019 on International Symposium on Symbolic and Algebraic Computation. ISSAC ’19*. Beijing, China: Association for Computing Machinery, 2019, 363–370. URL: <https://doi.org/10.1145/3326229.3326264>.