



HAL
open science

Isolating the Singularities of the Plane Projection of Generic Space Curves and Applications in Robotics

George Krait

► **To cite this version:**

George Krait. Isolating the Singularities of the Plane Projection of Generic Space Curves and Applications in Robotics. Computational Geometry [cs.CG]. Université de Lorraine (Nancy), 2021. English. NNT: . tel-03241794v1

HAL Id: tel-03241794

<https://theses.hal.science/tel-03241794v1>

Submitted on 31 May 2021 (v1), last revised 15 Jun 2021 (v2)

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

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

Isolating the Singularities of the Plane Projection of Generic Space Curves and Applications in Robotics

THÈSE DE DOCTORAT

présentée et soutenue publiquement le 04 mai 2021

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention informatique)

par

George Krait

Composition du jury

Président : Pierrick Gaudry (CNRS, Loria)

Rapporteurs : Nicolas Delanoue (Université d'Angers, ISTIA)
Sonia Pérez-Díaz (University of Alcalá)

Examineurs : Sylvain Lazard -directeur- (INRIA Nancy Grand Est, Loria)
Guillaume Moroz -co-directeur- (INRIA Nancy Grand Est, Loria)
Erika Ottaviano (University of Cassino and Southern Lazio)

Invité : Marc Pouget (INRIA Nancy Grand Est, Loria)

Remerciements

Entreprendre cette thèse a été une riche expérience, qui m'a révélé non seulement les aspects académiques de ma propre personnalité, mais aussi des caractéristiques intimes, des faiblesses et des forces dont je n'avais jamais pris conscience auparavant. Une thèse de doctorat est souvent décrite comme un travail solitaire; cependant, les noms qui suivent démontrent absolument le contraire. De fait, le chemin vers mon but a été éclairé par l'appui de personnes qui n'ont rien épargné pour m'aider, me motiver et partager des connaissances.

Ce travail a été réalisé avec l'orientation de mes superviseurs: Sylvain Lazard, Guillaume Moroz et Marc Pouget. Avec votre inestimable expertise, la recherche et la méthode allèrent de l'avant. Votre apport perspicace m'incita au raffinement de ma pensée et orienta mon travail vers un niveau supérieur. Merci pour votre amitié, votre patience et votre rigueur académique.

Ma gratitude s'adresse aussi aux membres du jury, qui ont pris le temps de lire cette thèse et ont formulé des commentaires importants et des suggestions, qui m'aident pour l'amélioration de cette contribution et la poussent vers l'avant. Outre l'appui académique, j'ai aussi bénéficié de l'assistance amicale de Joel Veldkamp et de Juan Pedro de Gandt sur le plan linguistique. Vos efforts, bien appréciés, ont enrichi et consolidé la formulation de mon texte, soulignant et renforçant la rigueur et la précision du langage utilisé.

Une profonde gratitude va aussi vers ma famille, pour qui la distance n'a pas été un obstacle aux encouragements et aux preuves d'amour. J'ai aussi bénéficié d'un appui familial de la part de Pierre Kreit et de sa famille, auprès de qui j'ai trouvé de sages conseils et une oreille sympathisante, étant pour moi, en France, comme un second père, en m'offrant, avec Madame Christine Kreit, ce qui me fit sentir chez moi.

Le dernier remerciement s'adresse au lecteur de ces pages, quel qu'il soit. Tout commentaire, toute suggestion ou critique seront aussi des contributions pour ce travail.

This thesis is dedicated to the big family I miss in Syria and to the small one I am making in France.

Résumé

L'isolation des points singuliers d'une courbe plane est la première étape vers le calcul de sa topologie. Pour cela, les méthodes numériques sont efficaces mais non certifiées en général. Nous sommes intéressés par le développement d'algorithmes numériques certifiés pour isoler les singularités. Pour ce faire, nous limitons notre attention au cas particulier des courbes planes qui sont des projections de courbes lisses en dimensions supérieures. Ce type de courbes apparaît naturellement dans les applications robotiques et la visualisation scientifique. Dans ce cadre, nous montrons que les singularités peuvent être encodées par un système carré et régulier dont les solutions peuvent être isolées avec des méthodes numériques certifiées. Notre analyse est conditionnée par des hypothèses que nous démontrons comme étant génériques en utilisant la théorie de la transversalité; nous fournissons également un semi-algorithme pour vérifier leur validité. Enfin, nous présentons des expériences de visualisation et de robotique, dont certaines ne sont pas accessibles par d'autres méthodes, et discutons de l'efficacité de notre méthode.

Mots-clés: Transversalité, Singularités Génériques, Algorithmes Numériques Certifiés, Arithmétique d'Intervalles, Topologie de Courbe Singulière

Abstract

Isolating the singularities of a plane curve is the first step towards computing its topology. For this, numerical methods are efficient but not certified in general. We are interested in developing certified numerical algorithms for isolating the singularities. In order to do so, we restrict our attention to the special case of plane curves that are projections of smooth curves in higher dimensions. This type of curves appears naturally in robotics applications and scientific visualization. In this setting, we show that the singularities can be encoded by a regular square system whose solutions can be isolated with certified numerical methods. Our analysis is conditioned by assumptions that we prove to be generic using transversality theory. We also provide a semi-algorithm to check their validity. Finally, we present experiments in visualization and robotics, some of which are not reachable by other methods, and discuss the efficiency of our method.

Keywords: Transversality, Generic Singularities, Certified Numerical Algorithms, Interval Arithmetic, Singular Curve Topology

Contents

List of Tables **xi**

List of Figures **xiii**

| | |
|---|----------|
| Chapter 1 | |
| Introduction | 1 |
| 1.1 Motivation | 2 |
| 1.2 Problem statement | 5 |
| 1.2.1 Preserved properties when approximating plane curves | 5 |
| 1.2.2 Isolating singularities | 7 |
| 1.2.3 Formal statement of the problem | 8 |
| 1.3 Related work | 10 |
| 1.3.1 Curves and singularities | 11 |
| 1.3.2 Zero-dimensional systems | 12 |
| 1.3.3 Singularities of implicit plane curves | 16 |
| 1.3.4 Singularities of the plane projections of implicit space curves | 16 |
| 1.4 Contribution | 18 |
| 1.4.1 Generic assumptions | 19 |
| 1.4.2 Enclosing singularities by the Ball system | 19 |
| 1.4.3 Algorithmic contribution | 20 |
| 1.4.4 Implementation, experiments and applications | 21 |

Part I Singularity modelling **23**

| | |
|--|----|
| Chapter 2 | |
| Preliminaries in geometry and algebra | |
| 2.1 Real algebraic geometry | 28 |
| 2.1.1 The ring of real polynomials | 28 |

| | | |
|-------|---|----|
| 2.1.2 | Determinantal varieties | 29 |
| 2.2 | Differential geometry | 29 |
| 2.2.1 | Smooth functions and manifolds | 29 |
| 2.2.2 | Germ of smooth functions | 31 |
| 2.2.3 | The multiplicity of smooth systems of equations | 32 |
| 2.2.4 | Transversality theorems | 33 |

| |
|--|
| <p>Chapter 3 Generic space curve</p> |
|--|

| | | |
|-------|---|----|
| 3.1 | Notation and assumptions | 38 |
| 3.1.1 | Notation | 38 |
| 3.1.2 | Assumptions | 38 |
| 3.2 | Genericity of the assumptions | 39 |
| 3.2.1 | Genericity of the assumptions for a curve in \mathbb{R}^n | 41 |
| 3.2.2 | Genericity of the assumptions for the silhouette of a surface in \mathbb{R}^n | 49 |

| |
|---|
| <p>Chapter 4 Modelling system</p> |
|---|

| | | |
|-----|--|----|
| 4.1 | Encoding the singular points of the plane projection | 57 |
| 4.2 | Singularities induced by \mathcal{L}_n | 64 |
| 4.3 | Singularities induced by \mathcal{L}_c | 64 |
| 4.4 | Regularity of the Ball system | 73 |

| |
|---|
| <p>Chapter 5 Semi-algorithms to check the assumptions and isolate singularities</p> |
|---|

| | | |
|-------|--|----|
| 5.1 | Interval arithmetic | 84 |
| 5.2 | Semi-algorithms to check assumptions and isolate singularities | 85 |
| 5.3 | Semi-algorithms improvements | 92 |
| 5.3.1 | Subdivision-based solver on a high-dimensional system | 92 |
| 5.3.2 | Evaluating the operator D | 98 |
| 5.3.3 | Improvement integration | 99 |

Part II Software, experiments and application in robotics 101

| |
|---|
| <p>Chapter 6 Implementation</p> |
|---|

| | | |
|-----|--|-----|
| 6.1 | Third-party libraries | 106 |
| 6.2 | The Python software <i>Isolating_singularities</i> | 107 |

Chapter 7

Experiments

7.1 Experiment 1: Analytic curve in \mathbb{R}^3 generating one node and one ordinary cusp 112

7.2 Experiment 2: Analytic curve in \mathbb{R}^4 with many nodes 113

7.3 Experiment 3: High degree algebraic curve in \mathbb{R}^4 114

7.4 Experiment 4: Two close lines in \mathbb{R}^3 generating a node 115

Chapter 8

Applications in robotic mechanisms

8.1 Preliminaries in robotics 118

8.2 Description of *RRRRR* robot 122

8.3 Results 123

8.3.1 The parallel singularity $S_{parallel}$ 123

8.3.2 The serial singularity S_{serial} 128

Conclusion

133

Chapter 9

Conclusion

Bibliography

List of Tables

| | | |
|-----|---|-----|
| 7.1 | Timings (in seconds) and numbers of boxes in Experiments 1 to 3. | 112 |
| 7.2 | Experiment 4: Performances for different values of ϵ . | 116 |
| 8.1 | The list of certified nodes enclosure that <i>enclosing_singularities</i> returns, where every row represents an enclosing box of a solution of $\text{Ball}(P_1)$. Notice that <i>enclosing_singularities</i> returns exactly one solution of every pair of twin solutions. | 127 |
| 8.2 | The list of certified nodes enclosure that <i>enclosing_singularities</i> returns, where every row represents an enclosing box of a solution of $\text{Ball}(P_2)$. Notice that <i>enclosing_singularities</i> returns exactly one solution of every pair of twin solutions. | 130 |

List of Figures

| | | |
|-----|--|----|
| 1.1 | Left: Illustration of a configuration space of a 2-degrees-of-freedom robot. In this example, it is a surface embedded in \mathbb{R}^3 . Generically, the silhouette of this surface is a (smooth) curve. Right: Projection of the silhouette on the motor space. | 3 |
| 1.2 | An approximation of the curve in Figure 1.1 (right) that preserves the topology and the locations of the critical points. The approach used for this goal is to isolate the critical points, and then, to connect them properly using the state-of-the-art methods since the remaining part is smooth. Additional black vertices are added to improve the visualization. The number on each face represents the number of solutions of the direct kinematic problem. | 4 |
| 1.3 | Approximating the given curve preserving its topology, but losing relative positions of the connected components. | 6 |
| 1.4 | An approximation of the plane curve that is ambient-isotopic to the given curve. | 7 |
| 1.5 | An approximation of the plane curve where the vertices are boxes isolating singular or smooth critical points connected by segments approximating the curve (in blue). | 8 |
| 1.6 | The curve defined by $y^2 - (x^3 + x^2) = 0$ | 9 |
| 1.7 | Illustration of a node and cusps in the plane projection of a smooth curve. | 21 |
| 1.8 | Left: An example of a box in \mathbb{R}^n that satisfies (a) and whose projection contains a node. Right: A simplified illustration of pruning the domain of the Ball system. The brown and orange boxes stratify (a), also the pair of red boxes satisfy (b). Using these boxes, the blue boxes in \mathbb{R}^{2n-1} are computed. | 22 |
| 2.1 | An example in \mathbb{R}^2 where f and g define the same germ. | 31 |

| | | |
|-----|---|----|
| 2.2 | Left: At an A_1 singularity, two branches of the curve intersect transversally. Right: At an A_{2k+1} singularity with $k > 1$, the tangent lines of the two branches at the intersection point coincide. | 32 |
| 3.1 | Illustration of the assumptions. | 40 |
| 3.2 | Illustration of the proof of Lemma 3.2.6. | 45 |
| 3.3 | Proof of Lemma 3.2.7: The first case where v and v' are independent. | 46 |
| 3.4 | Proof of Lemma 4.3.3: The second case where v and v' are co-linear. | 47 |
| 3.5 | Example: The silhouette of the torus in \mathbb{R}^3 with the coordinates (x, y, z) (source: [IMP2016a]) | 50 |
| 3.6 | Illustration of Lemma 2.2.17 which is used in the proof of Proposition 3.2.12. Since φ is a submersion over $Reg(W)$ and since $j^1\tilde{P}$ is, generically, transversal to $Reg(W)$, the lemma implies that the composition $\tilde{P} = \varphi \circ j^1\tilde{P}$ is generically a submersion. | 52 |
| 4.1 | Illustration of a node and cusps in the plane projection of a smooth curve. | 58 |
| 4.2 | The curve \mathcal{C} (red) and its plane projection $\pi_{\mathcal{C}}(\mathcal{C})$ (blue) of Example 4.1.11 displaying a cusp singularity. | 63 |
| 4.3 | The curve \mathcal{C} (red) and its plane projection $\pi_{\mathcal{C}}(\mathcal{C})$ (blue) of Example 4.1.12 displaying a node singularity. | 65 |
| 4.4 | An example of Lemma 4.3.1 for $n = 3$ | 68 |
| 4.5 | Comparing the singularities considered in Lemma 4.3.4 with A_{2k+1} types. | 72 |
| 4.6 | Comparing the curve \mathcal{C} defined in Remark 4.4.5 with \mathcal{C}' that is defined by $x_1 - (x_3 - 1)^2 = x_2 - (x_3 - 1)^3 = 0$ where both of them have the same plane projection which contains an ordinary cusp; however, the multiplicities of S (defined in the same remark) at q differ. In this example, the pair (q, q) satisfies Assumption $\mathcal{A}_5^{-'}$ for \mathcal{C}' , however, it does not for \mathcal{C} | 78 |
| 5.1 | An example of Case (ii) for a curve \mathcal{C} in \mathbb{R}^n . The box \mathfrak{B} contains a smooth part of \mathcal{C} that projects to a loop with a node. This implies that two points q_1 and q_2 are in \mathfrak{B} such that $\det(M_1(q_1)) = 0$ and $\det(M_2(q_2)) = 0$. Hence, the tangent space plane projection of q_1 (resp. q_2) is parallel to x_2 -axis (resp. x_1 -axis). | 94 |

| | | |
|-----|---|-----|
| 5.2 | An example of \mathfrak{B} and \mathfrak{B}' satisfying (2) in Lemma 5.3.1. Their projections have a node in the boundary. The curve \mathcal{C} is tangent to the boundaries of both \mathfrak{B} and \mathfrak{B}' at q and q' respectively. The curve and the boxes intersect at q'' such that q'' does not have the same plane projection as q and q' | 95 |
| 5.3 | An example of \mathfrak{B} and \mathfrak{B}' satisfying (2) in Lemma 5.3.1. The projection has a loop. This implies that there are a x_1 -critical point and a x_2 -critical point in the union of \mathfrak{B} and \mathfrak{B}' . In this example, these points are different and the projection has a node. | 95 |
| 5.4 | A simplified illustration of pruning $\overline{B}_{\text{Ball}}$ as described in Proposition 5.3.3. The red boxes satisfy (1). On the other hand, the brown (resp. orange) box satisfies (2). Hence, we solve the Ball system over f_{Ball} pre-images (in blue) of the red boxes, the brown box and the orange box. | 97 |
| 7.1 | Experiment 1: Plane projection of an analytic curve in \mathbb{R}^3 with one node and one ordinary cusp. | 113 |
| 7.2 | Experiment 2: Plane projection of an analytical curve \mathcal{C} in \mathbb{R}^4 . Each of the 43 black boxes contains a node of $\pi_{\mathcal{C}}(\mathcal{C})$ and is the projection of a box in \mathbb{R}^7 containing one solution of $\text{Ball}(P)$ | 114 |
| 7.3 | Experiment 3: High degree algebraic curve in \mathbb{R}^3 generating 7 nodes. | 115 |
| 8.1 | A non-singular configuration of \underline{PRRP} where there is a local bijection between poses and commands. | 120 |
| 8.2 | Geometric modelling of \underline{PRRP} motion. | 120 |
| 8.3 | The parallel singular poses of \underline{PRRP} where the pose x admits two commands. | 121 |
| 8.4 | The serial singular poses of \underline{PRRP} where the command q admits two poses. | 121 |
| 8.5 | \underline{RRRRR} robot in a non-singular configuration. | 122 |
| 8.6 | Illustration of the projections of S_{serial} on different variable subspaces. | 124 |
| 8.7 | Illustration of the projections of S_{parallel} on different variable subspaces. | 125 |
| 8.8 | Enclosing the plane projection of S_{parallel} with respect to x_1 and x_2 | 126 |
| 8.9 | The plane plotting of $\text{Isolating_singularities}$ output for S_{parallel} with respect to q_1 and q_2 | 128 |

| | | |
|------|--|-----|
| 8.10 | Plotting the result of applying <i>Isolating_singularities</i> to S_{serial} with respect to x_1, x_2 , where it plots only red boxes since neither Assumption \mathcal{A}_2 nor Assumption \mathcal{A}_1 is satisfied. | 129 |
| 8.11 | An example of two different (serial) singular configurations of <u>RRRRR</u> that define the same pose (x_1, x_2) | 129 |
| 8.12 | An illustration of <i>enclosing_singularities</i> output for the serial singularity S_{serial} of the robot <u>RRRRR</u> . The projection is with respect to the (q_1, q_1) -plane (the command variables). The certification of the geometric singularities is done only where our strong assumptions are satisfied. The red boxes contain the projections of the (geometric) singular points of S_{serial} | 131 |
| 9.1 | An example where our visualization tool does not provide a correct topology for some precision. A: Our software returns a set of boxes that encloses the projected curve without reporting any node (or unknown-status boxes). B & C: Two different scenarios of the plot in A after enough refinements. | 138 |

