

# COMPLEXITY ANALYSIS OF POLYNOMIAL ALGORITHMS

FINAL DEGREE PROJECT



**Ignacio Iker Prado Rujas**  
Complutense University of Madrid (UCM)  
Double Degree in Mathematics and Computer Science

July 1<sup>st</sup>, 2016

**Tutors:**

**Juan R. Delgado, José F. Fernando & José M. Gamboa**  
Department of Algebra, Faculty of Mathematics.



## Abstract

In this work we gather three different proofs appearing in [FG], [FGU] and [FU1] of the following result: let  $R$  be a real closed field and consider the open quadrant  $\mathcal{Q} := \{x > 0, y > 0\} \subset R^2$ . Then, there exists a polynomial map  $f : R^2 \rightarrow R^2$  such that  $f(R^2) = \mathcal{Q}$ . In addition, we compare in the last chapter the computational complexity of the three maps  $f$  satisfying the equality  $f(R^2) = \mathcal{Q}$  looking for the most efficient one.

The first proof, which uses some basics in real algebraic geometry such as Sturm's theorem and the Curve selection lemma, requires the aid of a computer to determine the sign of some univariate polynomials in small intervals of the type  $(0, \varepsilon)$ . The second one has a purely algebraic character and to check that the image of  $R^2$  is  $\mathcal{Q}$  is rather easy. However its execution time is high and the image of suitable grids is not thick enough. The third proof involves more sophisticated mathematical arguments, most of them concerning algebraic topology, but it seems to be very efficient from the computational view point.

As announced above we lastly perform a computational comparison in chapter 5, using SageMath, between the three polynomial maps that are used in the previous chapters, on the search for the most efficient and the quickest one.

*Keywords.* Polynomial map, polynomial image, semialgebraic set, open quadrant, complexity, optimal algebraic structure, optimal multiplicative complexity.

## Resumen

En este trabajo reunimos tres demostraciones distintas, que aparecen en [FG], [FGU] y [FU1], del siguiente resultado: dado un cuerpo real cerrado  $R$  el cuadrante abierto  $\mathcal{Q} := \{x > 0, y > 0\} \subset R^2$  es la imagen de una aplicación polinómica  $f : R^2 \rightarrow R^2$ . Además comparamos en el último capítulo la complejidad computacional de las tres aplicaciones cuya imagen es el cuadrante abierto, tratando de encontrar la más eficiente.

La primera demostración, que usa resultados elementales de la geometría algebraica real tales como el teorema de Sturm o el Lema de selección de curvas, requiere de la ayuda de un ordenador para estudiar el signo de ciertos polinomios en una variable en intervalos muy pequeños del tipo  $(0, \varepsilon)$ . La segunda tiene carácter algebraico y la comprobación de que la imagen es el cuadrante abierto es muy sencilla. Sin embargo su coste en tiempo de ejecución es alto y el tamizado de la imagen de un cuadrado centrado en el origen de  $R^2$  no es tan denso como en los restantes casos. Por último la tercera prueba, que es la que usa argumentos matemáticos más sofisticados, entre ellos algunos propios de la topología algebraica, parece ser muy eficiente desde el punto de vista computacional.

La comparación de las complejidades a las que hemos hecho referencia en el párrafo anterior se realiza mediante SageMath, es el contenido del capítulo 5 y tiene por objetivo determinar cual de las tres aplicaciones es la más eficiente y la más rápida en rellenar el cuadrante.

*Palabras clave.* Aplicación polinomial, imagen polinomial, conjunto semialgebraico, cuadrante abierto, complejidad, estructura algebraica óptima, complejidad multiplicativa óptima.



## Contents

<b>1</b>	<b>Introduction to polynomial images of <math>R^n</math></b>	<b>1</b>
1.1	General background and structure of the memoir . . . . .	1
1.1.1	Structure of this memoir . . . . .	2
1.2	Statement of the problem and main results . . . . .	2
1.2.1	Necessary conditions and examples . . . . .	3
1.2.2	Statement of the main results . . . . .	5
1.3	Complementary set of a finite set . . . . .	7
<b>2</b>	<b>The open quadrant <math>\mathcal{Q}</math> problem: First proof</b>	<b>9</b>
2.1	How to find a “potential” map to solve the problem . . . . .	9
2.2	The first proof for the open quadrant problem . . . . .	11
<b>3</b>	<b>A short proof for the open quadrant <math>\mathcal{Q}</math> problem</b>	<b>21</b>
3.1	A new approach . . . . .	21
3.1.1	The new polynomial maps . . . . .	21
3.2	The new proof for the open quadrant problem . . . . .	23
3.2.1	The first lemma . . . . .	23
3.2.2	The second lemma . . . . .	23
3.2.3	The third lemma . . . . .	25
3.2.4	The second proof for the open quadrant problem . . . . .	26
<b>4</b>	<b>A topological proof for the open quadrant <math>\mathcal{Q}</math> problem</b>	<b>27</b>
4.1	A topological perspective . . . . .	27
4.1.1	The new polynomial map . . . . .	27
4.2	Topological tools . . . . .	27
4.2.1	Some definitions . . . . .	27
4.2.2	Auxiliary results . . . . .	29
4.3	The topological proof for the open quadrant problem . . . . .	31
4.3.1	Factorizing $\mathcal{F}$ . . . . .	31
4.3.2	Proof of Claim ( $\diamond$ ) . . . . .	32

---

<b>5 Complexity analysis</b>	<b>41</b>
5.1 Introduction to the computational comparison . . . . .	41
5.2 Complexity of the given polynomial maps . . . . .	42
5.2.1 The map from the first proof . . . . .	42
5.2.2 The map from the short proof . . . . .	42
5.2.3 The map from the topological proof . . . . .	43
5.2.4 Optimal algebraic structure of the given polynomial maps . . . . .	43
5.2.5 Optimal multiplicative complexity of the given polynomial maps . . . . .	43
5.3 Computational analysis . . . . .	45
5.3.1 Uniformly distributed points contained in a square . . . . .	45
5.3.2 Using families of curves . . . . .	48
5.3.3 Randomly distributed points contained in a disc . . . . .	49
5.4 Conclusions and future research . . . . .	51
5.4.1 Comments on further reductions . . . . .	53
<b>Bibliography</b>	<b>55</b>
<b>A Auxiliary definitions and results</b>	<b>57</b>
A.1 Real algebraic geometry basics . . . . .	57
A.2 Root finding algorithms . . . . .	59
A.3 Fundamental group . . . . .	61
<b>B Python code</b>	<b>63</b>
B.1 Laguerre's Method . . . . .	63
B.2 Examples . . . . .	64
<b>C Sage code</b>	<b>67</b>
C.1 Computations related to Chapter 1 . . . . .	67
C.2 Computations related to Chapter 2 . . . . .	67
C.3 Computations related to Chapter 3 . . . . .	76
C.4 Computations related to Chapter 5 . . . . .	79
C.4.1 The set $\mathcal{P}_1$ . . . . .	80
C.4.2 The set $\mathcal{P}_2$ . . . . .	81
C.4.3 The set $\mathcal{P}_3$ . . . . .	82
C.4.4 The set $\mathcal{P}_4$ . . . . .	82
C.4.5 The set $\mathcal{P}_5$ . . . . .	83
C.4.6 The set $\mathcal{P}_6$ . . . . .	84
C.4.7 Expanded vs. Composed . . . . .	85
C.4.8 Using $f_2(\overline{\mathcal{Q}})$ on the third map. . . . .	85
C.4.9 Families of curves composed with $g$ and $f_2$ . . . . .	86
C.4.10 Randomly distributed points contained in a disc . . . . .	88
C.4.11 Families of curves composed with the new maps: $\mathcal{N}_1$ and $\mathcal{N}_2$ . . . . .	90
C.4.12 Expansion and grouping of the given polynomial maps . . . . .	91

## List of Figures

1.1	Plots of the regions defined in Example 1.8 (ii). . . . .	5
2.1	$h(x, y) = (xy - 1)^2 + x^2$ . . . . .	9
2.2	Plots of the polynomial functions $\mathcal{F}, \mathcal{G} : \mathbb{R}^2 \rightarrow \mathbb{R}$ . . . . .	11
2.3	$y^+(x, v)$ and $y^-(x, v)$ for $v := 0.8$ . . . . .	12
2.4	$y^+(x, v)$ and $y^-(x, v)$ for $v := 1$ . . . . .	13
2.5	Numerators of $\gamma_v^+$ and $\gamma_v^-$ for $x := 0$ . . . . .	14
2.6	Notice the value of $\gamma_1^-(x)$ at $x = 0$ . . . . .	15
2.7	Plot of the univariate function $v(x)$ . . . . .	16
2.8	Idea of what we are saying with (♠). Here $v := 0.1$ . . . . .	17
2.9	Plots of $N_v, \delta_v < 0$ and $\Delta(N_v, v), \Delta(\delta_v, v) > 0$ for $v \in (0, 0.28^2)$ . . . . .	18
2.10	Plot of $\{(N_v, v)\}, \{(\delta_v, v)\} \subset D_v$ , for $0 < v < 0.28^2$ . . . . .	19
3.1	The polynomial maps from the second proof: $(\mathcal{H} \circ \mathcal{G} \circ \mathcal{F})(\mathbb{R}^2) = \mathcal{Q}$ . . . . .	22
3.2	Relevant sets for Lemmas 3.1, 3.2 and 3.3. . . . .	24
3.3	Idea of how we represent the set $\mathcal{B}$ in Lemmas 3.2 and 3.3. . . . .	25
4.1	The loop $\beta = \beta_0 * \beta_1$ meets transversally once the disk $\mathbb{D}_A$ . . . . .	30
4.2	The set $M := S \cup D$ and a generator $\alpha$ of $\pi_1(M, x_0)$ from Lemma 4.5. . . . .	31
4.3	The homeomorphism $\zeta$ for $\xi_1(x, y) := \sqrt{B^2 - \min(y^2, B^2)}$ acting on $\mathbb{R}^3$ . . . . .	33
4.4	Idea of the identification in Construction 4.10. . . . .	34
4.5	Left and right views of $\phi_M(\mathcal{R}_M) \subset \mathcal{S}$ . . . . .	35
4.6	Behavior of the map $\mu : \bar{\mathcal{R}}_M \rightarrow \bar{\mathcal{T}}_M$ . . . . .	36
4.7	The boundary map $\partial \tilde{\phi}_M : \partial \tilde{\mathcal{R}}_M \rightarrow \mathbb{R}^3$ meets transversally once $\mathcal{D}_1$ . . . . .	37
4.8	Behavior of the paths $\beta_1$ and $\beta_2$ , surrounding $\bar{\mathcal{R}}_M$ . . . . .	37
4.9	The boundary map $\partial \tilde{\phi}_M : \partial \tilde{\mathcal{R}}_M \rightarrow \mathbb{R}^3$ meets transversally once $\mathcal{D}_2$ . . . . .	38
5.1	The sets $g(\mathcal{P}_4), g(\mathcal{P}_3), f(\mathcal{P}_3)$ and $\mathcal{F}(\mathcal{P}_3)$ . . . . .	46
5.2	Images of the $4 \cdot 10^6$ points from $\mathcal{P}_6$ through $g, f$ and $\mathcal{F}$ . . . . .	47
5.3	Images of families of curves for $g$ and $f_2$ . . . . .	50

5.4	Images of points in $\mathbb{D}_1$ through $g, f$ and $\mathcal{F}$ .	51
5.5	Images of families of curves for $\mathcal{N}_1$ and $\mathcal{N}_2$ .	53

## List of Tables

5.1	The first polynomial map expanded. . . . .	43
5.2	The second polynomial map expanded. . . . .	44
5.3	The third polynomial map expanded. . . . .	44
5.4	Total degree, total number of monomials and non-escalar complexity. . . . .	45
5.5	Execution times of the computation of points contained in a square. . . . .	48
5.6	Comparison between $g$ on its expanded form or as a composition. . . . .	48
5.7	Comparison between $f$ on its expanded form or as a composition. . . . .	48
5.8	Execution times when computing the image of curves through $g$ . . . . .	49
5.9	Execution times when computing the image of curves through $f_2$ . . . . .	49
5.10	Execution times of the computation of points contained in $\mathbb{D}_1$ and $\mathbb{D}_{100}$ . .	49
5.11	Properties of the new maps. . . . .	52
5.12	Execution times when computing the image of curves through $\mathcal{N}_1$ . . . . .	52
5.13	Execution times when computing the image of curves through $\mathcal{N}_2$ . . . . .	52



# 1

## Introduction to polynomial images of $R^n$

### 1.1 General background and structure of the memoir

Although it is usually said that the first work in Real Geometry is due to Harnack [H], who obtained an upper bound for the number of connected components of a non-singular real algebraic curve in terms of its genus, modern Real Algebraic Geometry was born with Tarski's article [T], where it is proved that the image of a semialgebraic set under a polynomial map is a semialgebraic set. In the last decade J.F. Fernando, J.M. Gamboa and C. Ueno have intensively studied what might be called the ‘inverse problem’. In the 1990 *Oberwolfach Reelle algebraische Geometrie* week [G] Gamboa proposed to characterize the subsets of  $R^m$  that are polynomial images of some  $R^n$ .

Open semialgebraic sets deserve a special attention in connection with the real Jacobian Conjecture [P]. In addition it can be easily seen that they constitute a not at all trivial challenge. Observe for instance that the closed half-line  $[0, +\infty)$  is the image of  $R$  under the polynomial map  $f : R \rightarrow R, x \mapsto x^2$  but the open half-line  $(0, +\infty)$  is not the image of any polynomial map  $R \rightarrow R$ , although it is the image of

$$f : R^2 \rightarrow R, (x, y) \mapsto (xy - 1)^2 + y^2.$$

As open sets with piecewise linear boundary seem to be friendly it is natural to begin by studying them. Indeed it is easily seen that the open upper half-plane  $\mathcal{H} := \{y > 0\} \subset R^2$  is the image of the polynomial map

$$R^2 \rightarrow R^2, (x, y) \mapsto (y(xy - 1), (xy - 1)^2 + x^2).$$

These examples explain why in [G] the so called ‘open quadrant problem’ was proposed, that is, to determine whether the open quadrant  $\mathcal{Q} := \{x > 0, y > 0\} \subset R^2$  is a polynomial image of  $R^2$ .

This problem stimulated the interest of many specialists in the field. However, only after twelve years a first solution was found in [FG] and presented by the first author in the 2002 *Oberwolfach Reelle algebraische Geometrie* week [F].

The open quadrant problem was the germ of a more systematic study of ‘Polynomial images of Euclidean spaces’ developed during the last decade, and was the topic of the Ph.D. Thesis of C. Ueno [U] supervised by J.F. Fernando. It is worthwhile mentioning

that the work on polynomial and regular images constitutes the germ of the recent proof by Fernando, [F1] of the so called “Shiota’s conjecture”. The result is the following. *A subset  $S \subset \mathbb{R}^n$  is the image of a Nash map  $f : \mathbb{R}^m \rightarrow \mathbb{R}^n$  if and only if  $S$  is semialgebraic, pure dimensional of dimension  $d \leq m$  and there exists an analytic path  $\alpha : [0, 1] \rightarrow S$  whose image meets all the connected components of the set of regular points of  $S$ .*

Coming back to polynomial and regular maps of euclidean spaces a lot of work has been done after the discovery that  $\mathcal{Q}$  is a polynomial image of  $\mathbb{R}^2$ , with two main objectives:

- ★ Finding obstructions to be an either polynomial or regular image of an euclidean space.
- ★ Proving (constructively) that large families of semialgebraic sets with piecewise linear boundary (convex polyhedra, their interiors, complements and the interiors of their complements) are either polynomial or regular images of some Euclidean space. The positive answer to the open quadrant problem has been a recurrent starting point for this approach.

This is the main reason to search the “best” polynomial map  $\mathbb{R}^2 \rightarrow \mathbb{R}^2$  whose image is  $\mathcal{Q}$ . As easy such a map is, as easy they would be those polynomial or regular maps  $\mathbb{R}^n \rightarrow \mathbb{R}^m$  whose image is a prescribed convex polyhedron, its interior or their complements.

### 1.1.1 Structure of this memoir

This memoir is organized as follows. In this first chapter we collect some examples of polynomial images of euclidean spaces and present some obstructions to be such an image. In Chapters 2, 3 and 4 we present very detailed proofs of the main results in, respectively, [FG], [FU1] and [FGU]. The first one has a strong computational character, the second one is purely algebraic whereas the third one involves algebraic topology arguments. It must be pointed out that concerning the proof in Chapter 2 we have complemented the use of Sturm sequences in the original work [FG] by the use of Laguerre’s algorithm to approximate the roots of an univariate polynomial with complex coefficients.

The most original part of the work is Chapter 5, where we compare from distinct view points the complexity of the maps presented in the precedent chapters having  $\mathcal{Q}$  as image. We have found two ways to compare the effectiveness of two such maps. Namely, the execution time needed to fill the quadrant, and the density in  $\mathcal{Q}$  of the image of a thick enough grid. To perform the corresponding computations we have used Sage. To finish we present without proof simpler maps  $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  that, maybe, satisfy the equality  $f(\mathbb{R}^2) = \mathcal{Q}$ . If this were so one of them would probably be the map with smallest *total degree* among those having the open quadrant as image.

## 1.2 Statement of the problem and main results

**Definition 1.1.** Given a **real closed field**  $R$  and positive integers  $m, n$  it is said that  $f := (f_1, \dots, f_n) : R^m \rightarrow R^n$  is a *polynomial map* if  $f_i \in R[\mathbf{x}_1, \dots, \mathbf{x}_m]$  for  $i = 1, \dots, n$ .

We refer the reader to Appendix A to recall several notions and results appearing there in order to lighten the presentation and soften the reading. For example the concept of

real closed field is introduced in Appendix A; now let us simply say that the field  $\mathbb{R}$  of real numbers is an example (the most important one) of real closed field.

A very famous theorem by Tarski and Seidenberg states the following:

**Theorem 1.2 (Tarski-Seidenberg)**

The image of every polynomial map  $f : R^m \rightarrow R^n$  is a **semialgebraic subset** of  $R^n$ .

As we have commented above, in this work we study a sort of converse of this statement proposed by J.M. Gamboa [G] in the 1990 *Oberwolfach Reelle algebraische Geometrie* week:

**Problem.** Characterize the semialgebraic subsets of  $R^n$  that are polynomial images of some  $R^m$ .

We need to mention to which topology we refer to when we talk about closures, boundaries, etc. More specifically, the *exterior boundary* of a subset  $S \subset R^n$  is  $\delta S := \overline{S} \setminus S$ , with  $\overline{S}$  being the *closure* of  $S$  in the usual topology of  $R^n$ . In addition we will denote by  $\overline{S}^{\text{zar}}$  the closure of  $S$  with respect to the **Zariski topology** of  $R^n$ . We will say that a semialgebraic subset  $A \subset R^n$  is *Zariski-irreducible* if its Zariski closure  $\overline{A}^{\text{zar}}$  is an irreducible algebraic set.

### 1.2.1 Necessary conditions and examples

To begin working on this idea, we provide some necessary conditions for a set  $S \subset R^n$  to be a polynomial image of  $R^m$ .

If  $m = n = 1$ , that is, for a polynomial function  $f : R \rightarrow R$ , its image  $f(R)$  is either a *singleton*, that is, a set with a unique point if the function  $f$  is constant, or an unbounded closed interval. For example if  $f(x) = x^2$  we have  $f(R) = [0, +\infty)$ , and  $f(R) = R$  if  $f$  is a polynomial of odd degree.

In the general case, by **Tarski-Seidenberg**,  $S$  must be a semialgebraic set and, as  $R^n$  is semialgebraically connected,  $S$  is semialgebraically connected too. In addition, by the identity principle for polynomials,  $S$  is Zariski-irreducible and **pure dimensional**.

But to be a polynomial image of some  $R^m$  is a restrictive condition, and there are more constraints than those quoted above.

**Definition 1.3.** A polynomial map  $f : R^m \rightarrow R^n$  is said to be *semialgebraically proper at a point*  $p \in R^n$  if there exists an open neighborhood  $K$  of  $p$  such that the restriction  $f|_{f^{-1}(K)} : f^{-1}(K) \rightarrow K$  is a **semialgebraically proper map**.

**Definition 1.4.** A *parametric semiline* of  $R^n$  is the image of  $R$  under a non-constant polynomial map  $R \rightarrow R^n$ .

It is clear that every parametric semiline is semialgebraically closed, since every polynomial map from  $R$  to  $R^n$  is semialgebraically proper. Let  $\mathcal{S}_f$  denote the set of points  $p \in R^n$  at which  $f$  is **not** semialgebraically proper.

**Theorem 1.5 (Jelonek)**

Let  $f : R^2 \rightarrow R^2$  be a **dominant** polynomial map. Then  $\mathcal{S}_f$  is a finite union of parametric semilines.

With these ideas in mind, we present in the following proposition some obstructions for a semialgebraic set to be a polynomial image of  $R^n$ .

**Proposition 1.6.** *Let  $f : R^m \rightarrow R^n$  be a polynomial map and  $S := f(R^m)$ .*

(1)  $\delta S \subset \mathcal{S}_f$ .

*Proof.* Suppose  $p \in \delta S \setminus \mathcal{S}_f$ . Since  $p \notin \mathcal{S}_f$ , there exists an open neighborhood  $K$  of  $p$  such that the restriction  $f|_{f^{-1}(K)} : f^{-1}(K) \rightarrow K$  of  $f$  is proper. Thus its image  $K \cap S$  is a closed subset of  $K$ . Hence,  $p \in K \cap \overline{S} = K \cap (\overline{K \cap S}) = K \cap S$ , which yields in a contradiction.

(2) *Let  $m = n = 2$  and  $\Gamma$  be a 1-dimensional irreducible component of  $\overline{\delta S}^{\text{zar}}$ . Then  $\Gamma$  is the Zariski closure of a parametric semiline of  $R^2$ .*

*Proof.* As  $f$  is a dominant map,  $\mathcal{S}_f$  is, by Theorem 1.5, a finite union of parametric semilines, say  $M_1, \dots, M_s$  in  $R^2$ . Using (1) we get:  $\Gamma \subset \overline{\delta S}^{\text{zar}} \subset \overline{\mathcal{S}_f}^{\text{zar}} = \bigcup_{i=1}^s \overline{M_i}^{\text{zar}}$ . Lastly, using that both  $\Gamma$  and the  $\overline{M_i}^{\text{zar}}$ 's are irreducible,  $\Gamma = \overline{M_i}^{\text{zar}}$  for some index  $i = 1, \dots, s$ .

(3) *Let  $p : R^n \rightarrow R$  be a polynomial map which is non-constant on  $S$ . Then  $p(S)$  is unbounded.*

*Proof.* For each  $a \in R^m$  let  $\varphi_a : R \rightarrow R$  defined as  $\varphi_a(t) := p(f(ta))$ . Then  $p(S)$  would contain the image  $\varphi_a(R)$  for all  $a \in R^m$ . Suppose now that  $\varphi_a(R)$  is bounded. Then  $\varphi_a(R)$  would be a point  $r_a$ , and given two points  $a, b \in R^m$  we would have

$$\varphi_a(1) = p(f(a)) = r_a = \varphi_a(0) = \varphi_b(0) = r_b = p(f(b)) = \varphi_b(1).$$

Then  $p$  would be constant on  $S$ , which is a contradiction.

**Corollary 1.7.** *Because of statement (3) in Proposition 1.6, all linear projections of a polynomial image  $S = f(R^m)$  are either a point or unbounded. Thus,  $S$  is also unbounded or a point.*

### Examples 1.8.

- (i) The exterior of the closed unit disc  $S := \{u^2 + v^2 > 1\} \subset R^2$  is **not** a polynomial image of  $R^2$ . This is so because the only irreducible component of  $\overline{\delta S}^{\text{zar}}$  is the circle  $\{u^2 + v^2 = 1\} \subset R^2$  and this set is not a parametric semiline because it is bounded.
- (ii) Let  $S_1 := \{uv < 1\} \subset R^2$  and  $S_2 := \{uv > 1, u > 0\} \subset R^2$  (see Figure 1.1). They both **are not** polynomial images of  $R^2$  since the Zariski closure of their exterior boundaries  $\overline{\delta S_1}^{\text{zar}} = \overline{\delta S_2}^{\text{zar}}$  is the hyperbola  $\{uv = 1\} \subset R^2$ , which is not a parametric semiline.
- (iii) Let  $S := R^2 \setminus \{(0, 0)\}$  be the punctured plane. Then  $S$  is the image of the polynomial map  $(x, y) \mapsto (xy - 1, (xy - 1)x^2 - y)$ .
- (iv) The open upper half-plane  $\mathcal{H} := \{v > 0\} \subset R^2$  is the image of the polynomial map

$$R^2 \rightarrow R^2, (x, y) \mapsto (y(xy - 1), (xy - 1)^2 + x^2).$$

This implies that every open half-plane is a polynomial image of  $R^2$ . This is probably the simplest polynomial map whose image is  $\mathcal{H}$ .

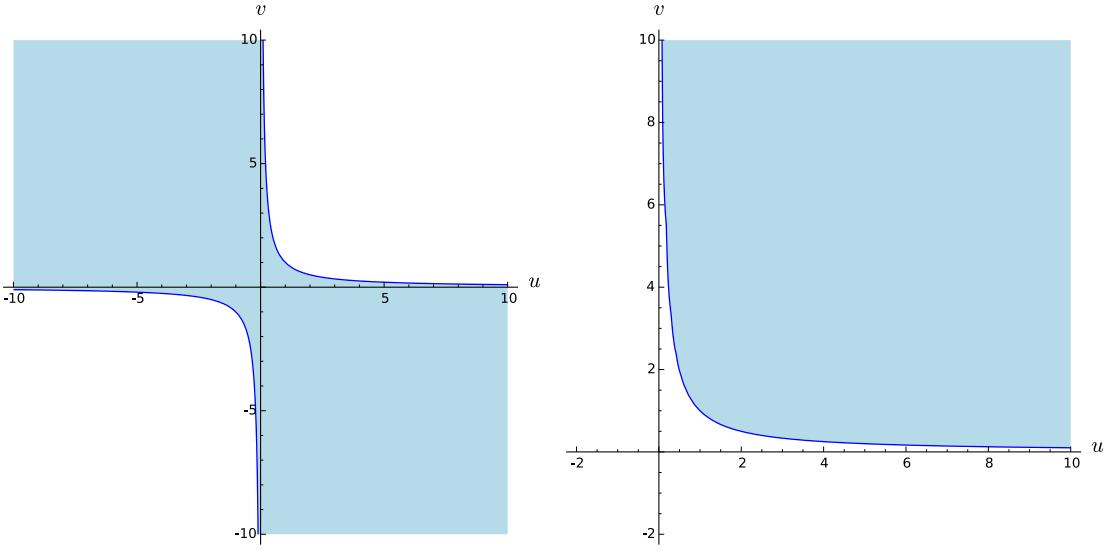


Figure 1.1: Plots of the regions defined in Example 1.8 (ii).

### 1.2.2 Statement of the main results

The main results of this chapter are generalizations of items (iii) and (iv) from Examples 1.8.

#### Theorem 1.9

Let  $n \geq 2$ . For every finite set  $F \subset R^n$ , the semialgebraic set  $R^n \setminus F$  is a polynomial image of  $R^n$ .

#### Theorem 1.10

Let  $n \geq 2$ . Given independent linear forms  $h_1, \dots, h_r$  of  $R^n$ , the open semialgebraic set  $\{h_1 > 0, \dots, h_r > 0\}$  is a polynomial image of  $R^n$ .

Before the paper [FG] was published, the exterior boundary of all open sets that were known to be polynomial images of  $R^2$  was Zariski-irreducible, and all of them were deformations of  $\mathcal{H}$ . J.M. Gamboa outlined the problem of finding if the open quadrant  $\mathcal{Q} := \{x > 0, y > 0\} \subset R^2$  is a polynomial image of  $R^2$  or not, since its exterior boundary is Zariski-reducible. The solution of this problem is a key particular case of the content of Theorem 1.10. Before that, the closest known approach to look for a solution of this problem was the transformation

$$\psi : R^2 \rightarrow R^2, (x, y) \mapsto (x^4y^2, x^2y^4)$$

whose image is  $\mathcal{Q} \cup \{(0, 0)\}$ . The answer to the first intriguing problem in this field was given in the following theorem:

**Theorem 1.11**

The open quadrant  $\mathcal{Q}$  is a polynomial image of  $R^2$ .

**Remark 1.12.** The first proof of Theorem 1.11 consists of two parts:

- ★ Choosing a “good” candidate to be a polynomial map whose image is close enough to  $\mathcal{Q}$ , and giving the reasons behind this choice (see section 2.1).
- ★ Checking that the image of the map is  $\mathcal{Q}$  indeed. After some arguments, this can be reduced to prove the non-existence of real roots of certain univariate polynomials on certain intervals, and to compare some rational functions on those intervals. In order to do this, we use symbolic computations with tools like Sage and Maple. Because of the high degree of the involved polynomials, the actual checking of the non-existence of roots is done with a Maple package that performs **Sturm algorithm** and a Python programme that implements **Laguerre’s method**.

Let us see how Theorem 1.10 follows from Theorem 1.11.

[Proof of Theorem 1.10] After a linear change of coordinates we can suppose that  $h_1 := \mathbf{x}_1, \dots, h_r := \mathbf{x}_r$ , so we only have to prove that for every pair of positive integers  $r \leq n$  the semialgebraic set  $\{x_1 > 0, \dots, x_r > 0\} \subset R^n$  is a polynomial image of  $R^r$ . This is reduced to prove the following two steps:

- ★  $\mathcal{H} := \{x_1 > 0\} \subset R^2$  and  $\mathcal{Q} := \{x_1 > 0, x_2 > 0\} \subset R^2$  are polynomial images of  $R^2$  by Example 1.8 (iv) and Theorem 1.11.
- ★ This implies that  $\mathcal{O} := \{x_1 > 0, x_2 > 0, x_3 > 0\} \subset R^3$  is a polynomial image of  $R^3$ . Indeed, let  $f : R^2 \rightarrow R^2$ ,  $(x_1, x_2) \mapsto (f_1(x_1, x_2), f_2(x_1, x_2))$  and  $g : R^2 \rightarrow R^2$ ,  $(x_1, x_2) \mapsto (g_1(x_1, x_2), g_2(x_1, x_2))$  be polynomial maps whose respective images are  $\mathcal{H}$  and  $\mathcal{Q}$ . Let us define:

$$\begin{aligned} (f, \text{id}_R) : R^3 &= R^2 \times R \rightarrow R^3 = R^2 \times R \\ (x_1, x_2, x_3) &\mapsto (f_1(x_1, x_2), f_2(x_1, x_2), x_3) \\ (\text{id}_R, g) : R^3 &= R \times R^2 \rightarrow R^3 = R \times R^2 \\ (x_1, x_2, x_3) &\mapsto (x_1, g_1(x_2, x_3), g_2(x_2, x_3)). \end{aligned}$$

Then,  $\mathcal{O}$  is the image of the polynomial map

$$\begin{aligned} H := (\text{id}_R, g) \circ (f, \text{id}_R) : R^3 &\rightarrow R^3 \\ (x_1, x_2, x_3) &\mapsto (f_1(x_1, x_2), g_1(f_2(x_1, x_2), x_3), g_2(f_2(x_1, x_2), x_3)). \end{aligned}$$

Once the case  $n = 3$  is solved the case of arbitrary  $n$  follows straightforwardly, by taking  $g \times \cdots \times g : R^n \rightarrow R^n$  when  $n = 2k$  and  $H \times g \times \cdots \times g : R^n \rightarrow R^n$  when  $n = 2k - 1$ .  $\square$

The original proofs of Theorems 1.9 and 1.11 are written for  $R := \mathbb{R}$ . As for both theorems explicit polynomial maps are given, the results can be extended to arbitrary real closed field  $R$  by the **Transfer Principle**.

### 1.3 Complementary set of a finite set

We proceed to prove Theorem 1.9:

[Proof of Theorem 1.9] Let  $F := \{p_1, \dots, p_k\}$ . Let us see that it suffices to prove the result for points of the form  $p_j := (a_j, \vec{0}) \in \mathbb{R} \times \mathbb{R}^{n-1}$ . After a linear change of coordinates we can assume that the first coordinates of the given points are pairwise distinct.

In other words, we denote  $p_j := (a_{1j}, \dots, a_{nj})$  and we may suppose that  $a_{1j} \neq a_{1\ell}$  when  $j \neq \ell$ . Then, there exists a polynomial  $P_1 \in \mathbb{R}[\mathbf{t}]$  such that  $P_1(a_{1j}) = a_{nj}$ , with  $j = 1, \dots, n$ , and denoting  $x' := (x_1, \dots, x_{n-1})$ , we define the polynomial map

$$h_1 : \mathbb{R}^n \rightarrow \mathbb{R}^n, (x', x_n) \mapsto (x', x_n + P_1(x_1)).$$

Indeed  $h_1$  is bijective. Note first that every point of  $\mathbb{R}^n$  has a preimage in  $\mathbb{R}^n$ , namely if  $x := (x_1, \dots, x_n)$ , then  $z := (x', x_n - P_1(x_1))$  satisfies  $h_1(z) = x$  and  $h_1$  is onto. As for being injective, let  $x, y \in \mathbb{R}^n$  such that

$$h_1(x) = (x_1, \dots, x_n + P_1(x_1)) = (y_1, \dots, y_n + P_1(y_1)) = h_1(y).$$

Then  $x_i = y_i$  for  $i = 1, \dots, n-1$ . Also  $x_n + P_1(x_1) = y_n + P_1(y_1)$  and  $P_1(x_1) = P_1(y_1)$  because  $x_1 = y_1$ . Therefore  $x_n = y_n$  and  $x = y$ .

Now, for  $p'_j := (a_{1j}, \dots, a_{(n-1)j}, 0)$  we have  $h_1(p'_j) = p_j$ . Analogously, there exists  $P_2 \in \mathbb{R}[\mathbf{t}]$  such that  $P_2(a_{1j}) = a_{(n-1)j}$ , and define the polynomial bijection

$$h_2 : \mathbb{R}^n \rightarrow \mathbb{R}^n, (x'', x_{n-1}, x_n) \mapsto (x'', x_{n-1} + P_2(x_1), x_n),$$

where  $x'' := (x_1, \dots, x_{n-2})$ . Then  $h_2(p''_j) = p'_j$  for  $p''_j = (a_{1j}, \dots, a_{(n-2)j}, 0, 0)$ . In this way the polynomial bijection  $h_1 \circ h_2 : \mathbb{R}^n \rightarrow \mathbb{R}^n$  satisfies

$$(h_1 \circ h_2)(p''_j) = h_1(h_2(p''_j)) = h_1(p'_j) = p_j,$$

and we can inductively construct a polynomial bijection  $h : \mathbb{R}^n \rightarrow \mathbb{R}^n$  such that  $h(q_j) = p_j$  for  $q_j := (a_{1j}, \vec{0}) \in \mathbb{R} \times \mathbb{R}^{n-1}$ . Now let  $G := \{q_1, \dots, q_k\}$  and suppose that there exists a polynomial map  $g : \mathbb{R}^n \rightarrow \mathbb{R}^n$  such that  $g(\mathbb{R}^n) = \mathbb{R}^n \setminus G$ . Then  $(h \circ g)(\mathbb{R}^n) = \mathbb{R}^n \setminus F$ , which concludes the first part of the proof. Thus in what follows we suppose that  $p_j := (a_j, \vec{0})$ .

We claim that the image of the polynomial map  $f := (f_1, \dots, f_n)$  defined as:

$$f(\mathbf{x}) := (\mathbf{x}_1 \mathbf{x}_2 - r + a_1, \mathbf{x}_1^4 \rho(\mathbf{x}) + \mathbf{x}_1^2 \sigma(\mathbf{x}) + \mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_n) \quad \text{for } \mathbf{x} := (\mathbf{x}_1, \dots, \mathbf{x}_n) \in \mathbb{R}^n$$

is  $\mathbb{R}^n \setminus F$ , where  $r$  is an integer such that  $r \neq a_1 - a_j$  for  $j = 1, \dots, k$ ,

$$\sigma(\mathbf{x}) := \sum_{j=3}^n \mathbf{x}_j^2 \quad \text{and} \quad \rho(\mathbf{x}) := \prod_{j=1}^k (\mathbf{x}_1 \mathbf{x}_2 - r + a_1 - a_j).$$

First, suppose that there exists  $b := (b_1, \dots, b_n) \in R^n$  with  $f(b) = p_\ell$  for some  $\ell = 1, \dots, k$ . Then  $f_1(b) = b_1 b_2 - r + a_1 = a_\ell$ . Thus the  $\ell^{th}$ -factor of the polynomial  $\rho$  evaluated at  $x := b$  is

$$b_1 b_2 - r + a_1 - a_\ell = a_\ell - a_\ell = 0,$$

and  $\rho(b) = 0$ . In addition, the equality  $f(b) = p_\ell = (a_\ell, \vec{0})$  implies that  $f_i(b) = 0$  for  $i = 2, \dots, n$ . In particular, as  $f_i \equiv \text{id}$  for  $i = 3, \dots, n$  we get  $b_i = 0$  for  $i = 3, \dots, n$ .

Consequently  $\sigma(b) = 0$ . Since  $\sigma(b) = \rho(b) = 0$  we get  $b_2 = f_2(b) = 0$ , so  $b_2 = 0$  and  $a_\ell = f_1(b) = a_1 - r$ , that is,  $r = a_1 - a_\ell$ , which is a contradiction. So  $\text{im}(f) \subset \mathbb{R}^n \setminus F$ . Conversely, let  $u := (u_1, \dots, u_n) \in \mathbb{R}^n \setminus F$ . We must prove that the system of polynomial equations:

$$\begin{cases} f_1(\mathbf{x}) = \mathbf{x}_1 \mathbf{x}_2 - r + a_1 = u_1 \\ f_2(\mathbf{x}) = \mathbf{x}_1^4 \rho(\mathbf{x}) + \mathbf{x}_1^2 \sigma(\mathbf{x}) + \mathbf{x}_2 = u_2 \\ f_j(\mathbf{x}) = \mathbf{x}_j = u_j, \quad j \geq 3 \end{cases}$$

has a solution.

(i) If  $u_1 = a_1 - r$  then  $f(0, u_2, \dots, u_n) = u$ .

(ii) If  $u_1 \neq a_1 - r$  we use the first equation to substitute

$$\mathbf{x}_2 = \frac{u_1 - a_1 + r}{\mathbf{x}_1} \quad \text{and} \quad \mathbf{x}_j = u_j \quad \text{for } j \geq 3.$$

Next, we expand  $f_2(\mathbf{x})$ :

$$\mathbf{x}_1^4 \rho(\mathbf{x}) + \mathbf{x}_1^2 \sigma(\mathbf{x}) - u_2 = -\mathbf{x}_2 = -\frac{u_1 - a_1 + r}{\mathbf{x}_1},$$

and multiplying by  $\mathbf{x}_1$  we get

$$\mathbf{x}_1^5 \rho(\mathbf{x}) + \mathbf{x}_1^3 \sigma(\mathbf{x}) - u_2 \mathbf{x}_1 + (u_1 - a_1 + r) = 0. \quad (1.1)$$

Then,  $\rho(\mathbf{x}) = \prod_{j=1}^k (u_1 - a_j)$  and  $\sigma(\mathbf{x}) = \sigma(u)$ . Now it is clear that each solution  $x_1$  of equation (1.1) is a nonzero root of the polynomial:

$$Q(t) = \left( \prod_{j=1}^k (u_1 - a_j) \right) t^5 + \sigma(u) t^3 - u_2 t + (r - a_1 + u_1),$$

which has odd degree, and so it has a real root, unless

$$\prod_{j=1}^k (u_1 - a_j) = \sigma(u) = u_2 = 0.$$

If this were the case, then  $u_1 = a_j$  for some  $j = 1, \dots, k$  and  $u_2 = u_3 = \dots = u_k = 0$ . This is not possible because  $u \notin F$ . Thus,  $Q(t)$  has a real root, say  $b_1$ , and in fact  $b_1 \neq 0$  because  $Q(0) = r - a_1 + u_1 \neq 0$ . Finally,

$$f \left( b_1, \frac{u_1 - a_1 + r}{b_1}, u_3, \dots, u_n \right) = u,$$

as required. □

# 2

## The open quadrant $\mathcal{Q}$ problem: First proof

### 2.1 How to find a “potential” map to solve the problem

It is worthwhile mentioning that though the open interval  $I := (0, +\infty)$  is the image of  $\mathbb{R}^2$  under the polynomial map  $h : \mathbb{R}^2 \rightarrow \mathbb{R}$ ,  $(x, y) \mapsto (xy - 1)^2 + x^2$ , see Figure 2.1, the interval  $I$  is not a polynomial image of  $\mathbb{R}$  because polynomial images of the real line are closed subsets of  $\mathbb{R}$ .

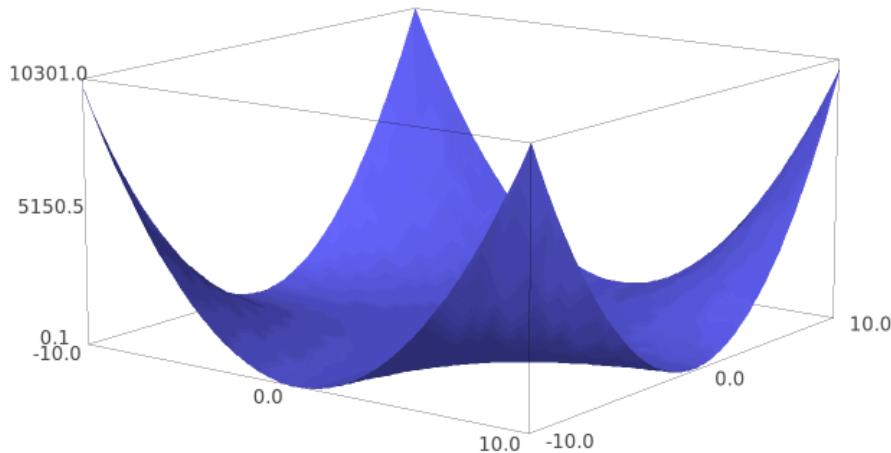


Figure 2.1:  $h(x, y) = (xy - 1)^2 + x^2$ .

However, although  $h(\mathbb{R}^2) = I$ , the polynomial  $h$  does not help to obtain  $\mathcal{Q}$  at all:

**Remark 2.1.** There is no polynomial map  $f := (P_1, P_2) : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  satisfying  $f(\mathbb{R}^2) = \mathcal{Q}$  and  $P_1(x, y) = (xy - 1)^2 + x^2$ .

The proof of this remark relies on a suitable use of the Curve Selection Lemma ([ABR, VIII.2.6]) to approach a point  $(\lambda^2, 0) \in \overline{\mathcal{Q}}$  with  $\lambda > 0$ , to get a contradiction.

On the topic of finding a polynomial map  $\Phi : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  that satisfies  $\Phi(\mathbb{R}^2) = \mathcal{Q}$ , “a major difficulty” is the following:

*The closure of its image must contain the positive half-axes.*

(♣)

**Remark 2.2.** Using Theorem 1.9, we just need to find a polynomial map

$$\mathcal{P} = (\mathcal{F}, \mathcal{G}) : \mathbb{R}^2 \rightarrow \mathbb{R}^2$$

such that  $\mathcal{P}(\mathbb{R}^2)$  is the disjoint union of  $\mathcal{Q}$  and a set with finite preimage, say  $\mathcal{P}(\mathbb{R}^2) = \mathcal{Q} \sqcup F$  with  $\mathcal{P}^{-1}(F)$  a finite set. Suppose for a while this is proved. Then, by Theorem 1.9, there exists a polynomial map  $\varphi : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  such that  $\varphi(\mathbb{R}^2) = \mathbb{R}^2 \setminus \mathcal{P}^{-1}(F)$  and the polynomial map  $\Phi := \mathcal{P} \circ \varphi$  satisfies

$$\Phi(\mathbb{R}^2) = \mathcal{P}(\varphi(\mathbb{R}^2)) = \mathcal{P}(\mathbb{R}^2 \setminus \mathcal{P}^{-1}(F)) = \mathcal{P}(\mathbb{R}^2) \setminus F = (\mathcal{Q} \sqcup F) \setminus F = \mathcal{Q}.$$

We are going to define a map  $\mathcal{P} := (\mathcal{F}, \mathcal{G})$  that accomplish this task, with the set  $F$  being  $F := \{(-1, 0), (0, -1)\}$ . If we were able to find such map  $\mathcal{P}$ , then condition (♣) will immediately be satisfied.

Suppose for a while that such a map  $\mathcal{P}$  exists. Then, for every  $\lambda, \mu \geq 0$  there will exist **Nash half-branch curve germs**  $\alpha_\lambda(s), \beta_\mu(s)$  which cannot be extended to 0 and such that:

$$\lim_{s \rightarrow 0} P(\alpha_\lambda(s)) = (\lambda^2, 0) \quad \text{and} \quad \lim_{s \rightarrow 0} P(\beta_\mu(s)) = (0, \mu^2).$$

We can try parameterizations of the form:

$$\alpha_\lambda(s) := \left( s^{n_\lambda}, \frac{a_{\lambda 0} + a_{\lambda 1}s + \dots}{s^{m_\lambda}} \right) \quad \text{and} \quad \beta_\mu(s) := \left( \frac{b_{\mu 0} + b_{\mu 1}s + \dots}{s^{\ell_\mu}}, s^{k_\mu} \right).$$

Then  $a_{\lambda 0}, b_{\mu 0}$  must be constants (except maybe for finitely many values of  $\lambda$  and  $\mu$ ). This leads us to choose curves of the type:

$$\alpha_\lambda(s) := \left( s^{n_\lambda}, \frac{1 + a_{\lambda 1}s + \dots}{s^{m_\lambda}} \right) \quad \text{and} \quad \beta_\mu(s) := \left( \frac{1 + b_{\mu 1}s + \dots}{s^{\ell_\mu}}, s^{k_\mu} \right),$$

and among them we make the simplest choice:

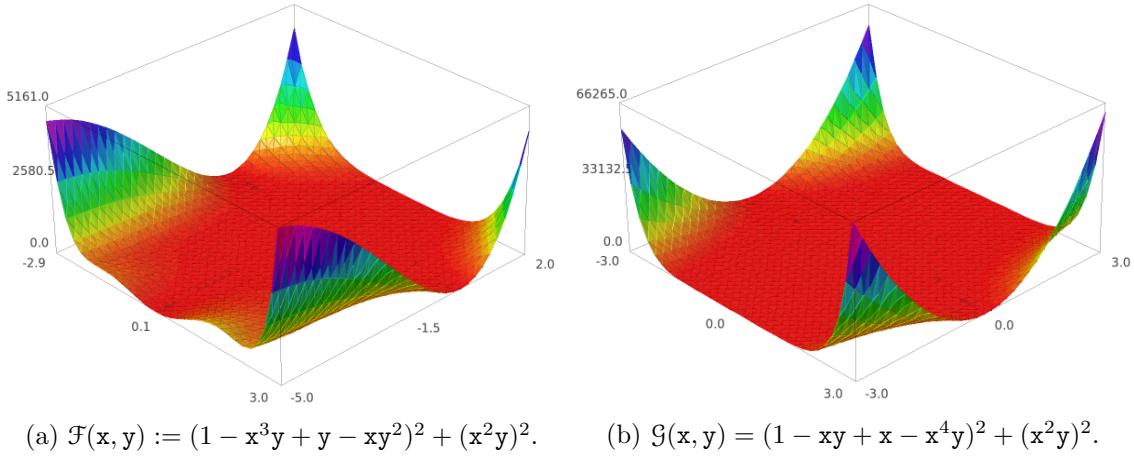
$$\alpha_\lambda(s) := \left( s, \frac{1 + a_\lambda s}{s} \right) \quad \text{and} \quad \beta_\mu(s) := \left( \frac{1 + b_\mu s}{s}, s^3 \right).$$

The following pair of polynomials:

$\mathcal{F}(x, y) := (1 - x^3y + y - xy^2)^2 + (x^2y)^2 = \mathcal{F}_1^2 + \mathcal{F}_2^2$
$\mathcal{G}(x, y) := (1 - xy + x - x^4y)^2 + (x^2y)^2 = \mathcal{G}_1^2 + \mathcal{G}_2^2$

enjoy a nice behavior along these curves, as Figure 2.2 show. Next, notice that

- (a)  $\cdot \mathcal{F}_1 \circ \alpha_\lambda = 1 - a_\lambda - a_\lambda^2 s - s^2 - a_\lambda s^3 \in \mathbb{R}[s, a_\lambda], \quad \mathcal{F}_1 \circ \alpha_\lambda(0) = 1 - a_\lambda.$   
 $\cdot \mathcal{F}_1 \circ \beta_\mu = -3b_\mu s - 3b_\mu^2 s^2 - (b_\mu^3 - 1)s^3 - s^5 - b_\mu s^6 \in \mathbb{R}[s, b_\mu], \quad \mathcal{F}_1 \circ \beta_\mu(0) = 0.$
- (b)  $\cdot \mathcal{G}_1 \circ \alpha_\lambda = (1 - a_\lambda)s - s^3 - a_\lambda s^4 \in \mathbb{R}[s, a_\lambda], \quad \mathcal{G}_1 \circ \alpha_\lambda(0) = 0.$   
 $\cdot \mathcal{G}_1 \circ \beta_\mu = 1 - 3b_\mu - 6b_\mu^2 s - (4b_\mu^3 + 1)s^2 - (b_\mu^4 + b_\mu)s^3 \in \mathbb{R}[s, b_\mu], \quad \mathcal{G}_1 \circ \beta_\mu(0) = 1 - 3b_\mu.$
- (c)  $\cdot \mathcal{F}_2 \circ \alpha_\lambda = s + a_\lambda s^2 = \mathcal{G}_2 \circ \alpha_\lambda \in \mathbb{R}[s, a_\lambda].$   
 $\cdot \mathcal{F}_2 \circ \beta_\mu = s + 2b_\mu s^2 + b_\mu^2 s^3 = \mathcal{G}_2 \circ \beta_\mu \in \mathbb{R}[s, b_\mu].$   
 $\cdot \mathcal{F}_2 \circ \alpha_\lambda(0) = \mathcal{G}_2 \circ \alpha_\lambda(0) = \mathcal{F}_2 \circ \beta_\mu(0) = \mathcal{G}_2 \circ \beta_\mu(0) = 0.$


 Figure 2.2: Plots of the polynomial functions  $F, G : \mathbb{R}^2 \rightarrow \mathbb{R}.$ 

All of these map compositions were computed by Sage. Thus, we get the following properties:

- (i) The polynomial functions  $F, G$  are non-negative in  $\mathbb{R}^2.$
- (ii)  $\cdot F^{-1}(0) = F_1^{-1}(0) \cap F_2^{-1}(0) = \{(0, -1)\} \xrightarrow{\mathcal{P}} \{(0, 1)\}.$   
 $\cdot G^{-1}(0) = G_1^{-1}(0) \cap G_2^{-1}(0) = \{(-1, 0)\} \xrightarrow{\mathcal{P}} \{(1, 0)\}.$
- (iii)  $\cdot \mathcal{P} \circ \alpha_\lambda = (F \circ \alpha_\lambda, G \circ \alpha_\lambda) =$   

$$(a_\lambda^2 - 2a_\lambda + 1 + 2(a_\lambda^3 - a_\lambda^2)s + (a_\lambda^4 + 2a_\lambda - 1)s^2 + a_\lambda^2 s^3 + (2a_\lambda^3 + a_\lambda^2 + 1)s^4 + 2a_\lambda s^5 + a_\lambda^2 s^6,$$

$$(a_\lambda^2 - 2a_\lambda + 2)s^2 + 2a_\lambda s^3 + (a_\lambda^2 + 2a_\lambda - 2)s^4 + 2(a_\lambda^2 - a_\lambda)s^5 + s^6 + 2a_\lambda s^7 + a_\lambda^2 s^8).$$
  
 $\cdot \mathcal{P} \circ \beta_\mu = (F \circ \beta_\mu, G \circ \beta_\mu) =$   

$$(9b_\mu^2 + 1)s^2 + 2(9b_\mu^3 + 2b_\mu)s^3 + 3(5b_\mu^4 + 2b_\mu^2 - 2b_\mu)s^4 + 2(3b_\mu^5 + 2b_\mu^3 - 3b_\mu^2)s^5 + (b_\mu^6 + b_\mu^4 - 2b_\mu^3 + 6b_\mu + 1)s^6 + 12b_\mu^2 s^7 + 2(4b_\mu^3 - 1)s^8 + 2(b_\mu^4 - b_\mu)s^9 + s^{10} + 2b_\mu s^{11} + b_\mu^2 s^{12},$$

$$9b_\mu^2 - 6b_\mu + 1 + 12(3b_\mu^3 - b_\mu^2)s + (60b_\mu^4 - 8b_\mu^3 + 6b_\mu - 1)s^2 + 2(27b_\mu^5 - b_\mu^4 + 9b_\mu^2 + b_\mu)s^3 + (28b_\mu^6 + 20b_\mu^3 + 6b_\mu^2 + 1)s^4 + 2(4b_\mu^7 + 5b_\mu^4 + 2b_\mu^3 + b_\mu)s^5 + (b_\mu^8 + 2b_\mu^5 + b_\mu^4 + b_\mu^2)s^6).$$

The polynomials  $F \circ \alpha_\lambda, G \circ \alpha_\lambda \in \mathbb{R}[s, a_\lambda]$  and  $F \circ \beta_\mu, G \circ \beta_\mu \in \mathbb{R}[s, b_\mu]$  were computed with Sage. As we anticipated before, by condition (ii) the set  $F = \{(-1, 0), (0, -1)\}.$

## 2.2 The first proof for the open quadrant problem

[Proof of Theorem 1.11] We are going to prove that  $\mathcal{Q} \subset \mathcal{P}(\mathbb{R}^2).$  To do this it is enough to fix  $v > 0$  and to see that the image under  $\mathcal{F}$  of the curve  $\{G = v\}$  contains the open

half-line  $(0, +\infty)$ .

**Step 1** *Parametrization of the curve  $\{\mathcal{G} - v = 0\}$ .*

We start by solving the equation  $\mathcal{G} - v = 0$ , that is,

$$(1 - xy + x - x^4y)^2 + (x^2y)^2 - v = 0.$$

As it has degree 2 with respect to  $y$ , we compute its roots  $y^+(x, v)$  and  $y^-(x, v)$  given by:

$$y^+(x, v) := \frac{1 + x + x^3 + x^4 + \sqrt{\Delta(x, v)}}{x(x^2 + (x^3 + 1)^2)}$$

$$y^-(x, v) := \frac{1 + x + x^3 + x^4 - \sqrt{\Delta(x, v)}}{x(x^2 + (x^3 + 1)^2)}$$

where  $\Delta(x, v) = \Delta_v(x) := v(x^2 + (x^3 + 1)^2) - x^2(x + 1)^2$ ,  $\deg_x(\Delta) = 6$ . We can see on Figure 2.3 how  $y^+$  and  $y^-$  look like for instance for  $v := 0.8$ . As we can see on Figure 2.4, for  $v := 1$  there are no singularities on  $y^-$  because  $\lim_{x \rightarrow 0} y^-(x, 1) = 1$ . This observation is used later, in Step 2.

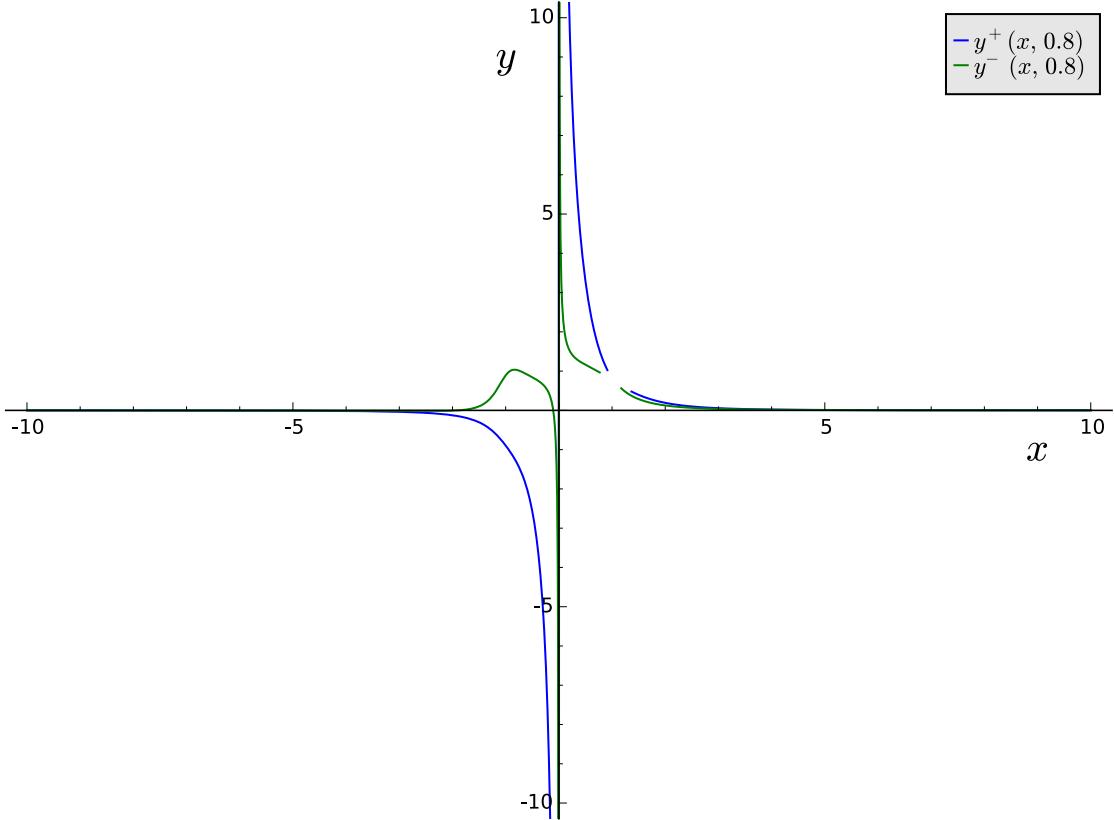
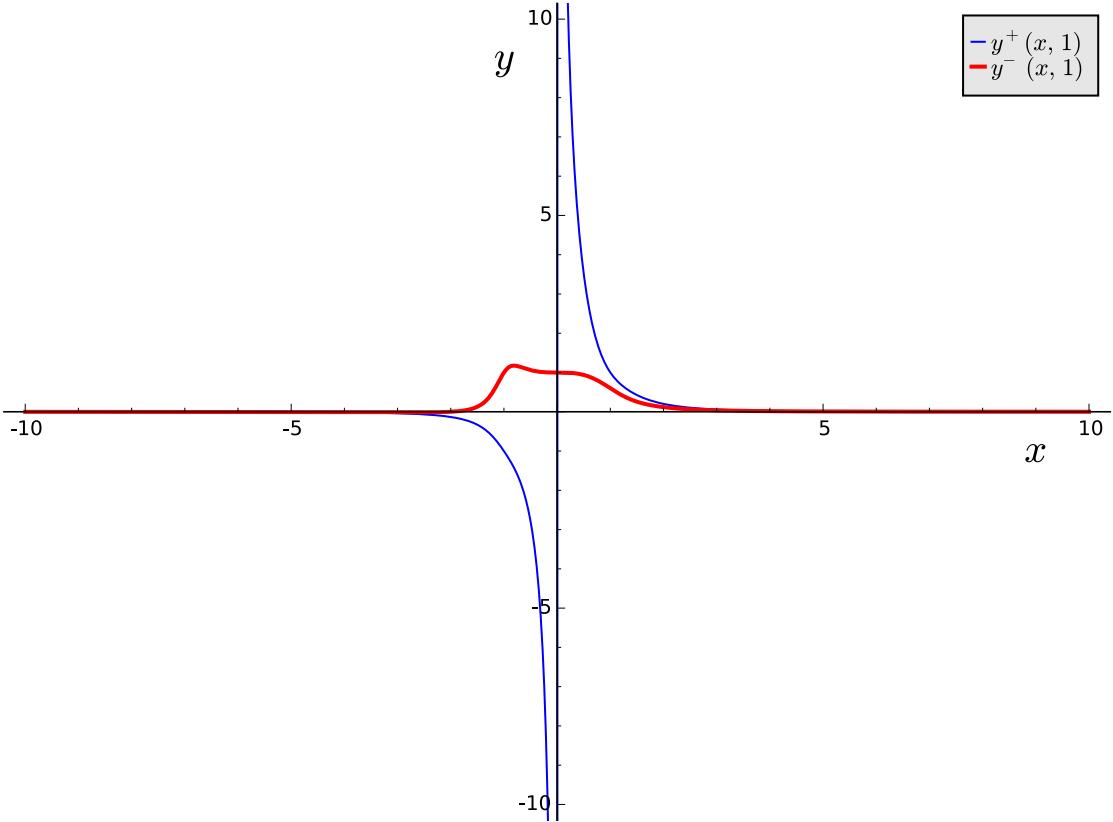


Figure 2.3:  $y^+(x, v)$  and  $y^-(x, v)$  for  $v := 0.8$ .

The common domain of these two functions is the set

$$D_v := \{x \in R : \Delta(x, v) \geq 0, x \neq 0\}.$$


 Figure 2.4:  $y^+(x, v)$  and  $y^-(x, v)$  for  $v := 1$ .

Notice that the only real root of the denominator is  $x_0 := 0$ .<sup>1</sup> Let

$$\gamma_v^+ : D_v \rightarrow \mathbb{R}, x \mapsto \mathcal{F}(x, y^+(x, v)) \quad \text{and} \quad \gamma_v^- : D_v \rightarrow \mathbb{R}, x \mapsto \mathcal{F}(x, y^-(x, v))$$

Note that  $\mathcal{F}(\{\mathcal{G} = v\}) = \text{im}(\gamma_v^+) \cup \text{im}(\gamma_v^-)$ , so all reduce to prove the inclusion

$$(0, +\infty) \subset \text{im}(\gamma_v^+) \cup \text{im}(\gamma_v^-).$$

**Step 2** Main properties of  $\gamma_v^+$  and  $\gamma_v^-$ .

In this section we are going to prove that:

$$(i) \lim_{x \rightarrow \pm\infty} \gamma_v^+(x) = \lim_{x \rightarrow \pm\infty} \gamma_v^-(x) = 0.$$

$$(ii) \lim_{x \rightarrow 0} \gamma_v^+(x) = +\infty, \quad \lim_{x \rightarrow 0} \gamma_v^-(x) = \begin{cases} +\infty & \text{for } v \neq 1 \\ 4 & \text{for } v = 1 \end{cases}$$

Using Sage we can symbolically check how  $\gamma_v^+$  and  $\gamma_v^-$  look like, getting polynomials  $A_1, A_2, B_1, B_2 \in \mathbb{R}[\mathbf{x}, v]$  and  $C \in \mathbb{R}[\mathbf{x}]$  such that:

$$(a) \gamma_v^+(x) = \frac{A_1(x, v) + B_1(x, v)\sqrt{\Delta(x, v)}}{C(x)}, \quad \gamma_v^-(x) = \frac{A_2(x, v) + B_2(x, v)\sqrt{\Delta(x, v)}}{C(x)},$$

<sup>1</sup>We checked with Laguerre's method, implemented with Python 2.7, that the polynomial  $x^7 + 2x^4 + x^3 + x$  has 6 complex roots.

$$\begin{aligned}
 A_1(x, v) &= A_2(x, v), & \deg_x(A_1) = \deg_x(A_2) &= 24 \\
 (b) \quad B_1(x, v) &= -B_2(x, v), & \deg_x(B_1) = \deg_x(B_2) &= 21 \\
 C(x) &= x^2(x^2 + (x^3 + 1)^2)^4, & \deg_x(C) &= 26.
 \end{aligned}$$

We proceed to study  $\gamma_v^+$  and  $\gamma_v^-$  at the origin. Since  $\Delta$  has even degree and positive leading coefficient with respect to  $x$ , it is positive for  $|x|$  large enough, so (i) holds.

Now, for  $x = 0$ , we get  $\Delta(0, v) = v > 0$ , thus  $0 \in \overline{D}_v$ . Also:

- ★  $A_1(0, v) + B_1(0, v)\sqrt{\Delta(0, v)} = v(1 + \sqrt{v})^2 > 0$ .
- ★  $A_2(0, v) + B_2(0, v)\sqrt{\Delta(0, v)} = v(1 - \sqrt{v})^2 \geq 0$ , and equality holds if and only if  $v = 1$ .

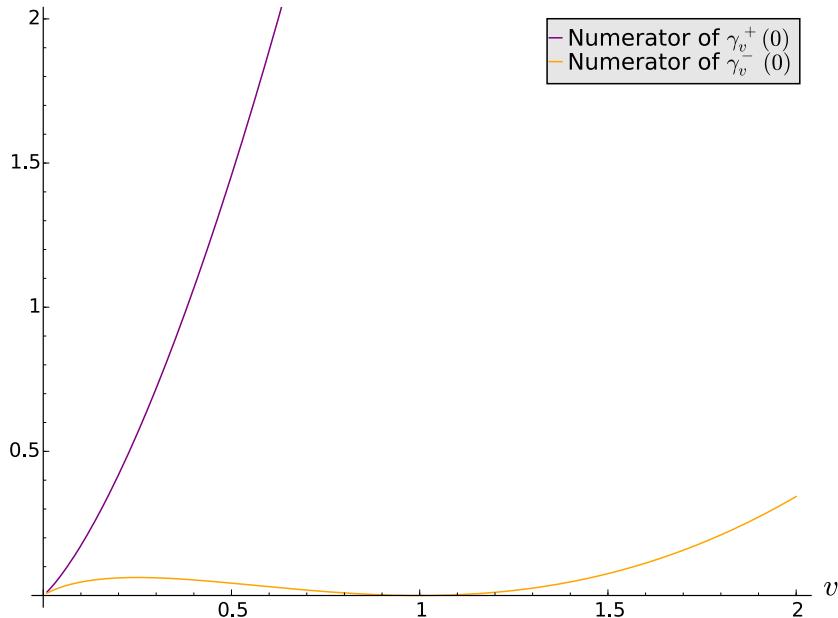


Figure 2.5: Numerators of  $\gamma_v^+$  and  $\gamma_v^-$  for  $x := 0$ .

Thus, (ii) holds (we also checked it with Sage). The result for  $v := 1$  in (ii) is not relevant here (see Figure 2.6).

**Step 3** When  $v \geq 0.28^2$  we have  $(0, +\infty) \subset \text{im}(\gamma_v^+)$ .

In order to see whether  $(0, +\infty) \subset \text{im}(\gamma_v^+) \cup \text{im}(\gamma_v^-)$  or not we are now going to study the domain  $D_v$ . To that end we need to study when  $\Delta(x, v) = 0$ , so it seems convenient to define:

$$v(x) := \frac{x^2(x+1)^2}{x^2 + (x^3 + 1)^2},$$

whose graph can be seen in Figure 2.7. If  $x \in (-\infty, 0)$  we checked using Laguerre's method that the polynomial

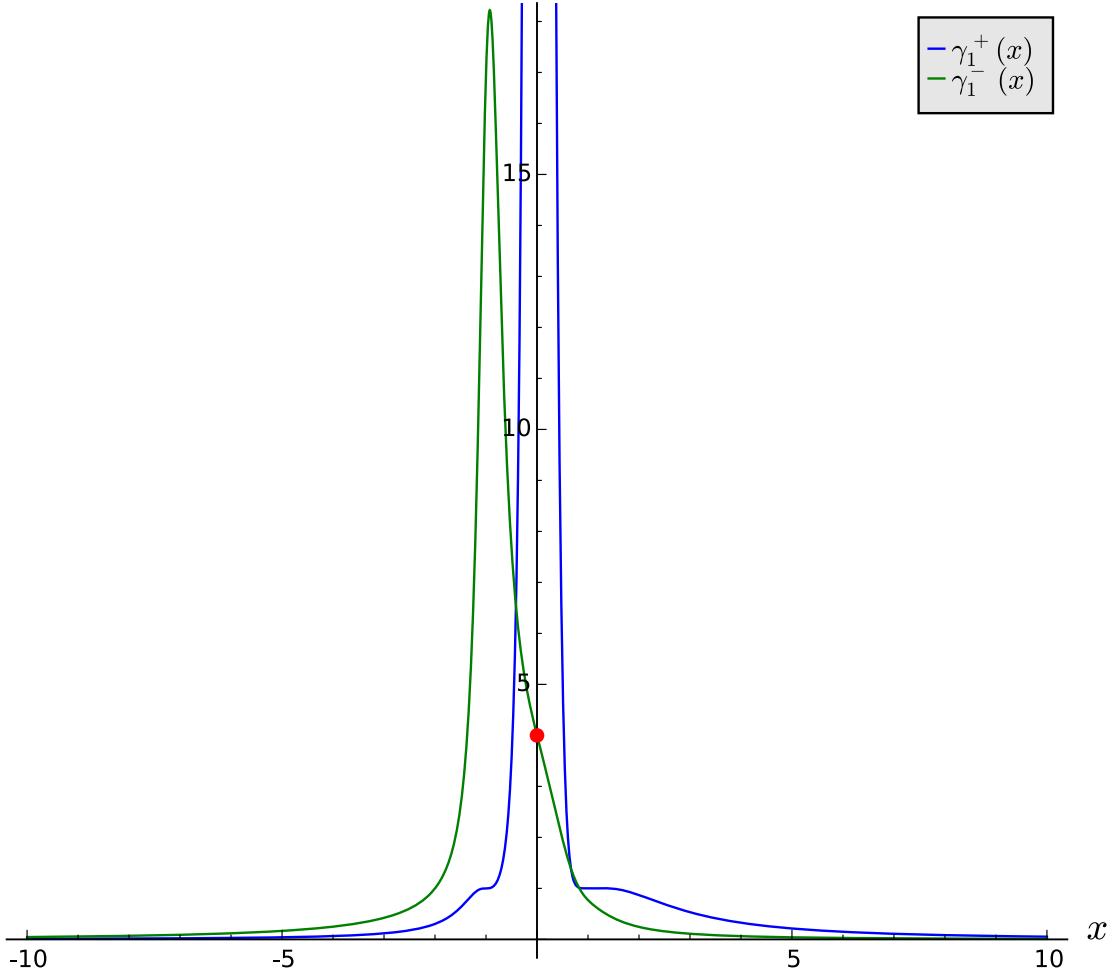


Figure 2.6: Notice the value of  $\gamma_1^-(x)$  at  $x = 0$ .

$$\Delta(x, 0.28^2) = 0.0784 x^6 - x^4 - 1.8432 x^3 - 0.9216 x^2 + 0.0784$$

has 4 complex roots and 2 real ones<sup>2</sup>. These last are  $\delta_0 \approx 0.236$  and  $\delta_1 \approx 4.336$ . Thus  $\Delta(x, v)$  has no negative roots for  $v \geq 0.28^2$  and, in addition, it only attains positive values. Therefore  $(-\infty, 0) \subset D_v$ . But then, as  $\gamma_v^+$  is continuous and recalling the limits computed in Step 2, we get

$$(0, +\infty) \subset \text{im}(\gamma_v^+) \subset \text{im}(\gamma_v^+) \cup \text{im}(\gamma_v^-).$$

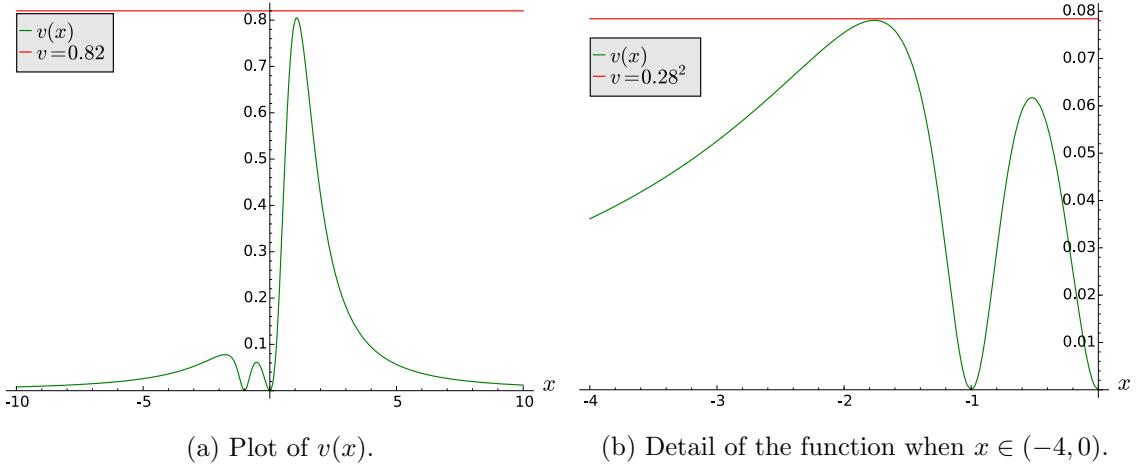
**Step 4** When  $0 < v < 0.28^2$  we have  $(0, +\infty) \subset \text{im}(\gamma_v^-)$ .

To prove that for  $0 < v < 0.28^2$  the inclusion  $(0, +\infty) \subset \text{im}(\gamma_v^-)$  holds it is enough to prove the existence of  $N_v, \delta_v \in \mathbb{R}$  satisfying

$$N_v < \delta_v, \quad (-\infty, N_v] \cup [\delta_v, +\infty) \subset D_v \text{ and } \gamma_v^-(N_v) > \gamma_v^+(\delta_v) \quad (\spadesuit)$$

See Figure 2.8 to get an idea of what we are saying here.

<sup>2</sup>The value  $v_0 := 0.28^2$  comes from a careful observation of the plot from Figure 2.7b.


 Figure 2.7: Plot of the univariate function  $v(x)$ .