1

Introduction

Computational geometry is the branch of computer science that studies theoretical and applied aspects of geometric algorithms. A classical problem in computational geometry is to visualize curves, in particular, plane curves. The importance of curve visualization comes from the fact that curves are very common in engineering, physics and mathematical problems. In general, visualizing a geometric object consists in representing it by an implementable structure which can be handled by computers. In particular, (plane) curve visualization is usually based on approximating a given curve into a piecewise-linear graph that preserves some properties of the curve. A question that might come to mind is: What are the properties that an approximation is supposed to preserve? The classical properties that are considered are that the approximation should be “close enough” to the original curve and that the original curve and its approximation should have the same topologies, in particular, connected components and self-intersections should not be missed.

Depending on the applications, preserving the topology and some other geometric features may be critical or of interest while in some other applications, this may not be relevant. In this regard, one can find in the literature two main types of approaches for approximating curves:

- Non-certified approaches in the sense that the topology of the approximation does not necessarily coincide with the one of the original curve. In particular, connected components and self-intersection points might be missed or spuriously added. Many numerical non-certified approaches exist in the literature such as marching cubes and continuation methods; see Section [1.3](#).
- Certified approaches in the sense that they ensure that the topologies of the curve and its approximation are the same. Few certified approaches exist in the literature. Most of these approaches are algebraic, although some numer-

ical approaches exist if the input curves satisfy some assumptions. In Section 1.3, we provide more details about these approaches. In particular, we focus on a critical sub-problem that appears in all known approaches of this type, namely, isolating the singularities of plane curves.

In this thesis, we address the problem of certified plane curve approximation. Numerical approaches are usually efficient for approximating curves. However, when the considered curves contain singularities, numerical methods do not, in general, provide certified approximations. Our goal is to design a fast numerical method that is certified for approximating and visualizing plane curves. With this goal in mind, we restrict ourselves to a special case of $2D$ curves, namely, those that are the projections of smooth curves living in \mathbb{R}^n , with $n > 2$. This problem of approximating and visualizing 2D projections of smooth curves in higher dimensions is rather specific and we start by presenting the application in robotics that motivates our work.

1.1 Motivation

Before introducing the problem in details, we describe the context that motivates our study. The motivation we present revolves around robotics design and analysis. For this goal, we recall some preliminaries [LP2017]:

An end-effector is the end robotic part that attaches to robotic arms or appendages. The configuration space is the space of possible positions an end-effector may attain. The degrees of freedom of a robot refer to the number of movable joints or links of this robot. Equivalently, the degrees of freedom refer to the dimension of the configuration space. The motor (or command) space is the set of positions that the engines of a robot may attain. The direct kinematic problem refers to determining the position of the end-effector, given the values for the motor variables of a robot. In particular, for a robot with 2 degrees of freedom, the configuration space can be seen as a surface embedded in an n -dimensional space, where $n \in \mathbb{N}$ represents the number of variables that define and control the robot. Moreover, the engines' possible positions can be represented in \mathbb{R}^2 (as a sub-space of \mathbb{R}^n).

Figure 1.1 (left) shows how a 2-degrees-of-freedom robot configuration space looks like when $n = 3$. The first two coordinates x and y represent the engine variables. The third one z represents, for example, the position of an end-effector. This figure is for illustrative purposes and is not meant to refer to any robot. Every point of this surface represents a solution of the direct kinematic problem, given its projection to the (x, y) -plane. The silhouette is the set of points in the surface at which

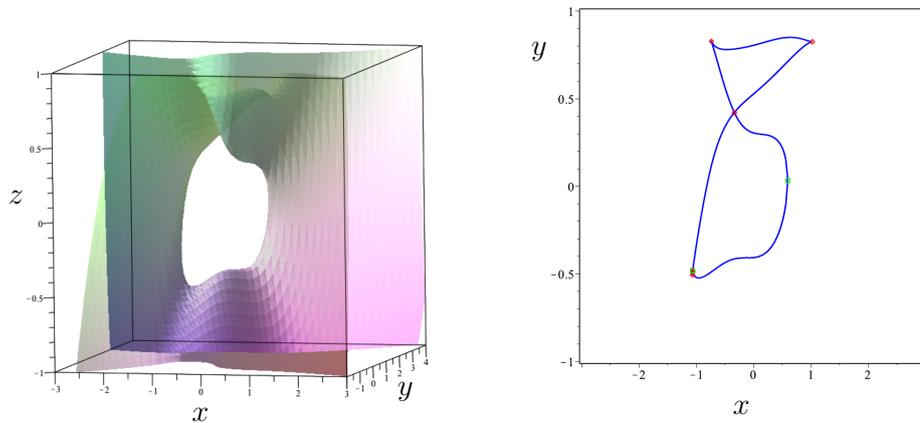


Figure 1.1: Left: Illustration of a configuration space of a 2-degrees-of-freedom robot. In this example, it is a surface embedded in \mathbb{R}^3 . Generically, the silhouette of this surface is a (smooth) curve. Right: Projection of the silhouette on the motor space.

the tangent plane is vertical. Its projection to (x, y) -plane contains the boundary of the engine possible motions. In the neighbourhood of a point on the silhouette, the number of solutions of the direct kinematic problem is likely to change. Figure 1.1 (right) shows the projection of the configuration space's silhouette on the motor space.

Moving the robot from a position to another amounts to convey the engines accordingly. Thus, a path in the motor space is drawn for this motion. If such a path traverses the projection of the silhouette, bad scenarios might occur. For example, suppose that the approximation misses topological information about the silhouette projection such as the part numbered with zero in Figure 1.2 (right). This part surrounds points where the direct kinematic problem has no solution. In this case, paths traversing this part might cause robot damage. One of the approaches used for this problem is based on approximating the silhouette projection into a piecewise-linear graph. This approximation is supposed to preserve several properties, more precisely, the topology and geometric features such as the location of critical points and their types. By critical points we mean those where the tangent line is not well-defined (in this case, the points are called singular) or the tangent line is parallel to one of the axes (smooth critical points). Preserving these properties allows to understand how the silhouette plane projection divides the motor space topologically. This allows to classify the motor space by the number of solutions of the direct kinematic problem. With the help of this information, bad scenarios as the one

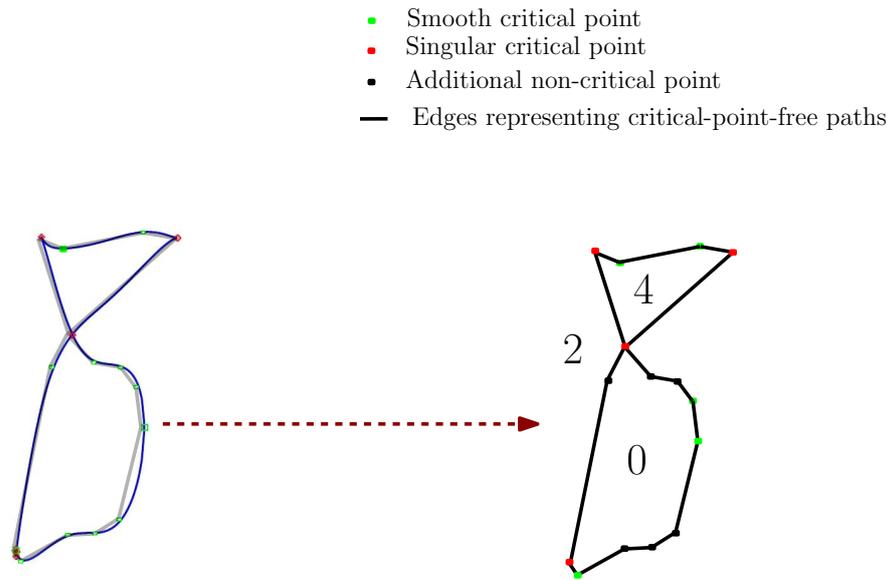


Figure 1.2: An approximation of the curve in Figure 1.1 (right) that preserves the topology and the locations of the critical points. The approach used for this goal is to isolate the critical points, and then, to connect them properly using the state-of-the-art methods since the remaining part is smooth. Additional black vertices are added to improve the visualization. The number on each face represents the number of solutions of the direct kinematic problem.

mentioned above can be avoided. This type of approximation is also useful when computing paths in the command space for preserving a safe distance between the paths and the boundary of the silhouette. In other words, a topologically-correct approximation of the silhouette projection leads to approximately computing simply connected subsets of the motor space. In every component, the robot is safely movable from and to every two points in that component. More information about this approach is described in Chapter 8.

Recall that, as mentioned in the introduction, if the projections of the silhouette curves contain singularities, numerical methods such as [Sny1992, LOD2001, PV2004, MGGJ2013] fail to ensure a correct topology in general. On the other hand, in the case of algebraic curves, symbolic methods can be used for this goal (see Section 1.3 for more details) but, in general, they are slower compared with numerical methods.

However, a specificity of the problem described above is that, although the sil-

houette projections generically have singularities, the silhouette curves before the projection are in general smooth. This is why we aim at a certified numerical method for approximating the plane projections of smooth curves in \mathbb{R}^n .

Although the above example with a configuration space in \mathbb{R}^3 is a toy example, other 2-degrees-of freedom robots appear in real-life for different purposes such as plotting, pick-and-place and other industrial tasks. Such robots usually have two motor variables but they tend to have more than one variable for representing the position of an end-effector. Hence, their configuration spaces are surfaces embedded in \mathbb{R}^n , with n is larger than 3. This explains why we are interested in approximating the projections of silhouette curves in \mathbb{R}^n and not only in \mathbb{R}^3 .

Even with robots of k degrees of freedom, with $k > 2$, our approach might be helpful by fixing $k - 2$ motors. Then, the motor space is embedded in \mathbb{R}^2 and the configuration space is restricted to a surface. We can thus use our approach for approximating the projection of its silhouette. For path planning, this could be used to design safe paths with sequences of motions that only move 2 motors simultaneously.

1.2 Problem statement

This section is dedicated to state the problem we consider more formally. We start with explaining the properties that we want to preserve when approximating a possibly-singular plane curve. Then, we describe a natural approach to approximate the targeted type of (plane) curves. Then, we focus on the main sub-problem of this approach, namely, to isolate the singular points of that curve.

1.2.1 Preserved properties when approximating plane curves

Recall that the set of critical points consists of singularities and smooth (non-singular) points with tangent lines parallel to one of the axis. The curve approximation that we wish is a plane graph where every critical point of the curve is identified by a vertex in that graph. In addition, we previously discussed the properties that we want to certify in order to tackle the robotics problem in Section 1.1.

In the following, we state these properties more formally and justify their significance. To avoid technicality, we describe our purposes by examples.

Topology. We first consider “simple” properties such as the number of connected components, intersection points, etc. More formally, we wish to find a plane graph that is homeomorphic to the given curve. Figure 1.3 (left) shows a plane curve to be approximated. An approximation of the latter curve that fulfills these conditions

is illustrated in the right-hand side. However, it is clear that this approximation does not look similar to the curve since the relative positions of the connected components are not preserved. This leads us to another interesting property which is beyond topology.

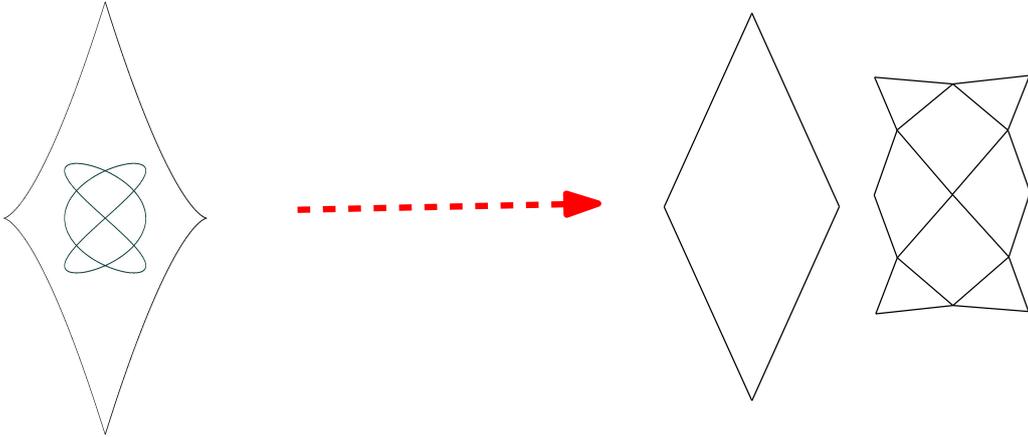


Figure 1.3: Approximating the given curve preserving its topology, but losing relative positions of the connected components.

To improve the approximation, we require an additional property. This property is about the relative position of the connected components. More formally, we want that the curve and the graph are ambient-isotopic, that is, the curve can be continuously deformed towards the graph such that stopping the deformation at any point gives a curve that has the same topology as the graph.

Geometric properties. By geometric properties we mean the location of critical points. More precisely, for every critical point in the curve and any positive real ϵ , we want to find a box in \mathbb{R}^2 , with a width at most ϵ that contains this point and does not contain any other critical point. This box is called an isolating box. In addition, we also want to determine the type of this point, e.g., smooth critical point, node, cusp, etc. Figure 1.5 shows an approximation of the plane curve mentioned above preserving its geometric properties. Having the critical points isolated, the approximation can be easily improved by adding vertices on critical-point-free parts of the curve. Consequentially, we can control the Hausdorff distance between the curve and its approximation and make it as small as we wish.

From the above discussion, the approach we consider for approximating a plane curve consists in computing an isolating box for every critical point, as well as additional boxes added in the case where there are different edges that have the same endpoints. Then, we identify these boxes as vertices in the approximating

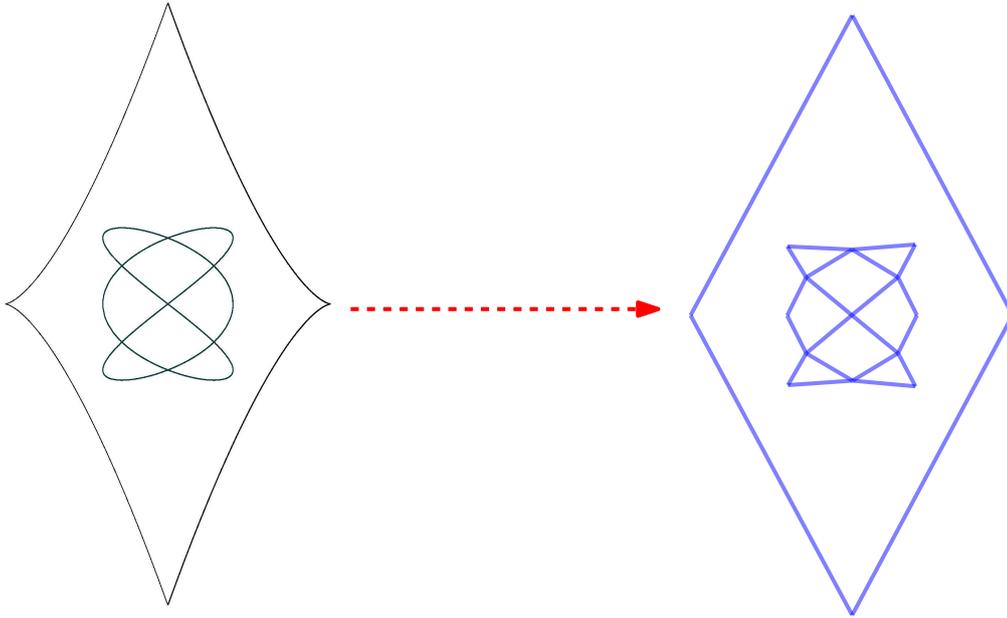


Figure 1.4: An approximation of the plane curve that is ambient-isotopic to the given curve.

plane graph. This approach allows us to preserve the topological and geometric properties mentioned in the previous paragraph.

Thus, the first step (and the main one) in this approach is to compute isolating boxes for critical points. In the next section we discuss the difficulties of this step.

1.2.2 Isolating singularities

Recall that for implicit plane curves, the critical points are described by systems of equations. In order to detect them efficiently, one may wish to use certified numerical methods such as Newton interval methods [Neu1991]. Recall that these methods require that the system is (a) square, that is it has as many variables as equations, and (b) regular, that is, the system has no multiple solutions; equivalently, the Jacobian matrix at every solution is regular. Generically, critical points can be described by square regular systems in smooth curves. On the other hand, the usual system that encodes singular points (where tangent lines are not well-defined), which is obtained by the Jacobian criteria, does not satisfy these conditions in general.

For example, take the curve defined by $f(x, y) = y^2 - (x^3 + x^2) = 0$ (Figure 1.6). The singularities are encoded by the system $f(x, y) = \frac{\partial f}{\partial x}(x, y) = \frac{\partial f}{\partial y}(x, y) = 0$,

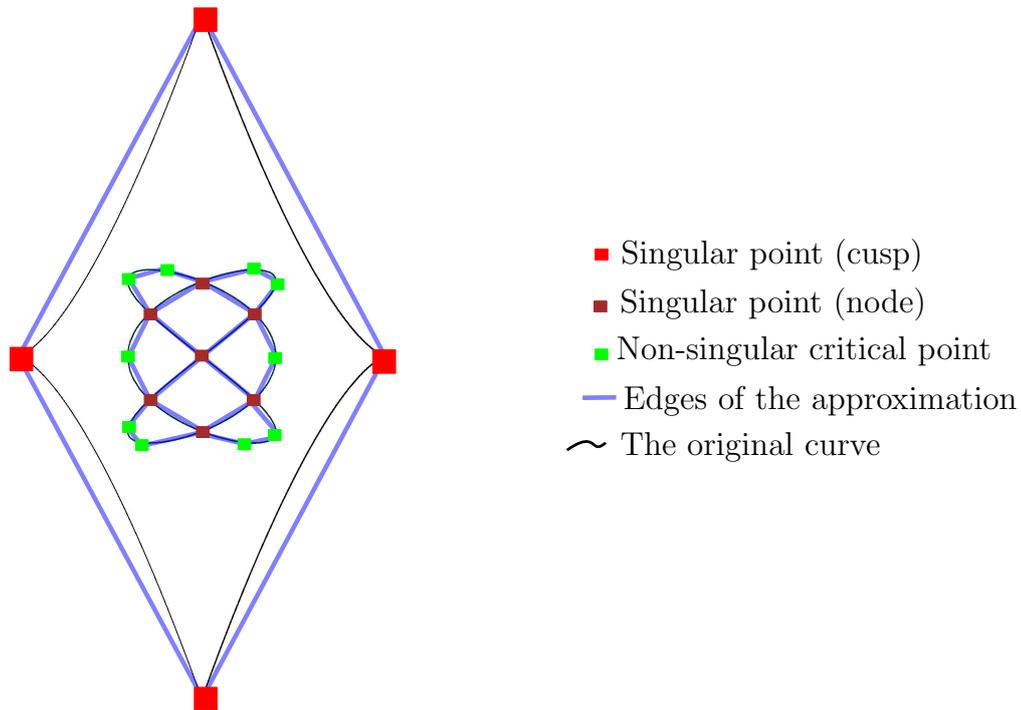


Figure 1.5: An approximation of the plane curve where the vertices are boxes isolating singular or smooth critical points connected by segments approximating the curve (in blue).

that is $y^2 - (x^3 + x^2) = -(3x^2 + 2x) = 2y = 0$ which is not square. Hence, we cannot apply Newton interval methods to certify the correctness of the solutions.

For a system of polynomial equations, symbolic methods are available to certify all solutions of any zero-dimensional system (in particular, to certify curve singularities). However, they show, in general, a slower performance compared with numerical methods; see Section 1.3.2.

From the above discussion, we deduce that the problem of isolating the singularities of a plane curve is itself a challenge towards approximating it. In fact, the main focus of this thesis is to numerically isolate singular points of a special type of plane curves using certified efficient methods.

1.2.3 Formal statement of the problem

In this section, we state the problem more formally and precisely. Before that, we want to determine what we mean by concepts such as “input”, “curve” and “solution”. In other words, we first mention several related considerations that

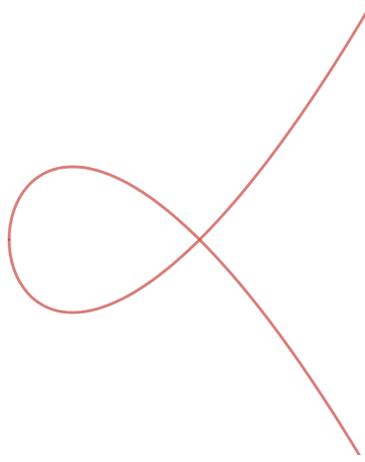


Figure 1.6: The curve defined by $y^2 - (x^3 + x^2) = 0$.

play an important role in solving this type of problems.

The first factor to consider is the input of the problem. More precisely, the way in which the curve is given. For example, the curve can be parametrized or, more generally, given by an implicit form, i.e., as the zero locus of a function $f(x, y)$. In some other problems (as the one mentioned in Section 1.1), the plane curve is given as the projection of a curve in a higher-dimensional space. The plane curves in the previous examples are described by functions. This leads us to another factor, the type of functions used to define that curve, e.g., polynomials, analytic or more general differentiable functions. Moreover, one also considers the domains in which these functions are defined, e.g., real numbers, complex numbers, finite fields, etc. Finally, an important factor is how we want to output singularities (e.g., parametrized, contained in isolating boxes, etc.) and how reliable the output is.

These considerations are parameters of a family of problems, each of which having its own state of the art. In Section 1.3, we discuss the state of the art of the problems that are related to our goal.

Genericity. The formal description of the problem concerning this thesis requires us to mention the concept of genericity. In vague terms, a property is generic if it is satisfied for almost all instances. More formally, the set of C^∞ functions from \mathbb{R}^n to \mathbb{R}^{n-1} is equipped with the weak topology [Dem2000, §3.9.2] (see Section 2.2.4). The convergence in this topology is defined as follows: For a sequence of C^∞ -functions $f_i \rightarrow f$, every partial derivative of f_i uniformly converges to the corresponding partial derivative of f over every compact subset of \mathbb{R}^n . A property over C^∞ -functions is generic if it is satisfied by a countable intersection of open dense subsets of functions with respect to this topology.

Formal statement of the problem. Let \mathcal{C} be a generic (in particular, smooth; see Lemma 3.2.1) curve defined by a C^∞ -function P from \mathbb{R}^n to \mathbb{R}^{n-1} , that is, if P maps $x = (x_1, \dots, x_n) \rightarrow (P_1(x), \dots, P_{n-1}(x))$, the curve \mathcal{C} is the intersection of the $n - 1$ hypersurfaces of the implicit equations $P_i(x) = 0$ for $i = 1, \dots, n - 1$. In particular, \mathcal{C} is not necessarily algebraic. Let $\pi_{\mathcal{C}} : \mathcal{C} \rightarrow \mathbb{R}^2$ be the projection sending every point to its first two coordinates. We want to find a set of pairwise disjoint boxes in \mathbb{R}^2 (a product of two closed intervals) such that every singular point of $\pi_{\mathcal{C}}(\mathcal{C})$ is contained in a unique box of this set and every box of this set contains exactly one singular point of $\pi_{\mathcal{C}}(\mathcal{C})$.

Plane curves $\pi_{\mathcal{C}}(\mathcal{C})$ of the type described in the latter paragraph appear naturally in visualizing the apparent contour of surfaces which is also related to mechanical design. As mentioned in the motivation (Section 1.1), obtaining information about the silhouette projection to the command space eases the analysis and design of safe motion.

1.3 Related work

This Section is dedicated to the state of the art of the methods for approximating plane curves and in particular the plane projections of higher-dimensional curves. We also focus on the sub-problem of detecting their singularities. We are in particular interested in methods that ensure the exactness of the topology and that correctly detect all the singularities.

There are two main approaches for approximating the plane projections of curves in \mathbb{R}^n . Either the curve is approximated in \mathbb{R}^n and then projected or it is projected and then approximated.

Most classical numerical methods for approximating curves in \mathbb{R}^n are based on marching cube (see [GVJ⁺2009, Chapter 7]) and continuation methods (see [GVJ⁺2009, Chapter 6]). However, even when these approaches may ensure to preserve the topology of the curve in \mathbb{R}^n [KX1994, FM2007, BL2013, MGGJ2013, TBV2019], they do not ensure to preserve the topology of their projections and we do not detail these methods in this state of the art.

On the other hand, algebraic methods for approximating curves in \mathbb{R}^n are to the best of our knowledge based on first projecting these curves in the plane (see e.g. [MPS⁺2006, Section 3.7]) and they are thus of no interest in the context of our problem.

In summary, there are very few approaches that approximate with a certified topology the plane projections of higher-dimensional curves without first projecting

them in the plane. We only know three of such contributions, which we detail in Section 1.3.4: Plantinga and Vegter [PV2006] consider the contour generator projections of implicit surfaces. Delanoue and Lagrange [DL2014] consider the apparent contours of smooth surfaces in \mathbb{R}^4 and Imbach et al. [IMP2016b] handle the plane projections of smooth curves in \mathbb{R}^3 using a subdivision scheme locally in four dimensions; the latter one is of particular interest for us because our work is a generalization of theirs.

Consider now the approaches that first project the curves in \mathbb{R}^2 . When the curves are algebraic, projecting them can be done using elimination theory [CLO2007, Chapter 3]. Then the main problem for approximating the plane curve is to isolate the singularities of the projected curve, which amounts to solving an algebraic zero-dimensional system of equations, for which there is a large state of the art (see Section 1.3.2). However, computing the projection (using elimination theory) has a rather high complexity and it is quite time consuming in practice. Furthermore, when the input curves are not algebraic, existing approaches of elimination theory cannot be used.

Our approach is based on the idea of encoding the singularities of the projected curve by a *square* and *regular* system that can be solved using certified numerical methods. In fact, the same idea has been already used by Imbach et al. [IMP2016b] for the plane projections of generic curves in \mathbb{R}^3 . We provide more details about this work in Section 1.3.4. The same idea was also considered to deal with singular surfaces. More precisely, Diatta et al. [DMP2019] used this idea to encode the curves of singularities that appear on surfaces obtained by projecting generic surfaces in \mathbb{R}^4 to \mathbb{R}^3 .

The rest of this section is organized as follows. In Section 1.3.1 we discuss how to characterize the singularities, given different types of plane curves. Based on that, in Section 1.3.2, we consider the problem of zero-dimensional systems of equations of which singularities detection can be seen as a special case. Then, in Section 1.3.3, we provide a brief state of the art on the isolation of singularities in implicit plane curves. Finally, Section 1.3.4 is dedicated to explain the approaches that are more related to our goal.

1.3.1 Curves and singularities

As mentioned in Section 1.2.3, there are several parameters that must be taken into account when treating the problem of isolating the singular points. When the curve is real or complex and described in the “language of functions” (e.g., as solution set of implicit equations, as a locally or globally parametrized curve, a projection of a

curve embedded in a higher dimension, etc.), the Jacobian criteria can be used to encode singularities as the solution set of a zero-dimensional system of equations. If the curve is parametrized by $(f_1(t), f_2(t))$, then every singular point corresponds to a t -value with $(f_1'(t), f_2'(t)) = (0, 0)$. If the curve is given by $f(x, y) = 0$, with f is differentiable, then the singularities are the solutions set of the system $f(x, y) = \frac{\partial f}{\partial x}(x, y) = \frac{\partial f}{\partial y}(x, y) = 0$. For the case where the curve is the projection of an implicit curve in a higher dimension, it is more complicated. As mentioned in the preamble of this section, one may project the curve to \mathbb{R}^2 to compute an implicit equation that defines the projection by which the singularities can be characterized. Another way to characterize the singularities can be done in terms of points in the upper curve as in [IMP2016b, DL2014] which involves zero-dimensional systems of equations.

From the foregoing, our considered problem can be seen as a special case of solving a zero-dimensional system of equations. This general problem is fundamental in algebra and has itself parameters that change many aspects. So, in addition to the state of the art of singularities detection, it is useful to go through the history of solving a zero-dimensional system of equations. This allows us to better understand the different available approaches of singularities detection. In general, two main types of approaches are used to deal with both problems. In the following section, we clarify these types. We discuss their strategies, pros and cons that interest us.

1.3.2 Zero-dimensional systems

By a zero-dimensional system we mean a system of which the solutions set forms a zero-dimensional manifold. This is a very active area of research involving, in general, two main types of approaches. These approaches are numerical and symbolic. In the following, we explain these types in details.

Numerical methods

Numerical methods are tools to solve mathematical problems where the output is an approximated form of the targeted object. In the context of solving system of equations (in particular, isolating singularities of plane curves), the basic strategy in numerical methods is to construct sequences of objects that converge towards the solutions. This gives rise to an important question, that is whether the convergence is always guaranteed. Unfortunately, the answer is no in general. Moreover, the solutions are usually sought in a given area of interest. This makes searching for roots in precise regions efficient. However, if the goal is finding all solutions,

numerical methods do not give a complete answer, unless bounds on the solutions are given. On the other hand, numerical methods provide fairly fast approaches compared with the other category (which we introduce next). Moreover, in terms of the type of equations, numerical methods can solve a wide range of systems such as analytic and differentiable.