To prove the existence of such  $N_v$  and  $\delta_v$ , we must compute the roots of  $\Delta_v(\mathbf{x})$  in an algebraic closure of the field of rational functions  $\mathbb{R}(v)$ . Such an algebraic closure is the field of Puiseux series  $\mathbb{C}(\{v^*\})$ , see A.9. These roots are power series in  $\mathbb{C}(\{w^*\})$  with  $w := \sqrt{v}$ , and we consider the largest and the smallest negative roots  $\eta_v, \xi_v \in \mathbb{R}(\{v^*\})$  of  $\Delta_v$  with respect to the unique ordering in  $\mathbb{R}(\{v^*\})$  that makes  $v$  positive and infinitesimal with respect to  $\mathbb{R}$ . These roots are:

$$\begin{cases} \eta_v := -\frac{1}{w} + 1 + w + w^2 + \frac{5}{2}w^3 + \dots \\ \xi_v := -w - w^2 - \frac{5}{2}w^3 - 6w^4 + \dots \end{cases}$$

Notice that, by the definition of the ordering in  $\mathbb{R}(\{v^*\})$ , the first coefficient of a series is the “most meaningful orderwise”. In particular  $\eta_v < \xi_v$ . To perform calculations, we handle suitable truncations of the involved series. Here the word suitable means “as short as possible but order preserving”; in other words, we look for  $N_v$  and  $\delta_v$  with  $N_v < \eta_v < \xi_v < \delta_v$ , and in fact we choose

$$\begin{cases} N_v := -\frac{1}{w} + 1 + w + w^2 = \eta_v - \left(\frac{5}{2}w^3 + \dots\right) < \eta_v \\ \delta_v := -w - w^2 - \frac{5}{2}w^3 = \xi_v - (-6w^4 + \dots) > \xi_v. \end{cases}$$

We checked with Sage that  $-\infty < N_v < \delta_v < 0$  for  $v \in (0, 0.28^2)$  that is, for  $w \in (0, 0.28)$ , see Figure 2.9a. Now we can focus on proving (♦). Since  $\Delta(N_{w^2}, w^2)$  and  $\Delta(\delta_{w^2}, w^2)$  are positive (see Figure 2.9b) for  $w \in (0, 0.28)$ , we get that  $N_v, \delta_v \in D_v$ . For the first part, let

$$D := \bigcup_{v>0} D_v = \bigcup_{v>0} \{x \in \mathbb{R} : \Delta(x, v) \geq 0, x \neq 0\},$$

whose boundary is the union of the axis  $\{x = 0\} \subset \mathbb{R}^2$  and the curve given by the equation

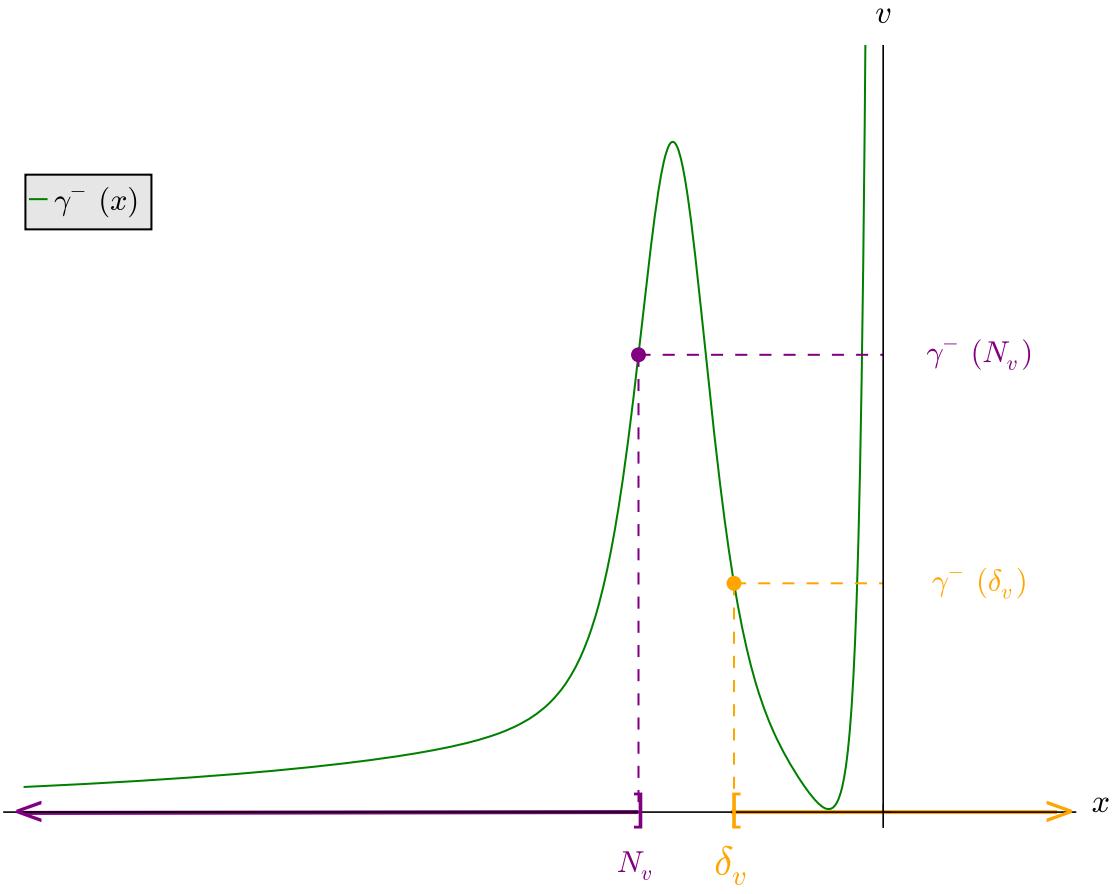


Figure 2.8: Idea of what we are saying with  $(\spadesuit)$ . Here  $v := 0.1$ .

$\Delta(x, v) = 0$ , that is, the graph of the regular function

$$v(x) = \frac{x^2(x+1)^2}{x^2 + (x^3 + 1)^2}.$$

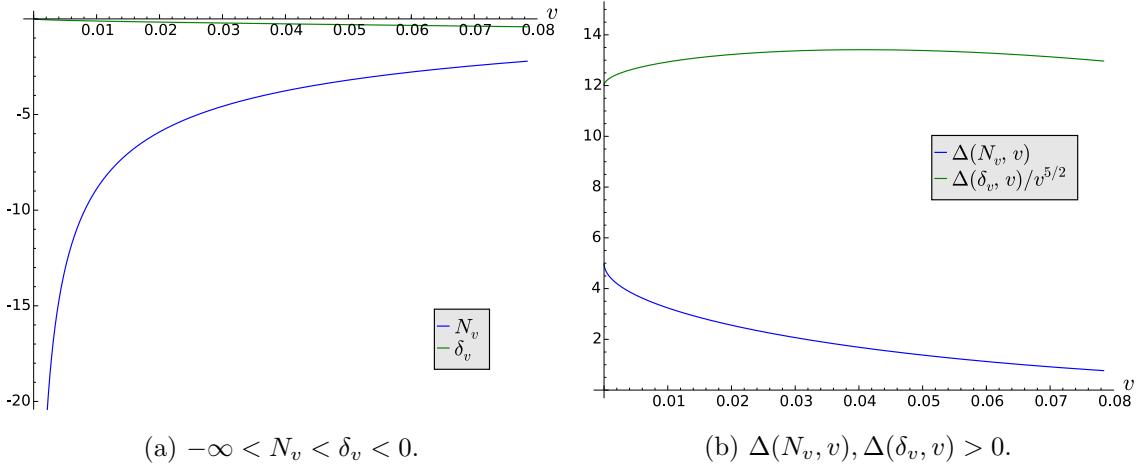
This graph is above the axis  $\{v = 0\} \subset R^2$ . Then,  $(-\infty, N_v]$  and  $[\delta_v, 0)$  are contained in the interior of  $D_v$  for  $v \in (0, 0.28^2)$ , because the curves

$$\{(\delta_v, v) : 0 < v < 0.28^2\} \text{ and } \{(N_v, v) : 0 < v < 0.28^2\}$$

are contained in  $D$ , they are graphs above the vertical axis  $\{x = 0\} \subset R^2$ , and  $\delta_v < \xi_v$  and  $N_v < \eta_v$  as we saw before. Look at Figure 2.10.

So the only thing left to do is checking that  $\gamma_v^-(N_v) > \gamma_v^-(\delta_v)$ . Recall that

$$\gamma_v^-(x) = \frac{A_2(x, v) + B_2(x, v)\sqrt{\Delta(x, v)}}{C(x)},$$


 Figure 2.9: Plots of  $N_v, \delta_v < 0$  and  $\Delta(N_v, v), \Delta(\delta_v, v) > 0$  for  $v \in (0, 0.28^2)$ .

with  $\deg_x(A_2) = 24$ ,  $\deg_x(B_2) = 21$ ,  $\deg_x(\Delta) = 6$  and  $\deg_x(C) = 26$ . Consider:

$$\begin{array}{ll} \cdot f_1(w) = A_2(N_{w^2}, w^2) \cdot w^{24} & \cdot f_2(w) = A_2(\delta_{w^2}, w^2) \\ \cdot g_1(w) = B_2(N_{w^2}, w^2) \cdot w^{21} & \cdot g_2(w) = B_2(\delta_{w^2}, w^2) \\ \cdot q_1(w) = \Delta(N_{w^2}, w^2) & \cdot q_2(w) = \Delta(\delta_{w^2}, w^2) \\ \cdot h_1(w) = C(N_{w^2}) \cdot w^{26} & \cdot h_2(w) = C(\delta_{w^2}). \end{array}$$

Thus, we need to prove that for  $w \in (0, 0.28)$ :

$$\frac{f_1 \cdot (w^{24})^{-1} + g_1 \cdot (w^{21})^{-1} \sqrt{q_1}}{h_1 \cdot (w^{26})^{-1}} > \frac{f_2 + g_2 \sqrt{q_2}}{h_2} \iff$$

$$\frac{w^2 h_2 f_1 - f_2 h_1}{h_1 h_2} + \frac{w^5 g_1 \sqrt{q_1}}{h_1} - \frac{g_2 \sqrt{q_2}}{h_2} > 0,$$

and we are going to prove that

$$\Lambda_1 := \frac{w^2 h_2 f_1 - f_2 h_1}{h_1 h_2}, \quad \Lambda_2 := \frac{w^5 g_1 \sqrt{q_1}}{h_1} \quad \text{and} \quad \Lambda_3 := -\frac{g_2 \sqrt{q_2}}{h_2}$$

are positive in the given interval, which only contains positive values. As  $q_1, q_2$  are positive, we can clear away  $w^5$  and  $\sqrt{q_1}$  from  $\Lambda_2$ , and  $\sqrt{q_2}$  from  $\Lambda_3$ . Furthermore, the polynomial  $C(x) = x^2(x^2 + (x^3 + 1)^2)^4 > 0$ , so we can also remove  $h_1$  and  $h_2$  from  $\Lambda_1, \Lambda_2, \Lambda_3$ . Thus it suffices to see that

$$L := \frac{w^2 h_2 f_1 - f_2 h_1}{w^4}, \quad g_1, \quad K := -\frac{g_2}{w^3}$$

are positive for  $w \in (0, 0.28)$ .

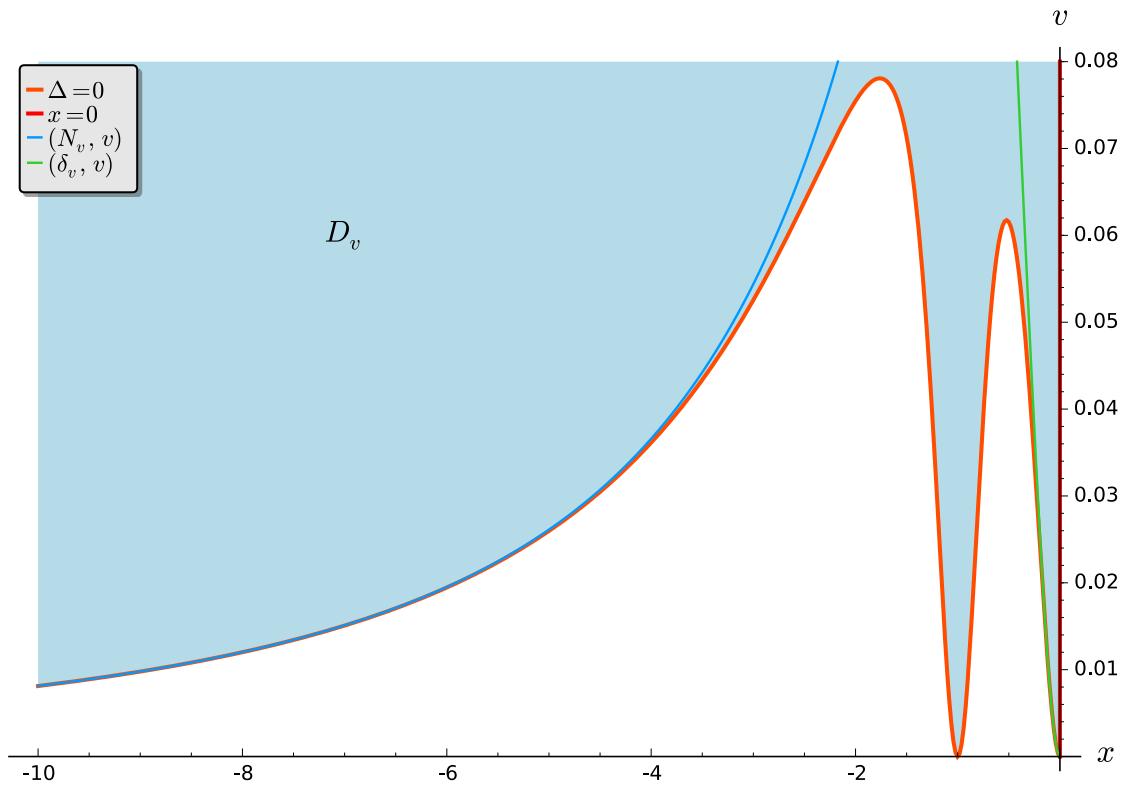
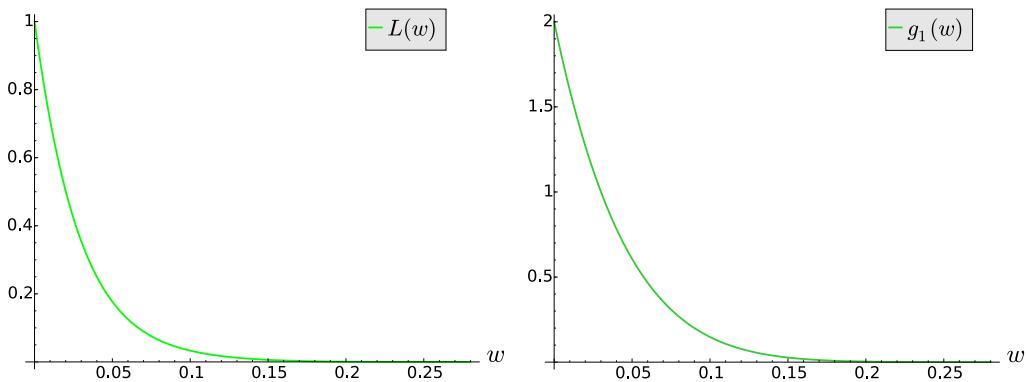
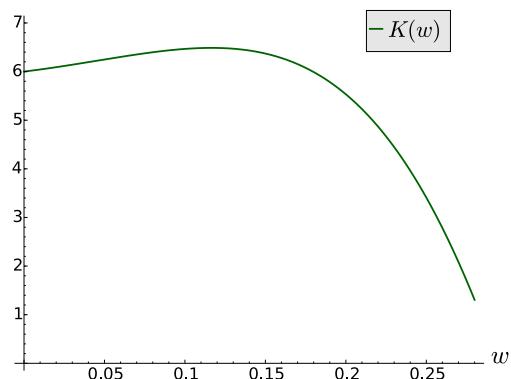


Figure 2.10: Plot of  $\{(N_v, v)\}, \{(\delta_v, v)\} \subset D_v$ , for  $0 < v < 0.28^2$ .

As we see in the figures they indeed are. This has been also checked with Sturm algorithm and numerically with Sage. Thus, ( $\spadesuit$ ) holds and the result is proved.





□

# 3

## A short proof for the open quadrant $\mathcal{Q}$ problem

### 3.1 A new approach

In the previous chapter, we proved that  $\mathcal{Q}$  is the image of  $\mathbb{R}^2$  under the polynomial map  $\mathcal{P} := (\mathcal{F}, \mathcal{G})$ . This fact alongside Theorem 1.9 were key for proving Theorem 1.10. To achieve Theorem 1.11, we needed the aid of a computer in order to check that certain polynomials do not have roots on particular intervals or that they are positive on them. Although legit, this procedure is controversial, and Fernando and Ueno kept on working on the problem of characterizing which semialgebraic subsets  $\mathcal{S} \subset \mathbb{R}^m$  are polynomial images of  $\mathbb{R}^n$  and wrote a new paper [FU] with a much simpler proof.

In this third chapter we present a new approach to the open quadrant problem. We will show that  $\mathcal{Q}$  is the image of the composition of three simple polynomial maps:  $\mathcal{F}$ ,  $\mathcal{G}$  and  $\mathcal{H}$ . The proof of Theorem 1.11 will be conducted by inspecting at the images of the aforementioned polynomials, albeit the lack of precise tools to determine the image of a polynomial map.

#### 3.1.1 The new polynomial maps

In this subsection we introduce new polynomial maps that satisfy the requirements to prove Theorem 1.11 in a different way. To be more precise, we define

$$f := \mathcal{H} \circ \mathcal{G} \circ \mathcal{F} : \mathbb{R}^2 \rightarrow \mathbb{R}^2,$$

where

$$\boxed{\begin{aligned}\mathcal{F} : \mathbb{R}^2 &\rightarrow \mathbb{R}^2, (x, y) \mapsto ((xy - 1)^2 + x^2, (xy - 1)^2 + y^2), \\ \mathcal{G} : \mathbb{R}^2 &\rightarrow \mathbb{R}^2, (x, y) \mapsto (x, y(xy - 2)^2 + x(xy - 1)^2), \\ \mathcal{H} : \mathbb{R}^2 &\rightarrow \mathbb{R}^2, (x, y) \mapsto (x(xy - 2)^2 + \frac{1}{2}xy^2, y).\end{aligned}}$$

We can see how the graphs of the components of these maps look like in Figure 3.1, and appreciate the symmetry between the  $x$  and  $y$  variables while the transformation is performed.

In the next section we proceed to develop the proof, which splits into three lemmas.

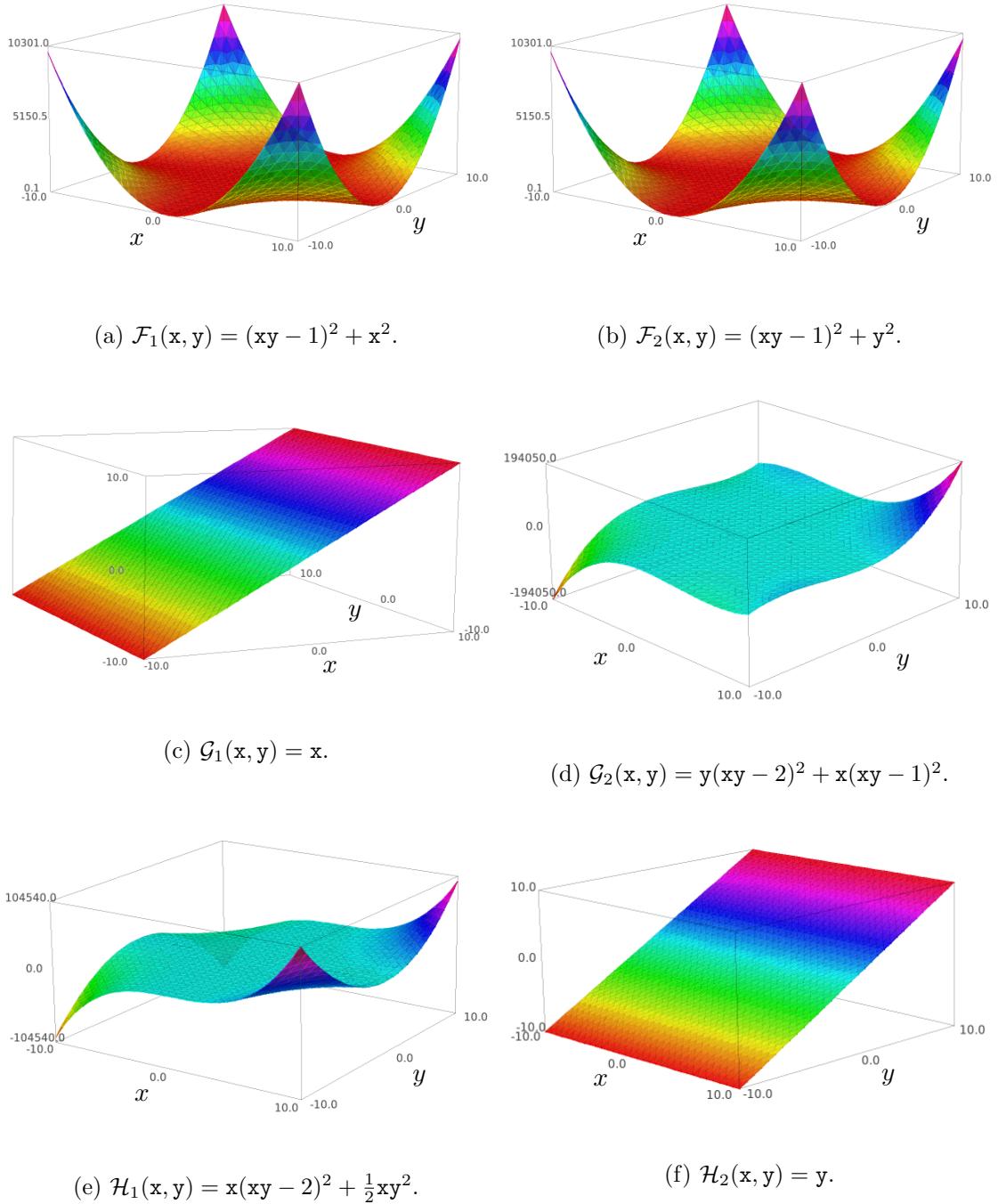


Figure 3.1: The polynomial maps from the second proof:  $(\mathcal{H} \circ \mathcal{G} \circ \mathcal{F})(\mathbb{R}^2) = \mathcal{Q}$ .

## 3.2 The new proof for the open quadrant problem

As we have anticipated before,  $\mathcal{Q}$  is the image of the composition  $\mathcal{H} \circ \mathcal{G} \circ \mathcal{F}$ , but before proving this fact we need three auxiliar lemmas in order to shed some light on the properties that the images of  $\mathcal{F}$ ,  $\mathcal{G}$  and  $\mathcal{H}$  enjoy.

### 3.2.1 The first lemma

**Lemma 3.1.** *Let  $\mathcal{A} := \{xy \geq 1\} \cap \mathcal{Q}$ . Then the image of the map*

$$\mathcal{F} := (\mathcal{F}_1, \mathcal{F}_2) : \mathbb{R}^2 \rightarrow \mathbb{R}^2, (x, y) \mapsto ((xy - 1)^2 + x^2, (xy - 1)^2 + y^2)$$

satisfies that  $\mathcal{A} \subset \mathcal{F}(\mathbb{R}^2) \subset \mathcal{Q}$ .

*Proof.* Since  $\mathcal{F}_1$  and  $\mathcal{F}_2$  are positive on  $\mathbb{R}^2$ , the inclusion  $\mathcal{F}(\mathbb{R}^2) \subset \mathcal{Q}$  is evident. To prove the other inclusion, we must show that for  $(a, b) \in \mathcal{A}$  the system of polynomial equations

$$\begin{cases} (xy - 1)^2 + x^2 = a \\ (xy - 1)^2 + y^2 = b \end{cases} \quad (3.1)$$

has a solution  $(x_0, y_0) \in \mathbb{R}^2$ . Set  $z := xy - 1$  in order to rewrite the system 3.1 in terms of  $x$  and  $z$ . Since  $y = \frac{z+1}{x}$  we get:

$$\begin{cases} z^2 + x^2 = a \\ z^2 + \frac{(z+1)^2}{x^2} = b. \end{cases} \quad (3.2)$$

By eliminating  $x$  on the system 3.2 we deduce that  $z$  must be a root of the polynomial

$$P(z) := z^4 - (a + b + 1)z^2 - 2z + (ab - 1) = 0.$$

Taking into account that  $(a, b) \in \mathcal{A}$  satisfy that  $a, b > 0$  and  $ab \geq 1$ , we notice that  $P$  is a monic polynomial of even degree satisfying

$$P(0) = ab - 1 \geq 0 \quad \text{and} \quad P(\sqrt{a}) = -2\sqrt{a} - a - 1 < 0.$$

Thus,  $P$  has a real root  $z_0$  such that  $0 \leq z_0 < \sqrt{a}$ , so we set:

$$x_0 := \sqrt{a - z_0^2} \quad \text{and} \quad y_0 := \frac{z_0 + 1}{x_0}.$$

Then  $F(x_0, y_0) = (a, b)$  and  $\mathcal{A} \subset \mathcal{F}(\mathbb{R}^2)$ , as required.  $\square$

### 3.2.2 The second lemma

**Lemma 3.2.** *Let  $\mathcal{B} := \mathcal{A} \cup \{y \geq x > 0\}$ . Then, the map*

$$\mathcal{G} := (\mathcal{G}_1, \mathcal{G}_2) : \mathbb{R}^2 \rightarrow \mathbb{R}^2, (x, y) \mapsto (x, y(xy - 2)^2 + x(xy - 1)^2)$$

satisfies that  $\mathcal{B} \subset \mathcal{G}(\mathcal{A}) \subset \mathcal{G}(\mathcal{Q}) \subset \mathcal{Q}$ .

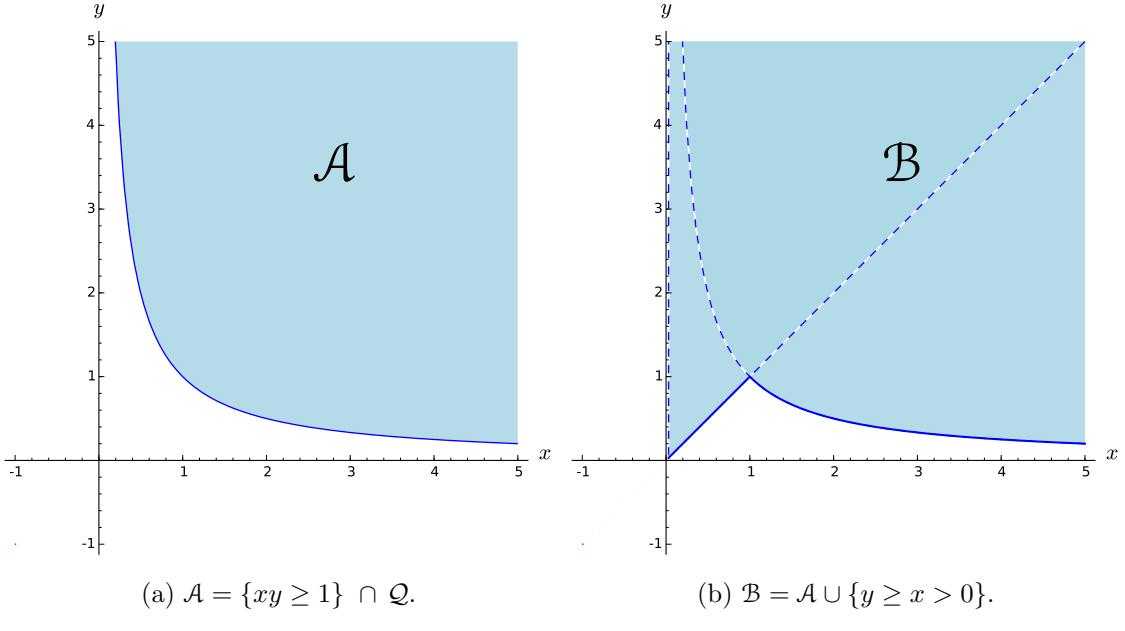


Figure 3.2: Relevant sets for Lemmas 3.1, 3.2 and 3.3.

*Proof.* The inclusion  $\mathcal{G}(\mathcal{A}) \subset \mathcal{G}(\mathcal{Q})$  is trivial because  $\mathcal{A} \subset \mathcal{Q}$ . The last inclusion  $\mathcal{G}(\mathcal{Q}) \subset \mathcal{Q}$  also holds since  $\mathcal{G}_1$  and  $\mathcal{G}_2$  are strictly positive on  $\mathcal{Q}$ : if  $x_0, y_0 > 0$  then  $\mathcal{G}_1(x_0, y_0) = x_0 > 0$  and  $\mathcal{G}_2(x_0, y_0) = y_0(x_0 y_0 - 2)^2 + x_0(x_0 y_0 - 1)^2 > 0$ . Notice that  $\mathcal{G}_2(x_0, y_0) \neq 0$ . Otherwise there would exist a solution of the system of equations

$$\begin{cases} xy - 1 = 0 \\ xy - 2 = 0, \end{cases}$$

which is not possible.

Now we can focus on proving  $\mathcal{B} \subset \mathcal{G}(\mathcal{A})$ . First of all, notice that we can express the set  $\mathcal{B}$  as a union of closed half-lines. Precisely, for each  $x > 0$  let  $\mathcal{B}_x := [y_x, +\infty)$ , where  $y_x := \min\{x, 1/x\}$  and notice that

$$\mathcal{B} = \bigsqcup_{x>0} (\{x\} \times \mathcal{B}_x).$$

That is, we are “slicing” the set  $\mathcal{B}$  vertically, depending on  $x$ . Now look at the definition of the polynomial  $\mathcal{G}_2$  and consider for each  $x > 0$  the polynomial  $\phi_x \in \mathbb{R}[y]$  (dependent on the variable  $y$ ):

$$\phi_x(y) := y(xy - 2)^2 + x(xy - 1)^2 = x^2y^3 + (x^3 - 4x)y^2 + (4 - 2x^2)y + x.$$

The polynomials  $\phi_x(y)$  have odd degree and positive leading coefficient since  $x > 0$ . Now notice that if we fix  $x_0 > 0$  then we get the following sequence of inclusions:

$$\begin{aligned} \phi_{x_0}([1/x_0, +\infty)) &\supset^1 \phi_{x_0}([2/x_0, +\infty)) \\ &\cup^2 \quad \cup^3 \\ [1/x_0, +\infty) &\qquad \qquad [x_0, +\infty) \end{aligned}$$

Inclusion <sup>1</sup> follows from the fact that  $[1/x_0, +\infty) \supset [2/x_0, +\infty)$  (notice that for  $x > 0$  the graph of the map  $2/x$  is “above” the one of  $1/x$ ). Inclusion <sup>2</sup> and <sup>3</sup> follow from the fact that  $\phi_{x_0}(y)$  has positive leading coefficient and the computation of the images of  $1/x_0$  and  $2/x_0$  through  $\phi_{x_0}(y)$ :

$$\begin{aligned}\phi_{x_0}\left(\frac{1}{x_0}\right) &= \frac{1}{x_0} + \left((x_0^3 - 4x_0)\frac{1}{x_0^2} + (4 - 2x_0^2)\frac{1}{x_0} + x_0\right) = \frac{1}{x_0} \\ \phi_{x_0}\left(\frac{2}{x_0}\right) &= \left(\frac{8}{x_0} + (x_0^3 - 4x_0)\frac{4}{x_0^2} + (4 - 2x_0^2)\frac{2}{x_0}\right) + x_0 = x_0.\end{aligned}$$

Now it is clear that:

$$\mathcal{B}_x = [y_x, +\infty) \subset \phi_x([1/x, +\infty)).$$

Then we can prove the desired inclusion in this way:

$$\mathcal{B} = \bigsqcup_{x>0} (\{x\} \times \mathcal{B}_x) \subset \bigsqcup_{x>0} (\{x\} \times \phi_x([1/x, +\infty))) = \bigsqcup_{x>0} \mathcal{G}(\{x\} \times [1/x, +\infty)) = \mathcal{G}(\mathcal{A}),$$

which concludes the proof.  $\square$

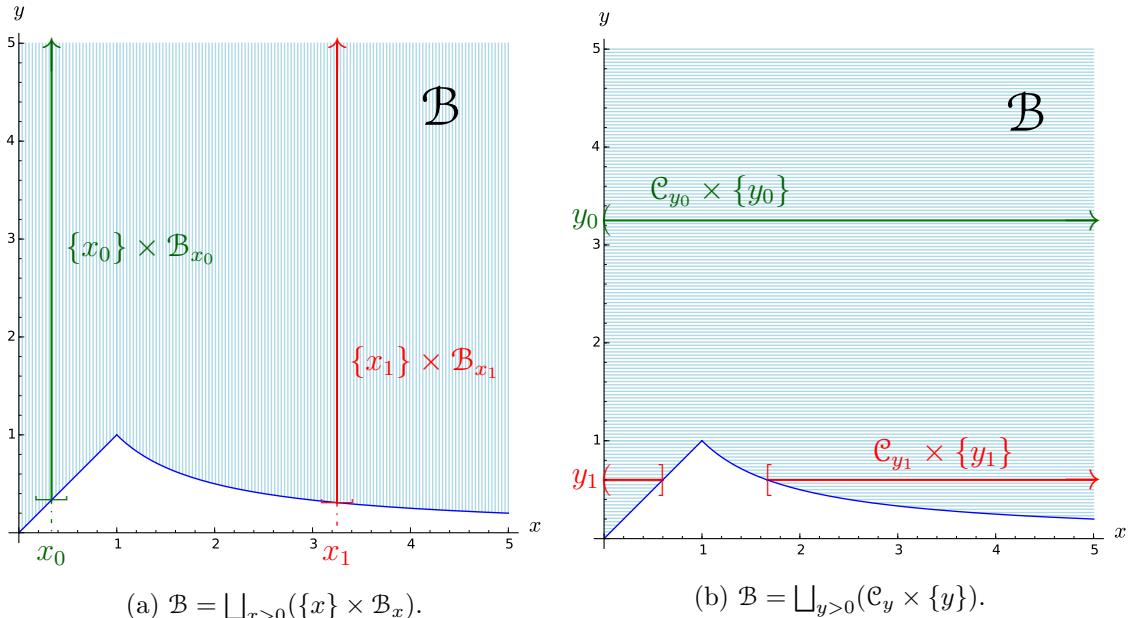


Figure 3.3: Idea of how we represent the set  $\mathcal{B}$  in Lemmas 3.2 and 3.3.

### 3.2.3 The third lemma

**Lemma 3.3.** *The polynomial map*

$$\mathcal{H} := (\mathcal{H}_1, \mathcal{H}_2) : \mathbb{R}^2 \rightarrow \mathbb{R}^2, (x, y) \mapsto (x(xy - 2)^2 + \frac{1}{2}xy^2, y)$$

satisfies  $\mathcal{H}(\mathcal{B}) = \mathcal{H}(\mathcal{Q}) = \mathcal{Q}$ .

*Proof.* The inclusion  $\mathcal{H}(\mathcal{B}) \subset \mathcal{H}(\mathcal{Q})$  is trivial because  $\mathcal{B} \subset \mathcal{Q}$ . As for  $\mathcal{H}(\mathcal{Q}) \subset \mathcal{Q}$ , since  $\mathcal{H}_1$  and  $\mathcal{H}_2$  are strictly positive on  $\mathcal{Q}$  the inclusion holds.

We proceed now to prove that  $\mathcal{Q} \subset \mathcal{H}(\mathcal{B})$  and therefore get  $\mathcal{Q} \subset \mathcal{H}(\mathcal{B}) \subset \mathcal{H}(\mathcal{Q}) \subset \mathcal{Q}$ , which implies that  $\mathcal{H}(\mathcal{B}) = \mathcal{H}(\mathcal{Q}) = \mathcal{Q}$ .

Firstly, notice that the set  $\mathcal{B}$  can be expressed as follows:

$$\mathcal{B} = \bigsqcup_{y>0} (\mathcal{C}_y \times \{y\}), \quad \text{where } \mathcal{C}_y := \begin{cases} (0, +\infty) & \text{if } y \geq 1, \\ (0, y] \cup [1/y, +\infty) & \text{if } 0 < y < 1. \end{cases}$$

Intuitively, we are “slicing” the set  $\mathcal{B}$  horizontally, depending on  $y$ . Looking at the definition of  $\mathcal{H}_1$  we define for each  $y > 0$  the polynomial  $\psi_y$  dependent on the variable  $x$ :

$$\psi_y(x) := x(yx - 2)^2 + \frac{1}{2}y^2x = y^2x^3 - 4yx^2 + (4 + \frac{1}{2}y^2)x.$$

Notice first that  $\psi_y(x)$  has odd degree, positive leading coefficient and the following properties:

- (i)  $\lim_{x \rightarrow +\infty} \psi_y(x) = +\infty.$
- (ii)  $\psi_y(0) = 0.$
- (iii)  $\psi_y(y) = y((y^2 - 2)^2 + \frac{1}{2}y^2) > y$  for  $0 < y < 1$ .
- (iv)  $\psi_y\left(\frac{2}{y}\right) = y.$

Property (iii) holds because  $(y^2 - 2)^2 + \frac{1}{2}y^2 > 1$  when  $0 < y < 1$ . Next, we are going to show that:

$$\psi_y(\mathcal{C}_y) = \begin{cases} \psi_y((0, +\infty)) =^1 (0, +\infty) & \text{if } y \geq 1, \\ \psi_y((0, y] \cup [\frac{1}{y}, +\infty)) \supset^2 (0, \psi_y(y)] \cup [\psi_y(\frac{2}{y}), +\infty) =^3 (0, +\infty) & \text{if } 0 < y < 1. \end{cases}$$

Equality <sup>1</sup> holds because of (i), (ii) and the fact that  $\psi_y$  is strictly positive on  $(0, +\infty)$ . Inclusion <sup>2</sup> follows from the fact that  $[1/y_0, +\infty) \supset [2/y_0, +\infty)$  for each fixed  $y_0$ . Finally, equality <sup>3</sup> holds because of (iii) and (iv):  $\psi_y(y) > y = \psi_y(\frac{2}{y})$ .

Then we can write:

$$\mathcal{Q} = \bigsqcup_{y>0} ((0, +\infty) \times \{y\}) \subset \bigsqcup_{y>0} (\psi_y(\mathcal{C}_y) \times \{y\}) = \bigsqcup_{y>0} \mathcal{H}(\mathcal{C}_y \times \{y\}) = \mathcal{H}(\mathcal{C}_y),$$

which concludes the proof. □

### 3.2.4 The second proof for the open quadrant problem

To wrap up this chapter, we can now write the proof of Theorem 1.11 relying on Lemmas 3.1, 3.2 and 3.3:

[Proof of Theorem 1.11] Applying the lemmas, we deduce that

$$\mathcal{Q} \stackrel{3.3}{=} \mathcal{H}(\mathcal{B}) \stackrel{3.2}{\subset} (\mathcal{H} \circ \mathcal{G})(\mathcal{A}) \stackrel{3.1}{\subset} (\mathcal{H} \circ \mathcal{G} \circ \mathcal{F})(\mathbb{R}^2) \stackrel{3.1}{\subset} (\mathcal{H} \circ \mathcal{G})(\mathcal{Q}) \stackrel{3.2}{\subset} \mathcal{H}(\mathcal{Q}) \stackrel{3.3}{=} \mathcal{Q},$$

which means

$$(\mathcal{H} \circ \mathcal{G} \circ \mathcal{F})(\mathbb{R}^2) = \mathcal{Q},$$

as required. □

# 4

## A topological proof for the open quadrant $\mathcal{Q}$ problem

### 4.1 A topological perspective

Although Fernando, Gamboa and Ueno proved that the open quadrant  $\mathcal{Q}$  is the image of  $\mathbb{R}^2$  under two distinct polynomial maps using very different techniques, they developed quite recently in [FGU] a completely new proof that relies on algebraic topology methods applied to a new polynomial map  $\mathcal{F} : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ . The genesis of this third proof appears in Ueno's Ph.D. [U] written under the supervision of Fernando.

In this chapter we will build up the topological machinery necessary to prove Theorem 1.11 in a very geometrical way, and completely avoiding the aid of computer calculations.

#### 4.1.1 The new polynomial map

Without further ado, the polynomial map we are going to work with is denoted  $\mathcal{F}$  and it is defined as:

$$\mathcal{F}(x, y) := ((x^2y^4 + x^4y^2 - y^2 - 1)^2 + x^6y^4, (x^6y^2 + x^2y^2 - x^2 - 1)^2 + x^6y^4).$$

As we announced before, in Section 4.2 we will introduce some definitions and prove some facts about a family of topological subspaces of  $\mathbb{R}^3$  homeomorphic to a closed disc, and paths that go through them.

Afterwards, after factorizing  $\mathcal{F} = f_2 \circ f_1$  with  $f_2 = h \circ g$ , it will be clear that the proof is reduced to check that the boundaries of two certain topological subspaces of  $\mathbb{R}^3$  homeomorphic to a closed disc meet  $g(\overline{\mathcal{Q}})$ , where  $\overline{\mathcal{Q}} := \{x \geq 0, y \geq 0\}$  is the closed quadrant.

### 4.2 Topological tools

#### 4.2.1 Some definitions

We begin by defining the key topological manifolds we are going to work with.

**Definitions 4.1.** (i) Let  $\mathbb{D}_A := \{(x, y) : x^2 + y^2 \leq A^2\} \subset \mathbb{R}^2$  be the closed disc of center the origin and radius  $A > 0$ . A *warped disc* is a subset

$$\mathcal{D}_{A,\xi} := \{(x, y, z) : x^2 + y^2 \leq A^2, z = \xi(x, y)\} \subset \mathbb{R}^3$$

where  $\xi : \mathbb{R}^2 \rightarrow \mathbb{R}$  is a continuous function. Let  $\zeta$  be the homeomorphism

$$\zeta : \mathbb{R}^3 \rightarrow \mathbb{R}^3, (x, y, z) \mapsto (x, y, z - \xi(x, y)).$$

Clearly  $\zeta(\mathcal{D}_{A,\xi}) = \mathbb{D}_A \times \{0\}$  is the projection or “flattening” of the warped disc  $\mathcal{D}_{A,\xi}$  onto the plane  $\{z = 0\} \subset \mathbb{R}^3$ . We will also call a warped disc the image of  $\mathcal{D}_{A,\xi}$  under a permutation of the variables of  $\mathbb{R}^3$ ; namely

$$\begin{aligned} \mathcal{D}_{A,\xi} &:= \{(x, y, z) : y^2 + z^2 \leq A^2, x = \xi(y, z)\} \text{ or} \\ \mathcal{D}_{A,\xi} &:= \{(x, y, z) : x^2 + z^2 \leq A^2, y = \xi(x, z)\}, \end{aligned}$$

with  $\xi$  defined as above.

(ii) For each  $\varepsilon > 0$  let us define the following open neighborhood of  $\mathbb{D}_A$ :

$$\mathbb{D}_A(\varepsilon) := \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 < (A + \varepsilon)^2\} \times (-\varepsilon, \varepsilon) \subset \mathbb{R}^3.$$

As  $\zeta$  is a homeomorphism,  $\mathcal{D}_{A,\xi}(\varepsilon) := \zeta^{-1}(\mathbb{D}_A(\varepsilon)) \subset \mathbb{R}^3$  is an open neighborhood of the warped disc  $\mathcal{D}_{A,\xi}$ .

We refer the reader to Appendix A.3 for further details concerning the *fundamental group* of a (path connected) topological space. Next we explain the meaning of *meeting transversally once a warped disc*, which is just a formalization of our intuition.

**Definitions 4.2.** (i) A path  $\alpha : [a, b] \rightarrow \mathbb{R}^3$  meets transversally once the warped disc  $\mathcal{D}_{A,\xi}$  if there exist  $s_0 \in (a, b)$  and  $\varepsilon > 0$  such that

$$J := \alpha^{-1}(\mathcal{D}_{A,\xi}(\varepsilon)) = (s_0 - \varepsilon, s_0 + \varepsilon)$$

is an open subinterval of  $[a, b]$  and

$$(\zeta \circ \alpha)|_J(t) = (0, 0, t - s_0),$$

where obviously  $(\zeta \circ \alpha)|_J(t) \subset \mathbb{D}_A(\varepsilon)$  because  $\alpha|_J(t) \in \zeta^{-1}(\mathbb{D}_A(\varepsilon)) = \mathcal{D}_{A,\xi}(\varepsilon)$ .

(ii) Let  $C$  be a topological space homeomorphic to a closed disc and let  $\phi : C \rightarrow \mathbb{R}^3$  be a continuous map. The restriction  $\partial\phi := \phi|_{\partial C}$  is called the *boundary map* of  $\phi$ . We say that the boundary map  $\partial\phi$  meets transversally once a warped disc  $\mathcal{D}_{A,\xi} \subset \mathbb{R}^3$  if there exists a parameterization  $\beta : [a, b] \rightarrow \partial C \cong \mathbb{S}^1 := \{x^2 + y^2 = 1\} \subset \mathbb{R}^2$  such that the path  $\alpha := \phi \circ \beta : [a, b] \rightarrow \mathbb{R}^3$  meets transversally once the warped disc  $\mathcal{D}_{A,\xi}$ .

**Remark 4.3.** If the path  $\alpha : [a, b] \rightarrow \mathbb{R}^3$  meets transversally once the warped disc  $\mathcal{D}_{A,\xi}$ , then  $\alpha([a, b]) \cap \partial\mathcal{D}_{A,\xi} = \emptyset$ . This will be helpful in the proof of Proposition 4.7.

### 4.2.2 Auxiliary results

We proceed now to prove some facts that will be helpful later on to prove Theorem 1.11. Firstly, we prove in Lemma 4.4 that if a loop meets transversally once a warped disc, then the topological space  $\mathbb{R}^3$  minus the border of the warped disc has  $\mathbb{Z}$  as fundamental group. Figure 4.1 helps to visualize the idea of the proof.

**Lemma 4.4.** *Let  $\mathcal{D}_{A,\xi} \subset \mathbb{R}^3$  be a warped disc and let  $X := \mathbb{R}^3 \setminus \partial\mathcal{D}_{A,\xi}$ . Let  $\alpha : [a, b] \rightarrow X$  be a loop with base point  $x_0 \in X$  that meets transversally once  $\mathcal{D}_{A,\xi}$ . Then  $[\alpha]$  is a generator of  $\pi_1(X, x_0) \cong \mathbb{Z}$ .*

*Proof.* Using the notation introduced in Definitions A.13 and 4.2, let  $s_0 \in (a, b)$  and  $\varepsilon > 0$  be such that

$$J := \alpha^{-1}(\mathcal{D}_{A,\xi}(\varepsilon)) = (s_0 - \varepsilon, s_0 + \varepsilon) \subset [a, b]$$

and  $(\zeta \circ \alpha)|_J(t) = (0, 0, t - s_0)$ . We can assume  $s_0 = 0$  after a reparameterization of  $\alpha$ ; so  $J = (-\varepsilon, \varepsilon)$  and  $(\zeta \circ \alpha)|_J(t) = (0, 0, t)$ . Note that we can write the interval  $[a, b]$  as  $[a, -\varepsilon] \cup (-\varepsilon, \varepsilon) \cup [\varepsilon, b]$ .

Since  $\zeta$  is a homeomorphism of  $\mathbb{R}^3$ , instead of working in the warped disc we will prove the statement for  $\mathbb{D}_A$  where  $\beta := \zeta \circ \alpha$ ,  $Y := \mathbb{R}^3 \setminus \partial\mathbb{D}_A$  and the new base point is  $y_0 := \beta(-\varepsilon) = (0, 0, -\varepsilon)$ . Consider the path  $\gamma : [0, 1] \rightarrow \mathbb{R}^3$  (see Figure 4.1) defined in the following way:

$$\gamma(t) := \begin{cases} (3(A + \varepsilon)t, 0, \varepsilon) & \text{if } 0 \leq t \leq \frac{1}{3}, \\ (A + \varepsilon, 0, \varepsilon - (t - \frac{1}{3})6\varepsilon) & \text{if } \frac{1}{3} < t \leq \frac{2}{3}, \\ (A + \varepsilon - 3(A + \varepsilon)(t - \frac{2}{3}), 0, -\varepsilon) & \text{if } \frac{2}{3} < t \leq 1. \end{cases}$$

Let  $\beta_0 := \beta|_{[-\varepsilon, \varepsilon]} = \beta|_{(-\varepsilon, \varepsilon)}$  and  $\beta_1 := \beta|_{[\varepsilon, b]} * \beta|_{[a, -\varepsilon]}$ . Using the properties of path concatenation we get:

$$[\beta] = [\beta_0 * \beta_1] = [\beta_0 * (\gamma * \gamma^{-1}) * \beta_1] = [\beta_0 * \gamma] \cdot [\gamma^{-1} * \beta_1].$$

We claim that  $[\beta_0 * \gamma] = g$  and  $[\gamma^{-1} * \beta_1] = e$ , where  $g$  and  $e$  are, respectively, a generator of  $\pi_1(Y, y_0) \cong \mathbb{Z}$  and the identity element.

Firstly, the loop  $\gamma^{-1} * \beta_1$  with base point  $y_0$  is contained in  $\mathbb{R}^3 \setminus \mathbb{D}_A$ , which is a simply connected space. Thus the loop  $\gamma^{-1} * \beta_1$  is null-homotopic and  $[\gamma^{-1} * \beta_1] = e \in \pi_1(Y, y_0)$ .

On the other hand, the class  $[\beta_0 * \gamma]$  generates  $\pi_1(Y, y_0)$ . Indeed,  $Y$  has as deformation retract the set  $Z := \partial\mathbb{D}_A(\varepsilon) \cup I_\varepsilon$  where  $\partial\mathbb{D}_A(\varepsilon)$  is the boundary of  $\mathbb{D}_A(\varepsilon)$  and we denote  $I_\varepsilon := \{(0, 0, z) : -\varepsilon \leq z \leq \varepsilon\}$ . Finally, since  $Z$  is homeomorphic to the union of a sphere and one of its diameters we deduce from Lemma 4.5 that  $[\beta_0 * \gamma]$  is a generator of  $\pi_1(Z, y_0) \cong \pi_1(Y, y_0) \cong \mathbb{Z}$ , as required.  $\square$

**Lemma 4.5.** *Let  $S := \{x^2 + y^2 + z^2 = A^2\} \subset \mathbb{R}^3$  be the sphere of center the origin and radius  $A > 0$ , let  $D$  be one of its diameters and let  $M := S \cup D$ . If  $x_0 \in M$  then  $\pi_1(M, x_0) \cong \mathbb{Z}$ .*

*Proof.* Let  $x_0 \in S \setminus D$  and let  $U := M \setminus \{x_0\}$ . Also, let  $V$  be an small enough neighborhood of  $x_0$  homeomorphic to a closed disc such that it does not meet the diameter  $D$ . It is clear that  $M \cong U \cup V$ . Consider the group homomorphism

$$\varphi : \pi_1(U \cap V, x_0) \rightarrow \pi_1(U, x_0)$$

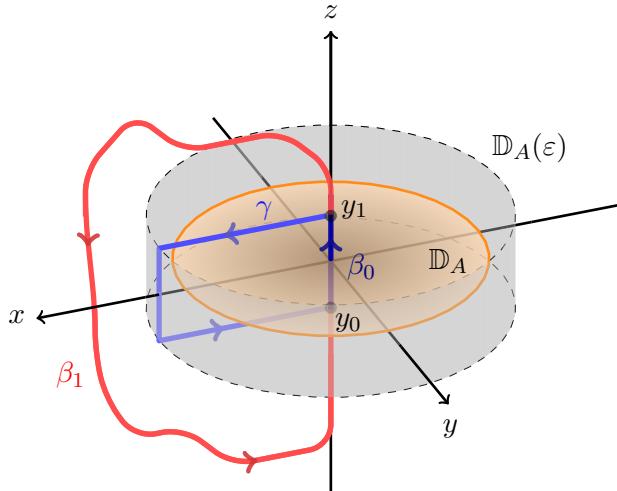


Figure 4.1: The loop  $\beta = \beta_0 * \beta_1$  meets transversally once the disk  $D_A$ . Here  $y_0 := \beta(-\varepsilon)$  and  $y_1 := \beta(\varepsilon)$ , so the path  $\beta_0$  goes from  $y_0$  to  $y_1$  (in purple),  $\gamma$  goes from  $y_1$  to  $y_0$  (in blue) and  $\beta_1$  goes from  $y_1$  to  $y_0$  (in red).

induced by the inclusion map  $j : (U \cap V) \hookrightarrow U$ . As  $V$  is a simply connected space, we deduce from the Theorem of Seifert and Van Kampen [M, Chapter IV: §4, page 95] that

$$\pi_1(M, x_0) = \pi_1(U, x_0) / \text{im}(\varphi).$$

Next, notice that  $U$  is homeomorphic to the union of a closed disc with an arc that joints two points of the disc and that doesn't touch its border. Thus,  $U$  has  $S^1$  as deformation retract and  $\pi_1(U, x_0) \cong \mathbb{Z}$ . In addition, as  $U \cap V \cong V \setminus \{x_0\}$  the set  $U \cap V$  is homeomorphic to a punctured disc, so  $\pi_1(U \cap V, x_0) \cong \pi_1(S^1, x_0) \cong \mathbb{Z}$ , and notice that a generator of this fundamental group is a loop contained in  $S \setminus \{x_0\} \subset U$ . As  $S \setminus \{x_0\}$  is contractible,  $\varphi(\pi_1(U \cap V, x_0)) = 0$ , hence  $\pi_1(M, x_0) = \pi_1(U, x_0)$  is generated by the class of a loop  $\alpha$  as in Figure 4.2. Consequently,  $\pi_1(M, x_0) = \mathbb{Z}$  as required.  $\square$

**Lemma 4.6.** *Let  $C$  be a topological space homeomorphic to a closed disc and let  $z_0 \in \partial C$ . Let  $\phi : C \rightarrow X$  be a continuous map and consider a parameterization  $\beta : [a, b] \rightarrow \partial C$  such that  $\beta(a) = z_0 = \beta(b)$ . Then  $[\phi \circ \beta]$  is the identity element of  $\pi_1(X, \phi(z_0))$ .*

*Proof.* Let  $\psi : C \rightarrow \{x^2 + y^2 \leq 1\} \subset \mathbb{R}^2$  be a homeomorphism. Let  $H$  be the continuous map

$$H : [0, 1] \times [a, b] \rightarrow X, (\rho, t) \mapsto (\phi \circ \psi^{-1})(\rho \cdot (\psi \circ \beta)(t) + (1 - \rho) \cdot \psi(z_0)).$$

Notice that  $H(0, t) = \phi(z_0)$  and  $H(1, t) = \phi(\beta(t))$ . Thus  $H$  is a homotopy map between  $\phi \circ \beta$  and the constant path, as required.  $\square$

**Proposition 4.7.** *Let  $C$  be a topological space homeomorphic to a closed disc and let  $\phi : C \rightarrow \mathbb{R}^3$  be a continuous map. If the boundary map  $\partial\phi$  of  $\phi$  meets transversally once a warped disc  $\mathcal{D} \subset \mathbb{R}^3$ , then  $\partial\mathcal{D} \cap \phi(C) \neq \emptyset$ .*

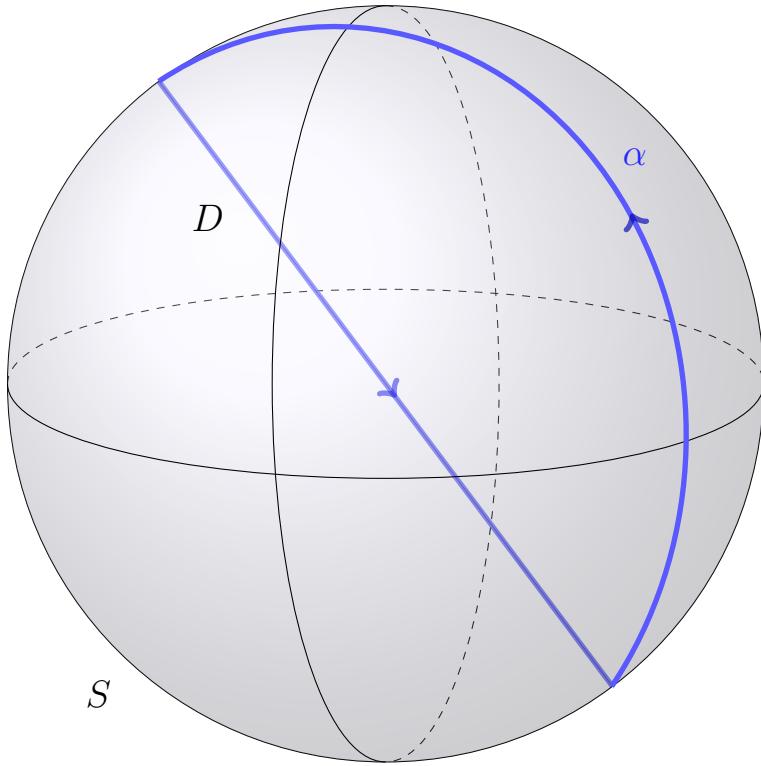


Figure 4.2: The set  $M := S \cup D$  and a generator  $\alpha$  of  $\pi_1(M, x_0)$  from Lemma 4.5.

*Proof.* Suppose instead that  $\partial\mathcal{D} \cap \phi(\overset{\circ}{C}) = \emptyset$ . Since  $\partial\phi$  meets transversally once  $\mathcal{D}$  it follows from Remark 4.3 that the image of  $\partial C$  under the map  $\phi$  does not intersect  $\partial\mathcal{D}$ . This alongside our initial assumption gives us that  $\phi(C) \subset X := \mathbb{R}^3 \setminus \partial\mathcal{D}$ .

Let  $\beta : [a, b] \rightarrow \partial C$  be a parameterization such that  $\beta(a) = z_0 = \beta(b)$  for some point  $z_0 \in \partial C$  such that  $\phi \circ \beta$  meets transversally once  $\mathcal{D}$ . By Lemma 4.6 the class  $[\phi \circ \beta]$  is the identity element of  $\pi_1(X, \phi(z_0))$ . However  $[\phi \circ \beta]$  generates, by Lemma 4.4, the group  $\pi_1(X, \phi(z_0)) \cong \mathbb{Z}$ , which leads to a contradiction. Thus,  $\partial\mathcal{D} \cap \phi(\overset{\circ}{C}) \neq \emptyset$ , as desired.  $\square$

## 4.3 The topological proof for the open quadrant problem

### 4.3.1 Factorizing $\mathcal{F}$

The authors observed in [FGU, Section 3] that the map  $\mathcal{F}$  can be factorized as  $\mathcal{F} = f_2 \circ f_1$ , where the polynomial maps  $f_1, f_2$  are defined as follows:

$$\begin{aligned} f_1 : \mathbb{R}^2 &\rightarrow \mathbb{R}^2, \quad (x, y) \mapsto (x^2, y^2), \\ f_2 : \mathbb{R}^2 &\rightarrow \mathbb{R}^2, \quad (x, y) \mapsto ((xy^2 + x^2y - y - 1)^2 + x^3y^2, (x^3y + xy - x - 1)^2 + x^3y^2). \end{aligned}$$

Since  $f_1(\mathbb{R}^2) = \overline{\mathcal{Q}}$  it is then clear that we just need to prove that

$$f_2(\overline{\mathcal{Q}}) = \mathcal{Q}. \quad (4.1)$$

Note that both components of  $f_2$  are strictly positive on  $\overline{\mathcal{Q}}$  so the inclusion  $f_2(\overline{\mathcal{Q}}) \subset \mathcal{Q}$  holds. We need to prove the converse inclusion:

$$\mathcal{Q} \subset f_2(\overline{\mathcal{Q}}). \quad (4.2)$$

Indeed  $f_2 = h \circ g$ , where the continuous semialgebraic maps  $g, h$  are defined as:

$$\begin{aligned} g : \overline{\mathcal{Q}} &\rightarrow \mathbb{R}^3, (x, y) \mapsto (xy^2 + x^2y - y - 1, x^{3/2}y, x^3y + xy - x - 1), \\ h : \mathbb{R}^3 &\rightarrow \mathbb{R}^2, (x, y, z) \mapsto (x^2 + y^2, y^2 + z^2). \end{aligned}$$

In this way we can reduce the proof to show that for each pair of positive real numbers  $(A^2, B^2) \in \mathcal{Q}$  there exists a point  $(x_0, y_0) \in \overline{\mathcal{Q}}$  such that  $(h \circ g)(x_0, y_0) = (A^2, B^2)$ . This is equivalent to check that

$$\boxed{\forall (A^2, B^2) \in \mathcal{Q} : h^{-1}(\{(A^2, B^2)\}) \cap g(\overline{\mathcal{Q}}) \neq \emptyset.}$$

Let  $\mathcal{S} := g(\overline{\mathcal{Q}})$  and fix values  $B \geq A > 0$ . By the definition of  $h$  we know that

$$\begin{aligned} h^{-1}(\{(A^2, B^2)\}) &= \{x^2 + y^2 = A^2, y^2 + z^2 = B^2\} \subset \mathbb{R}^2 \text{ and} \\ h^{-1}(\{(B^2, A^2)\}) &= \{y^2 + z^2 = A^2, x^2 + y^2 = B^2\} \subset \mathbb{R}^2. \end{aligned}$$

Both of them are the intersection of two orthogonal cylinders. Besides, these sets contain, respectively, the boundaries of the warped discs

$$\mathcal{D}_1 := \{x^2 + y^2 \leq A^2, z = \xi_1(x, y)\} \subset \mathbb{R}^3 \quad \text{and} \quad (4.3)$$

$$\mathcal{D}_2 := \{y^2 + z^2 \leq A^2, x = \xi_2(y, z)\} \subset \mathbb{R}^3, \quad (4.4)$$

for the continuous and semialgebraic functions

$$\xi_1 : \mathbb{R}^2 \rightarrow \mathbb{R}, (x, y) \mapsto \sqrt{B^2 - \min\{y^2, B^2\}}, \quad (4.5)$$

$$\xi_2 : \mathbb{R}^2 \rightarrow \mathbb{R}, (y, z) \mapsto \sqrt{B^2 - \min\{y^2, B^2\}}, \quad (4.6)$$

see Figure 4.3.

Now it is clear that our aim is to prove the following fact:

$$\boxed{\text{For fixed values } B \geq A > 0 \text{ the intersections } \partial\mathcal{D}_1 \cap \mathcal{S} \text{ and } \partial\mathcal{D}_2 \cap \mathcal{S} \text{ are non-empty.}} \quad (\diamond)$$

### 4.3.2 Proof of Claim $(\diamond)$

In order to prove  $(\diamond)$ , we first introduce an auxiliary map  $\phi$  and study some of its properties. Let  $\phi := (\phi_1, \phi_2, \phi_3) : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ , whose components are defined as

$$\phi_1(\rho, \theta) := \cos \theta \sin \theta (\cos \theta - \sin \theta)^2 + \rho(2 \cos^4 \theta \sin \theta + \cos \theta \sin^4 \theta + \cos^5 \theta) + \rho^2 \cos^5 \theta \sin \theta,$$

$$\phi_2(\rho, \theta) := \sqrt{\cos \theta \sin \theta} (\cos \theta + \sin \theta + \rho \cos \theta \sin \theta),$$

$$\phi_3(\rho, \theta) := \rho \sin \theta.$$

Let also  $\mathcal{R} := [0, +\infty) \times (0, \frac{\pi}{2})$  and let  $\overline{\mathcal{R}} := [0, +\infty) \times [0, \frac{\pi}{2}]$  be its closure.

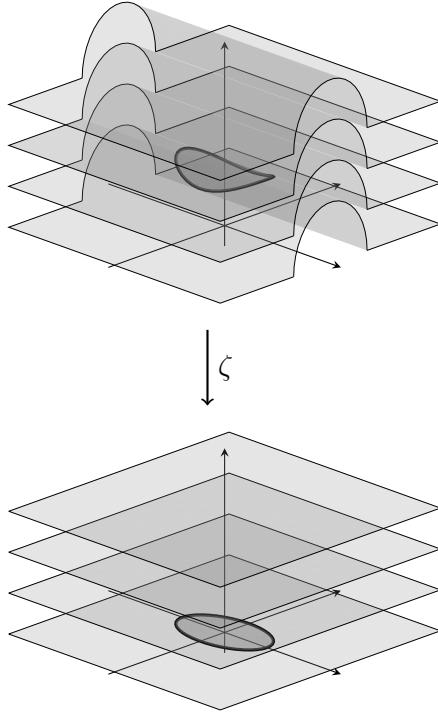


Figure 4.3: The homeomorphism  $\zeta$  for  $\xi_1(x, y) := \sqrt{B^2 - \min(y^2, B^2)}$  acting on  $\mathbb{R}^3$ . Notice how it “flattens” the warped disc  $\mathcal{D}_1$ .

**Lemma 4.8.** With the notation introduced above,  $\phi(\mathcal{R}) \subset \mathcal{S}$ .

*Proof.* Let  $\psi$  be the analytic map

$$\psi := (\psi_1, \psi_2) : \mathcal{R} \rightarrow \mathcal{Q}, \quad (\rho, \theta) \mapsto \left( \frac{\sin \theta}{\cos \theta}, \frac{(\cos \theta + \sin \theta + \rho \cos \theta \sin \theta) \cos^2 \theta}{\sin \theta} \right).$$

Note that  $g \circ \psi = \phi|_{\mathcal{R}}$ . If  $\theta \in (0, \frac{\pi}{2})$  then  $\cos \theta, \sin \theta \in (0, 1)$ , so  $\psi$  is well-defined. Since  $\rho \in [0, +\infty)$  we deduce that  $\psi_1, \psi_2$  just attain positive values, so  $\psi(\mathcal{R}) \subset \mathcal{Q} \subset \overline{\mathcal{Q}}$ . Thus,  $\phi(\mathcal{R}) = g(\psi(\mathcal{R})) \subset g(\overline{\mathcal{Q}}) = \mathcal{S}$ .  $\square$

**Lemma 4.9.** For all  $(\rho, \theta) \in \overline{\mathcal{R}}$  the inequality  $\phi_1^2(\rho, \theta) + \phi_3^2(\rho, \theta) \geq \frac{\rho^2}{4}$  holds. Thus,

$$\forall (\rho, \theta) \in \overline{\mathcal{R}} : \text{ such that } \text{dist}(\phi(\rho, \theta), 0) \geq \frac{\rho}{2}. \quad (4.7)$$

*Proof.* Notice first that if the inequality  $\phi_1^2(\rho, \theta) + \phi_3^2(\rho, \theta) \geq \frac{\rho^2}{4}$  holds, then

$$\text{dist}(\phi(\rho, \theta), 0) = \sqrt{\phi_1^2(\rho, \theta) + \phi_3^2(\rho, \theta) + \phi_2^2(\rho, \theta)} \geq \sqrt{\frac{\rho^2}{4} + \phi_2^2(\rho, \theta)} \geq \sqrt{\frac{\rho^2}{4}} \geq \frac{\rho}{2},$$

so 4.7 holds. Now, for the first part, if  $(\rho, \theta) \in \overline{\mathcal{R}}$  then  $\cos \theta, \sin \theta \in [0, 1]$  and  $\rho \geq 0$ , so we

have:

$$\begin{aligned}\phi_1(\rho, \theta) &\geq \rho(\cos \theta \sin^4 \theta + \cos^5 \theta) = \rho \cos \theta (\cos^4 \theta + \sin^4 \theta) \\ &=^1 \rho \cos \theta (1 - 2 \cos^2 \theta \sin^2 \theta) =^2 \rho \cos \theta \left(1 - \frac{\sin^2(2\theta)}{2}\right) \geq^3 \frac{\rho}{2} \cos \theta.\end{aligned}$$

In the equality <sup>1</sup> we have used that  $1^2 = (\cos \theta^2 + \sin \theta^2)^2 = \cos \theta^4 + \sin \theta^4 + 2 \cos \theta^2 \sin \theta^2$ . Equality <sup>2</sup> holds because  $2 \cos \theta \sin \theta = \sin(2\theta)$ . Also, as  $\theta \in [0, \frac{\pi}{2}]$  we have  $2\theta \in [0, \pi]$ , so  $\sin(2\theta) \in [0, 1]$ . Thus  $\sin^2(2\theta) \in [0, 1]$  and  $\frac{\sin(2\theta)}{2} \in [0, \frac{1}{2}]$  and inequality <sup>3</sup> holds.

In addition,  $\phi_3(\rho, \theta) = \rho \sin \theta \geq \frac{\rho}{2} \sin \theta$ , so we obtain the desired inequality:

$$\phi_1^2(\rho, \theta) + \phi_3^2(\rho, \theta) \geq \frac{\rho^2}{4} \cos^2 \theta + \frac{\rho^2}{4} \sin^2 \theta = \frac{\rho^2}{4}.$$

□

**Construction 4.10.** Recall the elementary equalities

$$\sin \theta = \cos \left( \frac{\pi}{2} - \theta \right) \text{ and } \cos \theta = \sin \left( \frac{\pi}{2} - \theta \right) \quad \forall \theta \in \mathbb{R}.$$

Thus  $\phi(0, \theta) = \phi(0, \frac{\pi}{2} - \theta)$  for  $\theta \in [0, \frac{\pi}{2}]$ . This induces us to make an identification described as follows. Fix  $M > 0$  and consider the rectangle

$$\overline{\mathcal{R}}_M := [0, M] \times [0, \frac{\pi}{2}] \subset \overline{\mathcal{R}}.$$

Let  $\phi_M = \phi|_{\overline{\mathcal{R}}_M}$ . Now, identify the points  $(0, \theta)$  and  $(0, \frac{\pi}{2} - \theta)$  for  $\theta \in [0, \frac{\pi}{2}]$  (see Figure 4.4) and endow the quotient space  $\tilde{\mathcal{R}}_M$  with the quotient topology. Observe that, as a topological manifold with boundary, the interior of  $\tilde{\mathcal{R}}_M$  is the quotient space  $\tilde{\mathcal{R}}_M$  obtained identifying the points  $(0, \theta)$  and  $(0, \frac{\pi}{2} - \theta)$  of  $\mathcal{R}_M := [0, M] \times (0, \frac{\pi}{2})$ , where  $\theta \in (0, \frac{\pi}{2})$ .

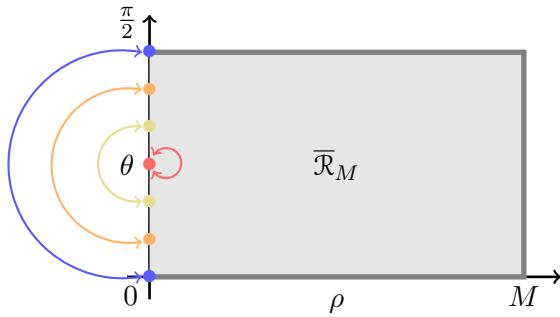


Figure 4.4: Idea of the identification in Construction 4.10.

Consider the canonical projection  $\pi_M : \overline{\mathcal{R}}_M \rightarrow \tilde{\mathcal{R}}_M$ , which is continuous. Since  $\phi_M$  is compatible with  $\pi_M$ , there exists a continuous map  $\tilde{\phi}_M : \tilde{\mathcal{R}}_M \rightarrow \mathbb{R}^3$  such that the following diagram is commutative:

$$\begin{array}{ccc}
 \mathcal{R}_M & \hookrightarrow & \bar{\mathcal{R}}_M \\
 \pi_M|_{\mathcal{R}_M} \downarrow & & \downarrow \pi_M \\
 \tilde{\mathcal{R}}_M & \hookrightarrow & \tilde{\bar{\mathcal{R}}}_M \xrightarrow{\tilde{\phi}_M} \mathbb{R}^3
 \end{array}$$

In addition  $\tilde{\phi}_M(\tilde{\mathcal{R}}_M) = \phi(\mathcal{R}_M) \subset \phi(\mathcal{R}) \subset \mathcal{S}$  because of Lemma 4.8 and the fact that  $\mathcal{R}_M \subset \mathcal{R}$ .

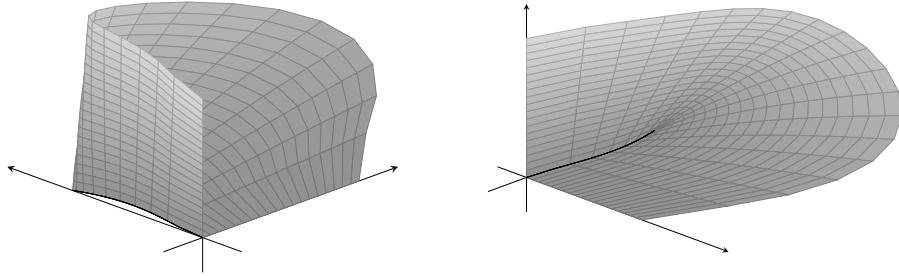


Figure 4.5: Left and right views of  $\phi_M(\mathcal{R}_M) \subset \mathcal{S}$ .

**Lemma 4.11.** The set  $\tilde{\bar{\mathcal{R}}}_M$  is homeomorphic to a disc and its boundary is the set

$$\pi_M(\{\rho = M\} \cup \{\theta = 0\} \cup \{\theta = \frac{\pi}{2}\}).$$

*Proof.* Identify  $\mathbb{R}^2$  with  $\mathbb{C}$  (interchanging the order of the variables  $(\rho, \theta) \rightsquigarrow (\theta, \rho)$ ) and consider the continuous map defined as

$$\mu : \mathbb{C} \rightarrow \mathbb{C}, z := \theta + i\rho \mapsto w := u + iv = \left(\frac{4}{\pi}z - 1\right)^2,$$

where  $i$  is the imaginary unit  $\sqrt{-1}$ . The restriction  $\mu|_{\{\rho \geq 0\}} : \{\rho > 0\} \rightarrow \mathbb{C} \setminus ([0, +\infty) \times \{0\})$  is a homeomorphism. Now observe that the image of  $\mathcal{R}_M \setminus \{\rho = 0\}$  is

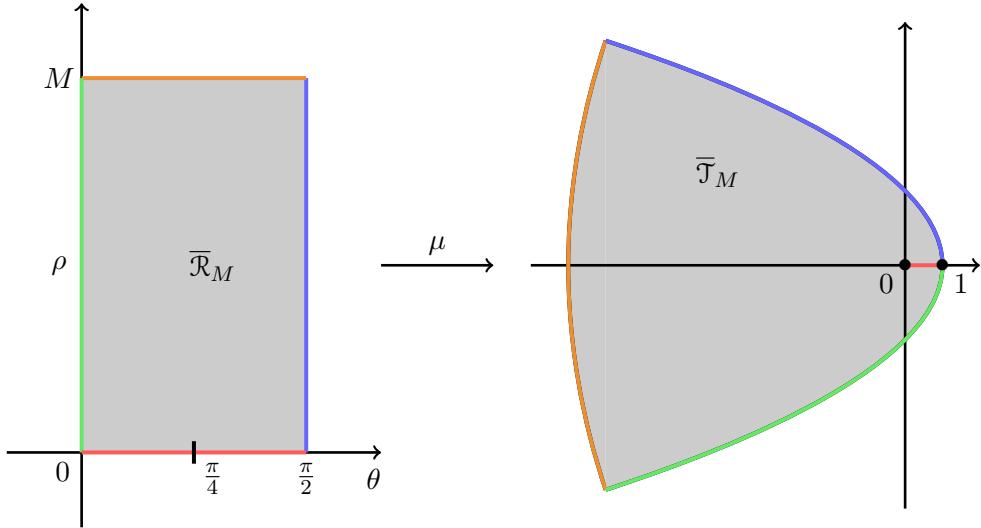
$$\mathcal{T}_M := \{(u, v) \in \mathbb{R}^2 : \left(\frac{\pi v}{8M}\right)^2 - \left(\frac{4M}{\pi}\right)^2 \leq u \leq 1 - \left(\frac{v}{2}\right)^2\} \setminus \{(u, 0) : u \in [0, 1]\}.$$

The closure  $\bar{\mathcal{T}}_M$  of  $\mathcal{T}_M$  is a compact convex set because it is a closed bounded intersection of two convex sets, see Figure 4.6. This implies, by [B, Corollary 11.3.4], that  $\bar{\mathcal{T}}_M$  is homeomorphic to a closed disc. In addition, the map

$$\mu|_{\{\rho=0\}} : \{\rho = 0\} \rightarrow \{(u, 0) : u \in [0, +\infty)\}, \theta \mapsto \left(\frac{4}{\pi}\theta - 1\right)^2$$

maps the segment  $[0, \frac{\pi}{2}] \times \{0\}$  onto the interval  $[0, 1] \times \{0\}$  and the fiber  $(\mu|_{\{\rho=0\}})^{-1}(t_0)$  for  $t_0 \in [0, 1]$  consists of two points, namely:

$$\{\theta_1 := \frac{\pi}{4}(1 + \sqrt{t_0}), \theta_2 := \frac{\pi}{4}(1 - \sqrt{t_0})\},$$


 Figure 4.6: Behavior of the map  $\mu : \overline{\mathcal{R}}_M \rightarrow \overline{\mathcal{T}}_M$ .

which makes sense considering the identification described in Construction 4.10.

Following Construction 4.10 and since  $\theta_1 = \frac{\pi}{2} - \theta_2$ , the map  $\lambda := \mu|_{\overline{\mathcal{R}}_M} : \overline{\mathcal{R}}_M \rightarrow \overline{\mathcal{T}}_M$  factors through  $\tilde{\mathcal{R}}_M$  and there exists a continuous map  $\tilde{\lambda} : \tilde{\mathcal{R}}_M \rightarrow \overline{\mathcal{T}}_M$  such that the following diagram is commutative:

$$\begin{array}{ccc} \mathcal{R}_M & \hookrightarrow & \overline{\mathcal{R}}_M \\ \pi_M|_{\mathcal{R}_M} \downarrow & & \pi_M \downarrow \\ \tilde{\mathcal{R}}_M & \hookrightarrow & \tilde{\mathcal{R}}_M \xrightarrow{\tilde{\lambda}} \mathcal{T}_M \end{array}$$

The map  $\tilde{\lambda}$  is continuous and bijective and in fact it is a homeomorphism because  $\tilde{\mathcal{R}}_M$  is compact and  $\overline{\mathcal{T}}_M$  is a Hausdorff space. Thus,  $\tilde{\mathcal{R}}_M$  is homeomorphic to a disc and its boundary is

$$\pi_M(\{\rho = M\} \cup \{\theta = 0\} \cup \{\theta = \frac{\pi}{2}\}),$$

which concludes the proof.  $\square$

We are ready to prove the main result of this section, which will be the key to prove Theorem 1.11.