One can find in the literature classical approaches based on Newton (-Raphson) or Fixed-Point methods; see for example [Epp2013, (§) 7.8] for details. These methods are not certified in general. However, under some conditions, the convergence can be guaranteed. Concerning our goals in this thesis, we focus on certified methods. An important type of numerical methods is subdivision-based methods; see for example [EH2004, SP1993, GS2001b] and the references therein. Subdivision can be seen as a generalization of binary search. Starting from a given domain, the subdivision consists in recursively bisecting the area until existence and uniqueness criteria are satisfied or the non-existence of solutions is guaranteed. One can distinguish two main types of subdivision: algebraic and box-function-based. Algebraic subdivision handles systems of polynomials. One exploits representations of polynomials such as Bernstein form or B-splines [ZSP1993, GS2001b, GS2001a, MP2009]. On the other hand, box-function-based subdivision [Neu1991, PVHK1997, Kea2013] is based on interval evaluations of the given functions (and possibly their derivatives) which supports a broader class of functions.

In these approaches, different tests are used. In the case of differentiable functions of which box functions are given (i.e., interval enclosures of the function evaluations; see Definition 5.1.1), some approaches are based on interval computation techniques such as interval evaluation, Newton interval method, Krawczyk operator [Neu1991, MKC2009]. These operators offer existence, uniqueness and exclusion tests (see [Neu1991, Theorems 5.1.7 & 5.1.8]). Other approaches use tests such as fixed point theorem of Brouwer, in the form of Miranda's Theorem [Kul1997]. For example, in [XY2019], Miranda and Jacobian test are used to verify the existence and uniqueness respectively. In addition, the exclusion test of the previous paper is based on interval evaluation.

Another type of numerical methods is based on homotopy continuation; see for example [BL2013, VDH2015, TBV2019]. Using a system S' with known solutions, homotopy-based methods build a continuous path between the solutions of a given system S and the solutions of S' .

Symbolic methods

Roughly speaking, symbolic methods are algorithms with exact input and output based on a formal language and algebraic-combinatorial approaches. In the case of solving a zero-dimensional system of equations (in particular, isolating singularities), these methods are usually based on elimination strategies. Thus, they can be seen as generalizations of the classical Gauss elimination method. The basic idea of this type of methods is, first, to transfer the given problem to a lower dimensional one using algebraic-combinatorial methods, returning a formal representation of the solutions. This step is usually called a symbolic or a projection step. The second step is to transfer this characterization of solutions into an approximated numerical presentation. This step is called numerical or lifting step.

In contrast to general numerical methods, symbolic methods are certified and provide a complete answer about the solution set. Moreover, they do not require initial conditions to guarantee the correctness or completeness of the output. On the other hand, in most well-conditioned problems, the techniques used in the projection step are usually slow. compared with the numerical ones. This makes symbolic methods unpractical for systems of high dimensions and degrees. Moreover, most algebraic techniques apply only for polynomials which limits the generality of such approaches.

In general, two main types of symbolic methods are used for zero-dimensional systems: those that are based on resultant techniques or Gröbner basis. Using these approaches, solving a zero-dimensional system amounts to, roughly speaking, computing a formal representation of the solutions such as a univariate representation or a triangular representation. To stay in the context of our thesis, we present in the following a short survey of the contributions considering mainly the bivariate case. For the general zero-dimensional case, we refer the reader to [CLO2007] and [RZ2009].

Resultant techniques. The resultant of two univariate polynomials is a polynomial expression of their coefficients that is zero if and only if the two polynomials have a common root. This idea can be generalized to higher dimensions. In the latter case, the resultant is used for computing the projection of varieties. For solving zero-dimensional systems (in particular isolating singularities), the usual approach using resultant is to compute the projection of solutions along several directions. Then, to obtain candidates which are then filtered until we get only the solutions. In the context of implicit plane curves, methods based on genericity are presented in [GVK1996, SW2005, CGL2009, Vil2018]. Other papers use resultant techniques without generic position assumption such as [DET2009, BES2011].

Gröbner basis. The Gröbner basis technique was introduced by Bruno Buchberger [Buc1965].¹ Given a system of polynomial equations, a Gröbner basis of this system is a set of polynomials satisfying the following: A polynomial f vanishes in all solutions of the system iff it is divisible by the Gröbner basis, that is, the remainder of f , over the Gröbner basis polynomials, by the Euclidean division, is zero. In addition to zero-dimensional systems, Gröbner basis techniques are used in many applications in computer algebra; see for example [GM1989, Laz1992, Sva2014] and the references therein.

Based on the previous techniques, the following representations can be used to solve zero-dimensional systems:

Univariate representation. Given a zero-dimensional system of polynomial equations, this representation consists of, first, a univariate polynomial. The roots of this polynomial are in one-to-one correspondence with the given system solutions. Second, a set of univariate rational mappings that parameterize the coordinates of the solutions, that is, the mappings send the roots of the univariate polynomials to the solutions of the system. This type of representation started in end of 19th century with Kronecker's work. This approach usually consists of two steps: The first one is to compute a polynomial (called separating polynomial) that takes pairwise different values when evaluated in distinct solutions. Second, to compute the polynomials defining the univariate representation.

Recently, this type of techniques has been widely studied for both solving zero-dimensional polynomial systems and singularity detection of implicit plane curves; see for example [ABRW1996, Rou1998, BSS2001, Bou2014, BLM⁺2016] and the references therein.

Triangular representation. The basic idea of this method is to recursively rewrite the system that we want to solve in a triangular way, that is, in an ordered list of equations such that every equation has one variable less than the one before. Hence, in the manipulated system, there is a univariate equation which can be solved numerically. This allows to analogously compute approximations of the other variables. In the bivariate case, this amounts to saying that the manipulated system is of the shape $f_1(x, y) = f_2(x) = 0$. The function f_2 is solved using univariate numerical methods [Epp2013, §3]. Then, f_1 is solved in the same way using the computed approximations of f_2 -solutions; see for example [Laz1992, Kal1993, WCB2001, vdHL2018] and the references therein for more details.

¹Wolfgang Gröbner was Buchberger's thesis advisor.

1.3.3 Singularities of implicit plane curves

By an implicit plane curve we mean the zero locus of a bivariate function $f(x, y)$. To approximate such curves, the literature is rich of uncertified numerical methods that compute approximations of given curves without any guarantee in the topology. Since this type of methods is out of our interests for this thesis, we simply refer the reader to [GVJ⁺2009] for more details.

For approximating plane curves in a certified way, one can find in the literature many approaches that are based on subdivision such as in [Sny1992] or homotopy such as in [BL2013]. These approaches are certified under the smoothness condition. Some software is also available to certify smooth curves such as [Ibexsolve](#) [Nin2015]. This software provides a tool to enclose smooth plane curves, even for some non-algebraic curves, more precisely, for those that are defined by [the Minibex language](#) (e.g., sin, cos, exp...).² For singular curves, ISOTOP [CLP⁺2010] can be used to certify the topology of algebraic curves.

Concerning the problem of isolating singularities, recall that, for plane curves, the singularities are encoded by the system $f(x, y) = \frac{\partial f}{\partial x}(x, y) = \frac{\partial f}{\partial y}(x, y) = 0$. This system does not satisfy the conditions required for applying certified numerical methods such as the interval Newton methods.

A numerical approach to solve this issue is to use combinations of these equations to translate this system into a square one. Non-regular solutions can be handled through deflation systems (see for example [OWM1983, LVZ2006, MM2011, HMS2015]). However, spurious solutions may appear in this process. We are not aware about any approach that finds the singularities of a general implicit plane curve numerically in a certified way. Recall that, in Section 1.3.2, we discussed the symbolic methods for isolating the singularities of plane curves in the context of solving zero-dimensional systems of equations.

1.3.4 Singularities of the plane projections of implicit space curves

As mentioned in Section 1.1, such curves appear naturally in real-life problems as in robotics. In the polynomial case, for curves given as the intersection of two surfaces in \mathbb{R}^3 , the plane projection can be defined by the resultant of the polynomials defining the surfaces. Its singular locus can be related to the first sub-resultant. Imbach et al. [IMP2016a] use the latter facts and present a deflation-based approach to generically isolate the singularities of the plane projection (i.e., the zero set of the

²<http://www.ibex-lib.org/doc/minibex.html>.

resultant). This approach involves obtaining a square regular system that encodes the singularities. The idea of this system is based on a relation between the singularities of the plane projection and sub-resultants under some generic assumptions. In other words, under some generic assumptions presented in [IMP2016a, §2], the authors provide a square regular system [IMP2016a, Theorem 2], based on sub-resultants that can be used to numerically isolate singularities in a certified way.

A different approach is presented in [IMP2016b] for more general curves, namely, those that are defined by analytic functions. Similarly to [IMP2016a], the goal of the latter work is to numerically isolate the singularities of the plane projection of a curve in \mathbb{R}^3 . On the other hand, the approach in [IMP2016b] uses information about the projection function Π_{xy} from the space curve onto \mathbb{R}^2 . More precisely, under some generic assumptions [IMP2016b, §2.1], the authors show that a node (resp. an ordinary cusp) appears in the plane projection exactly where Π_{xy} is not injective (resp. not immersion; see Definition 2.2.6). Based on these statements, a regular square system that encodes singularities is presented. This system is called the Ball system and is solved using subdivision approaches. Comparing with the system presented on [IMP2016a], the Ball system is of higher dimension which may mean that it is more costly to solve. On the other hand, it is more general since it holds for analytic curves. An algorithmic version of this approach in addition to an implementation (for polynomials with interval coefficients) are presented in [IMP2018]. Even though subdivision approaches may suffer in practice from the curse of dimensionality, Imbach et al. observe experimentally that, for algebraic curves, their approach is more efficient than computing the implicit equation of the projected plane curves and its singularities using symbolic methods. In fact, our work in this thesis can be seen as a generalization of this approach for generic space curves in \mathbb{R}^n , for $n > 2$ defined by C^∞ functions.

Plantinga and Vegter [PV2006] provided for the first time explicit conditions on implicit surfaces in \mathbb{R}^3 to exhibit generic singularities in the contour generators (the curves on the surfaces which separate front-facing from back-facing regions) and their projections. They presented a certified algorithm to approximate the contour generator and its projection. The algorithm is based on a theoretical analysis which provides local models of the singularities.

Delanoue and Lagrange [DL2014], compute the topology of the apparent contour (the set of critical values) of a smooth mapping from \mathbb{R}^2 to \mathbb{R}^2 . The reader can notice the connection of this work with our considered problem by the following: The graph of a generic function $f : (x, y) \rightarrow (X = f_1(x, y), Y = f_2(x, y))$ can be seen as a parametric surface in \mathbb{R}^4 , defined by $\{(x, y, X, Y) \in \mathbb{R}^4 \mid X = f_1(x, y), Y = f_2(x, y)\}$. The silhouette is the set of those points in the surface

where the projection of the tangent onto \mathbb{R}^2 is not surjective. The plane projection of the silhouette is equal to the apparent contour. This approach is based on result proved by Whitney that, generically, critical points of a smooth map are folds or cusps [Whi1955]. Using interval Newton method, Delanoue and Lagrange solve two systems of equations which represent ordinary cusp and nodes respectively. The first one is based on the idea that f restricted to the apparent contour around an ordinary cusp is equal (up to an equivalence) to $(x, y) \rightarrow (x, xy + y^3)$. The second system is based on characterizing distinct points of the domain that have the same image under f .

1.4 Contribution

In this section we give a detailed overview of the results we obtained. Recall that we are interested in the problem of isolating the singularities of the plane projection of a generic curve \mathcal{C} of \mathbb{R}^n , with $n > 2$. Our approach is based on the following: First, we find proper conditions on \mathcal{C} that lead to characterizing the singularity in a useful way. These conditions are general enough in the sense that we prove that they are generic. Hence, after finding these conditions, we prove their genericity. Then, under these conditions, we introduce a regular-square system that encodes the singularities of $\pi_{\mathcal{C}}(\mathcal{C})$. Based on this system, we develop a (semi) algorithm that, first, checks whether a given curve \mathcal{C} satisfies our assumptions. Second, it returns a set (or a superset) of boxes in \mathbb{R}^2 that isolates the singularities of $\pi_{\mathcal{C}}(\mathcal{C})$. In addition, we provide an implementation of our (semi) algorithm and we examine its performance using examples of implicit curves and an application in robotics.

The main novelty of our work is that we consider types of curves for which no certified methods used to exist. Furthermore, our approach allows us to handle classes of curves that were in practice not reachable by other methods in a certified way. Namely, we consider curves defined by C^∞ -functions and not only algebraic or even analytic functions. Moreover, for curves that live in \mathbb{R}^n for n larger than 3, no certified numerical methods existed even for algebraic curves. For such types of algebraic curves, our preliminary experiments show that our approach allows us to handle, in practice, curves that are unreachable by classical algebraic certified approaches.

We give in Section 1.4.1 a summary of the assumptions we consider. Then, in Section 1.4.2 we explain our idea for encoding the singularities in a square and regular system, which we call the Ball system. Section 1.4.3 is dedicated to briefly describing the specifications of the algorithm we designed for checking the assumptions and isolating the singularities of the projected curve. In addition, we show how

we overcame the curse of dimensionality when solving the Ball system. Finally in Section 1.4.4, we outline the properties of our implementation and the experiments we perform to measure its efficiency.

1.4.1 Generic assumptions

Our goal is to design a numerical algorithm for isolating the singularities that appear in the plane projection of a curve \mathcal{C} in \mathbb{R}^n . Numerical algorithms usually cannot handle degenerate cases, that is, singularities in our context. However, under some assumptions on \mathcal{C} (which are formally defined in Section 3.1.2), we succeed to handle in a certified way some, although not all, singularities of the projection. Namely, we require that the singularities are “generic”, that is, only nodes can appear in the projection. Our other assumptions on \mathcal{C} are, roughly speaking, that it is smooth, that its projection only has a discrete set of singularities, and that at most two points of \mathcal{C} project on each singularity. We call this set of assumptions strong. We prove in Section 3.2 that these assumptions are generically satisfied.

In addition, we consider a weaker assumption that relaxes the constraint that only nodes can appear. This assumption allows for ordinary cusps in $\pi_{\mathcal{C}}(\mathcal{C})$. Our motivation for also considering this weak assumption is dual. First, our certified algorithm for isolating the singularities of the projection of curves satisfying the strong assumptions also works, to some extent, if only the weak assumptions hold: namely, it outputs a *superset* of the isolating boxes of the singularities. Second, we conjecture that our weak assumptions are satisfied by silhouette curves of generic surfaces (see Proposition 3.2.17 and Conjecture 3.2.18). A detailed explanation about the assumptions and their genericity are presented in Chapter 3.

1.4.2 Enclosing singularities by the Ball system

We consider here our set of weak assumptions, that is, nodes and cusps are allowed in the projected curve $\pi_{\mathcal{C}}(\mathcal{C})$. Our goal is to encode the singularities of $\pi_{\mathcal{C}}(\mathcal{C})$ by a square and regular (see Definition 4.4.1) system so that it is solvable with certified numerical methods.

Intuitively, a node appears when two points of \mathcal{C} project to the same point and a cusp appears when projecting a point with a tangent line orthogonal to the projection plane (see Figure 1.7). The idea to encode the nodes is to design a system whose variables are the coordinates of two different points in \mathbb{R}^n constrained to be on \mathcal{C} and so that they have the same plane projection. To encode a cusp, we design a system whose variables are the coordinates of one point in \mathbb{R}^n constrained to be on \mathcal{C} with a tangent orthogonal to the projection plane.