**Lemma 4.12.** Fix  $B \geq A > 0$  and consider the warped discs  $\mathcal{D}_1$  and  $\mathcal{D}_2$  introduced in (4.3) and (4.4). Then, there exists  $M > 0$  such that the boundary map  $\partial\tilde{\phi}_M : \partial\tilde{\mathcal{R}}_M \rightarrow \mathbb{R}^3$  meets transversally once both discs  $\mathcal{D}_1$  and  $\mathcal{D}_2$ .

*Proof.* First, since  $\mathcal{D}_1$  and  $\mathcal{D}_2$  are bounded, there exists  $M_0 > 0$  such that

$$\mathcal{D}_1 \cup \mathcal{D}_2 \subset \{(x, y, z) : \|(x, y, z)\| < M_0\} \subset \mathbb{R}^3.$$

Take  $M := 4M_0$  and consider the set  $\tilde{\mathcal{R}}_M$  and the continuous map  $\tilde{\phi}_M$  introduced in Construction 4.10.

We are going to prove that the boundary map  $\partial\tilde{\phi}_M : \partial\tilde{\mathcal{R}}_M \rightarrow \mathbb{R}^3$  meets transversally once  $\mathcal{D}_1$ .

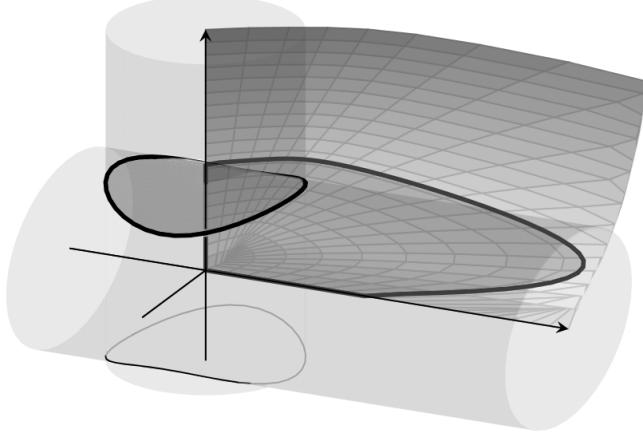


Figure 4.7: The boundary map  $\partial\tilde{\phi}_M : \partial\tilde{\mathcal{R}}_M \rightarrow \mathbb{R}^3$  meets transversally once  $\mathcal{D}_1$ .

Consider the parameterization of  $\partial\tilde{\mathcal{R}}_M$  given by

$$\beta_1(t) := \begin{cases} \pi_M(t, \frac{\pi}{2}) & \text{if } 0 \leq t \leq M, \\ \pi_M(M, M + \frac{\pi}{2} - t) & \text{if } M < t \leq M + \frac{\pi}{2}, \\ \pi_M(2M + \frac{\pi}{2} - t, 0) & \text{if } M + \frac{\pi}{2} < t \leq 2M + \frac{\pi}{2}. \end{cases}$$

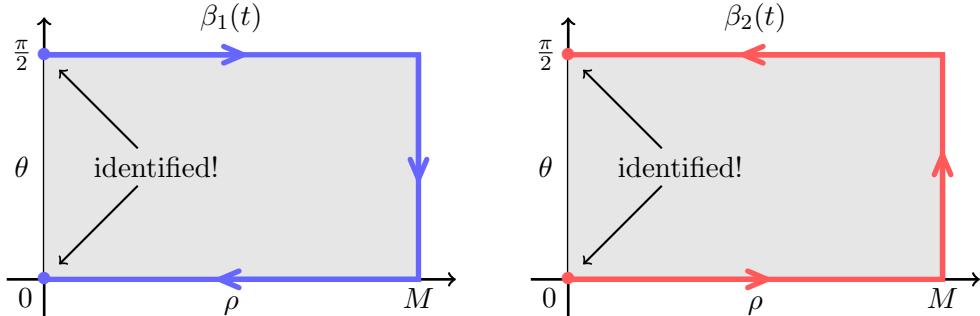


Figure 4.8: Behavior of the paths  $\beta_1$  and  $\beta_2$ , surrounding  $\tilde{\mathcal{R}}_M$ .

Let  $\alpha_1(t) := \tilde{\phi}_M \circ \beta_1(t)$  defined as:

$$\alpha_1(t) = \begin{cases} \phi(t, \frac{\pi}{2}) & \text{if } 0 \leq t \leq M, \\ \phi(M, M + \frac{\pi}{2} - t) & \text{if } M < t \leq M + \frac{\pi}{2}, \\ \phi(2M + \frac{\pi}{2} - t, 0) & \text{if } M + \frac{\pi}{2} < t \leq 2M + \frac{\pi}{2}. \end{cases}$$

Choose  $\varepsilon$  such that  $0 < \varepsilon < \min\{B, M_0 - B\}$  and consider the homeomorphism

$$\zeta_1 : \mathbb{R}^3 \rightarrow \mathbb{R}^3, (x, y, z) \mapsto (x, y, z - \xi_1(x, y)),$$

where  $\xi_1$  is the continuous semialgebraic function introduced in (4.5). Let us abbreviate  $\mathcal{D}_1(\varepsilon) := \xi_1^{-1}(\mathbb{D}_A(\varepsilon))$ . As we have done in other proofs we can rely on  $\mathbb{D}_A(\varepsilon)$  to conclude this part of the proof, so is enough to check the equality:

$$\alpha_1^{-1}(\mathcal{D}_1(\varepsilon)) = (B - \varepsilon, B + \varepsilon).$$

If  $t_0 \in [0, 2M + \frac{\pi}{2}]$ , pick  $p_0 := \alpha_1(t_0) \in \text{im}(\alpha_1)$ . The definition of  $\alpha_1$  leads us to distinguish three cases:

(i) If  $0 \leq t_0 \leq M$ , then

$$\zeta_1(p_0) = \zeta_1(\alpha(t_0)) = \zeta_1(\phi(t_0, 0)) = \zeta_1(\phi(t_0, \frac{\pi}{2})) = (0, 0, t_0 - B),$$

where  $\phi$  was introduced at the beginning of 4.3.2. Thus,  $\zeta_1(p_0) \in \mathbb{D}_A(\varepsilon)$  if and only if  $-B < -\varepsilon < t_0 - B < \varepsilon < M - B$ , that is,  $0 < B - \varepsilon < t_0 < B + \varepsilon < M$ .

(ii) If  $M < t_0 \leq M + \frac{\pi}{2}$  we have by (4.7), for every point  $q \in \mathcal{D}_1(\varepsilon)$ ,

$$\text{dist}(p_0, 0) = \text{dist}(\phi(M, M + \frac{\pi}{2} - t), 0) \geq \frac{M}{2} = 2M_0 > \sqrt{2}M_0 > \text{dist}(q, 0).$$

Therefore  $p_0 \notin \mathcal{D}_1(\varepsilon)$ .

(iii) If  $M + \frac{\pi}{2} < t_0 \leq 2M + \frac{\pi}{2}$ , then

$$p_0 = \alpha_1(t_0) = \phi(2M + \frac{\pi}{2} - t_0, 0) = (2M + \frac{\pi}{2} - t_0, 0, 0),$$

so  $\zeta_1(p_0) = (2M + \frac{\pi}{2} - t_0, 0, -B)$ . As  $\varepsilon < B$ , we have  $\zeta_1(p_0) \notin \mathbb{D}_A(\varepsilon)$  and  $p_0 \notin \mathcal{D}_1(\varepsilon)$ .

We conclude  $\alpha_1^{-1}(\mathcal{D}_1(\varepsilon)) = (B - \varepsilon, B + \varepsilon)$ , so  $\alpha_1$  meets transversally once  $\mathcal{D}_1$ .

Analogously one shows that the boundary map  $\partial\tilde{\phi}_M : \partial\tilde{\mathcal{R}}_M \rightarrow \mathbb{R}^3$  meets transversally once  $\mathcal{D}_2$ .

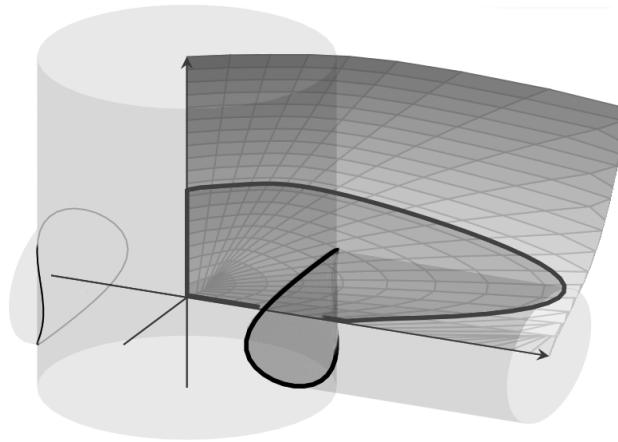


Figure 4.9: The boundary map  $\partial\tilde{\phi}_M : \partial\tilde{\mathcal{R}}_M \rightarrow \mathbb{R}^3$  meets transversally once  $\mathcal{D}_2$ .

Consider in this case the parameterization of  $\partial\tilde{\mathcal{R}}_M$  given by

$$\beta_2(t) := \begin{cases} \pi_M(t, 0) & \text{if } 0 \leq t \leq M, \\ \pi_M(M, t - M) & \text{if } M < t \leq M + \frac{\pi}{2}, \\ \pi_M(2M + \frac{\pi}{2} - t, \frac{\pi}{2}) & \text{if } M + \frac{\pi}{2} < t \leq 2M + \frac{\pi}{2}, \end{cases}$$

(see Figure 4.8). Then we define  $\alpha_2(t) := \tilde{\phi}_M \circ \beta_2(t)$  as:

$$\alpha_2(t) = \begin{cases} \phi(t, 0) & \text{if } 0 \leq t \leq M, \\ \phi(M, t - M) & \text{if } M < t \leq M + \frac{\pi}{2}, \\ \phi(2M + \frac{\pi}{2} - t, \frac{\pi}{2}) & \text{if } M + \frac{\pi}{2} < t \leq 2M + \frac{\pi}{2}. \end{cases}$$

Proceed as above keeping the same values for  $A$  and  $\varepsilon$  and using in this case the homeomorphism

$$\zeta_2 : \mathbb{R}^3 \rightarrow \mathbb{R}^3, (x, y, z) \mapsto (z, y, x - \xi_2(z, y)),$$

where  $\xi_2$  is the continuous semialgebraic function introduced in (4.6), to prove that  $\alpha_2$  meets transversally once the warped disk  $\mathcal{D}_2$ . This concludes the proof.  $\square$

Now we are ready to prove  $(\diamond)$  and this way conclude the topological proof.

[Proof of Theorem 1.11] Note first that, by Lemma 4.11,  $\bar{\mathcal{R}}_M$  is homeomorphic to a closed disc. By Proposition 4.7 applied to the continuous map  $\phi_M : \bar{\mathcal{R}}_M \rightarrow \mathbb{R}^3$  and Lemma 4.12, we deduce that the boundaries of both warped discs  $\mathcal{D}_1$  and  $\mathcal{D}_2$  meet  $\phi_M(\mathcal{R}_M) \subset \mathcal{S}$ . Thus,  $(\diamond)$  holds, and this concludes the proof.  $\square$



# 5

## Complexity analysis

### 5.1 Introduction to the computational comparison

In this chapter we set up the theoretical basis to discuss how “good” or how “bad” the three polynomial maps  $\mathbb{R}^2 \rightarrow \mathbb{R}^2$  having  $\mathcal{Q}$  as image are. Thus, we first need to clearly specify what “good” and “bad” means here. Afterwards, and using a computer, we will compare the images under the maps  $\mathcal{P}$  (Chapter 2),  $\mathcal{H} \circ \mathcal{G} \circ \mathcal{F}$  (Chapter 3) and  $\mathcal{F}$  (Chapter 4) of certain subsets of  $\mathbb{R}^2$ .

We will start with some naif grids of integers contained in the square  $[-10, 10] \times [-10, 10]$  and in  $[-100, 100] \times [-100, 100]$ , and some grids in  $[-1, 1] \times [-1, 1]$  and  $[0, 1] \times [0, 1]$ . Later, we will move into more specific sets provided by certain curves, following the ideas of Section 2.1. Lastly, we will use random sets of points contained, respectively, in the unit disc and in the disc of radius  $\rho = 100$ .

In the analysis, we will compare the execution times between the three maps, and also between computing the image as a composition of maps or as a fully expanded polynomial map. We will also take advantage of the proof in Chapter 4 for computing  $f_2(\overline{\mathcal{Q}})$  instead of computing  $\mathcal{F}(\mathbb{R}^2)$ .

One may wonder:

*Why is it interesting to represent the open quadrant in an easy way?*

Here the word *easy* means computationally efficient, so we are looking for the map that fills the open quadrant  $\mathcal{Q}$  quicker (in terms of the number of calculations) and the one  $f$  such that the image  $f(\mathcal{S})$  of suitable subsets  $\mathcal{S} \subset \mathbb{R}^2$  fill a bigger area.

The answer is that the representation of  $\mathcal{Q}$  as a polynomial image of  $\mathbb{R}^2$  is a key step to represent much more sophisticated subsets of  $\mathbb{R}^n$  as polynomial or regular images of some  $\mathbb{R}^m$ . Thus, if we are able to represent  $\mathcal{Q}$  easily we would be able to represent other semialgebraic sets easily too. This is the case, for instance, for the family of convex polyhedra, their interiors and their complementaries. The reason for this is that for constructing either polynomial or regular maps  $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$  producing polyhedral images, the first step relies on the construction of a polynomial map whose image is the open orthant  $\mathcal{O} := \{x_1 > 0, \dots, x_n > 0\} \subset \mathbb{R}^n$ . As we saw in Theorem 1.10 this heavily relies on using polynomial maps on the plane with the half-plane  $\mathcal{H}$  and the open quadrant  $\mathcal{Q}$  as

images. Afterwards, the strategy for obtaining polynomial maps with convex polyhedral images is usually to start from  $\mathcal{O}$ , which is the simplest convex polyhedron containing at least one vertex, and add new vertices inductively by composing with suitable either polynomial or regular maps.

That is why the better the constituents of the final map are, the better the final map will be. Thus, the representation of  $\mathcal{Q}$  as a polynomial image of  $\mathbb{R}^2$  is key in this process.

## 5.2 Complexity of the given polynomial maps

### 5.2.1 The map from the first proof

We need first to compute the coefficients and the shape of the polynomials we are going to use. Note for instance that the first polynomial map  $\mathcal{P}$  defined in 2.1 transforms  $\mathbb{R}^2$  into  $\mathcal{Q} \cup \{p_1 := (0, 1), p_2 = (1, 0)\}$  and then, relying on the proof of Theorem 1.9 we manage to get rid of  $p_1$  and  $p_2$  by means of a polynomial map (say  $H$ ) that is applied to  $\mathbb{R}^2$  before  $\mathcal{P}$  acts. Nevertheless the proof of Theorem 1.9 depends on the some values  $r, a_1, a_2, \dots$ ; so, how is  $H$  defined?

Fernando and Ueno answered this question in [FU1] where they proved the following Lemma:

**Lemma 5.1.** Let  $r \in \mathbb{R} \setminus \{0, 1\}$  and  $S := \{(0, 0), (-1, 0)\}$ . Then,  $\mathbb{R}^2 \setminus S$  is the image of the polynomial map

$$H = (H_1, H_2) : \mathbb{R}^2 \rightarrow \mathbb{R}^2, (x, y) \mapsto (xy - r, x^2(xy - r)(xy - r + 1) + y).$$

Here we take  $r := -1$  for simplicity. In fact the “controversial” set is the preimage through  $\mathcal{P}$  of the set  $\{(0, 1), (1, 0)\}$ , namely  $\{(0, -1), (-1, 0)\}$  and not  $\{(0, 0), (-1, 0)\}$ , but a change of coordinates

$$Q : \mathbb{R}^2 \rightarrow \mathbb{R}^2, (x, y) \mapsto (x, y - x - 1),$$

maps  $(-1, 0)$  to  $(-1, 0)$  and  $(0, 0)$  to  $(0, -1)$ . Now it is clear that the first map we will use for the computational comparison is

$$g = (g_1, g_2) := \mathcal{P} \circ Q \circ H : \mathbb{R}^2 \longrightarrow \mathbb{R}^2.$$

We wonder how the full expansion of  $g$  looks like, and the result can be observed in Table 5.1. In this chapter we will mostly be using the composition  $\mathcal{P} \circ Q \circ H$  to get the image of a given point, but we will also do a comparison between that composition and the full expansion as shown in Table 5.1.

### 5.2.2 The map from the short proof

As for the second map, we recall that it is defined as  $f := (f_1, f_2) := \mathcal{H} \circ \mathcal{G} \circ \mathcal{F}$ , see its definition in 3.1.1. The full expansion of  $f$  can be seen in Table 5.2. We will also be doing a comparison between the expanded map and the composition in the next section.

$$\begin{aligned}
 g_1(x, y) &= (x^{18} + 2x^{16} + x^{14})y^{10} + 2(7x^{17} + 11x^{15} + 2x^{14} + 3x^{13} + 3x^{12} - x^{11} + x^{10})y^9 + (87x^{16} + 98x^{14} + 44x^{13} - x^{12} + 48x^{11} - \\
 &\quad 10x^{10} + 6x^9 + 7x^8 - 2x^7 + x^6)y^8 + 2(158x^{15} + 104x^{13} + 104x^{12} - 49x^{11} + 69x^{10} + x^9 - 12x^8 + 19x^7 - 3x^6 + x^4)y^7 + (743x^{14} + 118x^{12} + \\
 &\quad 552x^{11} - 331x^{10} + 114x^9 + 116x^8 - 136x^7 + 61x^6 + 8x^5 - 14x^4 + 4x^3 + x^2)y^6 + 2(591x^{13} - 225x^{11} + 450x^{10} - 225x^9 - 132x^8 + 163x^7 - \\
 &\quad 103x^6 - 9x^5 + 26x^4 - 13x^3 - 2x^2 + x)y^5 + (1289x^{12} - 1250x^{10} + 924x^9 - 78x^8 - 786x^7 + 386x^6 - 10x^5 - 154x^4 + 66x^3 + 3x^2 - 14x + 1)y^4 + \\
 &\quad 2(476x^{11} - 770x^9 + 292x^8 + 282x^7 - 432x^6 + 63x^5 + 143x^4 - 76x^3 + x^2 + 19x - 4)y^3 + (456x^{10} - 1064x^8 + 208x^7 + 767x^6 - 456x^5 - \\
 &\quad 172x^4 + 290x^3 - 27x^2 - 52x + 23)y^2 + 4(32x^9 - 100x^7 + 8x^6 + 107x^5 - 24x^4 - 47x^3 + 23x^2 + 8x - 7)y + (16x^8 - 64x^6 + 92x^4 - 56x^2 + 13), \\
 g_2(x, y) &= x^{16}y^{12} + 2(7x^{15} - x^{13} + x^{12})y^{11} + (89x^{14} - 26x^{12} + 22x^{11} + x^{10} - 2x^9 + x^8)y^{10} + 2(171x^{13} - 76x^{11} + 54x^{10} + 6x^9 - \\
 &\quad 10x^8 + 4x^7)y^9 + (893x^{12} - 532x^{10} + 316x^9 + 64x^8 - 88x^7 + 28x^6)y^8 + 2(844x^{11} - 628x^9 + 311x^8 + 101x^7 - 114x^6 + 29x^5)y^7 + \\
 &\quad (2396x^{10} - 2140x^8 + 878x^7 + 427x^6 - 396x^5 + 81x^4)y^6 + 2(1299x^9 - 1369x^7 + 457x^6 + 326x^5 - 248x^4 + 40x^3)y^5 + (2148x^8 - \\
 &\quad 2668x^6 + 698x^5 + 751x^4 - 458x^3 + 55x^2)y^4 + 2(661x^7 - 976x^5 + 187x^4 + 327x^3 - 151x^2 + 12x)y^3 + (573x^6 - 1022x^4 + 126x^3 + \\
 &\quad 414x^2 - 130x + 5)y^2 + 4(39x^5 - 86x^3 + 5x^2 + 44x - 7)y + (20x^4 - 56x^2 + 40).
 \end{aligned}$$

Table 5.1: The first polynomial map expanded.

### 5.2.3 The map from the topological proof

Lastly and after such long polynomials, the map from the topological proof defined in 4.1.1 is showed in its expanded form in Table 5.3. Recall that  $\mathcal{F} = f_2 \circ f_1$  where  $f_1(\mathbb{R}^2) = \overline{\mathcal{Q}}$ , so we will also be testing how  $f_2(\overline{\mathcal{Q}})$  behaves.

### 5.2.4 Optimal algebraic structure of the given polynomial maps

In this section we calculate the total degree and the total number of (non-zero) monomials from the three polynomial maps, see Table 5.4. Recall that the *total degree* of a polynomial map is the sum of the degrees of its components. For instance  $\deg(\mathcal{F}) = \deg(\mathcal{F}_1) + \deg(\mathcal{F}_2) = 12 + 16 = 28$ .

### 5.2.5 Optimal multiplicative complexity of the given polynomial maps

When trying to evaluate a polynomial at a certain point in an efficient way, we can consider the number of non-scalar multiplications that the computer needs to perform to get its image. To do this, we assume that if the same product needs to be computed more than once in the same evaluation, the computer will only do it once. In any case we just compute an upper bound of the number of calculations, because depending on how the product is implemented on a certain programming language this value could be smaller.

For instance consider the first polynomial map  $g = \mathcal{P} \circ Q \circ H$ :

$$\begin{aligned}
 H : \mathbb{R}^2 &\rightarrow \mathbb{R}^2, (x, y) \mapsto (xy + 1, x^2(xy + 1)(xy + 2) + y), \\
 Q : \mathbb{R}^2 &\rightarrow \mathbb{R}^2, (x, y) \mapsto (x, y - x - 1), \\
 \mathcal{P} : \mathbb{R}^2 &\rightarrow \mathbb{R}^2, (x, y) \mapsto ((1 - x^3y + y - xy^2)^2 + x^4y^2, (1 - xy + x - x^4y)^2 + x^4y^2).
 \end{aligned}$$

Looking at  $H$  we see that the different non-scalar products that need to be computed are  $xy, x^2, (xy + 1)(xy + 2)$  and  $x^2(xy + 1)(xy + 2)$ , that makes 4 in total. In a similar way we get 0 for  $Q$  and 9 for  $\mathcal{P}$ . In total, an upper bound for  $g$  is  $4 + 0 + 9 = 13$ . Likewise, we

$$\begin{aligned}
 f_1(x, y) = & (4x^{26} + 20x^{24} + 41x^{22} + 44x^{20} + 26x^{18} + 8x^{16} + x^{14})y^{26} + (-104x^{25} - 480x^{23} - 902x^{21} - 880x^{19} - 468x^{17} - 128x^{15} \\
 & - 14x^{13})y^{25} + (32x^{26} + 1458x^{24} + 5839x^{22} + 9807x^{20} + 8554x^{18} + 4036x^{16} + 967x^{14} + 91x^{12})y^{24} + (-768x^{25} - 13876x^{23} - 46860x^{21} \\
 & - 69188x^{19} - 53264x^{17} - 22028x^{15} - 4564x^{13} - 364x^{11})y^{23} + (113x^{26} + 9382x^{24} + 97367x^{22} + 274164x^{20} + \frac{703621}{2}x^{18} + 236532x^{16} \\
 & + 84820x^{14} + 15018x^{12} + \frac{2003}{2}x^{10})y^{22} + (-2486x^{25} - 75768x^{23} - 525594x^{21} - 1229924x^{19} - 1360121x^{17} - 791240x^{15} - 243544x^{13} \\
 & - 36436x^{11} - 2007x^9)y^{21} + (231x^{26} + 27209x^{24} + 446529x^{22} + 2240074x^{20} + \frac{8710925}{2}x^{18} + \frac{8244713}{2}x^{16} + 2057114x^{14} + 538091x^{12} + \\
 & \frac{134535}{2}x^{10} + \frac{6051}{2}x^8)y^{20} + (-4620x^{25} - 193928x^{23} - 2018472x^{21} - 7667498x^{19} - 12393340x^{17} - 9976824x^{15} - 4233956x^{13} \\
 & - 931594x^{11} - 96204x^9 - 3492x^7)y^{19} + (301x^{26} + 45304x^{24} + \frac{1996081}{2}x^{22} + 7201629x^{20} + 21316526x^{18} + 28643722x^{16} + \frac{38977381}{2}x^{14} + \\
 & 6970385x^{12} + 1276041x^{10} + 107514x^8 + 3108x^6)y^{18} + (-5418x^{25} - 285964x^{23} - 3907609x^{21} - 20635184x^{19} - 48472730x^{17} \\
 & - 54092612x^{15} - 30891281x^{13} - 9219636x^{11} - 1387160x^9 - 94004x^7 - 2128x^5)y^{17} + (259x^{26} + 47243x^{24} + \frac{2580609}{2}x^{22} + \frac{23978613}{2}x^{20} + \\
 & 47980885x^{18} + 90479081x^{16} + \frac{167285973}{2}x^{14} + \frac{79543913}{2}x^{12} + 9792158x^{10} + 1193474x^8 + 63896x^6 + 1106x^4)y^{16} + (-4144x^{25} - \\
 & 262276x^{23} - 4387372x^{21} - 29332620x^{19} - 91032136x^{17} - 138703760x^{15} - 105761608x^{13} - 41463220x^{11} - 8305148x^9 - 805172x^7 \\
 & - 33232x^5 - 424x^3)y^{15} + (147x^{26} + 31738x^{24} + 1029734x^{22} + 11578342x^{20} + \frac{115563913}{2}x^{18} + 141146980x^{16} + 174192331x^{14} + \\
 & 108825883x^{12} + \frac{69479137}{2}x^{10} + 5560611x^8 + 418307x^6 + 12832x^4 + \frac{227}{2}x^2)y^{14} + (-2058x^{25} - 152936x^{23} - 3012828x^{21} - \\
 & 24111820x^{19} - 92027457x^{17} - 178500672x^{15} - 178178038x^{13} - 90313924x^{11} - 23084717x^9 - 2880652x^7 - 162050x^5 - 3480x^3 - \\
 & 19x)y^{13} + (53x^{26} + 13607x^{24} + 514692x^{22} + 6758603x^{20} + \frac{79906193}{2}x^{18} + \frac{236896489}{2}x^{16} + 183083134x^{14} + 146995602x^{12} + \\
 & \frac{119160879}{2}x^{10} + \frac{23809093}{2}x^8 + 1115393x^6 + 44130x^4 + \frac{1187}{2}x^2 + \frac{3}{2})y^{12} + (-636x^{25} - 55808x^{23} - 1273116x^{21} - 11799438x^{19} - \\
 & 52781280x^{17} - 122608140x^{15} - 150791284x^{13} - 96320302x^{11} - 30530204x^9 - 4590824x^7 - 302620x^5 - 7466x^3 - 48x)y^{11} + \\
 & (11x^{26} + 3544x^{24} + \frac{314743}{2}x^{22} + 2376631x^{20} + 16126054x^{18} + 55398078x^{16} + \frac{202163613}{2}x^{14} + 98174729x^{12} + 48935769x^{10} + \\
 & 11683851x^8 + \frac{2465761}{2}x^6 + 49595x^4 + 534x^2)y^{10} + (-110x^{25} - 12028x^{23} - 320343x^{21} - 3389132x^{19} - 17228062x^{17} - 45771560x^{15} - \\
 & 65325089x^{13} - 49297772x^{11} - 18508228x^9 - 3090272x^7 - 192077x^5 - 2508x^3 + 26x)y^9 + (x^{26} + 499x^{24} + \frac{54971}{2}x^{22} + \frac{963471}{2}x^{20} + \\
 & 3698783x^{18} + 14267155x^{16} + \frac{58589293}{2}x^{14} + \frac{64546337}{2}x^{12} + 18324283x^{10} + 4804470x^8 + \frac{907373}{2}x^6 + \frac{7397}{2}x^4 - 553x^2 - 3)y^8 + (- \\
 & 8x^{25} - 1344x^{23} - 44228x^{21} - 539160x^{19} - 3067508x^{17} - 9009888x^{15} - 14152800x^{13} - 11695504x^{11} - 4652360x^9 - 672008x^7 + \\
 & 7308x^5 + 4412x^3 + 72x)y^7 + (28x^{24} + 2366x^{22} + 50998x^{20} + 446884x^{18} + 1901034x^{16} + 4222671x^{14} + 4948005x^{12} + \frac{5751153}{2}x^{10} + \\
 & 639605x^8 - \frac{52037}{2}x^6 - 16587x^4 - \frac{1249}{2}x^2 - 1)y^6 + (-56x^{23} - 2828x^{21} - 42092x^{19} - 269596x^{17} - 854940x^{15} - 1404926x^{13} - \\
 & 1159452x^{11} - 389461x^9 + 18128x^7 + 30043x^5 + 2488x^3 + 9x)y^5 + (70x^{22} + 2310x^{20} + 24418x^{18} + 114536x^{16} + 265718x^{14} + \\
 & 306840x^{12} + \frac{302491}{2}x^{10} - \frac{3703}{2}x^8 - \frac{49077}{2}x^6 - \frac{9289}{2}x^4 - \frac{39}{2}x^2 + \frac{3}{2})y^4 + (-56x^{21} - 1264x^{19} - 9568x^{17} - 32360x^{15} - 52408x^{13} - \\
 & 37034x^{11} - 2472x^9 + 10104x^7 + 3718x^5 + 16x^3 - 20x)y^3 + (28x^{20} + 439x^{18} + 2350x^{16} + \frac{10971}{2}x^{14} + 5505x^{12} + 1064x^{10} - 2161x^8 - \\
 & 1411x^6 - 46x^4 + \frac{127}{2}x^2 + 1)y^2 + (-8x^{19} - 86x^{17} - 312x^{15} - 451x^{13} - 168x^{11} + 222x^9 + 252x^7 + 26x^5 - 34x^3 - 5x)y + (x^{18} + \\
 & 7x^{16} + \frac{31}{2}x^{14} + \frac{19}{2}x^{12} - 8x^{10} - 17x^8 - 4x^6 + 5x^4 + \frac{3}{2}x^2 + \frac{3}{2}),
 \end{aligned}$$

$$\begin{aligned}
 f_2(x, y) = & (2x^{10} + 5x^8 + 4x^6 + x^4)y^{10} + (-20x^9 - 40x^7 - 24x^5 - 4x^3)y^9 + (5x^{10} + 102x^8 + 149x^6 + 62x^4 + 6x^2)y^8 + (-40x^9 - 312x^7 - \\
 & 316x^5 - 84x^3 - 4x)y^7 + (4x^{10} + 149x^8 + 600x^6 + 395x^4 + 58x^2 + 1)y^6 + (-24x^9 - 316x^7 - 720x^5 - 276x^3 - 16x)y^5 + (x^{10} + 62x^8 + 397x^6 + \\
 & 504x^4 + 85x^2)y^4 + (-4x^9 - 84x^7 - 284x^5 - 168x^3)y^3 + (6x^8 + 60x^6 + 99x^4 + 5x^2 - 1)y^2 + (-4x^7 - 20x^5 - 8x^3 + 6x)y + (x^6 + 2x^4 - 2x^2 + 1).
 \end{aligned}$$

Table 5.2: The second polynomial map expanded.

$$\mathcal{F}_1(x, y) = x^4y^8 + 2(x^6 - x^2)y^6 + (x^8 + x^6 - 2x^4 - 2x^2 + 1)y^4 - 2(x^4 - 1)y^2 + 1,$$

$$\mathcal{F}_2(x, y) = (x^{12} + 2x^8 + x^6 + x^4)y^4 - 2(x^8 + x^6 + x^4 + x^2)y^2 + (x^4 + 2x^2 + 1).$$

Table 5.3: The third polynomial map expanded.

	Total degree	Total number of monomials	Non-escalar complexity
$g = \mathcal{P} \circ Q \circ H$	56	167	13
$f = \mathcal{H} \circ \mathcal{G} \circ \mathcal{F}$	72	350	11
$\mathcal{F} = f_2 \circ f_1$	28	22	11

Table 5.4: Total degree, total number of monomials and non-escalar complexity.

obtain  $4 + 4 + 3 = 11$  for  $f = \mathcal{H} \circ \mathcal{G} \circ \mathcal{F}$  rewritten as

$$\begin{aligned}\mathcal{F} : \mathbb{R}^2 &\rightarrow \mathbb{R}^2, (x, y) \mapsto ((xy - 1)^2 + x^2, (xy - 1)^2 + y^2), \\ \mathcal{G} : \mathbb{R}^2 &\rightarrow \mathbb{R}^2, (x, y) \mapsto (x, y((xy)^2 - 4xy + 4) + x((xy)^2 - 2xy + 1)), \\ \mathcal{H} : \mathbb{R}^2 &\rightarrow \mathbb{R}^2, (x, y) \mapsto (xy(x \cdot (xy) - 4x + 4 + \frac{1}{2}y), y),\end{aligned}$$

Recall that we have factorized  $\mathcal{F}$  as  $\mathcal{F} = f_2 \circ f_1$ , where the polynomial maps  $f_1, f_2$  are defined as follows:

$$\begin{aligned}f_1 : \mathbb{R}^2 &\rightarrow \mathbb{R}^2, (x, y) \mapsto (x^2, y^2), \\ f_2 : \mathbb{R}^2 &\rightarrow \mathbb{R}^2, (x, y) \mapsto ((xy^2 + x^2y - y - 1)^2 + x^3y^2, (x^3y + xy - x - 1)^2 + x^3y^2).\end{aligned}$$

Looking at  $f_1$  we have two different non-scalar products  $x^2, y^2$ . On the other hand, for  $f_2$ , we have  $y^2, x^2, xy^2, x^2y, x^2xy^2, xx^2y, xy, (xy^2 + x^2y - y - 1)^2, (x^3y + xy - x - 1)^2$ . Thus, we obtain  $2 + 9 = 11$ . See Table 5.4.

## 5.3 Computational analysis

After all these arguments and explanations, we did a few tests on Sage, in terms of timing and obtaining the resulting regions of  $\mathbb{R}^2$ . We split this analysis into three cases depending on the chosen domain.

### 5.3.1 Uniformly distributed points contained in a square

The tests we performed related to this were:

1. **The set of integers contained in  $[-10, 10] \times [-10, 10]$** , that is:

$$\mathcal{P}_1 := \{(x, y) : -10 \leq x, y \leq 10, x, y \in \mathbb{Z}\} \subset \mathbb{R}^2.$$

2. **The set of integers contained in  $[-100, 100] \times [-100, 100]$** , that is:

$$\mathcal{P}_2 := \{(x, y) : -100 \leq x, y \leq 100, x, y \in \mathbb{Z}\} \subset \mathbb{R}^2.$$

3. **Grid of  $10^6$  points uniformly distributed in  $[0, 1] \times [0, 1]$  with step  $h = \frac{1}{999}$** :

$$\mathcal{P}_3 := \{x : x = nh, 0 \leq n \leq 999\} \times \{y : y = nh, 0 \leq n \leq 999\} \subset \mathbb{R}^2.$$

4. **Grid of  $10^6$  points uniformly distributed in  $[-1, 1] \times [-1, 1]$  with step  $h = \frac{2}{999}$ :**

$$\mathcal{P}_4 := \{x : x = nh - 1, 0 \leq n \leq 999\} \times \{y : y = nh - 1, 0 \leq n \leq 999\} \subset \mathbb{R}^2.$$

5. **Grid of  $4 \cdot 10^4$  points uniformly distributed in  $[-10, 10] \times [-10, 10]$  with step  $h = \frac{20}{199}$ :**

$$\mathcal{P}_5 := \{x : x = nh - 10, 0 \leq n \leq 199\} \times \{y : y = nh - 10, 0 \leq n \leq 199\} \subset \mathbb{R}^2.$$

6. **Grid of  $4 \cdot 10^6$  points uniformly distributed in  $[-10, 10] \times [-10, 10]$  with step  $h = \frac{20}{1999}$ :**

$$\mathcal{P}_6 := \{x : x = nh - 10, 0 \leq n \leq 1999\} \times \{y : y = nh - 10, 0 \leq n \leq 1999\} \subset \mathbb{R}^2.$$

We considered to continue making the set of points bigger, namely a grid of  $4 \cdot 10^8$  points, but a quick estimation took this idea away from us, because it could take a few hours just for the first polynomial. The execution time results are displayed in Table 5.5.

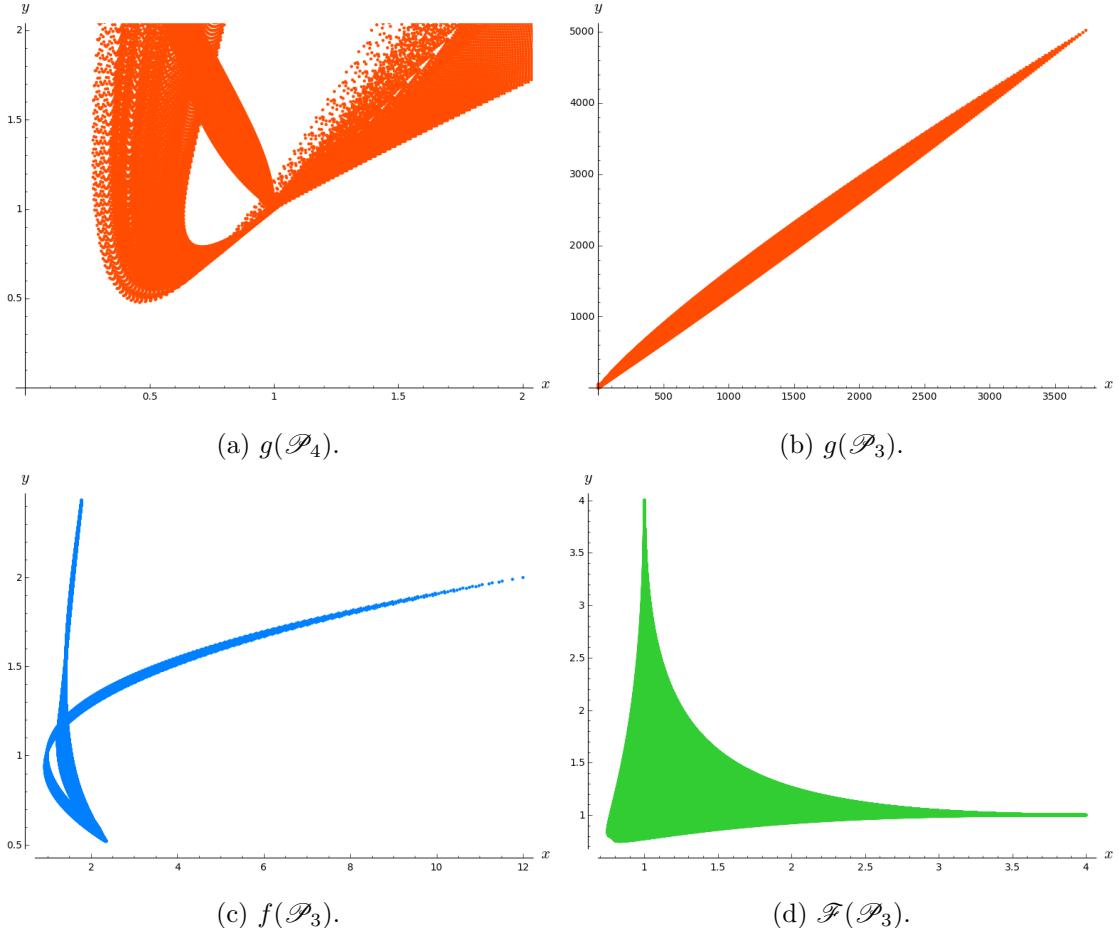


Figure 5.1: The sets  $g(\mathcal{P}_4)$ ,  $g(\mathcal{P}_3)$ ,  $f(\mathcal{P}_3)$  and  $\mathcal{F}(\mathcal{P}_3)$ .

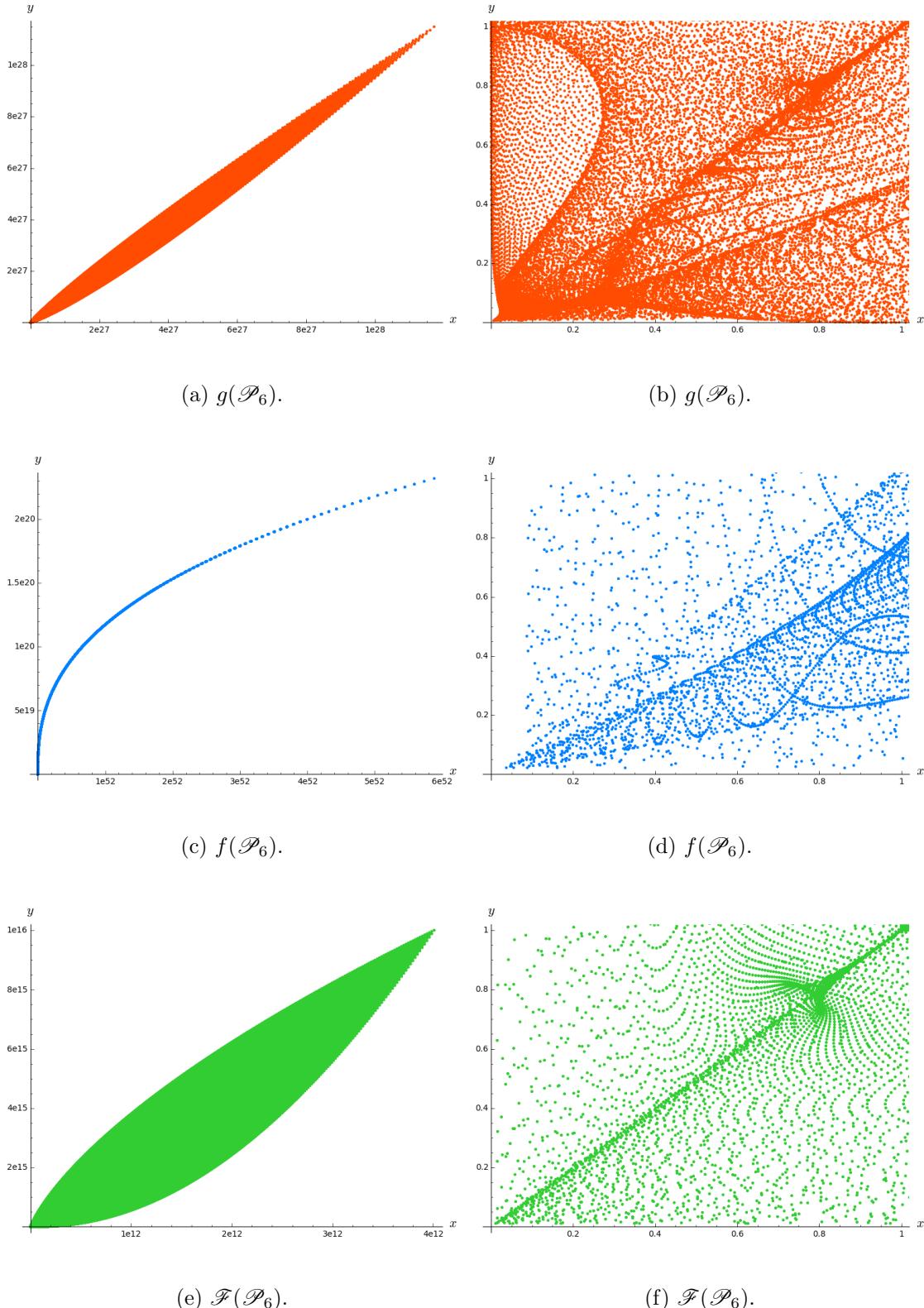


Figure 5.2: Images of the  $4 \cdot 10^6$  points from  $\mathcal{P}_6$  through  $g$ ,  $f$  and  $\mathcal{F}$ .

	$g = \mathcal{P} \circ Q \circ H$	$f = \mathcal{H} \circ \mathcal{G} \circ \mathcal{F}$	$\mathcal{F}$
$\mathcal{P}_1$	0.19 s	0.17 s	0.07 s
$\mathcal{P}_2$	17.38 s	16.23 s	6.88 s
$\mathcal{P}_3$	410.73 s (5.1b)	389.74 s (5.1c)	155.09 s (5.1d)
$\mathcal{P}_4$	408.88 s (5.1a)	392.34 s	155.29 s
$\mathcal{P}_5$	17.10 s	15.45 s	7.00 s
$\mathcal{P}_6$	1633.50 s (5.2a-5.2b)	1640.01 s (5.2c-5.2d)	625.11 s (5.2e-5.2f)

Table 5.5: Execution times of the computation of uniformly distributed points contained in a square.

$g$	Composition	Expanded
$\mathcal{P}_1$	0.19 s	0.23 s
$\mathcal{P}_5$	17.10 s	16.15 s
$\mathcal{P}_6$	1633.50 s	1582.08 s

Table 5.6: Comparison between  $g = \mathcal{P} \circ Q \circ H$  on its expanded form (Table 5.1) or as a composition.

$f$	Composition	Expanded
$\mathcal{P}_1$	0.17 s	0.40 s
$\mathcal{P}_5$	15.45 s	26.50 s
$\mathcal{P}_6$	1640.01 s	2650.13 s

Table 5.7: Comparison between  $f = \mathcal{H} \circ \mathcal{G} \circ \mathcal{F}$  on its expanded form (Table 5.2) or as a composition.

In this section we also compare the first and third maps as either a composition or a fully expanded expression, as shown in Table 5.1 and Table 5.2. The results are shown in Tables 5.6 and 5.7.

Lastly, we recall that  $\mathcal{F} = f_2 \circ f_1$ , where  $f_1(x, y) = (x^2, y^2)$ , so  $f_1(\mathbb{R}^2) = \overline{\mathcal{Q}}$ . Thus, we study  $f_2(\overline{\mathcal{Q}})$  instead of computing  $\mathcal{F}(\mathbb{R}^2)$ . We used a grid of  $10^6$  points uniformly distributed in  $[0, 10] \times [0, 10]$  with step  $h = \frac{10}{999}$ :

$$\mathcal{P}_7 := \{x : x = nh, 0 \leq n \leq 999\} \times \{y : y = nh, 0 \leq n \leq 999\} \subset \mathbb{R}^2.$$

It took 149.99 s for  $f_2$  to compute the image of the  $10^6$  points from  $\mathcal{P}_7$ , which is very close to the 155.09 s and 155.29 s that it took  $\mathcal{F}$  to compute the image of the  $10^6$  points from  $\mathcal{P}_3$  and  $\mathcal{P}_4$  respectively. Note that this makes sense since the only difference between computing the image of a set under  $\mathcal{F} = f_2 \circ f_1$  or  $f_2$  is that with  $\mathcal{F}$  you first compute the image under  $f_1$ , which should not take a lot of time because of its simplicity, and then you compute the image under  $f_2$  of the resulting set.

### 5.3.2 Using families of curves

In this section we recall the condition that any polynomial map that has  $\mathcal{Q}$  as an image must verify:

*The closure of its image must contain the positive half-axes.*



We used in Chapter 2 certain families of curves approaching the coordinate axes, namely:

$$\alpha_\lambda(s, a_\lambda) := \left( s, \frac{1 + a_\lambda s}{s} \right) \quad \text{and} \quad \beta_\mu(s, b_\mu) := \left( \frac{1 + b_\mu s}{s}, s^3 \right).$$

In this section we use as the domain of the curves the grid of  $10^6$  points uniformly distributed in  $[-10, 10] \times [-10, 10]$  with step  $h = \frac{20}{999}$ :

$$\mathcal{P}_8 := \{x : x = nh - 10, 0 \leq n \leq 999\} \times \{y : y = nh - 10, 0 \leq n \leq 999\} \subset \mathbb{R}^2.$$

First, for each point  $(s, a_\lambda) \in \mathcal{P}_8$  we computed  $\alpha_\lambda((s, a_\lambda))$ , and then we applied  $g$  to that set of points. We did the same for  $(s, b_\mu) \in \mathcal{P}_8$ , obtaining  $g(\beta_\mu((s, b_\mu)))$ . The timing tests are displayed in Table 5.8.

	Time
$\alpha_\lambda((s, a_\lambda))$	129.27 s
$\beta_\mu((s, b_\mu))$	133.76 s
$g(\alpha_\lambda((s, a_\lambda)))$	398.04 s
$g(\beta_\mu((s, b_\mu)))$	399.02 s

Table 5.8: Execution times when computing the image of curves through  $g$ .

	Time
$\gamma_\lambda((s, a_\lambda))$	133.64 s
$\delta_\mu((s, b_\mu))$	127.90 s
$f_2(\gamma_\lambda((s, a_\lambda)))$	141.92 s
$f_2(\delta_\mu((s, b_\mu)))$	143.21 s

Table 5.9: Execution times when computing the image of curves through  $f_2$ .

We can proceed in exactly the same way with a new family of curves for  $f_2$ :

$$\gamma_\lambda(s, a_\lambda) = \left( \frac{1 + a_\lambda s}{s}, s^2 \right) \quad \text{and} \quad \delta_\mu(s, b_\mu) = \left( s, \frac{1 + b_\mu s}{s} \right).$$

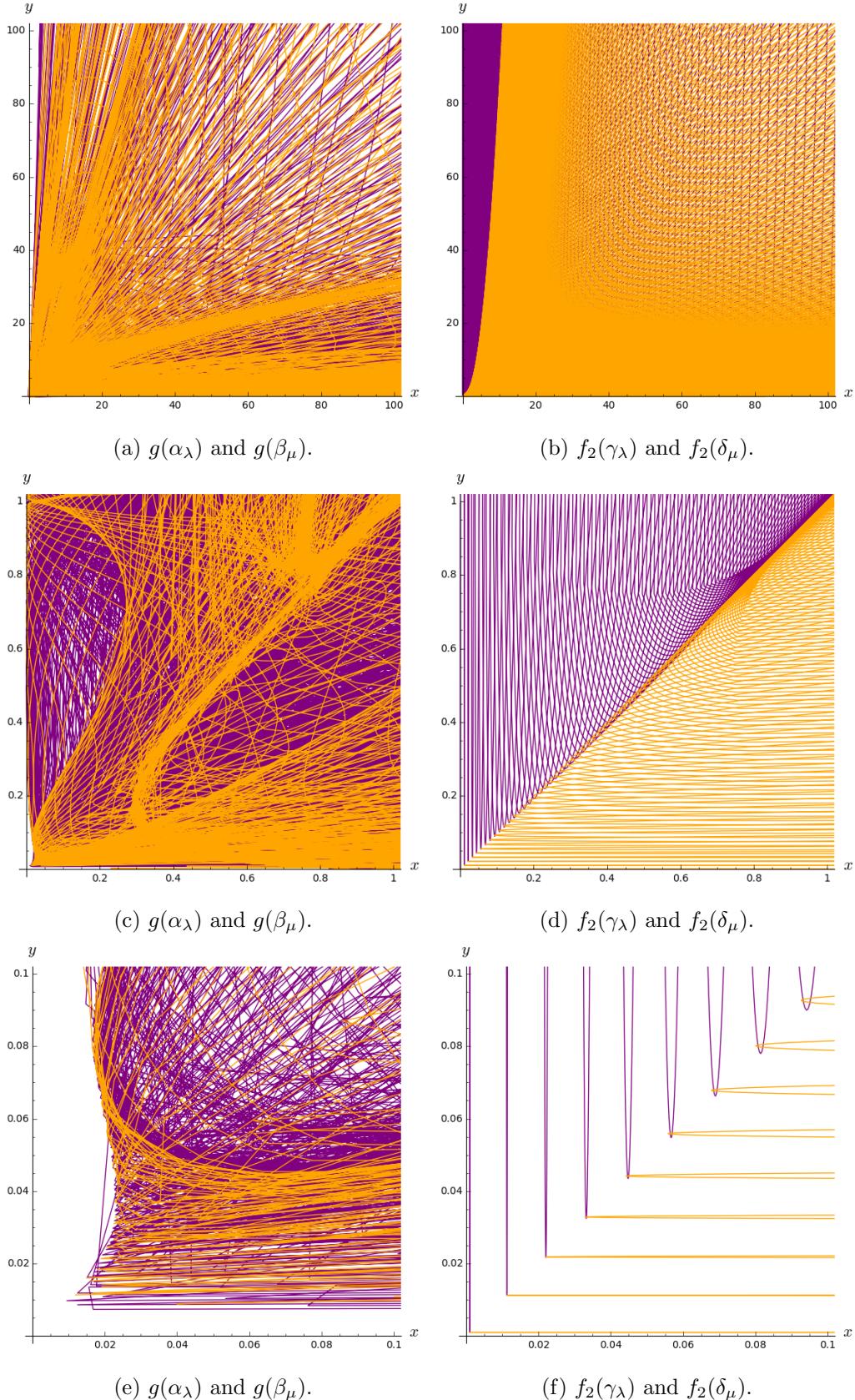
The computation times can be seen in Table 5.9, we used  $\mathcal{P}_7$  as the domain of these curves. The images of the curves through  $g$  and  $f_2$  are shown in Figure 5.3.

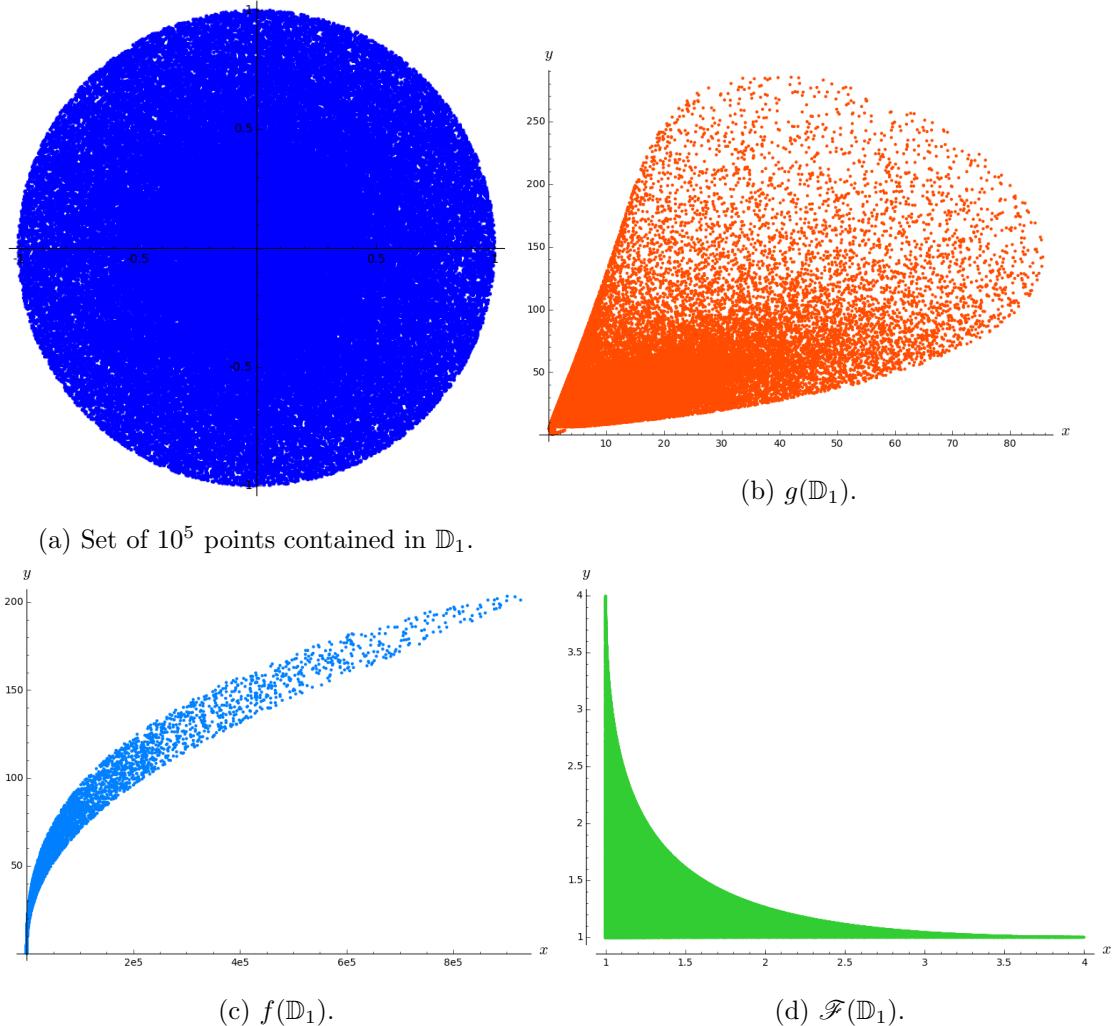
### 5.3.3 Randomly distributed points contained in a disc

When thinking of the possible domain that the polynomial maps could have, we thought of points randomly distributed on a square. However we opted for doing it on the unit disc. The points are generated by means of a random argument  $\theta \in [0, 2\pi]$  and a random radius; first with  $\rho \in [0, 1]$ , and  $\rho \in [0, 100]$  afterwards. Then, each point is generated as  $(\rho \cos \theta, \rho \sin \theta)$ . The result is a subset of  $10^5$  points of the disc that presents a higher concentration of points when we get closer to the origin. Of course this is not a problem for our study case. See Figure 5.4a. The timing tests are shown in Table 5.10.

	$g = \mathcal{P} \circ Q \circ H$	$f = \mathcal{H} \circ \mathcal{G} \circ \mathcal{F}$	$\mathcal{F}$
$\mathbb{D}_1$	131.97 s (5.4b)	99.56 s (5.4c)	32.02 s (5.4d)
$\mathbb{D}_{100}$	131.62 s	100.84 s	32.36 s

Table 5.10: Execution times of the computation of randomly distributed points in  $\mathbb{D}_1$  and  $\mathbb{D}_{100}$ .


 Figure 5.3: Images of families of curves:  $\alpha_\lambda$  and  $\gamma_\lambda$  in purple and  $\beta_\mu$  and  $\delta_\mu$  in orange.

Figure 5.4: Images of points in  $\mathbb{D}_1$  through  $g$ ,  $f$  and  $\mathcal{F}$ .

## 5.4 Conclusions and future research

As we have seen in the previous section, the third map  $\mathcal{F}$  outstrips the other two maps in the timing tests. This is to be expected from the ideas developed in Section 5.2. Also, even though the second map  $f$  was by far the one that provided the easiest proof for Theorem 1.11, the Figures from the previous section show that it is not very good at filling a big area of the open quadrant  $\mathcal{Q}$ . Lastly, the first map  $g$  was reasonably good in terms of covering  $\mathcal{Q}$ , but not so much in terms of execution time.

All of this could just be considered as an appetizer for finding a polynomial map with the least possible total degree and the least possible number of monomials:

**Problem.** (i) Which is the minimum total degree for a polynomial map  $\mathbb{R}^2 \rightarrow \mathbb{R}^2$  having  $\mathcal{Q}$  as image?

(ii) Which is the sparsest polynomial map  $\mathbb{R}^2 \rightarrow \mathbb{R}^2$  whose image is  $\mathcal{Q}$ ?

Fernando, Gamboa and Ueno have not stopped working on the problem, and we present

right now some new candidates of polynomial maps  $\mathbb{R}^2 \rightarrow \mathbb{R}^2$  whose images “should be”  $\mathcal{Q}$ :

- $\mathcal{N}_1 : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ ,  
 $(x, y) \mapsto (x^4y^4 + (x^2y + xy^2 - 1)^2(y^2 + 1), x^4y^4 + (x^2y + xy^2 - 1)^2(x^2 + 1)),$
- $\mathcal{N}_2 : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ ,  
 $(x, y) \mapsto (x^2y^2 + (x^2y + xy^2 - 1)^2(y^2 + 1), x^2y^2 + (x^2y + xy^2 - 1)^2(x^2 + 1)),$
- $\mathcal{N}_3 : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ ,  
 $(x, y) \mapsto (x^6y^4 + (x^2y + xy^2 - 1)^2(y^2 + 1), x^4y^6 + (x^2y + xy^2 - 1)^2(x^2 + 1)).$

It is remarkable how good their properties are in terms of total degree and number of monomials, see Table 5.11.

	Total degree	Total number of monomials	Non-escalar complexity
$\mathcal{N}_1$	16	24	10
$\mathcal{N}_2$	16	26	8
$\mathcal{N}_3$	20	26	13

Table 5.11: Properties of the new maps.

Note that the symmetry of the maps above converts them into good candidates to minimize the upper-bound of the non-escalar complexity. For instance, for  $\mathcal{N}_1$  we have  $x^2, x^4, y^2, y^4, x^4y^4, x^2y, xy^2, (x^2y+xy^2-1)^2, (x^2y+xy^2-1)^2(y^2+1), (x^2y+xy^2-1)^2(x^2+1)$ , which makes 10 in total. Also, note that for  $\mathcal{N}_3$  we don’t need to compute  $x^3$  and  $y^3$ , which reduces the non-scalar complexity. We have used the following pair of families of curves to approach the axes:

$$\sigma_\lambda(s, a_\lambda) = \left( s^2 + a_\lambda s^3, \frac{1}{s} \right) \quad \text{and} \quad \tau_\mu(s, b_\mu) = \left( \frac{1}{s}, s^2 + b_\mu s^3 \right).$$

The execution times for  $\mathcal{N}_1$  and  $\mathcal{N}_2$  are displayed respectively in Tables 5.12 and 5.13.

	Time
$\sigma_\lambda((s, a_\lambda))$	133.15 s
$\tau_\mu((s, b_\mu))$	129.80 s,
$\mathcal{N}_1(\sigma_\lambda((s, a_\lambda)))$	148.88 s
$\mathcal{N}_2(\tau_\mu((s, b_\mu)))$	153.14 s

Table 5.12: Execution times when computing the image of curves through  $\mathcal{N}_1$ .

	Time
$\sigma_\lambda((s, a_\lambda))$	133.15 s
$\tau_\mu((s, b_\mu))$	129.80 s,
$\mathcal{N}_1(\sigma_\lambda((s, a_\lambda)))$	148.23 s
$\mathcal{N}_2(\tau_\mu((s, b_\mu)))$	148.84 s

Table 5.13: Execution times when computing the image of curves through  $\mathcal{N}_2$ .

Note how good  $\mathcal{N}_1$  seems to be on approaching the positive half-axes for this family of curves. Even when checking on  $[0, 0.1] \times [0, 0.1]$  it fills a very dense region towards the axes. See Figures 5.5a and 5.5b.

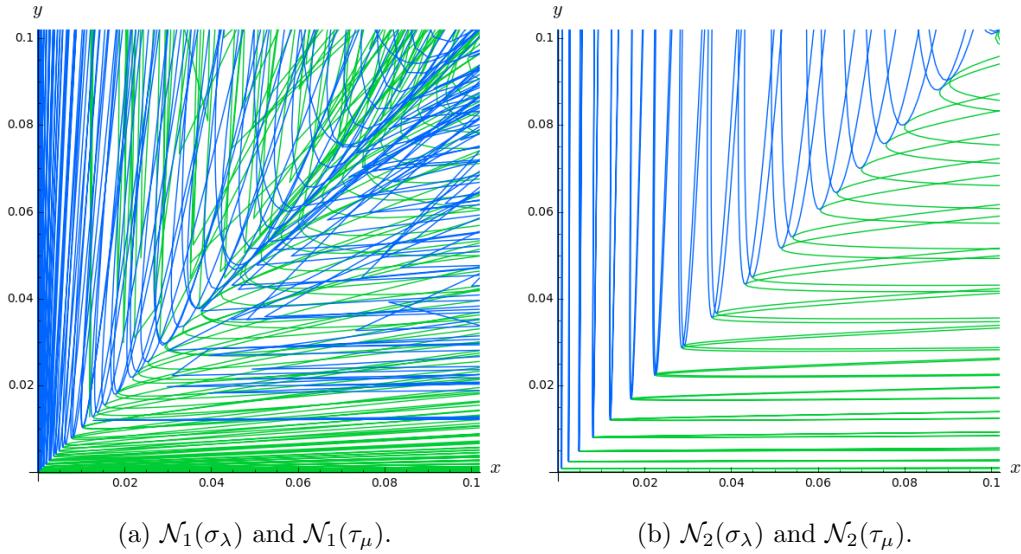


Figure 5.5: Images of families of curves for  $\mathcal{N}_1$  and  $\mathcal{N}_2$ :  $\sigma_\lambda$  in green and  $\tau_\mu$  in blue.

#### 5.4.1 Comments on further reductions

##### Reduction of multiplications using powers of 2

Aside from the optimal algebraic structure and the optimal multiplicative complexity, there is another measurement that could be considered when talking about the complexity of the given polynomials.

It is well known that for a computer it is far way easier to compute multiplications and divisions by powers of 2 than any other number, because that translates into a simple shift in the array of bits. For instance consider the number  $20 = 2^4 + 2^2$  expressed in base 2 with 8 bits:  $(20)_{10} = (0001\ 0100)_2$ . Then, if we want to multiply  $20 \cdot 4$  the computer can just shift the array two positions (because  $4 = 2^2$ ) to the left, namely:

$$(20 \cdot 4)_{10} = (0101\ 0000)_2 = (80)_{10}.$$

It is clear that the other way around also works for division, shifting to the right:

$$(80/16)_{10} = (0000\ 1010)_2 = (5)_{10}.$$

This is exactly what scholars do to multiply or divide an integer by a power of 10; they either add zeros or move the comma. Computers have had for years logic units that compute this shiftings flawlessly. Thus, we could focus on obtaining a polynomial map that only has powers of 2 as multiplicative factors on its components. Note that the three new maps and the one from the topological proof satisfy this condition. Indeed, this factor equals 2 for all of them.

##### Reduction of multiplications using hash tables and concurrency

There is an idea that could be implemented when computing the image of a given set of points through a map: relying on parallelism.

Let  $f(\mathbf{x}, \mathbf{y}) := (f_1(\mathbf{x}, \mathbf{y}), f_2(\mathbf{x}, \mathbf{y}))$  be any of the maps we were working with in this work. Taking advantage of the number of kernels computers have nowadays, we could use the readers-writers problem from concurrent computing and a data structure like a hash table. There would be one thread for  $f_1$  and one for  $f_2$ , and both threads would be both readers and writers on the hash table. The key of this hash table would be a monomial (or a part of it), for instance  $x^2$  or  $x^4y^2$ . The value associated to that key would be the computation of that monomial given the current point.

Let  $p := (x, y)$  be a point of the set of points we are using as a domain. When the programme starts, we would launch both threads. At each step the computation of  $f_1(p)$  involves to calculate a non-scalar multiplication (like the examples we just stated) it would look for it on the hash table. If its there, then there is no need to compute the multiplication, just get the value. If the key is not in the hash table, compute it and add a new entry to the hash table with this computation. The thread for  $f_2(p)$  would work in the same way. This way each non-scalar multiplication would be computed only once.

The remarkable fact is that this implementation could easily be generalized for computing  $f$  when it would be a composition of other maps, say  $f = g \circ h$ . The programme would act in the same way, starting computing  $h(p) := (h_1(p), h_2(p))$  for every point  $p$  by updating the hash table with each new point  $p$ . Then, when this phase would be over, we would have to delete the content in the hash table and proceed in the same way for  $g(h(p)) = g((h_1(p), h_2(p)))$ .

The only inconvenience would be working with symbolic expressions in Java or C++, which is far way easier with Sage (as far as I know). In any case, this implementation would probably represent an improvement in the execution time, because searching for a key in a (relatively small) hash table would be quicker than re-computing the multiplication. Indeed, searching a value on a hash table requires on average constant time  $O(1)$ , whereas computing the multiplication of two  $n$ -digits numbers is for example  $O(n^{1.585})$  for Karatsuba's algorithm. Thus if we are able to avoid computing two out of ten monomials for each point, the new total execution time would be roughly 20% less than the original time.

## Bibliography

- [A] F. S. Acton: Numerical Methods That Work, 2<sup>nd</sup> printing. Washington, DC: Math. Assoc. Amer., (1990).
- [ABR] C. Andradas, L. Bröcker, J.M. Ruiz: Constructible sets in real geometry. *Ergeb. Math.* **33**. Berlin, Heidelberg, New York: Springer Verlag, (1996).
- [BCR] J. Bochnak, M. Coste, M.-F. Roy: Géométrie algébrique réelle. *Ergeb. Math.* **12**, Springer-Verlag, Berlin, Heidelberg, New York (1987).
- [B] M. Berger: Geometry. I. *Universitext*. Springer-Verlag, Berlin: (1987).
- [F] J.F. Fernando: Polynomial images of  $\mathbb{R}^n$ . *Reelle algebraische und analytische Geometrie*, March, 17<sup>th</sup> – 23<sup>rd</sup> (2002), Oberwolfach.
- [F1] J.F. Fernando: On Nash images of Euclidean spaces. Submitted, arXiv:1503.07567.
- [FG] J.F. Fernando, J.M. Gamboa: Polynomial images of  $\mathbb{R}^n$ . *J. Pure Appl. Algebra*, **179** (2003), 241–254.
- [FGU] J.F. Fernando, J.M. Gamboa, C. Ueno: The open quadrant problem: A topological proof. A mathematical tribute to Professor José María Montesinos. Departamento de Geometría y Topología. Facultad de CC. Matemáticas UCM, (2016), 337–350.
- [FU] J.F. Fernando, C. Ueno: A short proof for the open quadrant problem. *Preprint RAAG*. Submitted to MEGA 2015, 8 pages.
- [FU1] J.F. Fernando, C. Ueno: On the open quadrant as a polynomial image of  $\mathbb{R}^2$  (revisited).
- [G] J.M. Gamboa: Algebraic images of the real plane. *Reelle algebraische Geometrie*, June, 10<sup>th</sup> – 16<sup>th</sup> (1990), Oberwolfach.
- [H] A. Harnack: Über die Vielheiligkeit der ebenen algebraischen Kurven. *Math. Ann.* **10** (1876), 189–198.

- [M] W. Massey: A Basic Course in Algebraic Topology, *Graduate Texts in Mathematics*, **127**. Springer-Verlag, New York: (1991).
- [P] S. Pinchuk: A counterexample to the real Jacobian Conjecture, *Math. Z.* **217** (1994), 1–4.
- [T] A. Tarski: A decisión method for elementary algebra and geometry. Prepared for publication by J.C.C. Mac Kinsey, Berkeley: (1951).
- [U] C. Ueno: Imágenes polinómicas y regulares de espacios euclídeos. Ph.D. Thesis UCM, (2012).
- [W] R.J. Walker: Algebraic curves. Berlin, Heidelberg, New York: Springer Verlag, (1978).



## Auxiliary definitions and results

### A.1 Real algebraic geometry basics

In this section we give a basic insight into the required concepts, definitions and results from real algebraic geometry. They are mostly used on the first chapter.

We begin with some definitions like *real closed field* and *semialgebraic set*. The *Transfer Principle* allows us to move our results from  $\mathbb{R}$  to an arbitrary real closed field  $R$ . We also give an overview of the *Zariski topology*, in order to talk about concepts like closure or *reducibility* in this topology. Finally, after defining other notions as *proper maps* and *germ Nash half-branch curves*, we finish the section recalling *Puiseux series*, since this is the nature of the solutions of the equation  $\Delta_v(\mathbf{x}) = 0$  in the first proof of Theorem 1.11.

**Definition A.1.** A *real closed field* is an ordered field  $R$  such that  $R(\sqrt{-1})$  is an algebraically closed field. There exist many characterizations of real closed fields. A very enlightening one is the following:  $R$  is a real closed field if it is an ordered field that shares with the field  $\mathbb{R}$  of real numbers its properties of the first-order language of ordered fields.

Indeed, [Tarski-Seidenberg Theorem](#) admits a useful formulation in model theory that explains accurately our last sentence.

**Definitions A.2.** (i) Let  $R$  be a real closed field. A subset  $S \subset R^n$  is *semialgebraic* if it is defined as a finite union of sets defined by a conjunction of polynomial equalities and inequalities:

$$\left\{ \begin{array}{l} P_1(\mathbf{x}_1, \dots, \mathbf{x}_n) = 0 \\ \vdots \\ P_r(\mathbf{x}_1, \dots, \mathbf{x}_n) = 0 \\ Q_1(\mathbf{x}_1, \dots, \mathbf{x}_n) > 0 \\ \vdots \\ Q_\ell(\mathbf{x}_1, \dots, \mathbf{x}_n) > 0 \end{array} \right.$$

It is easily seen that finite unions and intersections of semialgebraic sets are semialgebraic sets too. In addition, the complementary set of a semialgebraic set is also a semialgebraic set. The easy proof of these facts can be studied in [BCR, Ch.2].

(ii) A map  $f : S \rightarrow T$  between semialgebraic subsets  $S \subset R^n$  and  $T \subset R^m$  is *semialgebraic* if its graph is a semialgebraic subset of  $R^{m+n}$ .

(iii) A *semialgebraic homeomorphism* between two semialgebraic subsets  $S \subset R^n$  and  $T \subset R^m$  is a continuous and bijective semialgebraic map  $f : S \rightarrow T$ . It is easily seen that in such a case its inverse  $f^{-1} : T \rightarrow S$  is also semialgebraic.

### Theorem A.3 (Transfer Principle)

Let  $\mathcal{L}(R)$  be the first-order language of ordered fields with parameters in the real closed field  $R$  and let  $\Phi$  be a formula of  $\mathcal{L}(R)$ . Then, there exists a quantifier-free formula  $\Psi$  of  $\mathcal{L}(R)$  with the same free variables  $x_1, \dots, x_n$  as  $\Phi$  such that, for every real closed field extension  $K$  of  $R$  and every  $x \in K^n$ , the sentence  $\Phi(x)$  holds true if and only if  $\Psi(x)$  holds true.

**Definition A.4** (Zariski topology). (i) Let  $K$  be a field. A subset  $X \subset K^n$  is *algebraic* if it is the set of common zeros of a finite family of polynomials  $f_1, \dots, f_m \in K[x_1, \dots, x_n]$ .

(ii) The *Zariski topology* of an algebraic set  $X \subset K^n$  is the topology whose closed sets are the algebraic subsets of  $K^n$  contained in  $X$ . It is indeed a topology, that is, the arbitrary intersection of algebraic sets is also an algebraic set as a straightforward consequence of Hilbert's basis theorem.

(iii) An algebraic subset  $X \subset K^n$  is said to be *reducible* if there exist algebraic subsets  $Y \subsetneq X$  and  $Z \subsetneq X$  such that  $X = Y \cup Z$ . If  $X$  is not reducible it is said to be *irreducible*.

**Definition A.5.** (i) Let  $R$  be a real closed field and let  $S \subset R^n$  be a semialgebraic set. The *local dimension of  $S$  at a point  $p \in S$* , denoted  $\dim(S_p)$ , is the largest non-negative integer  $d$  such that for every open ball  $B$  centered at  $p$  the intersection  $S \cap B$  contains a semialgebraic subset semialgebraically homeomorphic to the cube  $[0, 1]^d$ .

It is said that  $S$  is *pure dimensional* if  $\dim(S_p) = \dim(S_q)$  for every pair of points  $p, q \in S$ .

**Definitions A.6.** (i) A continuous map  $f : X \rightarrow Y$  between topological spaces  $X$  and  $Y$  is said to be *proper* if  $f^{-1}(K)$  is a compact subspace of  $X$  for every compact subspace  $K$  of  $Y$ .

(ii) A semialgebraic map  $f : S \rightarrow T$  between semialgebraic sets  $S \subset R^n$  and  $T \subset R^m$  is said to be *semialgebraically proper* if  $f^{-1}(K)$  is bounded and closed in  $S$  for every bounded and closed in  $T$  subset  $K$  of  $T$ .