Furthermore, we need systems that are square and regular for using certified numerical solvers. To solve this issue and to gather the two systems into a single one, we first parameterize two different points of \mathcal{C} with the same projection by $(x_1, x_2, y + r\sqrt{t})$ and $(x_1, x_2, y - r\sqrt{t})$, with $x_1, x_2, t \in \mathbb{R}$, y, r in \mathbb{R}^{n-2} and $\|r\| = 1$, where $\|r\|$ denotes the Euclidean norm of r . Hence, the Ball system is of size $2n - 1$. Then, given any function f from \mathbb{R}^n to \mathbb{R} so that $f = 0$ is one of the $n - 1$ hypersurfaces that define \mathcal{C} , we introduce (Definition 4.1.1) two smooth functions $S \cdot f$ and $D \cdot f$. When $t > 0$, they return, roughly speaking, the arithmetic mean and difference of f at the above two points, hence they both vanish if and only if the two points are on the hypersurface $f = 0$. When $t = 0$, the two points coincide and $S \cdot f$ and $D \cdot f$ return, roughly speaking, f evaluated at this point and the gradient of f (at that point) scalar the “vertical” vector $(0, 0, r)$; hence, they both vanish if and only if the point is on the hypersurface $f = 0$ and its tangent hyperplane is normal to the plane of projection. It follows that given a curve defined by $P_1 = \dots = P_{n-1} = 0$, the solutions of the so-called Ball system of all $S \cdot P_i = D \cdot P_i = 0$ is the set of points on the curve that project to nodes and cusps (Theorem 4.1.3).

Note that we consider \sqrt{t} instead of t in the parameterization $(x_1, x_2, y \pm r\sqrt{t})$ for ensuring the regularity of the Ball system when $t = 0$ (because this ensures that the linear term of the Taylor expansion of $D \cdot f$, with respect to t , does not vanish). We refer the reader to Chapter 4 for a more formal explanation.

1.4.3 Algorithmic contribution

Based on interval arithmetic [MKC2009], we provide a semi-algorithm to check whether a given implicit curve in \mathbb{R}^n satisfies the weak (or strong) assumptions. More precisely, the semi-algorithm terminates if and only if the assumptions are satisfied. Moreover, under the strong assumptions the semi-algorithm provides a set of pairwise disjoint boxes in \mathbb{R}^2 each of which contains exactly one singular point (in this case, it is a node) of $\pi_{\mathcal{C}}(\mathcal{C})$ such that the boxes union contains all singularities. On the other hand, under the weak assumptions, the semi-algorithm returns a superset of the strong-assumption-case output. A full description of the semi-algorithm we designed is in Chapter 5.

Pruning the domain of the Ball system. As mentioned before, the Ball system is of size $2n - 1$. This is because the Ball system encodes (a) two distinct points in the case where the singular point is a node (see Figure 1.8, left) or (b) a point and a tangent vector for the case of an ordinary cusp. It is arguable whether solving a system of size $2n - 1$ for detecting singularities of a $2D$ curve is a good idea

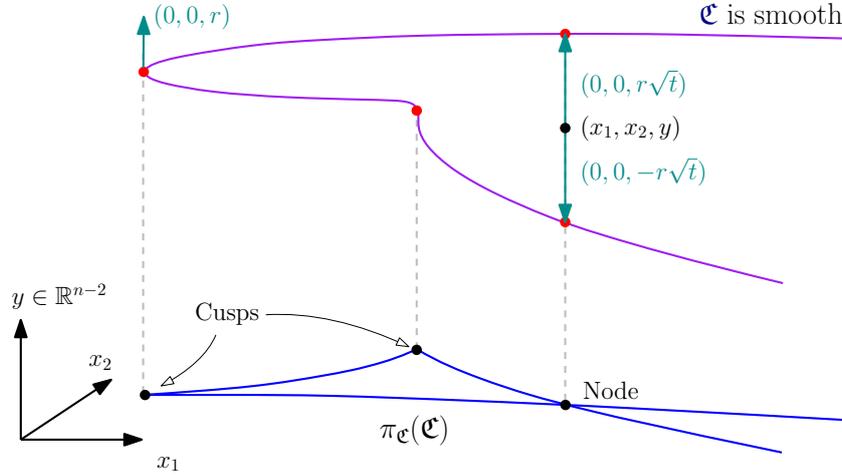


Figure 1.7: Illustration of a node and cusps in the plane projection of a smooth curve.

from an efficiency point of view. In fact, we succeed to overcome the curse of dimensionality by pruning the domain on which we solve the Ball system.

The idea is that we compute a set of n -dimensional boxes that cover \mathcal{C} . Then, every singular point of $\pi_{\mathcal{C}}(\mathcal{C})$ is contained, roughly speaking, in either (a) the plane projection of a box where the curve has (possibly equal) x_1 -critical points and x_2 -critical points or (b) a non-empty plane projection of two disjoint boxes. Then, we map these boxes to boxes in \mathbb{R}^{2n-1} ,³ which are the restricted domain on which we solve the Ball system (see Figure 1.8, right). A detailed description of the pruning in addition to other improvements are described in Section 5.3.

1.4.4 Implementation, experiments and applications

We provide a software *Isolating_singularities*⁴ written in Python that is an implementation of our semi-algorithm mentioned in Section 1.4.1. This software uses black boxes such as *Python-FLINT* [Joh2012b] and *Ibexsolve* [Nin2015] to deal with interval-arithmetic objects and to solve regular systems of non-linear equations. In addition, our software provides a way to visualize the projection of a space curve into \mathbb{R}^2 or \mathbb{R}^3 . We present our software in Chapter 6 and experiments and discussions on its performance in Chapter 7.

³Recall that $2n - 1$ is the size of the Ball system.

⁴The software is accessible at https://github.com/gkrait/Isolating_singularities.

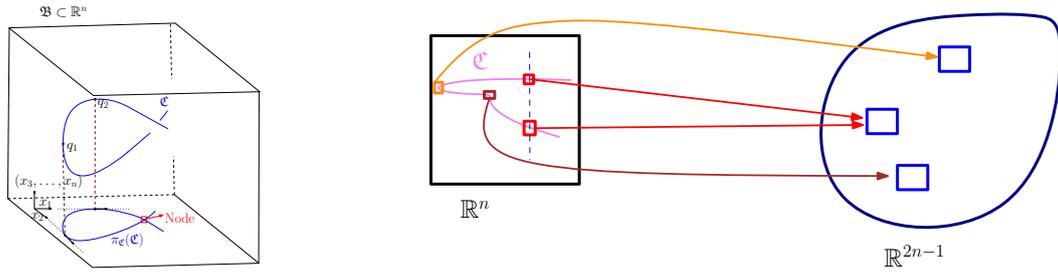


Figure 1.8: Left: An example of a box in \mathbb{R}^n that satisfies (a) and whose projection contains a node. Right: A simplified illustration of pruning the domain of the Ball system. The brown and orange boxes stratify (a), also the pair of red boxes satisfy (b). Using these boxes, the blue boxes in \mathbb{R}^{2n-1} are computed.

We terminate the summary of our contribution with exhibiting results regarding an application in robotics. Within this frame, we study the plane projection of the parallel and serial singularities (not to be confused with the geometric singularity; see Definition 8.1.1 (f)) of a parallel 2-degrees-of-freedom robot. In the considered example, the parallel and serial singularities are curves in \mathbb{R}^4 defined by analytic functions. With respect to different plane projections, we compute certified isolating boxes of singular (the geometric sense) points in the plane projection of the parallel and serial singularities (as long as our assumptions are satisfied). In the end, we provide 2D and 3D plots to visualize the results. See Chapter 8 for more details.

Part I

Singularity modelling

This part presents the main content of this thesis. It contains the theoretical aspects of our contribution and the results that prove the correctness of our approach. This part consists of local and global analysis of the plane projection curve in addition to (semi-) algorithms that are based on our theoretical results.

Before starting the analysis, we recall some facts in algebra and geometry in Chapter 2. Then, in Chapter 3, we introduce a set of conditions on curves in \mathbb{R}^n that guarantee the correctness of our approach. We prove the genericity of these conditions discussing, in the end of the chapter, the same conditions on the silhouette of a generic surface. In Chapter 4, we introduce the Ball system that encodes the singularities the plane projection of a space curve satisfying our conditions. We show, in addition, the relation between the regularity (Definition 4.4.1) of the Ball system and the generic conditions.

The algorithmic part consists of Chapter 5, where we provide the semi-algorithms for isolating the singularities of the plane projection of the space curve. We start the chapter with preliminaries in interval analysis. After presenting the semi-algorithms, we prove in Section 5.2 the correctness in addition to determining the necessary and sufficient conditions for termination. In Section 5.3, we explain improvements to the semi-algorithms preparing them to the implementation stage.

2

Preliminaries in geometry and algebra

Contents

| | |
|---|-----------|
| 2.1 Real algebraic geometry | 28 |
| 2.1.1 The ring of real polynomials | 28 |
| 2.1.2 Determinantal varieties | 29 |
| 2.2 Differential geometry | 29 |
| 2.2.1 Smooth functions and manifolds | 29 |
| 2.2.2 Germs of smooth functions | 31 |
| 2.2.3 The multiplicity of smooth systems of equations | 32 |
| 2.2.4 Transversality theorems | 33 |

This chapter is dedicated to recall some concepts and results in algebra and geometry. Most of the material in this chapter is well-known and can be found in any introductory textbook on commutative algebra, algebraic and differential geometry. We suggest to the reader the references [CLO2007, CLO2005, Dem2000]. The rest such as transversality theorems are less known.

We start this chapter with common properties of the ring of real polynomial and determinantal varieties. Then, we recall some results about manifolds and singularity theory. After that, we exhibit the concept of the multiplicity of non-linear system of equations. This concept is a generalization of the intersection multiplicity of the algebraic case. We terminate the chapter with transversality theorems which have a direct connection of our assumptions.

2.1 Real algebraic geometry

In this section, we recall concepts in algebraic geometry that we use in different places in this thesis. We assume that the reader is familiar with elementary notions and well-known results of commutative algebra such as polynomial rings, localization, ideals, Gröbner Bases, etc, and algebraic geometry such as affine varieties, Hilbert's Nullstellensatz, etc. We suggest to the reader the reference [CLO2007], in particular, Chapters 1, 2, 4, 5 and 9.

2.1.1 The ring of real polynomials

For a positive integer n and $x = (x_1, \dots, x_n)$, we consider the polynomial ring $\mathbb{R}[x]$ over the field of real numbers \mathbb{R} . In the following, we present the definition of the multiplicity of a zero-dimensional ideal over $\mathbb{R}[x]$ at a solution. This notion will be used to count the preimages of points in \mathbb{R}^2 under $\pi_{\mathcal{C}}$ (see Section 1.4.1). Loosely speaking, the multiplicity m at a solution a in this setup is a generalization of the univariate case where a is a repeated solution m times. For a point $q \in \mathbb{R}^n$, consider the ring $\mathbb{R}[x]_q$ that is the localization of $\mathbb{R}[x]$ at q .

Definition 2.1.1 ([CLO2005, Definition 4.2.1]). *For integers $m \geq n \geq 1$, let $G = (g_1(x), \dots, g_m(x))$ be a polynomial function from \mathbb{R}^n to \mathbb{R}^m and q be a solution of the system $\{G = 0\}$. Define the intersection multiplicity of q in the system $\{G = 0\}$ (or equivalently the multiplicity of the system $\{G = 0\}$ at q) to be the dimension of the real vector space $\frac{\mathbb{R}[x]_q}{I_G}$, where I_G is the ideal generated by the set $\{\frac{g_1}{1}, \dots, \frac{g_m}{1}\}$ in $\mathbb{R}[x]_q$.*

The previous definition is classical for the algebraic case. However, we are interested in curves defined as the zero locus of smooth functions. For this goal, we consider a more general definition in 2.2.10 for a system $S = \{f_1(x) = \dots = f_m(x) = 0\}$ with f_i is differentiable infinitely many times which is, unsurprisingly, more complicated and its computability is less clear. Definition 2.1.1 will be thus used when it is equivalent to Definition 2.2.10.

In some places of this thesis where we compute the multiplicity, we reduce the problem of computing the dimension of $\frac{\mathbb{R}[x]_q}{I_G}$ to computing the dimension of another quotient that is easier to handle. The following theorem is in this regards:

Theorem 2.1.2 ([CLO2005, Theorem 4.4.3, P. 177]). *Consider the real polynomial ring $\mathbb{R}[x]$ and its localization $\mathbb{R}[x]_q$ at some point $q \in \mathbb{R}^n$. Consider a local monomial ordering $>$ over $\mathbb{R}[x]_q$. Let I be an ideal in $\mathbb{R}[x]_q$ and $LT(I)$ be the ideal generated by the leading monomials of I . Then, the following statements are equivalent:*

(a) The dimension of the \mathbb{R} -vector $\frac{\mathbb{R}[x]_q}{I}$ space is finite.

(b) The dimension of the \mathbb{R} -vector $\frac{\mathbb{R}[x]_q}{LT(I)}$ space is finite.

Furthermore, when any of these conditions is satisfied, we have:

(c) The equality $\dim(\frac{\mathbb{R}[x]_q}{I}) = \dim(\frac{\mathbb{R}[x]_q}{LT(I)})$ holds.

2.1.2 Determinantal varieties

We study in this section a large and important class of varieties, those whose equations take the form of the minors of a matrix. This concept is useful in Section 3.2.2 to prove the genericity of properties for a silhouette curve, where those properties are based on determinantal relations.

Definition 2.1.3. Let $M(m, n)$ be the vector space of real matrices of size $m \times n$ and r be a positive integer such that $r < \min\{n, m\}$. The determinantal variety, $M_r(m, n)$, is the set of matrices in $M(m, n)$ that have rank less than $r + 1$.

Proposition 2.1.4 ([BV1988, Proposition 1.A.1.1]). Let $M_r(m, n)$ be a determinantal variety in $M(m, n)$. The following statements hold:

(a) $M_r(m, n)$ is an irreducible variety in $M(m, n)$.

(b) $M_r(m, n)$ is of dimension $r(n + m - r)$.

(c) The singular locus of $M_r(m, n)$ is $M_{r-1}(m, n)$.

Lemma 2.1.5 ([Bôc1964, §XIV.61 Theorem 1]). Let $n \geq 2$ be an integer, $\{x_{ij}\}_{1 \leq j, i \leq n}$ be a set of n^2 variables and $\mathbb{C}[x_{ij}]_{1 \leq j, i \leq n}$ be the ring of complex polynomials with the variables $\{x_{ij}\}$. Then, the determinant of the matrix $(x_{ij})_{1 \leq i, j \leq n}$ is an irreducible polynomial in $\mathbb{C}[x_{ij}]_{1 \leq j, i \leq n}$.

2.2 Differential geometry

In this section, we study real C^∞ (or smooth) functions, i.e., those that are differentiable infinity many times. Since the main goal of our work is to deal with the singular points of plane curves given in terms of smooth functions, it makes sense to study the behavior of this type of functions (and their zero locus) locally.

2.2.1 Smooth functions and manifolds

In this section we state the properties we need of the type of functions that interest us, namely, C^∞ functions. More formally, let n and m be positive integers. A

function f defined on an open subset of \mathbb{R}^n to \mathbb{R}^m is called C^∞ (or smooth) at a if it is differentiable infinitely many times at a , that is, all partial derivatives of f at a of all orders exist. For an open subset $U \subseteq \mathbb{R}^n$, the set of smooth functions from U to \mathbb{R}^m is denoted by $C^\infty(U, \mathbb{R}^m)$. In this section, we state useful statements about C^∞ functions and their zero locus.

Definitions 2.2.1 (Regular and singular points [Dem2000, Definition 2.2.2]). *Let $m \geq 1$ be an integer, V be a subset of \mathbb{R}^m and $p \in V$. We call p a regular (or smooth) point of V if V is a sub-manifold at p , that is, there exist a neighbourhood W of p in \mathbb{R}^m , an integer $k > 0$ and k smooth functions $\varphi_1, \dots, \varphi_k$ defined over W , such that $V \cap W$ is the set of solutions of $\{\varphi_1(x) = \dots = \varphi_k(x) = 0\}$ in W*

*and the rank of the matrix
$$\begin{pmatrix} \frac{\partial \varphi_1}{\partial x_1} & \cdots & \frac{\partial \varphi_1}{\partial x_m} \\ \vdots & \vdots & \vdots \\ \frac{\partial \varphi_k}{\partial x_1} & \cdots & \frac{\partial \varphi_k}{\partial x_m} \end{pmatrix},$$
 evaluated at q , is k . We call this matrix the Jacobian matrix of the system $\{\varphi_1(x) = \dots = \varphi_k(x) = 0\}$ and we denote it by $J_{(\varphi_1, \dots, \varphi_k)}$. If q is not a regular point of V , we call it a singular point (or singularity). If all points in V are regular, then V is called regular or smooth. Otherwise, V is called singular.*

For $\varphi = (\varphi_1, \dots, \varphi_k) \in C^\infty(\mathbb{R}^n, \mathbb{R}^k)$, we denote by $T_q\varphi$ its derivative (also known as the tangent map) at the point q . Note that the Jacobian matrix $J_\varphi = J_{(\varphi_1, \dots, \varphi_k)}$ is the expression of the derivative in the canonical bases of \mathbb{R}^n and \mathbb{R}^k .