**Definition A.7.** A polynomial map  $f : X \rightarrow Y$  between algebraic sets  $X$  and  $Y$  is said to be *dominant* if its image  $f(X)$  is a dense subset of  $Y$  in its Zariski topology.

**Definitions A.8.** (i) A function  $f : U \rightarrow \mathbb{R}$  defined in an open semialgebraic subset  $U \subset \mathbb{R}^n$  is a *Nash function* if it is analytic and semialgebraic.

(ii) A map  $f := (f_1, \dots, f_m) : U \rightarrow \mathbb{R}^m$  is a *Nash map* if each coordinate  $f_i : U \rightarrow \mathbb{R}$  is a Nash function.

(iii) Let  $\Gamma \subset \mathbb{R}^n$  be an algebraic curve and  $p \in \Gamma$ . For every small enough  $\varepsilon > 0$  the intersection  $B_\varepsilon \cap \Gamma \setminus \{p\}$ , where  $B_\varepsilon \subset \mathbb{R}^n$  is the open ball of radius  $\varepsilon$  centered at the point  $p$ , has finitely many connected components  $C_1, \dots, C_k$ . Each  $C_i$  is semialgebraically homeomorphic to the interval  $(0, 1]$  and in fact for  $i = 1, \dots, k$  there exists a Nash homeomorphism  $f_i : [0, 1] \rightarrow C_i \cup \{p\}$ , with  $f_i(0) = p$ .

Indeed, this result is a particular case of the local conic structure theorem of semialgebraic sets, [BCR, IX.3.6].

Observe that, by its very definition, it makes sense to define the germs  $C_{i,p}$  of  $C_i$  at the point  $p$  for  $i = 1, \dots, k$  as they are independent of the radius  $\varepsilon$ . These germs  $C_{i,p}$  are called the *germ Nash half-banches* of the curve  $\Gamma$  centered at  $p$ . For more details see [BCR, IX.5.2].

**Definition A.9.** It is proved in [W, pp. 98-102] that given an algebraically closed field  $K$  and an indeterminate  $t$  over  $K$ , the field  $K(\{t^*\})$  of *Puiseux series* with coefficients in  $K$  is algebraically closed. As it is an algebraic extension of the field  $K(t)$  of rational functions over  $K$ , the field  $K(\{t^*\})$  is an algebraic closure of the field  $F(t)$  for every subfield  $F$  of  $K$  such that the field extension  $K|F$  is algebraic. In particular, for  $F := \mathbb{R}$  it follows that  $\mathbb{C}(\{t^*\})$  is an algebraic closure of  $\mathbb{R}(t)$ .

## A.2 Root finding algorithms

The first proof of Theorem 1.11 requires frequently to check whether a given polynomial  $f$  has some root on a certain interval  $I$ . Sometimes we want to check that  $f$  is positive or negative on  $I$ . To that end it suffices to evaluate it at an arbitrary point  $a \in I$  and to check that  $f$  has no root on  $I$ .

Sturm's method is used to compute the number of different real roots of a polynomial in a given interval. Note that it is an algebraic method that doesn't rely on approximations.

**Definitions A.10.** (i) Let  $R$  be a real closed field and let  $f \in R[t]$ . The *Sturm sequence* (or *Sturm chain*) of  $f$  is the finite sequence of polynomials  $(f_0, f_1, \dots, f_k)$  defined as follows:

$$\begin{aligned} f_0 &:= f \\ f_1 &:= f' \\ &\dots \\ f_i &:= f_{i-1}q_i - f_{i-2}, \text{ with } q_i \in R[x] \text{ and } \deg(f_i) < \deg(f_{i-1}) \text{ for } i = 2, \dots, k. \end{aligned}$$

Then, by Euclid's Algorithm, there is an integer  $k$  satisfying  $f_k = \gcd(f, f')$ .

(ii) Given a sequence  $(a_0, a_1, \dots, a_k)$  of elements of  $R$  with  $a_0 \neq 0$ , we define the *number of sign changes in the sequence*  $(a_0, \dots, a_k)$  as follows: count one sign change every time  $a_i a_\ell < 0$ , with

$$\begin{aligned} \ell &= i + 1, \text{ or} \\ \ell &> i + 1 \text{ and } a_j = 0 \text{ for every } j \text{ satisfying } i < j < \ell. \end{aligned}$$

(iii) If  $a \in R$  is not a root of  $f$  and  $(f_0, \dots, f_k)$  is the Sturm sequence of  $f$ , we define  $v(f; a)$  to be the number of sign changes in  $(f_0(a), \dots, f_k(a))$ .

**Proposition A.11** (Sturm's Theorem). *Let  $R$  be a real closed field and  $f \in R[t]$ . Let  $a, b \in R$  be such that  $a < b$  and neither  $a$  nor  $b$  are roots of  $f$ . Then the number of roots of  $f$  in the interval  $(a, b)$  equals  $v(f; a) - v(f; b)$ .*

The proof of this proposition can be studied in [BCR, 1.2.10].

Laguerre's method differs from Sturm's method in that it is a numerical algorithm, rather than algebraic, to decide if a polynomial with real coefficients has a real root. It is remarkable that it converges to a root, with very few exceptions, from any initial value.

**Proposition A.12** (Laguerre's method). *Let  $f \in \mathbb{R}[\mathbf{x}]$  be a polynomial of degree  $n$  and write it as*

$$f(\mathbf{x}) = \prod_{i=1}^n (\mathbf{x} - x_i),$$

where  $x_i \in \mathbb{C}$ ,  $i = 1, \dots, n$ , are its roots. Consider

$$\ln|f(\mathbf{x})| = \ln \left| \prod_{i=1}^n (\mathbf{x} - x_i) \right| = \sum_{i=1}^n \ln|\mathbf{x} - x_i|.$$

Then, taking its derivative:

$$(\ln|f(\mathbf{x})|)' = \left( \sum_{i=1}^n \ln|\mathbf{x} - x_i| \right)' = \sum_{i=1}^n (\ln|\mathbf{x} - x_i|)' = \sum_{i=1}^n \frac{1}{\mathbf{x} - x_i} = \frac{f'(\mathbf{x})}{f(\mathbf{x})} := g(\mathbf{x}).$$

Also, consider

$$-g(\mathbf{x})' = - \sum_{i=1}^n \left( \frac{1}{\mathbf{x} - x_i} \right)' = \sum_{i=1}^n \frac{1}{(\mathbf{x} - x_i)^2} = g^2(\mathbf{x}) - \frac{f''(\mathbf{x})}{f(\mathbf{x})} := h(\mathbf{x}).$$

Now, making a “rather drastic set of assumptions” ([A, page 187]), suppose that the root we are seeking  $x_1$  is isolated and that the rest of them  $x_i$  are clustered at a certain distance from it, namely:  $a = \mathbf{x} - x_1$  and  $b = \mathbf{x} - x_i$ ,  $i = 2, \dots, n$ . Then, we can rewrite  $g$  and  $h$  as:

$$g(\mathbf{x}) = \frac{1}{\mathbf{x} - x_1} + \frac{n-1}{\mathbf{x} - x_i} = \frac{1}{a} + \frac{n-1}{b} \quad \text{and} \quad h(\mathbf{x}) = \frac{1}{(\mathbf{x} - x_1)^2} + \frac{n-1}{(\mathbf{x} - x_i)^2} = \frac{1}{a^2} + \frac{n-1}{b^2}.$$

Then, if we solve for  $a$ , we get a quadratic equation, whose solution is named Laguerre's formula:

$$a = \frac{n}{g(\mathbf{x}) \pm \sqrt{(n-1)(nh(\mathbf{x}) - g^2(\mathbf{x}))}}. \tag{A.1}$$

Equation A.1 defines an iterative algorithm in the following way:

- (1) Start with a trial random value  $x_0$ .
- (2) Compute  $f(x_k)$ , and if  $|f(x_k)|$  is smaller than a certain threshold, stop the algorithm and return  $x_k$ . If not, go to step (3).
- (3) Compute  $g(x_k)$  and  $h(x_k)$ . Compute  $a$  using A.1, by choosing the sign that results in the larger magnitude of the denominator. Take the new root to be:

$$x_{k+1} = x_k - \frac{n}{g(x_k) \pm \sqrt{(n-1)(nh(x_k) - g^2(x_k))}}.$$

- (4) Go to step (2).

A possible implementation for this algorithm is shown in Appendix B.1, using Python.

### A.3 Fundamental group

Let us recall, for the sake of completeness, the notions of *path* and *path concatenation*. Also, we recall the notion of *fundamental group* of a (path connected) topological space.

**Definitions A.13.** (i) Let  $X$  be a topological space. A continuous map  $\alpha : [a, b] \rightarrow X$ , where  $[a, b] \subset \mathbb{R}$  is a closed interval is a *path* in  $X$ . If  $\alpha(a) = \alpha(b) = x_0 \in X$ , we say that  $\alpha$  is a *loop* with base point  $x_0$ . In such a case,  $\alpha$  may be viewed as a map whose domain is the quotient space  $[a, b]/\sim$  obtained by identifying the end points of  $[a, b]$ . Notice that  $[a, b]/\sim$  is homeomorphic to the circumference  $\mathbb{S}^1 := \{x^2 + y^2 = 1\} \subset \mathbb{R}^2$ . After a linear change of variables we may assume, if necessary, that  $[a, b] = [0, 1]$ .

(ii) Given two paths  $\alpha : [0, 1] \rightarrow X$  and  $\beta : [0, 1] \rightarrow X$  such that  $\alpha(1) = \beta(0)$  the *concatenation* (also called *product*) of  $\alpha$  and  $\beta$  is defined as follows:

$$\alpha * \beta : [0, 1] \rightarrow X, t \mapsto \begin{cases} \alpha(2t) & \text{if } 0 \leq t \leq \frac{1}{2}, \\ \beta(2t - 1) & \text{if } \frac{1}{2} \leq t \leq 1. \end{cases}$$

The idea under this definition is clear. We use  $\alpha$  in the “first part of the path”, but the speed must be double of the original one to reach the point  $\alpha(1)$  at  $t := \frac{1}{2}$ . This explains the definition of the first part of the new path. We must use  $\beta$  for the second part. Necessarily  $(\alpha * \beta)(\frac{1}{2}) = \beta(0)$  and  $(\alpha * \beta)(1) = \beta(1)$ . To do that, we use the line that joints the points  $p := (\frac{1}{2}, 0)$  and  $q := (1, 1)$ , that is  $\gamma(t) := 2t - 1$ . Of course this is not at all the only way to construct a path joining  $p$  and  $q$ , but seems to be the simplest one.

It is easily checked that the concatenation of loops enjoys the associative property. Besides, fixed  $x_0 \in X$ , the constant loop  $e_{x_0} : [0, 1] \rightarrow X, t \mapsto x_0$  satisfies the equality  $e_{x_0} * \alpha = \alpha = \alpha * e_{x_0}$  for every loop  $\alpha$  in  $X$  of basis  $x_0$ . In addition, given a loop  $\alpha$  in  $X$  the loop  $\alpha^{-1} : [0, 1] \rightarrow X, t \mapsto \alpha(1-t)$  that travels the same curve as  $\alpha$  but in the opposite direction satisfies  $\alpha * \alpha^{-1} = \alpha^{-1} * \alpha = e_{x_0}$ .

(iii) Let  $I := [0, 1] \subset \mathbb{R}$  and let  $X, Y$  be two path-connected topological spaces. A *homotopy* is a continuous map  $H : X \times I \rightarrow Y$ . We associate to  $H$  the family of continuous maps  $\{H_s : X \rightarrow Y, x \mapsto H(x, s)\}_{s \in I}$ , and it is said that the maps  $H_0 : X \rightarrow Y$  and  $g := H_1 : X \rightarrow Y$  are *homotopic*. We denote this with  $f \xrightarrow{H} g$ . If, in addition,  $g$  is constant we say that  $f$  is *null-homotopic*.

(iv) Given a path-connected topological space  $X$  and a point  $x_0 \in X$  the *fundamental group of  $X$  at the base point  $x_0$*  is the quotient space of loops of base  $x_0$  modulo the “homotopy relation”, and it is denoted  $\pi_1(X, x_0)$ . If  $\alpha$  is a loop with base point  $x_0$ , then it represents an element of  $\pi_1(X, x_0)$  denoted with  $[\alpha]$ . The concatenation of loops is compatible with the homotopy relation and endows  $\pi_1(X, x_0)$  with a structure of group.

(v) We impose  $X$  to be path connected to be sure that the fundamental groups  $\pi_1(X, x_0)$  and  $\pi_1(X, x_1)$  corresponding to two distinct points  $x_0$  and  $x_1$  in  $X$  are isomorphic. Indeed, if  $\gamma$  is a path in  $X$  that connects  $x_0$  and  $x_1$  we associate to each loop  $\alpha$  with base  $x_0$  the loop of base  $x_1$  defined as the product  $\gamma \cdot \alpha \cdot \gamma^{-1}$ . This construction is compatible with the homotopy relation, so it induces a homomorphism  $\pi_1(X, x_0) \rightarrow \pi_1(X, x_1)$ . This is in fact an isomorphism because we can associate to every loop  $\beta$  with base  $x_1$  the loop of base  $x_0$  defined as the product  $\gamma^{-1} \cdot \beta \cdot \gamma$ . This is why it is usual to denote  $\pi_1(X)$  the fundamental group of  $X$  without any reference to the base point  $x_0$ .

(vi) A path connected space  $X$  is said to be *simply connected* if its fundamental group is trivial, that is, all loops are null-homotopic. Intuitively this means that all loops in  $X$  are contractible to a point.



## Python code

### B.1 Laguerre's Method

..../Laguerre/polyRoots.py

```
1 # module polyRoots
2 ''' roots = polyRoots(a).
3     Uses Laguerre's method to compute all the roots of
4     a[0] x^n + a[1] x^(n-1) + ... + a[n-1] x + a[n] = 0.
5     The roots are returned in the array 'roots'.
6 '''
7 import numpy as np
8 import cmath
9 from random import random
10
11
12 def polyRoots(a, tol=1.0e-12):
13
14     def laguerre(a, tol):
15         # Starting value (random number)
16         x = random()
17         n = len(a) - 1
18         for i in range(30):
19             p = np.polyval(a, x)
20             dp = np.polyval(np.polyder(a, 1), x)
21             ddp = np.polyval(np.polyder(a, 2), x)
22             if abs(p) < tol:
23                 return x
24             g = dp / p
25             h = g * g - ddp / p
26             f = cmath.sqrt((n - 1) * (n * h - g * g))
27             if abs(g + f) > abs(g - f):
28                 dx = n / (g + f)
```

```

29         else:
30             dx = n / (g - f)
31             x = x - dx
32             if abs(dx) < tol:
33                 return x
34             print('Too many iterations...')

35
36     def deflPoly(a, root): # Deflates a polynomial
37         n = len(a) - 1
38         b = [(0.0 + 0.0j)] * n
39         # The leading coefficient still unchanged:
40         b[0] = a[0]
41         for i in range(1, n):
42             b[i] = a[i] + root * b[i - 1]
43         return b
44
45     n = len(a) - 1
46     roots = np.zeros((n), dtype=complex)
47     for i in range(n):
48         x = laguerre(a, tol)
49         if abs(x.imag) < tol:
50             x = x.real
51             roots[i] = x
52             a = deflPoly(a, x)
53     return roots

```

## B.2 Examples

..../Laguerre/polyExample.py

```

1 #!/usr/bin/python
2
3 from polyRoots import *
4 import numpy as np
5
6 # Example 1: x^4 - 5x^3 - 9x^2 + 155x - 250
7 c = np.array([1.0, -5.0, -9.0, 155.0, -250.0])
8 rootList = polyRoots(c)
9 print('x^4 - 5x^3 - 9x^2 + 155x - 250\nRoots are:\n')
10 print '[%s]' % ',\n'.join(map(str, rootList)), '\n'
11
12
13 # Example 2: x^7 + 2x^4 + x^3 + x
14 c = np.array([1, 0, 0, 2, 1, 0, 1, 0])
15 rootList = polyRoots(c)

```

```

16 print('x^7 + 2x^4 + x^3 + x\nRoots are:\n')
17 print '[%s]' % ',\n '.join(map(str, rootList)), '\n'
18
19 # Example 3: Delta(x, 0.28^2) =
20 #           0.0784*x^6-x^4-1.8432*x^3-0.9216*x^2+0.0784
21 c = np.array([0.0784, 0, -1.0, -1.8432, -0.9216, 0, 0.0784])
22 rootList = polyRoots(c)
23 print('Delta(x, 0.28^2)\nRoots are:\n')
24 print '[%s]' % ',\n '.join(map(str, rootList)), '\n'

```

Output for the given examples:

\* **Example 1:**

```

x^4 - 5x^3 - 9x^2 + 155x - 250
Roots are:
[(2+0j),
 (4-3j),
 (4+3j),
 (-5+0j)]

```

\* **Example 2:**

```

x^7 + 2x^4 + x^3 + x
Roots are:
[(0.224542684789-0.690314534559j),
 (-9.29688929465e-14+0j),
 (0.224542684789+0.690314534559j),
 (0.780156137307+1.03569383548j),
 (-1.0046988221+0.345379300925j),
 (0.780156137307-1.03569383548j),
 (-1.0046988221-0.345379300925j)]

```

\* **Example 3:**

```

Delta(x, 0.28^2)
Roots are:
[(0.235691061376+0j),
 (-0.527292033951-0.194675359692j),
 (-0.527292033951+0.194675359692j),
 (-1.7584904885+0.0707240369592j),
 (-1.7584904885-0.0707240369592j),
 (4.33587398352+0j)]

```



# C

## Sage code

All the computations performed in this Appendix were done with a server provided by the Faculty of Mathematics of the Complutense University of Madrid (UCM): [link to website](#).

### C.1 Computations related to Chapter 1

Graphs from Example 1.8 (ii).

```
var('x, y')
hyper = region_plot([x*y<1], (x, -10, 10), (y, -10, 10), alpha
=0.9, bordercol='blue', incol='lightblue', legend_label='
\$S_1\$')
hyper.axes_labels(['$u$', '$v$'])
hyper.save('ch1_01_S_1.svg')

hyper_other = region_plot([x*y>1, x>0], (x,-2,10), (y,-2,10),
alpha=0.9, bordercol='blue', incol='lightblue')
hyper_other.axes_labels(['$u$', '$v$'])
hyper_other.save('ch1_02_S_2.svg')
```

### C.2 Computations related to Chapter 2

Define the polynomials  $\mathcal{F}_1, \mathcal{F}_2, \mathcal{G}_1, \mathcal{G}_2, \mathcal{F}, \mathcal{G}$  and  $\mathcal{P}$  in  $\mathbb{R}[x, y]$ .

```
var('a, b, s, x, y')
F_1(x, y) = 1 - x**3 * y + y - x*y**2
F_2(x, y) = x**2 * y
G_1(x, y) = 1 - x * y + x - x**4 * y
G_2(x, y) = x**2 * y
F(x, y) = F_1**2 + F_2**2; show('F: R^2 -> R'); show(F)
G(x, y) = G_1**2 + G_2**2; show('G: R^2 -> R'); show(G)
P(x, y) = (F, G); show('P = (F, G): R^2 -> R^2'); show(P)
```

Graph of  $\mathcal{F}(x, y)$ .

```
F(x, y) = (1 - x**3*y + y - x*y**2)**2 + (x**2*y)**2
plot3d(F, (x, -2.9, 3), (y, -5, 2), adaptive=True, mesh=True,
        opacity=0.8, color=rainbow(100, 'rgbtuple'))
```

Graph of  $\mathcal{G}(x, y)$ .

```
G(x, y) = (1 - x*y + x - x**4*y)**2 + (x**2*y)**2
plot3d(G, (x, -3, 3), (y, -3, 3), adaptive=True, mesh=True,
        opacity=0.8, color=rainbow(100, 'rgbtuple'))
```

Define  $\alpha_\lambda(s)$  and  $\beta_\mu(s)$ .

```
alpha(s) = (s, (1+a*s)/s); show('alpha_lambda: R -> R^2')
show(alpha)
beta(s) = ((1+b*s)/s, s**3); show('beta_mu: R -> R^2')
show(beta)
```

$\mathcal{F}$  and  $\mathcal{G}$  behave well on those curves. 2.1: (a) and (b).

```
show('F_1(alpha) ='); show(F_1(*alpha).simplify_full())
show('F_1(beta) ='); show(F_1(*beta).simplify_full())
show('G_1(alpha) ='); show(G_1(*alpha).simplify_full())
show('G_1(beta) ='); show(G_1(*beta).simplify_full())
```

For printing the resulting polynomials in L<sup>A</sup>T<sub>E</sub>X style:

```
aux = F(*beta).simplify_full()
print "+".join(map(latex, sorted([f for f in aux.operands()],
key=lambda exp: exp.degree(s))))
```

2.1: (c).

```
show('F_2(alpha) =')
show(F_2(s, (1+a*s)/s).simplify_full())
show('G_2(alpha) =')
show(G_2(s, (1+a*s)/s).simplify_full())
show('F_2(beta) =')
show(F_2((1+b*s)/s, s**3).simplify_full())
show('G_2(beta) =')
show(G_2((1+b*s)/s, s**3).simplify_full())
```

2.1: (iii).

```
show('F(alpha) ='); show(F(*alpha).simplify_full())
show('G(alpha) ='); show(G(*alpha).simplify_full())
show('F(beta) ='); show(F(*beta).simplify_full())
show('G(beta) ='); show(G(*beta).simplify_full())
```

Now we proceed with the computations performed in the first proof of Theorem 1.11.

**Step 1:**

Roots of  $\mathcal{G} = v$  in  $y$ :  $y^+(x, v), y^-(x, v)$ .

We can see that the denominator only cancels in 0. This is checked with Python (Appendix B). The method `roots` returns the roots with its multiplicity:

```
var('v')
y_1, y_2 = solve(G-v, y)
show('y+: D_v -> R'); show(y_1)
show('y-: D_v -> R'); show(y_2)
show('The denominator, which is the same, only cancels in 0: ',
      expand(x * (x**2 + (x**3 + 1)**2)))
show((x * (x**2 + (x**3 + 1)**2)).roots())
#Actual check that it doesn't cancel on the rest of them:
plot(x**2+(x**3+1)**2, xmin=-2, xmax=2, ymin=-1, ymax=10)
```

Define  $\Delta(x, v) = v(x^2 + (x^3 + 1)^2) - x^2(x + 1)^2$ :

```
Delta(x, v) = v * (x**2 + (x**3 + 1)**2) - x**2*(x + 1)**2
show('Delta: R^2 -> R'); show(expand(Delta))
```

Define  $y^+(x, v), y^-(x, v)$  as functions in order to manipulate them. Recall that

$$D_v = \{x \in R : \Delta(x, v) \geq 0, x \neq 0\}.$$

```
y_1(x, v) = (1 + x + x **3 + x**4 + sqrt(Delta)) / (x * (x**2 +
(x**3 + 1)**2))
show('y+: D_v -> R'); show(y_1)
y_2(x, v) = (1 + x + x **3 + x**4 - sqrt(Delta)) / (x * (x**2 +
(x**3 + 1)**2))
show('y-: D_v -> R'); show(y_2)
```

Check of how they look like for instance for  $v = 0.8$  (Figure 2.3).

```
sols = plot(y_1(x, 0.8), xmin=-10, xmax=10, ymin=-10, ymax=10,
            legend_label='\$y^+(x, 0.8)$') +
       plot(y_2(x, 0.8), xmin=-10, xmax=10, ymin=-10, ymax=10,
            color='green', legend_label='\$y^-(x, 0.8)$') +
       text('\$y$', (-1, 9), fontsize=20, color='black') +
       text('\$x$', (9, -1), fontsize=20, color='black')
sols.save('ch1_06_sols.svg')
```

As for  $v = 1$ , it will be useful later on. We see that  $y(x, 1) \approx 1$  when  $x \rightarrow 0$ , in red. Thus, we have that  $\gamma_1(x) = F(x, y^-(x, 1)) \approx F(x, 1)$  if  $x \rightarrow 0$ , so we obtain  $= (1 + 1)^2 = 4$ .

```
sols_1 = plot(y_1(x, 1), xmin=-10, xmax=10, ymin=-10, ymax=10,
               legend_label='\$y^+(x, 1)$') +
        plot(y_2(x, 1), xmin=-10, xmax=10, ymin=-10, ymax=10, color=
             'red', thickness=2, legend_label='\$y^-(x, 1)$') +\
```

```

    text(''$y$'', (-1,9), fontsize=20, color='black') +\
    text(''$x$'', (9,-1), fontsize=20, color='black')
sols_1.save('ch1_07_sols_1.svg')

```

```
plot(y_1(x, 0.8), xmin=0, xmax=2, ymin=0, ymax=2) + plot(y_2(x, 0.8), xmin=0, xmax=2, ymin=0, ymax=2, color='green')
```

Animated pic of  $y^+(x, v)$ ,  $y^-(x, v)$  for fixed  $x$  and  $v$  varying between 0 and 5, step is 0.05.

```
G_sols = [plot(y_1(x, c), xmin=-10, xmax=10, ymin=-10, ymax=10) + plot(y_2(x, c), xmin=-10, xmax=10, ymin=-10, ymax=10, color='green') for c in sxrange(0, 5, .05)]
animate(G_sols)
```

### Step 2:

Define  $\gamma_v^+(x)$ ,  $\gamma_v^-(x) : D_v \rightarrow R$ .

```
gamma_1(x) = F(x, y_1).simplify_full(); show('gamma+: D_v -> R, ', gamma_1)
gamma_2(x) = F(x, y_2).simplify_full(); show('gamma-: D_v -> R, ', gamma_2)
```

Compute the limits from (i) y (ii) in Step 2.

```
show('lim of gamma+ when x -> infinity: ', lim(gamma_1, x=oo))
show('lim of gamma- when x -> infinity: ', lim(gamma_2, x=oo))
show('lim of gamma+ when x -> -infinity: ', lim(gamma_1, x=-oo))
show('lim of gamma- when x -> -infinity: ', lim(gamma_2, x=-oo))
show('lim of gamma+ when x -> 0: ', lim(gamma_1, x=0))
show('lim of gamma- when x -> 0: ', lim(gamma_2, x=0))
```

But we have to check what happens for  $v = 1$ , namely:

$$\lim_{x \rightarrow 0} \gamma_1^-(x).$$

Define  $\gamma_1^+$  as `gamma_1_aux` and  $\gamma_1^-$  as `gamma_2_aux`.

```
gamma_1_aux(x) = F(x, y_1(x, 1)).simplify_full()
show(gamma_1_aux)
gamma_2_aux(x) = F(x, y_2(x, 1)).simplify_full()
show(gamma_2_aux)
```

And the limit is:

```
show('lim de gamma+ cuando x -> 0 es: ', lim(gamma_1_aux, x = 0))
show('lim de gamma- cuando x -> 0 es: ', lim(gamma_2_aux, x = 0))
```

Now we are going to see how the numerator looks like for  $\gamma_v^+$  and  $\gamma_v^-$  when  $x := 0$ .

```

gamma_1_num(x, v) = F(x, y_1(x, v)).numerator()
show('Numerator of gamma+ when x = 0: ', gamma_1_num(0, v), ' =
', v*(1+sqrt(v))**2)
gamma_2_num(x, v) = F(x, y_2(x, v)).numerator()
show('Numerator of gamma- when x = 0: ', gamma_2_num(0, v), ' =
', v*(1-sqrt(v))**2)
gamma_den(x, v) = F(x, y_1(x, v)).denominator(); show(
    Denominators from both of them: ', gamma_den(x, v))

```

And we see that for  $v > 0$ , the first numerator is  $> 0$  (in purple) and the second one  $\geq 0$  (in orange).

```

numerators = plot(v*(1+sqrt(v))**2, xmin=0.01, xmax=2, ymin=0,
    ymax=2, color='purple', legend_label='Numerator of $\gamma_v^+')
+ \
plot(v*(1-sqrt(v))**2, xmin=0.01, xmax=2, ymin=0, ymax=2,
    color='orange', legend_label='Numerator of $\gamma_v^-')
, fontsize=13)
numerators.axes_labels(['$v$', ''])
numerators.set_legend_options(font_size=15, borderpad=0.35)
numerators.save('ch1_08_numerators.svg')

```

Plots of  $\gamma_1^+$  in blue, and  $\gamma_1^-$  in green, where we can see the value of the limit in  $x = 0$  (red point).

```

limit = circle((0,4), 0.12, fill=True, rgbcolor='red')
aux = plot(gamma_1_aux, xmin=-10, xmax=10, ymin=0, ymax=18,
    aspect_ratio=1, legend_label='$\gamma_1^+(x)$') +
plot(gamma_2_aux, xmin=-10, xmax=10, ymin=0, ymax=18, color=
    'green', legend_label='$\gamma_1^-(x)$') +
plot(limit, legend_label='Limit')
aux.axes_labels(['$x$', ''])
aux.save('ch1_09_limit.svg')

```

Animation of  $\gamma_1^+$  and  $\gamma_1^-$ .

```

gammas = [plot(gamma_1_num(x, c)/gamma_den(x, c), xmin=-10, xmax
=10, ymin=0, ymax=10) + plot(gamma_2_num(x, c)/gamma_den(x, c),
    xmin=-10, xmax=10, ymin=0, ymax=10, color='green') for c
    in srange(0.01, 5, .05)]
animate(gammas)

```

We proceed to study now the map

$$v(x) = \frac{(x+1)^2 x^2}{(x^3+1)^2 + x^2}.$$

```

uve(x) = (x**2 * (x + 1)**2) / (x**2 + (x**3 + 1)**2)
show('v: R -> R'); show(uve)
uve_plot = plot(uve, (x, -10, 10), color="green", legend_label='
    $v(x)$') + \
    plot(0.82, (x, -10, 10), color='red', legend_label='$v =
        0.82$', figsize=7, fontsize=13)
uve_plot.axes_labels(['$x$', ''])
uve_plot.set_legend_options(loc=2, font_size=18, borderpad=0.4)
uve_plot.save('ch1_10_uve.svg')

```

We estimate that  $v(x) < 0.28^2 = 0.0784$  when  $x \in (-\infty, 0)$ .

```

detail_uve = plot(uve, (x, -4, 0), ymin=0.00, ymax=0.08, color="
    green", legend_label='$v(x)$') + \
    plot(0.28**2, (x, -4, 0), color='red', thickness=1,
        legend_label='$v = 0.28^2$', figsize=7, fontsize=13)
detail_uve.axes_labels(['$x$', ''])
detail_uve.set_legend_options(loc=(0.02, 0.765), font_size=18,
    borderpad=0.4, shadow=False)
detail_uve.save('ch1_11_uve_detail.svg')

```

Test of an animated plot for  $\Delta(x, y)$  with  $y \in (0.0001, 5)$  and step 0.05.

```

Delta_plot = [plot(Delta(x, c), xmin=-1, xmax=1, ymin=0, ymax
    =10) for c in srange(0.0001, 5, .05)]
animate(Delta_plot)

```

We would like to prove that  $\Delta(x, 0.28^2)$  has no roots for  $x \in (-\infty, 0)$ . Since we didn't achieve it with Sage, we used Laguerre's method with Python (Appendix B).

```

show('Delta(x, 0.28^2) = ', Delta(x, 0.28**2).expand())
plot(Delta(x, 0.28**2), xmin=-3, xmax=0, ymin=0, ymax=0.2)

```

Idea of what we want to prove in **Step 4**.

```

gamma_what(x) = F(x, y_2(x, 0.1)).simplify_full()
idea = plot(gamma_what, (x, -4, 0), ymin=0, ymax=3.5, color='green',
    legend_label='$\gamma-(x)$', ticks=[[0], [0]]) + \
    text("]", (-1.16, 0), fontsize=20, color='purple') + \
    text("<", (-4, -0.004), fontsize=20, color='purple') + \
    text("$N_v$", (-1.15, -0.25), fontsize=15, color='purple') + \
    line([(-4, -0.004), (-1.15, 0)], thickness=2, color='purple') + \
    line([(-1.14, gamma_what(-1.14)), (-1.14, 0)], thickness=1,
        color='purple', linestyle='--') + \
    circle((-1.14, gamma_what(-1.14)), 0.03, fill=True, rgbcolor=
        'purple') + \
    line([(-1.14, gamma_what(-1.14)), (0, gamma_what(-1.14))],
        thickness=1, color='purple', linestyle='--') +

```

```

text("$\gamma^-(N_v)$", (0.45, gamma.what(-1.14)), fontsize=15, color='purple') + \
text("[", (-0.7, 0), fontsize=20, color='orange') + \
text(">", (0.8, -0.004), fontsize=20, color='orange') + \
text("$\delta_v$", (-0.7, -0.25), fontsize=20, color='orange') + \
line([(-0.7, 0), (0.8, 0)], thickness=2, color='orange') + \
line([(-0.695, gamma.what(-0.695)), (-0.695, 0)], thickness=1, color='orange', linestyle='--') + \
circle((-0.695, gamma.what(-0.695)), 0.03, fill=True, rgbcolor='orange') + \
line([(-0.695, gamma.what(-0.695)), (0, gamma.what(-0.695))], thickness=1, color='orange', linestyle='--') + \
text("$\gamma^-(\delta_v)$", (0.45, gamma.what(-0.695)), fontsize=15, color='orange')
idea.axes_labels(['$x$', '$v$'])
idea.set_legend_options(loc=(0.02, 0.765), font_size=15, borderpad=0.4, shadow=False)
idea.save('ch1_12.idea.svg')

```

Define  $N_v$  and  $\delta_v$ .

```

N(v) = -1/v**(1/2) + 1 + v**(1/2) + v
show('N_v: ', N)
delta(v) = - v**(1/2) - v - 5/2 * v**3/2
show('delta_v: ', delta)

```

And we see that  $-\infty < N_v < \delta_v < 0$  for  $0 < v < 0.28^2$ , with  $N_v$  in blue and  $\delta_v$  in green.

```

N_delta = plot(N, (x, 0, 0.28**2), ymin=-20, ymax=0, color='blue',
               legend_label='$N_v$') +
plot(delta, (x, 0, 0.28**2), ymin=-20, ymax=0, color='green',
      legend_label='$\delta_v$', figsize=7, fontsize=13)
N_delta.axes_labels(['$v$', ''])
N_delta.set_legend_options(loc=(0.8, 0.1), font_size=18,
                           borderpad=0.4, shadow=False)
N_delta.save('ch1_13_comp.svg')

```

Now we check that  $\Delta(N_{w^2}, w^2)$  and  $\Delta(\delta_{w^2}, w^2)$  are positive for  $w \in (0, 0.28)$ , that is  $v \in (0, 0.28^2)$ .

```

Delta_N_delta = plot(Delta(N, v), (x, 0, 0.28**2), ymin=0, ymax=15,
                      color='blue', legend_label='$\Delta(N_v, v)$') +
plot(Delta(delta, v)/v**(5/2), (x, 0, 0.28**2), ymin=0, ymax=15,
      color='green', legend_label='$\Delta(\delta_v, v^{5/2})$', figsize=7, fontsize=13)
Delta_N_delta.axes_labels(['$v$', ''])

```

```
Delta_N_delta.set_legend_options(loc=(0.65, 0.5), font_size=18,
    borderpad=0.4, shadow=False)
Delta_N_delta.save('ch1_14_positive.svg')
```

We take advantage of `numpy` library from Python, in order to use `linspace`.

```
import numpy as np
```

Compute the graphs of  $(N_v, v)$  and  $(\delta_v, v)$ .

```
vpoints1 = np.linspace(0.008, 0.08, 100)
vpoints2 = np.linspace(0.0001, 0.08, 100)
t_array = np.linspace(-10, 0, 500)
Npoints = [N(vpoints1[i]) for i in range(100)]
deltapoints = [delta(vpoints2[i]) for i in range(100)]
uvepoints = [uve(t_array[i]) for i in range(500)]
Njoin = line(zip(Npoints, vpoints1), rgbcolor=(0, 0.6, 1),
    thickness=1.2, legend_label='$(N_v, \ v)$')
deltajoin = line(zip(deltapoints, vpoints2), rgbcolor=(0.196078,
    0.803922, 0.196078), thickness=1.2, legend_label='$(\delta_v, \ v)$')
uve_plot = line(zip(t_array, uvepoints), rgbcolor=(1, 0.3, 0),
    thickness=2, legend_label='$\Delta = 0$')
```

And we draw it alongside the border of  $D$ , which is  $\{\Delta = 0\} \cup \{x = 0\}$ .

```
Delta_aux(x, y)=y * (x**2 + (x**3+1)**2) - x**2*(x+1)**2
```

```
xAxis = line([(0,0), (0,0.08)], color='red', thickness=2,
    legend_label='$x = 0$')
t_array = np.linspace(-10, 0, 0.01)
nice_plot = plot(uve_plot, marker=4) +
    plot(xAxis, (x, -10, 0), ymin=0, ymax=0.08) +
    plot(Njoin)+\
    plot(deltajoin)+\
    region_plot([Delta_aux > 0], (x, -10, 0), (y, 0, 0.08),
        alpha=0.9, bordercol='white', incol='lightblue',
        plot_points=1000, aspect_ratio=85)+ \
    text("$D_v$", (-7, 0.06), fontsize=15, color='black')
nice_plot.axes_labels(['$x$', '$v$'])
nice_plot.set_legend_options(loc=(0.01, 0.8), shadow=True,
    fancybox=True)
nice_plot.save('ch1_15_nice_plot_.svg')
```