A function $f : X \rightarrow Y$ between two manifolds is called diffeomorphism if f is a bijection and T_qf is full rank at every q in X . Two manifolds are called diffeomorphic if there exist a diffeomorphism from one of them to the other.

The following Theorem shows the compatibility between the dimension as a (smooth) real variety and the dimension as a manifold:

Theorem 2.2.1. [BCR1998, (Proposition 2.8.14)] *Let $V \subseteq \mathbb{R}^n$ be a manifold that is also an affine variety. Then its dimensions as a manifold and as a variety coincide.*

The following theorem by Whitney represents a nice property of even smooth functions. We use this theorem to rewrite functions locally and parametrize them in terms of variables of an even power.

Theorem 2.2.2 ([Whi1943, Theorem 1 & 2]). *Let f be an even (resp. odd) smooth function, then there exists a smooth function g such that $f(x) = g(x^2)$ (resp. $f(x) = x \cdot g(x^2)$).*

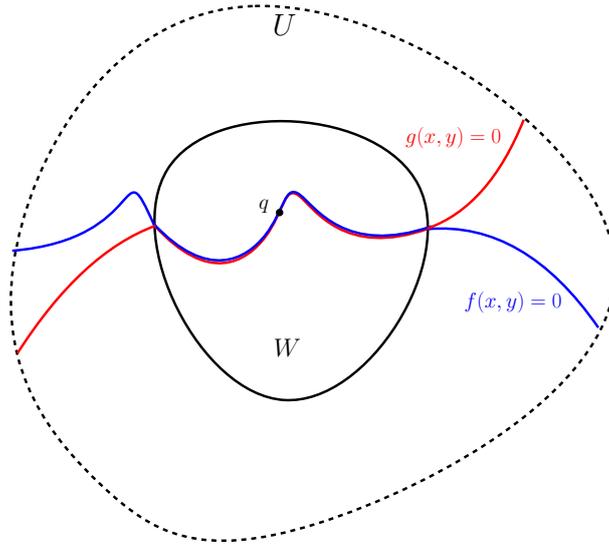


Figure 2.1: An example in \mathbb{R}^2 where f and g define the same germ.

2.2.2 Germs of smooth functions

As mentioned before, we are interested in the local behaviour of a smooth function that are defined in \mathbb{R}^n , that is, we want to study a function that is defined in a neighbourhood of a point $q \in \mathbb{R}^n$. For this reason, we can assume in this section, without loss of generality, that $q = 0 \in \mathbb{R}^n$.

Definition 2.2.3. [Dem2000, §5.2.1] *Let U be an open sub set of \mathbb{R}^n with $q \in U$ and $f, g : U \rightarrow \mathbb{R}$ be two smooth functions. The functions f, g define the same germ at q , if there exists an open subset W contained in U with $q \in W$ such that $f_W = g_W$, that is, f and g restricted to W coincide. Consequentially, the zero locus of f, g coincides in W (see Figure 2.1)*

Definition 2.2.4 ([AGZV2012, §17.1]). *For $i \in \{1, 2\}$, let C_i be a plane curve defined in a neighbourhood $U_i \subset \mathbb{R}^2$ of p_i by the zero set of a smooth function f_i . The pairs (p_1, C_1) and (p_2, C_2) are equivalent, and thus define the same plane curve singularity, if there exists a diffeomorphism φ from U_1 to U_2 such that $f_1 = f_2 \circ \varphi$ and $\varphi(p_1) = p_2$.*

In particular, a singularity is of type A_k if the curve is locally defined at the origin by the zero set of the function $x^2 - y^{k+1}$. As important special cases, A_1 is called a node singularity and A_2 is called an ordinary cusp singularity, see Figure 2.2. We are going to assume later that the plane curve that we consider has only nodes or ordinary cusps as singularities. After that, we prove that this assumption is “general enough”, more formally, generic; see Chapter 3.

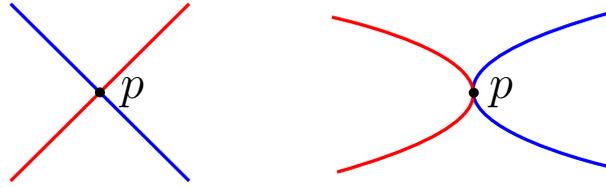


Figure 2.2: Left: At an A_1 singularity, two branches of the curve intersect transversally. Right: At an A_{2k+1} singularity with $k > 1$, the tangent lines of the two branches at the intersection point coincide.

Remark 2.2.5. *It is worthy to notice that a curve C has an ordinary cusp at a point p if C can be locally parametrized with (z^2, z^3) and p corresponds to the value $z = 0$. This remark is helpful to characterize ordinary cusps in Chapter 4.*

Definition 2.2.6. [GG1973, I.2.3] *Let X and Y be differentiable manifolds. Let $f : X \rightarrow Y$ be a differentiable function. Suppose that $T_q f$ is full rank. Then,*

- *if $\dim(X) \leq \dim(Y)$, f is an immersion at q ,*
- *if $\dim(X) \leq \dim(Y)$ and $f(X)$ is diffeomorphic to X , f is embedding*
- *if $\dim(X) \geq \dim(Y)$, f is an submersion at q ,*
- *if for every point $q \in X$, f is immersion (resp. submersion) at q , then f is an immersion (submersion).*

Proposition 2.2.7. [Dem2000, Proposition 2.9.6] *Preserving the same notation as in Definition 2.2.6, if f is an immersion at $q \in X$, there exists an open subset U in X containing q such that the restriction of f to U is embedding.*

2.2.3 The multiplicity of smooth systems of equations

We defined earlier the multiplicity of zero-dimensional systems of polynomials. Our goal now is to define the same notion for smooth functions. We first recall the definition of multiplicity in the univariate case before generalizing it to higher dimensions.

Definition 2.2.8. *Let f be a real smooth function at $a \in \mathbb{R}$. The order of f at a is the integer $\text{ord}_a(f(x)) = \min\{k \in \mathbb{N} \mid \frac{\partial^k f}{\partial x^k}(a) \neq 0\}$ if it exists, otherwise $\text{ord}_a(f(x)) = \infty$. For the case $a = 0$, we write for simplicity $\text{ord}(f) = \text{ord}_a(f)$.*

The following definition is introduced in [DLZ2011]. We are going to use the same notion of the previous paper:

Notation 2.2.9. Consider $x = (x_1, \dots, x_n)$, for $\tilde{k} = (k_1, \dots, k_n) \in \mathbb{Z}^n$ and $q \in \mathbb{R}^n$, define the differential operator $\partial_{\tilde{k}}[q] : C^\infty(U, \mathbb{R}) \rightarrow \mathbb{R}$ that sends f to $\frac{1}{k_1! \dots k_n!} \cdot \frac{\partial^{k_1 + \dots + k_n} f}{\partial x_1^{k_1} \dots \partial x_n^{k_n}}(q)$ if $\tilde{k} \geq 0 \in \mathbb{Z}^n$ and 0 otherwise. From now on, a linear combination of differential operator is called differential functional. The set of all differential functionals forms a vector space denoted by D_q over \mathbb{R} . Define the linear anti-derivative operator ϕ_j that maps $\frac{\partial^k}{\partial x_1^{k_1} \dots \partial x_j^{k_j} \dots \partial x_n^{k_n}}$ to $\frac{\partial^{k-1}}{\partial x_1^{k_1} \dots \partial x_j^{k_j-1} \dots \partial x_n^{k_n}}$ if $k_j > 0$ and to $\frac{\partial^0}{x_j^0}$ otherwise, where $k = \sum_{i=1}^n k_i$.

Let $S = \{f_1(x) = \dots = f_m(x) = 0\}$ be a system of smooth equations, q be a solution of S and k be a non-negative integer, we define the dual space of rank k , denoted by $D_q^k[S]$, to be the vector space of all linear combinations c of differential functionals $\frac{\partial^{k_1 + \dots + k_n}}{\partial x_1^{k_1} \dots \partial x_n^{k_n}}$ with $k_1 + \dots + k_n \leq k$ such that:

- (a) $D_q^0[S] = \text{span}(\{\frac{\partial^0}{\partial x_1^0 \dots \partial x_n^0}\})$,
- (b) c applied to f_i , evaluated at q is zero for all integers $1 \leq i \leq m$, and
- (c) for all $i \in \{1, \dots, n\}$, the anti-differentiation transformation ϕ_j applied to c is in $D_q^{k-1}[S]$.

Definition 2.2.10 ([DLZ2011, Definition 1]). Let $F \in C^\infty(\mathbb{R}^n, \mathbb{R}^k)$ such that $F^{-1}(0)$ is a finite set and let $a \in \mathbb{R}^n$ be a solution of the system $S = \{F = 0\}$. Consider the ascending chain of dual spaces $D_q^0[F] \subseteq D_q^1[F] \subseteq \dots \subseteq D_q^h[F] \subseteq \dots$. If there exists an integer α such that $D_q^\alpha[F] = D_q^{\alpha+1}[F]$, then the dimension of the vector space $D_q^\alpha[F]$ is called the multiplicity of q in the system S . If such an α does not exist, the multiplicity is, by convention, infinity.

As mentioned before, for polynomial systems, the previous definition is equivalent to Definition 2.1.1 [DLZ2011, Theorem 2]. In addition, the following proposition shows that algebraic tools can be used in the smooth case.

Proposition 2.2.11 ([DLZ2011, Corollary 3]). For an integer $k \geq n$, let $F = (f_1, \dots, f_k) \in C^\infty(\mathbb{R}^n, \mathbb{R}^k)$ and let $a \in \mathbb{R}^n$ be a solution of the system $\{F = 0\}$. Suppose that the multiplicity of a in $\{F = 0\}$ is $m < \infty$, then the intersection multiplicity at a of the polynomial system $\{G = (g_1, \dots, g_k) = 0\}$ is also m , where g_i is equal to the Taylor expansion of f_i at a up to degree at least m .

2.2.4 Transversality theorems

The goal of this section is to introduce the concept of genericity and some related results. This has of course a direct relation with our assumptions.

Let $U \subseteq \mathbb{R}^n$ be an open subset. The set $C^\infty(U, \mathbb{R})$ can be equipped with the so called weak topology that is defined in [Dem2000, §3.9.2] as follows: Let f_i be a sequence of elements in $C^\infty(U, \mathbb{R})$. The latter sequence is said to be convergent to a function f if over every compact subset of U , the sequence $\frac{\partial^k f_i}{\partial x_1^{k_1} \dots \partial x_n^{k_n}}$ uniformly converges to $\frac{\partial^k f}{\partial x_1^{k_1} \dots \partial x_n^{k_n}}$, with $k = \sum_{i=1}^n k_i$ for all k_1, \dots, k_n in \mathbb{N} . Having thus defined convergent sequences, we now know what are the closed and open sets in $C^\infty(U, \mathbb{R})$. Naturally, for a positive integer m , the set $C^\infty(U, \mathbb{R}^m)$ is equipped with the product topology as $C^\infty(U, \mathbb{R}^m)$ can be seen as is the Cartesian product of m copies of $C^\infty(U, \mathbb{R})$.

The key to prove the genericity of our assumptions is Thom's Transversality Theorem. We thus first recall, in this Section, the basics of transversality theory using the notation of Demazure's book [Dem2000].

We work with the set of smooth functions $C^\infty(\mathbb{R}^n, \mathbb{R}^{n-1})$ with the weak (or compact-open) topology [Dem2000, §3.9.2], that is, the convergence is understood as uniform on compact subsets and for any derivative. A subset of $C^\infty(\mathbb{R}^n, \mathbb{R}^{n-1})$ is called residual if it contains the intersection of a countable family of dense open subsets. The space $C^\infty(\mathbb{R}^n, \mathbb{R}^{n-1})$ is a Baire space [Dem2000, Proposition 3.9.3], that is, every residual subset of $C^\infty(\mathbb{R}^n, \mathbb{R}^{n-1})$ is dense.

Definition 2.2.12. *A property is generic in $C^\infty(\mathbb{R}^n, \mathbb{R}^{n-1})$ if it is satisfied by a residual subset.*

Definition 2.2.13 ([Dem2000, §3.8.3]). *Let $E \simeq \mathbb{R}^n$ and F be two finite-dimensional real vector spaces and let $r \geq 0$ be an integer. Let $P^r(E, F)$ be the vector space of polynomial functions of degree at most r from E to F . For an open subset U of E (with respect to the usual topology on E), let $J^r(U, F) = U \times P^r(E, F)$ be the space of jets of order r of functions from U to F . Notice that $J^r(U, F)$ can be identified with an open subset of \mathbb{R}^N for some positive integer N . Let $f : U \rightarrow F$ be a smooth function, the jet of order r of f is the function*

$$j^r f : U \subset \mathbb{R}^n \rightarrow J^r(U, F) \subseteq \mathbb{R}^N$$

$$x \mapsto \left(x, f(x), \frac{\partial f}{\partial x_1}(x), \dots, \frac{\partial f}{\partial x_n}(x), \frac{\partial^2 f}{\partial x_1 \partial x_2}(x), \dots, \frac{\partial^r f}{\partial x_n^r}(x) \right).$$

Let W be a sub-manifold of $J^r(U, F)$. We say that $j^r f$ is transverse to W if for all $a \in U$ either $j^r f(a) \notin W$ or every vector of \mathbb{R}^N can be written as a sum of a vector of $T_{j^r f(a)}W$ and a vector in the image of the function $T_a j^r f$, where $T_{j^r f(a)}W$ is the tangent space of W at $j^r f(a)$ and $T_a j^r f$ is the derivative function of $j^r f$ at a .

Theorem 2.2.14 (Thom's Transversality Theorem [Dem2000, Theorem 3.9.4]). *Let E and F be two finite-dimensional vector spaces with U an open set in E .*

Let $r \geq 0$ be an integer and W be a sub-manifold of $J^r(U, F)$. Then, the set of functions $f \in C^\infty(U, F)$ such that $j^r f$ is transverse to W is a dense residual subset of $C^\infty(U, F)$.

Proposition 2.2.15 ([Dem2000, Corollary 3.7.3]). *Let U be an open subset of \mathbb{R}^n , $N \geq 1$ be an integer and W be a sub-manifold of the vector space \mathbb{R}^N of pure co-dimension m . Assume that the smooth function $g : U \rightarrow \mathbb{R}^N$ is transverse to W , then $g^{-1}(W)$ is a (possibly empty) sub-manifold of dimension $n - m$.*

The idea of our proofs of genericity of our assumptions is to express each of them as a system of equations in the jet space. When this system defines a manifold, Thom's theorem applies directly to pull back the manifold from the jet space to the ambient space of the curve and obtain the subset where the assumption is satisfied together with its dimension according to Proposition 2.2.15. A difficulty occurs when the system does not define a manifold. The following corollary overcomes this difficulty in the special case where the system is defined by analytic functions, in other words the system defines an analytic variety. Such a variety does not need to be a manifold but, using the Whitney stratification theorem [Whi1965], the variety is written as a union of manifolds on which Thom's theorem is then applied.

Corollary 2.2.16. *Let E and F be two finite-dimensional vector spaces with E of dimension n and U an open set in E . Let $r \geq 0$ be an integer and W be an analytic variety of $J^r(U, F)$ with co-dimension larger than n , then for a generic $P \in C^\infty(U, F)$, the pre-image of W under $j^r P$ is empty.*

Proof. Let $W = \bigcup_{i=1}^m W_i$ be a Whitney stratification of W , where the W_i 's are sub-manifolds. Since $\text{codim}(W) > n$, we have that $\text{codim}(W_i) > n$ for any integer $1 \leq i \leq m$. Let $\Gamma_i = \{P \in C^\infty(U, F) \mid j^r P \text{ is transverse to } W_i\}$ and $\Gamma = \bigcap_{i=1}^m \Gamma_i$. By Theorem 2.2.14, Γ_i is residual and so is Γ . Moreover, by Proposition 2.2.15, for $P \in \Gamma_i$ the pre-image of W_i under $j^r P$ is empty. Hence, $(j^r P)^{-1}(W) = \bigcup_{i=1}^m (j^r P)^{-1}(W_i) = \emptyset$. \square

Lemma 2.2.17. [GG1973, Lemma II.4.3] *Let X, Y be smooth manifolds, $W \subset Y$ a sub-manifold, and $f : X \rightarrow Y$ be a smooth function. Let $q \in X$ and $f(q) \in W$. Suppose there is an open subset U of $f(q)$ in Y and a submersion $\phi : U \rightarrow \mathbb{R}^k$ ($k = \text{codim}(W)$) such that $W \cap U = \phi^{-1}(0)$. Then, f is transversal to W at q iff $\phi \circ f$ is submersion at q .*

We will also need a refined version of Thom's theorem in a multijet setting, that is for several points in the source space simultaneously. We give the formal def-

initions of the multijet space and function but we do not restate Theorem 2.2.14, Proposition 2.2.15 and Corollary 2.2.16 that also hold for multijets.

Definition 2.2.18 ([Dem2000, §3.9.6]). *Let U be an open subset of \mathbb{R}^n and $k \geq 1$ be an integer. We denote $\Delta_{(k)}(U)$ the subset of U^k consisting of sequences (a_1, \dots, a_k) of pairwise distinct points of U . For an integer $r \geq 0$ and a finite dimensional space F , the k -multijet space of order r , $J_{(k)}^r(U, F)$, is the subset of $J^r(U, F)^k = (U \times P^r(E, F))^k$ consisting of the k -tuples $((a_1, p_1), \dots, (a_k, p_k))$, with $(a_1, \dots, a_k) \in \Delta_{(k)}(U)$. Let $f : U \rightarrow F$ be a smooth function, the k -multijet of order r of f is the function*

$$\begin{aligned} j_{(k)}^r f : \Delta_{(k)}(U) &\rightarrow J_{(k)}^r(U, F) \\ (a_1, \dots, a_k) &\mapsto (j^r f(a_1), \dots, j^r f(a_k)). \end{aligned}$$

Theorem 2.2.19 (Multijet Transversality Theorem [Dem2000, Theorem 3.9.7]). *Let E and F be two finite dimensional vector spaces and let U be an open subset of E . Let $r \geq 0$ and $k \geq 1$ be two integers and W be a sub-manifold of $J_{(k)}^r(U, F)$. The set of $f \in C^\infty(U, F)$ such that $j_{(k)}^r f$ is transverse to W is a dense residual subset of $C^\infty(U, F)$.*

3

Generic space curve

Contents

| | |
|---|-----------|
| 3.1 Notation and assumptions | 38 |
| 3.1.1 Notation | 38 |
| 3.1.2 Assumptions | 38 |
| 3.2 Genericity of the assumptions | 39 |
| 3.2.1 Genericity of the assumptions for a curve in \mathbb{R}^n | 41 |
| 3.2.2 Genericity of the assumptions for the silhouette of a surface in \mathbb{R}^n | 49 |

The goal of this thesis is to design certified numerical methods to isolate the singularities of plane curves. Numerical algorithms usually cannot handle singularities. However, under some assumptions, we succeed to handle in a certified way some, although not all, singularities of projections of smooth curves defined in \mathbb{R}^n .

In this chapter, we introduce a set of assumptions on a curve \mathcal{C} in \mathbb{R}^n , with $n > 2$ (Section 3.1.2). We study, under these assumptions, the singularities of the plane projection of \mathcal{C} classifying them according to Arnold's list [AGZV2012]. This classification is later used to show a correspondence between the singularities of the plane projection of \mathcal{C} and the solutions of a regular and square system in \mathbb{R}^{2n-1} introduced in Chapter 4. In section 3.2.1, we prove that the assumptions mentioned above are satisfied by a generic curve in \mathbb{R}^n . In addition, in Section 3.2.2, we prove that the silhouette of a generic surface in \mathbb{R}^n satisfies a part of these assumptions. we conclude the chapter by a conjecture that the silhouette of a generic surface satisfies the rest of the assumptions.

3.1 Notation and assumptions

3.1.1 Notation

For a positive integer n , a closed (resp. an open) n -box is the Cartesian product of n closed (resp. open) interval. Assume that $n \geq 3$ and let B be an open n -box and \overline{B} be the topological closure of B with respect to the usual topology in \mathbb{R}^n . Let $C^\infty(\mathbb{R}^n, \mathbb{R}^{n-1})$ denote the set of smooth functions (i.e., differentiable infinitely many times) from \mathbb{R}^n to \mathbb{R}^{n-1} . Consider the function $P = (P_1, \dots, P_{n-1}) \in C^\infty(\mathbb{R}^n, \mathbb{R}^{n-1})$. We will denote by \mathfrak{C} (resp. $\overline{\mathfrak{C}}$) the solution set of the system $\{P_1(x) = \dots = P_{n-1}(x) = 0\}$, with $x = (x_1, \dots, x_n) \in B$ (resp. with $x \in \overline{B}$). Also, consider the projection $\pi_{\mathfrak{C}}$ (resp. $\pi_{\overline{\mathfrak{C}}}$) from \mathfrak{C} (resp. $\overline{\mathfrak{C}}$) to the (x_1, x_2) -plane. Unless otherwise stated, the plane projection of a point $x \in \mathbb{R}^n$ is (x_1, x_2) . If $\overline{\mathfrak{C}}$ is a smooth curve (see the definition below), define \mathfrak{L}_c (resp. \mathfrak{L}'_c) to be the set of points q in \mathfrak{C} (resp. $\overline{\mathfrak{C}}$) such that the tangent line, denoted by $T_q\mathfrak{C}$, (resp. $T_q\overline{\mathfrak{C}}$) is orthogonal to the (x_1, x_2) -plane. We also define the set \mathfrak{L}_n (resp. \mathfrak{L}'_n) to be the set of points q in \mathfrak{C} (resp. $\overline{\mathfrak{C}}$) such that the cardinality of the pre-image of $\pi_{\mathfrak{C}}(q)$ under $\pi_{\mathfrak{C}}$ (resp. $\pi_{\overline{\mathfrak{C}}}$) is at least two. We will see later that, under some generic assumption, \mathfrak{L}_c (resp. \mathfrak{L}_n) is equal to the set of points in \mathfrak{C} that project to a cusp (resp. node) which justifies the subscript c (resp. n).

3.1.2 Assumptions

Recall that our goal is to design a numerical algorithm for isolating the singularities that appear in the plane projection of a curve \mathfrak{C} in \mathbb{R}^n . Numerical algorithms usually cannot handle degenerate cases, that is, singularities in our context. However, under some assumptions on \mathfrak{C} , we succeed to isolate in a certified way some singularities of the projection. Namely, we require that the singularities are “generic”, that is, only nodes can appear in the projection (Assumption \mathcal{A}_5). Our other assumptions on \mathfrak{C} are, roughly speaking, that it is smooth (\mathcal{A}_1), that its projection only has a discrete set of singularities (\mathcal{A}_4), and that at most two points of \mathfrak{C} project on each singularity (\mathcal{A}_3). We will prove in Chapter 5 that our numerical algorithm is certified and terminates under these assumptions and in Section 3.2 that these assumptions are generically satisfied.

Recall that we denote by J_P the Jacobian matrix of the function P in $C^\infty(\mathbb{R}^n, \mathbb{R}^{n-1})$. Consider the following assumptions:

\mathcal{A}_1 – For all $q \in \overline{\mathfrak{C}}$, $\text{rank}(J_P(q)) = n - 1$. In particular, $\overline{\mathfrak{C}}$ is a smooth curve.⁵

⁵Note that the converse is not true as the vertical (double) line defined by $x_1^2 = x_2 = 0$ in \mathbb{R}^3 is smooth but the rank of its Jacobian is never full.

- \mathcal{A}_2 – The set $\mathcal{L}'_{\mathcal{C}}$ is discrete and does not intersect the boundary of B .
- \mathcal{A}_3 – For all points $p = (\alpha, \beta) \in \pi_{\overline{\mathcal{C}}}(\overline{\mathcal{C}})$, the pre-image of p under $\pi_{\overline{\mathcal{C}}}$ consists of at most two points in \overline{B} counted with multiplicities in the system $\{P(x) = 0 \in \mathbb{R}^{n-1}, x_1 - \alpha = x_2 - \beta = 0\}$.
- \mathcal{A}_4 – The set \mathcal{L}'_n is discrete and does not intersect the boundary of B .
- \mathcal{A}_5 – The singular points of $\pi_{\mathcal{C}}(\mathcal{C})$ are only nodes (see Definition 2.2.4).

We also consider a weaker version of Assumption \mathcal{A}_5 in which ordinary cusps can also appear in the projection:

- \mathcal{A}_5^- – The singular points of $\pi_{\mathcal{C}}(\mathcal{C})$ are only ordinary cusps or nodes (see Definition 2.2.4).

Definition 3.1.1. *Assumptions \mathcal{A}_{1-5} are called the strong assumptions and Assumptions \mathcal{A}_{1-4} and \mathcal{A}_5^- are called the weak assumptions.*

Figure 3.1 illustrates the assumptions that we have just introduced. Our goal in the next section is to prove that a generic curve \mathcal{C} satisfies our assumptions. Loosely speaking, the genericity implies that a “random” curve satisfies these assumptions. Moreover, if a curve does not satisfy the assumptions, there exists a sequence of generic curves that is converging to that curve.

Remark 3.1.2. *Our motivation for also considering these weak assumptions is dual. First, our certified algorithm for isolating the singularities of the projection of curves satisfying the strong assumptions also works, to some extent, if only the weak assumptions hold: namely, it outputs a superset of the isolating boxes of the singularities. Second, we conjecture that our weak assumptions are satisfied by silhouette curves of generic surfaces (see Proposition 3.2.17 and Conjecture 3.2.18).*

3.2 Genericity of the assumptions

Recall from the introduction (namely in Section 1.2.2) that a property in $C^\infty(\mathbb{R}^n, \mathbb{R}^{n-2})$ is generic if it is satisfied by all elements of a residual subset (a countable intersection of open dense subsets) in $C^\infty(\mathbb{R}^n, \mathbb{R}^{n-2})$. The key to prove the genericity of our assumptions mentioned on Section 3.1.2 is Thom’s Transversality Theorem 2.2.14. We mentioned, in Section 2.2.4, the basics of transversality theory using the notation of Demazure’s book [Dem2000]. We prove, in Section 3.2.1, that all assumptions of Section 3.1.2 are satisfied for a generic curve. In Section 3.2.2, we consider the special case where the curve is the silhouette of

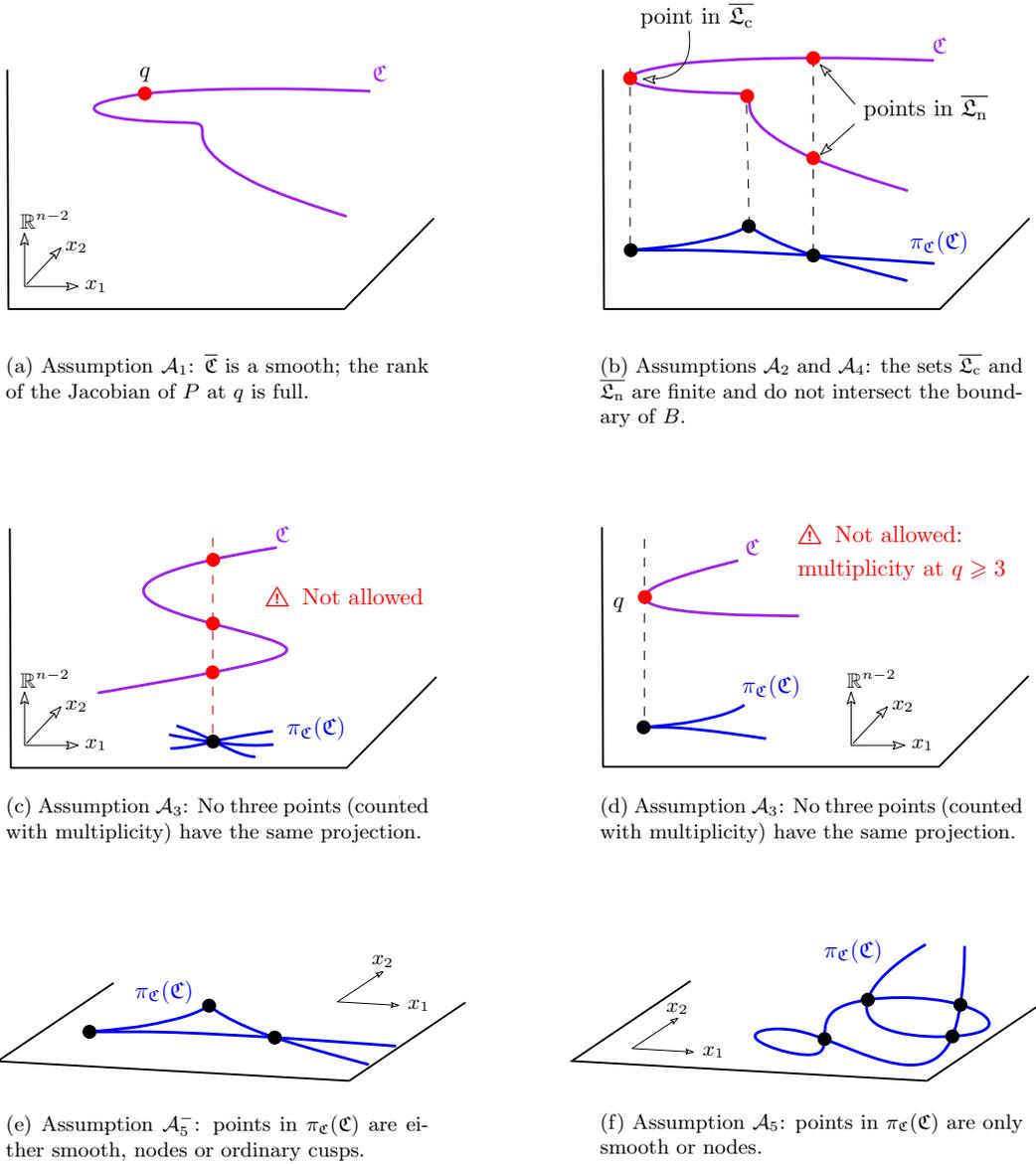


Figure 3.1: Illustration of the assumptions.

a surface and prove that Assumptions \mathcal{A}_1 , \mathcal{A}_2 , \mathcal{A}_4 are generically satisfied in this case; the genericity of Assumptions \mathcal{A}_3 and \mathcal{A}_5^- is left as a conjecture.

3.2.1 Genericity of the assumptions for a curve in \mathbb{R}^n

We are going to prove that each assumption in Section 3.1 is generic. Hence, the combination of these assumptions is also generic since a countable intersection of residual subsets in $C^\infty(\mathbb{R}^n, \mathbb{R}^{n-1})$ is residual.