Now we must check that  $\gamma_v^-(N_v) > \gamma_v^-(\delta_v)$ . In order to do that, recall the definition of

$$\gamma_v^-(x) = \frac{A_2(x, v) + B_2(x, v)\sqrt{\Delta(x, v)}}{C(x)}.$$

```

A_2(x, v) = (v + 2)*x**24 + (v - 4)*x**22 + 2*x**23 + 2*(2*v +
3)*x**21 + (3*v + 20)*x**20 + 2*(5*v - 3)*x**19 + (13*v - 3)*
x**18 + 2*(3*v + 23)*x**17 + (30*v + 11)*x**16 + 2*(19*v -
12)*x**15 + (18*v + 49)*x**14 + 2*(16*v + 23)*x**13 + (v**2 +
45*v - 30)*x**12 + 2*(21*v + 2)*x**11 + (28*v + 57)*x**10 +
2*(2*v**2 + v + 10)*x**9 + (2*v**2 + 28*v - 11)*x**8 + 12*(2*
v + 1)*x**7 + 2*(3*v**2 - 10*v + 13)*x**6 + 2*(2*v**2 - v +
3)*x**5 + (v**2 + 12*v - 9)*x**4 + 2*(2*v**2 - 3*v - 3)*x**3
+ (2*v**2 + v - 1)*x**2 + v**2 + 4*v*x + v

B_2(x, v) = - 2*(x**21 + x**20 - x**19 + 2*x**18 + 8*x**17 + (v
- 1)*x**15 + 3*x**16 + 9*x**14 + 3*(v + 1)*x**12 + 13*x**13 +
(2*v - 3)*x**11 + 2*(v + 1)*x**10 + (4*v + 7)*x**9 + 2*(v -
4)*x**8 + (7*v - 18)*x**7 + 3*(2*v - 1)*x**6 + (5*v - 11)*x
**4 + (5*v - 7)*x**3 + 2*v*x - x**2 + v)

C(x) = x**2*(x**2 + (x**3 + 1)**2)**4

show((A_2 + B_2 * sqrt(Delta))/C)

```

In order to simplify  $\sqrt{w^2} = w$  we need to tell Sage that  $w > 0$ , which is true because  $w^2 = v$  and recall that  $v > 0$ .

```

var('w')
assume(w > 0)

```

Define now:

$$\begin{aligned}
 f_1(w) &= A_2(N_{w^2}, w^2) \cdot w^{24}, & f_2(w) &= A_2(\delta_{w^2}, w^2), \\
 g_1(w) &= B_2(N_{w^2}, w^2) \cdot w^{21}, & g_2(w) &= B_2(\delta_{w^2}, w^2), \\
 q_1(w) &= \Delta(N_{w^2}, w^2), & q_2(w) &= \Delta(\delta_{w^2}, w^2), \\
 h_1(w) &= C(N_{w^2}) \cdot w^{26}, & h_2(w) &= C(\delta_{w^2}).
 \end{aligned}$$

```

f_1(w) = (A_2(N(w**2), w**2) * w**24).simplify().expand()
f_2(w) = A_2(delta(w**2), w**2).simplify().expand()
g_1(w) = (B_2(N(w**2), w**2) * w**21).simplify().expand()
g_2(w) = B_2(delta(w**2), w**2).simplify().expand()
q_1(w) = Delta(N(w**2), w**2).simplify().expand()
q_2(w) = Delta(delta(w**2), w**2).simplify().expand()
h_1(w) = (C(N(w**2)) * w**26).simplify().expand()
h_2(w) = C(delta(w**2)).simplify().expand()

```

And check that the maps

$$L(w) = \frac{w^2 h_2 f_1 - f_2 h_1}{w^2}, \quad g_1(w), \quad K(w) = \frac{-g_2}{w^3}$$

are positive for  $w \in (0, 0.28)$ .

```
L(w) = ((w**2*h_2*f_1 - f_2*h_1) / w**4).simplify().expand()
L_plot = plot(L, (x, 0, 0.28), ymin=0, ymax=1, rgbcolor=(0,1,0),
    legend_label='\$L(w)$', fontsize=18, thickness=2)
L_plot.axes_labels(['$w$', ''])
L_plot.set_legend_options(loc=(0.75, 0.9), font_size=25,
    borderpad=0.4)
L_plot.save('ch1_16_L.svg')
```

```
g_1_plot = plot(g_1, (x, 0, 0.28), ymin=0, ymax=2, rgbcolor
    =(0.196078, 0.803922, 0.196078), legend_label='\$g_1(w)$',
    fontsize=18, thickness=2)
g_1_plot.axes_labels(['$w$', ''])
g_1_plot.set_legend_options(loc=(0.75, 0.9), font_size=25,
    borderpad=0.4)
g_1_plot.save('ch1_17_g_1.svg')
```

```
K(w) = (-g_2 / w**3).simplify().expand()
K_plot = plot(K, (x, 0, 0.28), ymin=0, ymax=7, rgbcolor=(0,
    0.392157, 0), legend_label='\$K(w)$', fontsize=18, thickness
    =2)
K_plot.axes_labels(['$w$', ''])
K_plot.set_legend_options(loc=(0.75, 0.9), font_size=25,
    borderpad=0.4)
K_plot.save('ch1_18_K.svg')
```

There are some other ways to check that they have no roots in the given interval:

```
show(find_root(L, 0, 0.28))
show(find_root(K, 0, 0.28))
show(find_root(g_1, 0, 0.28))
```

Or:

```
import scipy
from scipy import optimize
scipy.optimize.brentq(K, 0, 0.28)
```

### C.3 Computations related to Chapter 3

The following code produces the graphs from Figure 3.1:  $\mathcal{F}_1, \mathcal{F}_2, \mathcal{G}_1, \mathcal{G}_2, \mathcal{H}_1$  and  $\mathcal{H}_2$ .

```
x, y = var('x, y')
F_1(x, y) = (x*y - 1)**2 + x**2
F_2(x, y) = (x*y - 1)**2 + y**2
G_1(x, y) = x
```

```
G_2(x, y) = y * (x*y - 2)**2 + x * (x*y - 1)**2
H_1(x, y) = x * (x*y - 2)**2 + 1/2*x*y**2
H_2(x, y) = y
```

```
plot3d(F_1, (x, -10, 10), (y, -10, 10), adaptive=True, mesh=True
, opacity=0.8, color=rainbow(100, 'rgbtuple'))
```

```
plot3d(F_2, (x, -10, 10), (y, -10, 10), adaptive=True, mesh=True
, opacity=0.8, color=rainbow(100, 'rgbtuple'))
```

```
plot3d(G_1, (x, -10, 10), (y, -10, 10), adaptive=True, mesh=True
, opacity=0.8, color=rainbow(100, 'rgbtuple'))
```

```
plot3d(G_2, (x, -10, 10), (y, -10, 10), adaptive=True, mesh=True
, opacity=0.8, color=rainbow(100, 'rgbtuple'))
```

```
plot3d(H_1, (x, -10, 10), (y, -10, 10), adaptive=True, mesh=True
, opacity=0.8, color=rainbow(100, 'rgbtuple'))
```

```
plot3d(H_2, (x, -10, 10), (y, -10, 10), adaptive=True, mesh=True
, opacity=0.8, color=rainbow(100, 'rgbtuple'))
```

Regions from Figure 3.2a and 3.2b.

```
A = region_plot([x*y >= 1, x > 0], (x, -1, 5), (y, -1, 5), alpha
=0.9, bordercol='blue', incol='lightblue')
A.axes_labels(['$x$', '$y$'])
A.show()
A.save('ch2_01_A.pdf')
```

```
B = region_plot([x*y >= 1, x > 0], (x, -1, 5), (y, -1, 5), alpha
=0.9, bordercol='blue', incol='lightblue', borderstyle='dashed
') + \
region_plot([y >= x, x > 0.03], (x, -1, 5), (y, -1, 5),
alpha=0.9, bordercol='blue', incol='lightblue',
borderstyle='dashed') + \
plot(x, (x, 0.03, 1), thickness=1.5) + \
plot(1/x, (x, 1, 5), thickness=1.5)
B.axes_labels(['$x$', '$y$'])
B.show()
B.save('ch2_02_B.pdf')
```

Graph that shows that  $1/x$  is “below”  $2/x$ :

```
plot(1/x, (x, 0, 5), ymin=0, ymax=5, aspect_ratio=1) +
plot(2/x, (x, 0, 5), ymin=0, ymax=5, color='green') + B
```

We use `numpy` as well here. First, generate points where we are going to evaluate:

```
import numpy as np
t_array_0 = np.linspace(0.2, 1, 25)
t_array_1 = np.linspace(0, 1, 30)
t_array_2 = np.linspace(1, 5, 120)
```

Then, compute the light blue lines from Figure 3.3a:

```
slice_vert_B_1 = [line([(x_0, x_0), (x_0, 5)], color='lightblue')
                  for x_0 in t_array_1]
slice_vert_B_2 = [line([(x_0, 1 / x_0), (x_0, 5)], color='lightblue')
                  for x_0 in t_array_2]
```

And draw the whole picture:

```
slice_vert_B = sum(slice_vert_B_1) + sum(slice_vert_B_2) + \
    line([(1/3, 1/3), (1/3, 5)], color='green', thickness=1.5) + \
    line([(1/3, 1/3), (1/3, 0)], color='green', linestyle='-.') + \
    line([(3.25, 1/3.25), (3.25, 5)], color='red', thickness
          =1.5) + \
    line([(3.25, 1/3.25), (3.25, 0)], color='red', linestyle='-.')
    line([(0, 0), (1, 1)]) +
plot(1/x, (x, 1, 5), aspect_ratio=1) +
text('$x_0$', (0.35, -0.2), fontsize=20, color='green') +
text('$x_1$', (3.25, -0.2), fontsize=20, color='red')
slice_vert_B.axes_labels(['$x$', '$y$'])
slice_vert_B.save('ch2_09_B_vert.pdf')
show(slice_vert_B)
```

Same for Figure 3.3b:

```
slice_hor_B_0 = [line([(0, y_0), (y_0, y_0)], color='lightblue')
                  for y_0 in t_array_1]
slice_hor_B_1 = [line([(1/y_0, y_0), (5, y_0)], color='lightblue')
                  for y_0 in t_array_0]
slice_hor_B_2 = [line([(0, y_0), (5, y_0)], color='lightblue')
                  for y_0 in t_array_2]
```

And the whole picture:

```
slice_hor_B = sum(slice_hor_B_0) + sum(slice_hor_B_1) + sum(
    slice_hor_B_2) +
    line([(0, 3.25), (5, 3.25)], color='green', thickness=1.5) + \
    line([(0, 0.6), (0.6, 0.6)], color='red', thickness=1.5) + \
    line([(1/0.6, 0.6), (5, 0.6)], color='red', thickness=1.5) + \
    line([(0, 0), (1, 1)]) +
plot(1/x, (x, 1, 5), aspect_ratio=1) +
text('$y_0$', (-0.2, 3.3), fontsize=20, color='green') + \
```

```

text('$y_1$', (-0.2, 0.65), fontsize=20, color='red')
slice_hor_B.axes_labels(['$x$', '$y$'])
slice_hor_B.save('ch2_10_B_hor.pdf')
show(slice_hor_B)

```

## C.4 Computations related to Chapter 5

We need to import `numpy` and define the polynomials we will be using. We define first the polynomial maps from the first proof. Note that the final map is  $P(Q(H(x, y)))$ , but we obtain it as the expanded composition, defined as  $P(G(H))$ , and as a fully expanded composition, which is  $P_{\text{aux}}$ .

```

Q(x, y) = (x, y - x - 1)
H(x, y) = (x*y + 1, x**2 * (x*y + 1) * (x*y + 2) + y)

F(x, y) = (1 - x**3 * y + y - x*y**2)**2 + (x**2 * y)**2
G(x, y) = (1 - x * y + x - x**4 * y)**2 + (x**2 * y)**2
P(x, y) = (F, G)
P_aux(x, y) =
(expand(F(*Q(*H))).collect(y), expand(G(*Q(*H))).collect(y))

show('P = (F, G): R^2 -> R^2')
show(P)

```

Now, the ones from the second one: the short proof.

```

FF_1(x, y) = (x*y - 1)**2 + x**2
FF_2(x, y) = (x*y - 1)**2 + y**2
FF(x, y) = (FF_1, FF_2)
show('FF = (FF_1, FF_2): R^2 -> R^2'); show(FF)

GG_1(x, y) = x
GG_2(x, y) = y * (x*y - 2)**2 + x * (x*y - 1)**2
GG(x, y) = (GG_1, GG_2)
show('GG = (GG_1, GG_2): R^2 -> R^2'); show(GG)

HH_1(x, y) = x * (x*y - 2)**2 + 1/2*x*y**2
HH_2(x, y) = y
HH(x, y) = (HH_1, HH_2)
show('HH = (HH_1, HH_2): R^2 -> R^2'); show(HH)

```

And we define `ff` to be the full expansion of the composition, grouping on the  $y$  variable:

```

ff_1(x, y) = 1/2*((((x*y - 1)**2 + x**2)*((x*y - 1)**2 + y**2) -
1)**2*((x*y - 1)**2 + x**2) + (((x*y - 1)**2 + x**2)*((x*y - 1)**2 + y**2) - 2)**2*((x*y - 1)**2 + y**2))**2*((x*y - 1)**2 + x**2) + (((((x*y - 1)**2 + x**2)*((x*y - 1)**2 + y**2)))

```

```

- 1)**2*((x*y - 1)**2 + x**2) + (((x*y - 1)**2 + x**2)*((x*y
- 1)**2 + y**2) - 2)**2*((x*y - 1)**2 + y**2))*((x*y - 1)**2
+ x**2) - 2)**2*((x*y - 1)**2 + x**2)

ff_2(x, y) = (((x*y - 1)**2 + x**2)*((x*y - 1)**2 + y**2) - 1)
**2*((x*y - 1)**2 + x**2) + (((x*y - 1)**2 + x**2)*((x*y - 1)
**2 + y**2) - 2)**2*((x*y - 1)**2 + y**2)

ff(x, y) = (expand(ff_1).collect(y), expand(ff_2).collect(y))
show('ff = HH(GG(FF)): R^2 -> R^2')#; show(ff)

```

And, lastly, the third polynomial map, the one from the topological proof.

```

FFF_1(x, y) = (x**2*y**4 + x**4*y**2 - y**2 - 1)**2 + x**6*y**4
FFF_2(x, y) = (x**6*y**2 + x**2*y**2 - x**2 - 1)**2 + x**6*y**4
FFF(x, y) = (FFF_1, FFF_2); show('FFF = (FFF_1, FFF_2): R^2 -> R
^2'); show(FFF)

```

```

f_1(x, y) = (x**2, y**2)
f_2(x, y) = ((x*y**2 + x**2*y - y - 1)**2 + x**3*y**2, (x**3*y +
x*y - x - 1)**2 + x**3*y**2)
g(x, y) = (x*y**2 + x**2*y - y - 1, x***(3/2)*y, x**3*y + x*y - x
- 1)
h(x, y, z) = (x**2 + y**2, y**2 + z**2)

```

Actual check of that  $\mathcal{F} = f_2 \circ f_1$  with  $f_2 = h \circ g$ .

```

show(expand(FFF_1))
show(expand(h(*g(*f_1(x, y))))[0]))

show(expand(FFF_2))
show(expand(h(*g(*f_1(x, y))))[1]))

```

### C.4.1 The set $\mathcal{P}_1$

First, define the set  $\mathcal{P}_1$ .

```

smallSq = [(i, j) for i in range(-10,11) for j in range(-10,11)]
point(smallSq)

```

For the map  $g$ :

```

time PQHsmallSq = [P(*Q(*H(*k))) for k in smallSq]
myplot = point(PQHsmallSq, rgbcolor=(1,0.3,0), axes_labels=[ '$x$',
    '$y$' ])
show(myplot)
myplot.save('ch5_01.pdf')

```

For the map  $f$ :

```
time HHGGFFsmallSq = [HH(*GG(*FF(*k))) for k in smallSq]
myplot = point(HHGGFFsmallSq, rbgcolor=(0,0.5,1), axes_labels=[ '$x$', '$y$'])
show(myplot)
myplot.save('ch5_02.pdf')
```

For the map  $\mathcal{F}$ :

```
time FFFsmallSq = [FFF(*k) for k in smallSq]
myplot = point(FFFsmallSq, rbgcolor=(0.196078,0.803922,0.196078),
               axes_labels=[ '$x$', '$y$'])
show(myplot)
myplot.save('ch5_03.pdf')
```

#### C.4.2 The set $\mathcal{P}_2$

First, define the set  $\mathcal{P}_2$ .

```
bigSq = [(i, j) for i in range(-100,101) for j in range
          (-100,101)]
point(bigSq)
```

For the map  $g$ :

```
time PQHbigSq = [P(*Q(*H(*k))) for k in bigSq]
myplot = point(PQHbigSq, rbgcolor=(1,0.3,0), axes_labels=[ '$x$', '$y$'])
show(myplot)
myplot.save('ch5_04.pdf')
```

For the map  $f$ :

```
time HHGGFFbigSq = [HH(*GG(*FF(*k))) for k in bigSq]
myplot = point(HHGGFFbigSq, rbgcolor=(0,0.5,1), axes_labels=[ '$x$', '$y$'])
show(myplot)
myplot.save('ch5_05.pdf')
```

For the map  $\mathcal{F}$ :

```
time FFFbigSq = [FFF(*k) for k in bigSq]
myplot = point(FFFbigSq, rbgcolor=(0.196078,0.803922,0.196078),
               axes_labels=[ '$x$', '$y$'])
show(myplot)
myplot.save('ch5_06.pdf')
```

### C.4.3 The set $\mathcal{P}_3$

First, define the set  $\mathcal{P}_3$ .

```
t_0 = np.linspace(0,1,1000)
zeroOne = [(i, j) for i in t_0 for j in t_0]
```

For the map  $g$ :

```
time PQHzeroOne = [P(*Q(*H(*k))) for k in zeroOne]
myplot = point(PQHzeroOne, bgcolor=(1,0.3,0), axes_labels=['$x$', '$y$'])
show(myplot)
myplot.save('ch5_07.pdf')
```

For the map  $f$ :

```
time HHGGFFzeroOne = [HH(*GG(*FF(*k))) for k in zeroOne]
myplot = point(HHGGFFzeroOne, bgcolor=(0,0.5,1), axes_labels=['$x$', '$y$'])
show(myplot)
myplot.save('ch5_08.pdf')
```

For the map  $\mathcal{F}$ :

```
time FFFzeroOne = [FFF(*k) for k in zeroOne]
myplot = point(FFFzeroOne, bgcolor=(0.196078,0.803922,0.196078),
               axes_labels=['$x$', '$y$'])
show(myplot)
myplot.save('ch5_09.pdf')
```

### C.4.4 The set $\mathcal{P}_4$

First, define the set  $\mathcal{P}_4$ .

```
t_1 = np.linspace(-1,1,1000)
minusOneOne = [(i, j) for i in t_1 for j in t_1]
```

For the map  $g$ :

```
time PQHminusOneOne = [P(*Q(*H(*k))) for k in minusOneOne]
myplot = point(PQHminusOneOne, bgcolor=(1,0.3,0), axes_labels=['$x$', '$y$'])
show(myplot)
myplot.save('ch5_10.pdf')
```

Take a closer look to the graph in  $[0, 2] \times [0, 2]$

```
myplot = point(PQHminusOneOne, bgcolor=(1,0.3,0), xmin=0, xmax=2,
               ymin=0, ymax=2, axes_labels=['$x$', '$y$'])
show(myplot)
#myplot.save('ch5_11_prime.pdf')
```

For the map  $f$ :

```
time HHGGFFminusOneOne = [HH(*GG(*FF(*k))) for k in minusOneOne]
myplot = point(HHGGFFminusOneOne, rbgcolor=(0,0.5,1),
               axes_labels=['$x$', '$y$'])
show(myplot)
myplot.save('ch5_12.pdf')
```

Looking closer:

```
myplot = point(HHGGFFminusOneOne, xmin=0, xmax=1, ymin=0, ymax
               =1, rbgcolor=(0,0.5,1), axes_labels=['$x$', '$y$'])
show(myplot)
myplot.save('ch5_13.pdf')
```

For the map  $\mathcal{F}$ :

```
time FFFminusOneOne = [FFF(*k) for k in minusOneOne]
myplot = point(FFFminusOneOne, rbgcolor
               =(0.196078,0.803922,0.196078), axes_labels=['$x$', '$y$'])
show(myplot)
myplot.save('ch5_14.pdf')
```

Looking closer:

```
myplot = point(FFFminusOneOne, xmin=0, xmax=1, ymin=0, ymax=1,
               rbgcolor=(0.196078,0.803922,0.196078), axes_labels=['$x$', '$y$'])
show(myplot)
myplot.save('ch5_15.pdf')
```

#### C.4.5 The set $\mathcal{P}_5$

First, define the set  $\mathcal{P}_5$ .

```
t_2 = np.linspace(-10,10,200)
minusTenTen = [(i, j) for i in t_2 for j in t_2]
```

For the map  $g$ :

```
time PQHminusTenTen = [P(*Q(*H(*k))) for k in minusTenTen]
myplot = point(PQHminusTenTen, rbgcolor=(1,0.3,0), axes_labels=['$x$', '$y$'])
show(myplot)
myplot.save('ch5_16.pdf')
```

For the map  $f$ :

```
time HHGGFFminusTenTen = [HH(*GG(*FF(*k))) for k in minusTenTen]
myplot = point(HHGGFFminusTenTen, rbgcolor=(0,0.5,1),
               axes_labels=['$x$', '$y$'])
```

```
show(myplot)
myplot.savefig('ch5_17.pdf')
```

For the map  $\mathcal{F}$ :

```
time FFFminusTenTen = [FFF(*k) for k in minusTenTen]
myplot = point(FFFminusTenTen, bgcolor=(0.196078, 0.803922, 0.196078), axes_labels=['$x$', '$y$'])
show(myplot)
myplot.savefig('ch5_18.pdf')
```

#### C.4.6 The set $\mathcal{P}_6$

First, define the set  $\mathcal{P}_6$ .

```
t_3 = np.linspace(-10, 10, 2000)
minusTenTen = [(i, j) for i in t_3 for j in t_3]
```

For the map  $g$ :

```
time PQHminusTenTen = [P(*Q(*H(*k))) for k in minusTenTen]
myplot = point(PQHminusTenTen, bgcolor=(1, 0.3, 0), axes_labels=['$x$', '$y$'])
show(myplot)
myplot.savefig('ch5_19.pdf')
```

Same graph in  $[0, 1] \times [0, 1]$ :

```
myplot = point(PQHminusTenTen, xmin=0, xmax=1, ymin=0, ymax=1,
               bgcolor=(1, 0.3, 0), axes_labels=['$x$', '$y$'])
show(myplot)
myplot.savefig('ch5_20.pdf')
```

And in  $[0, 0.1] \times [0, 0.1]$ :

```
myplot = point(PQHminusTenTen, xmin=0, xmax=0.1, ymin=0, ymax=0.1,
               bgcolor=(1, 0.3, 0), axes_labels=['$x$', '$y$'])
show(myplot)
myplot.savefig('ch5_21.pdf')
```

For the map  $f$ :

```
time HHGGFFminusTenTen = [HH(*GG(*FF(*k))) for k in minusTenTen]
myplot = point(HHGGFFminusTenTen, bgcolor=(0, 0.5, 1),
               axes_labels=['$x$', '$y$'])
show(myplot)
myplot.savefig('ch5_22.pdf')
```

Same graph in  $[0, 1] \times [0, 1]$ :

```
myplot = point(HHGGFFminusTenTen, xmin=0, xmax=1, ymin=0, ymax=1, rgbcolor=(0,0.5,1), axes_labels=['$x$', '$y$'])
show(myplot)
myplot.save('ch5_23.pdf')
```

For the map  $\mathcal{F}$ :

```
time FFFminusTenTen = [FFF(*k) for k in minusTenTen]
myplot = point(FFFminusTenTen, rgbcolor=(0.196078,0.803922,0.196078), axes_labels=['$x$', '$y$'])
show(myplot)
myplot.save('ch5_24.pdf')
```

And the same graph in  $[0, 1] \times [0, 1]$ :

```
myplot = point(FFFminusTenTen, xmin=0, xmax=1, ymin=0, ymax=1,
               rgbcolor=(0.196078,0.803922,0.196078), axes_labels=['$x$', '$y$'])
show(myplot)
myplot.save('ch5_25.pdf')
```

#### C.4.7 Expanded vs. Composed

Define the domain. For instance  $\mathcal{P}_6$ :

```
t_3 = np.linspace(-10,10,2000)
minusTenTen = [(i, j) for i in t_3 for j in t_3]
```

For the first map  $g$ :

```
time P_auxMinusTenTen = [P_aux(*k) for k in minusTenTen]
myplot = point(P_auxMinusTenTen, rgbcolor=(1,0.3,0), axes_labels=['$x$', '$y$'])
show(myplot)
#myplot.save('ch5_26_prime.pdf')
```

For the second one  $f$ :

```
time ffminusTenTen = [ff(*k) for k in minusTenTen]
myplot = point(ffminusTenTen, rgbcolor=(0,0.5,1), axes_labels=['$x$', '$y$'])
show(myplot)
#myplot.save('ch5_26.pdf')
```

#### C.4.8 Using $f_2(\overline{\mathcal{Q}})$ on the third map.

First, define  $\mathcal{P}_7$ :

```
t_4 = np.linspace(0,10,1000)
zeroTen = [(i, j) for i in t_4 for j in t_4]
```

And compute the result:

```
time f_2zeroTen = [f_2(*k) for k in zeroTen]
myplot = point(f_2zeroTen, rbgcolor=(0.196078,0.803922,0.196078)
               ,axes_labels=['$x$', '$y$'])
show(myplot)
myplot.savefig('ch5_27.pdf')
```

### C.4.9 Families of curves composed with $g$ and $f_2$ .

Define new variables for the curves, and the curves for the first map as in 2.1.

```
var('s, a, b')
alpha(s, a) = (s, (1+a*s)/s)
show('alpha_lambda: R^2 -> R^2'); show(alpha)
beta(s, b) = ((1+b*s)/s, s**3)
show('beta_mu: R^2 -> R^2'); show(beta)
```

Define the domain  $\mathcal{P}_8$  of the curves:

```
t_5 = np.linspace(-10,10,1000)
minusTenTen = [(i, j) for i in t_5 for j in t_5]
```

Compute the image of  $\mathcal{P}_8$  through the curves:

```
time alphaminusTenTen = [alpha(*k) for k in minusTenTen]
time betaminusTenTen = [beta(*k) for k in minusTenTen]
```

Plots of the curves  $\alpha_\lambda$  and  $\beta_\mu$  for  $(s, a) \in [-10, 10] \times [-10, 10]$  and 1000 points on each side:

```
myplot = line(alphaminusTenTen, color='purple', axes_labels=['$x$', '$y$']) +\
         line(betaminusTenTen, color='orange', axes_labels=['$x$', '$y$'])
show(myplot, xmin=-15, xmax=15, ymin=-15, ymax=15)
#myplot.savefig('ch5_28.pdf')
```

Compute the images of the curves through  $g$ :

```
time PQHalpha = [P(*Q(*H(*k))) for k in alphaminusTenTen]
time PQHbeta = [P(*Q(*H(*k))) for k in betaminusTenTen]
```

Plots of  $(P \circ Q \circ H)(\alpha_\lambda)$  and  $(P \circ Q \circ H)(\beta_\mu)$ .

```
myplot = line(PQHalpha, color='purple', axes_labels=['$x$', '$y$']) +\
```

```
line(PQHbeta, color='orange', axes_labels=['$x$', '$y$'])
show(myplot, xmin=0, xmax=1, ymin=0, ymax=1, aspect_ratio=1)
#myplot.save('ch5_29.pdf')
```

Same in  $[0, 100] \times [0, 100]$ :

```
myplot = line(PQHalpha, color='purple', axes_labels=['$x$', '$y$'])
+\
line(PQHbeta, color='orange', axes_labels=['$x$', '$y$'])
show(myplot, xmin=0, xmax=100, ymin=0, ymax=100, aspect_ratio=1)
```

Same in  $[0, 0.1] \times [0, 0.1]$ :

```
myplot = line(PQHalpha, color='purple', axes_labels=['$x$', '$y$'])
+\
line(PQHbeta, color='orange', axes_labels=['$x$', '$y$'])
show(myplot, xmin=0, xmax=0.1, ymin=0, ymax=0.1, aspect_ratio=1)
```

Now, we define the family of curves for the third map:

```
gamma(s, a) = ((1+a*s)/s, s**2)
show('gamma: R^2 -> R^2'); show(gamma)
delta(s, b) = (s, (1+b*s)/s)
show('delta: R^2 -> R^2'); show(delta)
```

The domain  $\mathcal{P}_8$ :

```
t_5 = np.linspace(0.001, 10, 1000)
zeroTen = [(i, j) for i in t_5 for j in t_5]
```

The image through the curves:

```
time gammazeroTen = [gamma(*k) for k in zeroTen]
time deltazeroTen = [delta(*k) for k in zeroTen]
```

The image through  $f_2$ :

```
time f_2gamma = [f_2(*k) for k in gammazeroTen]
time f_2delta = [f_2(*k) for k in deltazeroTen]
```

Plots of  $F(\gamma_\lambda)$  and  $F(\delta_\mu)$ :

```
myplot = line(f_2gamma, color='purple', axes_labels=['$x$', '$y$'])
+\
line(f_2delta, color='orange', axes_labels=['$x$', '$y$'])
show(myplot, xmin=0, xmax=100, ymin=0, ymax=100, aspect_ratio=1)
#myplot.save('ch5_31.pdf')
```

Same in  $[0, 4] \times [0, 4]$ :

```
myplot = line(f_2gamma, color='purple', axes_labels=['$x$', '$y$'])
+\
line(f_2delta, color='orange', axes_labels=['$x$', '$y$'])
show(myplot, xmin=0, xmax=4, ymin=0, ymax=4, aspect_ratio=1)
```

In  $[0, 1] \times [0, 1]$ :

```
myplot = line(f_2gamma, color='purple', axes_labels=['$x$', '$y$'])
          +\
line(f_2delta,color='orange',axes_labels=['$x$', '$y$'])
show(myplot, xmin=0, xmax=1, ymin=0, ymax=1, aspect_ratio=1)
```

And in  $[0, 0.1] \times [0, 0.1]$ :

```
myplot = line(f_2gamma, color='purple', axes_labels=['$x$', '$y$'])
          +\
line(f_2delta,color='orange',axes_labels=['$x$', '$y$'])
show(myplot, xmin=0, xmax=0.1, ymin=0, ymax=0.1, aspect_ratio=1)
```

#### C.4.10 Randomly distributed points contained in a disc

Generate the random points in the unit disc:

```
circle_cloud = []
circle_list = []
for i in range(0,100000):
    rho = RealField().random_element(0,1)
    theta = RealField().random_element(0,2*pi)
    circle_cloud = circle_cloud + [point((rho*cos(theta),
                                           rho*sin(theta)))]
    circle_list = circle_list + [(rho*cos(theta),
                                   rho*sin(theta))]
```

Plot it:

```
plot(circle_cloud, aspect_ratio=1)
```

Images through  $g$ :

```
time PQHcircle_list = [P(*Q(*H(*k))) for k in circle_list]
myplot = point(PQHcircle_list, rgbcolor=(1,0.3,0), axes_labels=['$x$', '$y$'])
show(myplot)
```

Looking closer:

```
show(myplot, xmin=0, xmax=10, ymin=0, ymax=10)
```

Images through  $f$ :

```
time HHGGFFcircle_list = [HH(*GG(*FF(*k))) for k in circle_list]
myplot = point(HHGGFFcircle_list, rgbcolor=(0,0.5,1),
               axes_labels=['$x$', '$y$'])
show(myplot)
```

Looking closer:

```
show(myplot, xmin=0, xmax=5, ymin=0, ymax=5)
```

Images through  $\mathcal{F}$ :

```
time FFFcircle_list = [FFF(*k) for k in circle_list]
myplot = point(FFFcircle_list, rgbcolor=(0.196078, 0.803922, 0.196078), axes_labels=['$x$', '$y$'])
show(myplot)
```

A closer look:

```
show(myplot, xmin=0, xmax=2, ymin=0, ymax=2)
```

Do the exact same thing for a disc with radius equals to 100:

```
circle_cloud = []
circle_list = []
for i in range(0,100000):
    rho = RealField().random_element(0,100)
    theta = RealField().random_element(0,2*pi)
    circle_cloud = circle_cloud + [point((rho*cos(theta),
        rho*sin(theta)))]
    circle_list = circle_list + [(rho*cos(theta),
        rho*sin(theta))]
```

Image through  $g$ :

```
time PQHcircle_list = [P(*Q(*H(*k))) for k in circle_list]
myplot = point(PQHcircle_list, rgbcolor=(1,0.3,0), axes_labels=['$x$', '$y$'])
show(myplot)
```

Looking closer:

```
show(myplot, xmin=0, xmax=10, ymin=0, ymax=10)
```

Images through  $f$ :

```
time HHGGFFcircle_list = [HH(*GG(*FF(*k))) for k in circle_list]
myplot = point(HHGGFFcircle_list, rgbcolor=(0,0.5,1),
    axes_labels=['$x$', '$y$'])
show(myplot)
```

Looking closer:

```
show(myplot, xmin=0, xmax=5, ymin=0, ymax=5)
```

Images through  $\mathcal{F}$ :

```
time FFFcircle_list = [FFF(*k) for k in circle_list]
myplot = point(FFFcircle_list, rgbcolor=(0.196078, 0.803922, 0.196078), axes_labels=['$x$', '$y$'])
show(myplot)
```

Looking closer:

```
show(myplot, xmin=0, xmax=10, ymin=0, ymax=10)
```

### C.4.11 Families of curves composed with the new maps: $\mathcal{N}_1$ and $\mathcal{N}_2$

We begin by defining the first new polynomial map  $\mathcal{N}_1$ :

```
newMap_1(x, y)=(x**4*y**4 + (x**2*y + x*y**2 - 1)**2*(y**2 + 1),
                  x**4*y**4 + (x**2*y + x*y**2 - 1)**2*(x**2 + 1))
show('newMap_1: R^2 -> R^2'); show(newMap_1)
```

And the new family of curves:

```
sigma(s, a) = (s**2 + a*s**3, 1/s)
show('sigma: R^2 -> R^2'); show(sigma)
tau(s, b) = (1/s, s**2 + b*s**3)
show('tau: R^2 -> R^2'); show(tau)
```

The domain:

```
t_5 = np.linspace(-10, 10, 1000)
minusTenTen = [(i, j) for i in t_5 for j in t_5]
```

The image through the curves:

```
time sigmaminusTenTen = [sigma(*k) for k in minusTenTen]
time tauminusTenTen = [tau(*k) for k in minusTenTen]
```

Plots of the curves  $\sigma_\lambda$  and  $\tau_\mu$  for  $(s, a) \in [-10, 10] \times [-10, 10]$  and 1000 points on each side:

```
myplot = line(sigmaminusTenTen, rgbcolor=(0, 0.8, 0.2),
               axes_labels=['$x$', '$y$']) +
    line(tauminusTenTen, rgbcolor=(0, 0.4, 1), axes_labels=['$x$', '$y$'])
show(myplot, xmin=-15, xmax=15, ymin=-15, ymax=15)
#myplot.save('ch5_32.pdf')
```

Image through  $\mathcal{N}_1$ :

```
time newMap_1sigma = [newMap_1(*k) for k in sigmaminusTenTen]
time newMap_1tau = [newMap_1(*k) for k in tauminusTenTen]
```

Plots of  $\mathcal{N}_1(\sigma_\lambda)$  and  $\mathcal{N}_1(\tau_\mu)$ :

```
myplot = line(newMap_1sigma, rgbcolor=(0, 0.8, 0.2), axes_labels=[
    '$x$', '$y$']) +
    line(newMap_1tau, rgbcolor=(0, 0.4, 1), axes_labels=['$x$', '$y$'])
show(myplot, xmin=0, xmax=0.1, ymin=0, ymax=0.1, aspect_ratio=1)
#myplot.save('ch5_33.pdf')
```

As for  $\mathcal{N}_2$ :

```
newMap_2(x,y)=(x**2*y**2 + (x**2*y + x*y**2 - 1)**2*(y**2 + 1),
               x**2*y**2 + (x**2*y + x*y**2 - 1)**2*(x**2 + 1))
show('newMap_2: R^2 -> R^2'); show(newMap_2)
```

We use the exact same curves we just did. Compute the image:

```
time newMap_2sigma = [newMap_2(*k) for k in sigmaminusTenTen]
time newMap_2tau = [newMap_2(*k) for k in tauminusTenTen]
```

Plots of  $\mathcal{N}_2(\sigma_\lambda)$  and  $\mathcal{N}_2(\tau_\mu)$ :

```
myplot = line(newMap_2sigma, rgbcolor=(0,0.8,0.2), axes_labels=[ '$x$', '$y$']) +\
         line(newMap_2tau, rgbcolor=(0,0.4,1), axes_labels=[ '$x$', '$y$'])
show(myplot, xmin=0, xmax=0.1, ymin=0, ymax=0.1, aspect_ratio=1)
#myplot.save('ch5_33.pdf')
```

The third map  $\mathcal{N}_3$  is:

```
newMap_3(x,y)=(x**6*y**4 + (x**2*y + x*y**2 - 1)**2*(y**2 + 1),
               x**4*y**6 + (x**2*y + x*y**2 - 1)**2*(x**2 + 1))
show('newMap_3: R^2 -> R^2'); show(newMap_3)
```

#### C.4.12 Expansion and grouping of the given polynomial maps

Computing the total degree and the total number of monomials.

First paper  $g$ :

- ★ Total degree: 56.
- ★ Total number of monomials: 167 (101 + 67).

```
show(expand(F(*Q(*H))).collect(y))
show(expand(G(*Q(*H))).collect(y))
```

Second map  $f$ :

- ★ Total degree: 72.
- ★ Total number of monomials: 350.

```
show(expand(ff[0]).collect(y))
show(expand(ff[1]).collect(y))
```

Third map  $\mathcal{F}$ :

- ★ Total degree: 28.

- \* Total number of monomials: 22.

```
show(expand(FFF[0]).collect(y))
show(expand(FFF[1]).collect(y))
```

First new map  $\mathcal{N}_1$ :

- \* Total degree: 16.
- \* Total number of monomials: 24.

```
show(expand(newMap_1[0]).collect(y))
show(expand(newMap_1[1]).collect(y))
```

Second new map  $\mathcal{N}_2$ :

- \* Total degree: 16.
- \* Total number of monomials: 26.

```
show(expand(newMap_2[0]).collect(y))
show(expand(newMap_2[1]).collect(y))
```

Third new map  $\mathcal{N}_3$ :

- \* Total degree: 20.
- \* Total number of monomials: 26.

```
show(expand(newMap_3[0]).collect(y))
show(expand(newMap_3[1]).collect(y))
```