Lemma 3.2.1. *Assumption \mathcal{A}_1 is generic.*

Proof. Consider the jet of order 1 of the function $P \in C^\infty(\mathbb{R}^n, \mathbb{R}^{n-1})$:

$$\begin{aligned} j^1 P : \mathbb{R}^n &\rightarrow J^1(\mathbb{R}^n, \mathbb{R}^{n-1}) = \mathbb{R}^n \times \mathbb{R}^{n-1} \times \mathbb{R}^{(n-1) \times n} \\ x &\mapsto (x, P(x), J_P(x)) = (x, y, z). \end{aligned}$$

We represent the jet space by the variables $x \in \mathbb{R}^n$, $y \in \mathbb{R}^{n-1}$ and $z \in \mathbb{R}^{(n-1) \times n}$. With abuse of notation, we can see the variable z as a $(n-1) \times n$ -matrix. Define the variety $W = \{(x, y, z) \in \mathbb{R}^n \times \mathbb{R}^{n-1} \times \mathbb{R}^{(n-1) \times n} \mid y = 0, \text{rank}(z) \leq n-2\}$. The variety W is a product of a determinantal variety in $\mathbb{R}^{(n-1) \times n}$ of dimension $n^2 - n - 2$ (by Proposition 2.1.4) and a linear space of dimension n in $\mathbb{R}^n \times \mathbb{R}^{n-1}$. Thus, W is a variety of co-dimension $n+1$ in $\mathbb{R}^n \times \mathbb{R}^{n-1} \times \mathbb{R}^{(n-1) \times n}$. Hence, by Corollary 2.2.16, there exists a residual subset $\Gamma_1 \subset C^\infty(\mathbb{R}^n, \mathbb{R}^{n-1})$, such that for $P \in \Gamma_1$ the pre-image of W under $j^1 P$ is empty. Consequently, for a generic $P \in C^\infty(\mathbb{R}^n, \mathbb{R}^{n-1})$ and any $q \in \bar{\mathcal{C}}$, we have that $q \notin (j^1 P)^{-1}(W) = \emptyset$, thus $\text{rank}(J_P(q)) = n-1$, which is Assumption \mathcal{A}_1 . \square

The following lemma proves a stronger claim than the genericity of Assumption \mathcal{A}_2 , namely, that generically, \mathcal{L}'_c is empty. However, as mentioned before, being \mathcal{L}'_c discrete, is the case for the Silhouette of a generic surface in \mathbb{R}^n (more precisely, generic P in $C^\infty(\mathbb{R}^n, \mathbb{R}^{n-2})$) which justify why we choose the weaker assumption.

Lemma 3.2.2. *Assumption \mathcal{A}_2 is generic. Moreover, generically, the set \mathcal{L}'_c is empty.*

Proof. We consider the jet of order 1 of the function $P \in C^\infty(\mathbb{R}^n, \mathbb{R}^{n-1})$ as is the proof of Lemma 3.2.1 with the same notation. Define the matrix $T_1(z)$ (resp. $T_2(z)$) to be the sub-matrix of z obtained by removing the first (resp. second) column. Consider the variety $W \subset J^1(\mathbb{R}^n, \mathbb{R}^{n-1})$ defined by $\{y = 0 \in \mathbb{R}^{n-1}, \det(T_1(z)) = \det(T_2(z)) = 0\}$. Notice that \mathcal{L}'_c is included in the pre-image of W under $j^1 P$ since \mathcal{L}'_c is the set of points of the curve $\bar{\mathcal{C}}$ that are both x_1 and x_2 -critical. By

Lemma 2.1.5, we have that both $\det(T_1(z))$ and $\det(T_2(z))$ are irreducible polynomials. By [CLO2007, §9.4 Prop 10], a proper sub-variety of an irreducible variety is of lower dimension, we deduce that the common zero locus of $\det(T_1(z))$ and $\det(T_2(z))$ is of co-dimension at least two. We deduce that $\text{codim}(W) > n$. By Corollary 2.2.16, there exists a residual subset $\Gamma_2 \subset C^\infty(\mathbb{R}^n, \mathbb{R}^{n-1})$, such that for $P \in \Gamma_2 \cap \Gamma_1$, the pre-image of W under $j^1 P$ is empty and hence \mathcal{L}'_c is empty, which implies Assumption \mathcal{A}_2 . \square

The following lemma provides a characterization of the points in \mathcal{L}'_c (under Assumption \mathcal{A}_1) in terms of multiplicity. Such an equivalence is useful to ease proofs and reduce technicalities.

Lemma 3.2.3. *Let $P = (P_1 \dots, P_{n-1}) \in C^\infty(\mathbb{R}^n, \mathbb{R}^{n-1})$ satisfy Assumption \mathcal{A}_1 . Let q be in $\bar{\mathcal{C}}$ such that the multiplicity of the system $S = \{P(x) = 0 \in \mathbb{R}^{n-1}, x_1 - \alpha = x_2 - \beta = 0\}$ at q is finite, where $(\alpha, \beta) = \pi_{\bar{\mathcal{C}}}(q) \in \mathbb{R}^2$. Then, $q \in \mathcal{L}'_c$ if and only if the multiplicity of the system S at q is at least two.*

Proof. Without loss of generality assume that $q = 0 \in \mathbb{R}^n$.

Sufficiency: Assume that $q \in \mathcal{L}'_c$. Let $v = (v_1, \dots, v_n)$ be a non-trivial vector of the tangent line of $\bar{\mathcal{C}}$ at q . Thus, $J_P(q) \cdot v^T = 0$. By the definition of \mathcal{L}'_c we have $v_1 = v_2 = 0$. Define the differential operator $c = \sum_{i=3}^n v_i \frac{\partial}{\partial x_i}$. Notice that $c \cdot P_j = \sum_{i=3}^n v_i \frac{\partial P_j}{\partial x_i}(q) = 0$ for all integers $1 \leq j \leq n-1$ (see [DLZ2011, 2.1] for the definition of $c \cdot P_j$). Moreover, by the definition of c and since $v_1 = v_2 = 0$, we have $c \cdot (x_1) = c \cdot (x_2) = 0$. Hence, $c \in D_q^1[S] \setminus D_q^0[S]$. Thus, $\dim(D_q^1) > 1$. Hence, the multiplicity of S at q is at least two.

Necessity: Assume that the multiplicity of S at q is at least two, then $D_q^0[S] \subsetneq D_q^1[S]$. This implies that there exists a non-trivial differential operator $c = \sum_{i=1}^n c_i \frac{\partial}{\partial x_i} \in D_q^1[S] \setminus D_q^0[S]$ such that:

- (a) We have that $c \cdot P_j = 0$ for all integers $1 \leq j \leq n-1$ which implies that if we write $v_i = c_i$, with $1 \leq i \leq n$, the non-trivial vector v is in the tangent space of $\bar{\mathcal{C}}$ at q .
- (b) We have that $c \cdot (x_1) = c \cdot (x_2) = 0$, equivalently, $c_1 = c_2 = 0$. Thus, $v_1 = v_2 = 0$.

The tangent line to the curve at q is thus orthogonal to the (x_1, x_2) -plane. Thus, $q \in \mathcal{L}'_c$. \square

Lemma 3.2.4. *Assumption \mathcal{A}_3 is generic.*

Proof. Let us consider the 3-multijet of order 0:

$$j_{(3)}^0 P : \Delta_{(3)}(\mathbb{R}^n) \rightarrow J_{(3)}^0(\mathbb{R}^n, \mathbb{R}^{n-1}) = (\mathbb{R}^n \times \mathbb{R}^{n-1})^3$$

$$(x, x', x'') \mapsto ((x, P(x)), (x', P(x')), (x'', P(x''))) = ((x, y), (x', y'), (x'', y''))$$

where every element in the jet space $J_{(3)}^0(\mathbb{R}^n, \mathbb{R}^{n-1})$ is of the form $((x, y), (x', y'), (x'', y''))$, where $x = (x_1, \dots, x_n)$, $x', x'' \in \mathbb{R}^n$ and $y, y', y'' \in \mathbb{R}^{n-1}$. Consider the linear sub-manifold $W = \{x_1 = x'_1 = x''_1, x_2 = x'_2 = x''_2, y = y' = y'' = 0\}$, the co-dimension of W is thus $3n + 1$ which is larger than the dimension of the source space $\Delta_{(3)}(\mathbb{R}^n)$ which is $3n$. Thus, by Corollary 2.2.16, there exists a residual subset $\Gamma_3 \subset C^\infty(\mathbb{R}^n, \mathbb{R}^{n-1})$, such that for $P \in \Gamma_3$, the pre-image of W by $j_{(3)}^0$ is empty, which translates to the fact that there are no pairwise distinct points q, q', q'' in $\bar{\mathcal{C}}$ such that $\pi_{\bar{\mathcal{C}}}(q) = \pi_{\bar{\mathcal{C}}}(q') = \pi_{\bar{\mathcal{C}}}(q'')$. This is also equivalent to say that the system $S = \{P(x) = 0 \in \mathbb{R}^{n-1}, x_1 - \alpha = x_2 - \beta = 0\}$ has at most two distinct solutions (without counting multiplicities) for any $(\alpha, \beta) \in \mathbb{R}^2$.

Using Γ_1, Γ_2 as defined in the proofs of Lemmas 3.2.1 & 3.2.2 and Γ_3 defined above, we define $\Gamma_4 = \Gamma_1 \cap \Gamma_2 \cap \Gamma_3$ which is thus a residual set and let P be in Γ_4 . Since P is in Γ_3 , the system S has at most two distinct solutions. In addition, since P is in $\Gamma_2 \cap \Gamma_1$, one has that \mathcal{L}'_c is empty and finally together with Lemma 3.2.3, since P is in Γ_1 , this implies that these solutions have multiplicity exactly 1 is S . For P in the residual set Γ_4 , the number of solutions counted with multiplicities of S is thus at most 2, which is Assumption \mathcal{A}_3 . \square

Lemma 3.2.5. *Assumption \mathcal{A}_4 is generic.*

Proof. Let us consider the 2-multijet of order 0 of P :

$$j_{(2)}^0 P : \Delta_{(2)}(\mathbb{R}^n) \rightarrow J_{(2)}^0(\mathbb{R}^n, \mathbb{R}^{n-1}) = (\mathbb{R}^n \times \mathbb{R}^{n-1})^2$$

$$(x, x') \mapsto ((x, P(x)), (x', P(x'))) = ((x, y), (x', y'))$$

where every element in the jet space $J_{(2)}^0(\mathbb{R}^n, \mathbb{R}^{n-1})$ is of the form $((x, y), (x', y'))$, where $x = (x_1, \dots, x_n)$, $x' \in \mathbb{R}^n$ and $y, y' \in \mathbb{R}^{n-1}$. Consider the linear sub-manifold $W = \{x_1 = x'_1, x_2 = x'_2, y = y' = 0\}$ of the jet space $J_{(2)}^0(\mathbb{R}^n, \mathbb{R}^{n-1})$. Notice that, $(j_{(2)}^0 P)^{-1}(W)$ contains the set $\widehat{\mathcal{L}}'_n = \{(q_1, q_2) \in \Delta_{(2)}(\mathbb{R}^n) \cap \bar{\mathcal{C}} \times \bar{\mathcal{C}} \mid \pi_{\bar{\mathcal{C}}}(q_1) = \pi_{\bar{\mathcal{C}}}(q_2)\}$ and \mathcal{L}'_n is the image of $\widehat{\mathcal{L}}'_n$ by the projection $(q_1, q_2) \rightarrow q_1$. We have $\dim(\Delta_{(2)}(\mathbb{R}^n)) = 2n$ and, since W is linear, its co-dimension is easily computed $\text{codim}(W) = 2(2n - 1) - (2 + 2(n - 1)) = 2n$. Proposition 2.2.15 thus yields that generically $(j_{(2)}^0 P)^{-1}(W)$ is a sub-manifold of dimension zero that is a discrete set in \mathbb{R}^n , and so is \mathcal{L}'_n .

Now, we prove that, generically, \mathcal{L}'_n does not intersect the boundary of B . The boundary ∂B of the box B is included in the union of the supporting hyperplanes

H_i of its 2^n faces of dimension $n - 1$, that is $\partial B = \cup_{i=1}^{2^n} H_i$. Define the linear sub-manifold $W_i = \{((x, y), (x', y')) \in W \mid x \in H_i \text{ or } x' \in H_i\}$, notice that this adds one equation to W and thus increases the co-dimension of W by one, thus $\text{codim}(W_i) = 2n + 1$. By Corollary 2.2.16, we have that, generically, the pre-image of W_i under $j_{(2)}^0 P$ is empty, which translates to the fact that there is no point of \mathcal{L}'_n on $\partial B \cap H_i$. This is also true for any i and thus, generically, \mathcal{L}'_n does not intersect the boundary of B . \square

Proving the genericity of Assumptions \mathcal{A}_5 and \mathcal{A}_5^- is a bit more complicated. We first study the singularity types that occur on the plane curve $\pi_{\mathcal{C}}(\mathcal{C})$ under Assumptions $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$ and \mathcal{A}_4 in Lemmas 3.2.6 and 3.2.7. Then, we translate their results into algebra (Lemma 3.2.9) which paves the way to use transversality theorem in proving the genericity of Assumptions \mathcal{A}_5 and \mathcal{A}_5^- in Corollary 3.2.10.

Lemma 3.2.6. *Under Assumptions $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$ and \mathcal{A}_4 , let $q \in \mathcal{C}$ and $p = \pi_{\mathcal{C}}(q)$. If $q \notin \mathcal{L}_c \cup \mathcal{L}_n$, then p is a smooth point of the plane curve $\pi_{\mathcal{C}}(\mathcal{C})$.*

Proof. As illustrated in Figure 3.2, since $q \notin \mathcal{L}_c$, the plane projection of $T_q \mathcal{C}$ is a line, or equivalently, the derivative $T_q \pi_{\mathcal{C}}$ of $\pi_{\mathcal{C}}$ at q is injective. Thus, $\pi_{\mathcal{C}}$ is an immersion at q . Hence, for a small enough neighbourhood U_0 of q in \mathbb{R}^n , we have that $\pi_{\mathcal{C}}$ restricted to $V = U_0 \cap \mathcal{C}$ is embedding (Proposition 2.2.7). We are going to prove that, assuming that U_0 is small enough, the curve $\pi_{\mathcal{C}}(\mathcal{C})$ has exactly one branch around $\pi_{\mathcal{C}}(q)$ which implies that $\pi_{\mathcal{C}}(\mathcal{C})$ is smooth at $\pi_{\mathcal{C}}(q)$ since \mathcal{C} is smooth at q by Assumption \mathcal{A}_1 .

To prove this claim, assume that there exists an open subset U'_0 in \mathbb{R}^n such that the set $V' = U'_0 \cap \mathcal{C}$ and V are disjoint, but $\pi_{\mathcal{C}}(q)$ is in the closure of $\pi_{\mathcal{C}}(V')$. Let q_k be a sequence of points in V' such that $\pi_{\mathcal{C}}(q_k)$ converges to $\pi_{\mathcal{C}}(q)$. Since \overline{B} is compact, there exists a convergent sub-sequence of q_k that has a limit q' in \overline{B} . Notice that $\pi_{\mathcal{C}}(q') = \pi_{\mathcal{C}}(q)$ by the continuity of $\pi_{\mathcal{C}}$. Hence, q, q' are both in \mathcal{L}'_n . However, since $q \notin \mathcal{L}_n$, we must have that $q' \notin B$. Hence, q' is in the boundary of B which contradicts Assumption \mathcal{A}_4 . Hence, the curve $\pi_{\mathcal{C}}(\mathcal{C})$ has exactly one smooth branch around $\pi_{\mathcal{C}}(q)$ which concludes the proof. \square

Regarding our goal about the singularities of $\pi_{\mathcal{C}}(\mathcal{C})$, by Lemma 3.2.6, we can concentrate our attention to the plane projections of the points in $\mathcal{L}_c \cup \mathcal{L}_n$.

Lemma 3.2.7. *Under Assumptions $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$ and \mathcal{A}_4 , if $q \in \mathcal{L}_n$, then $\pi_{\mathcal{C}}(q)$ is a singular point of the plane curve $\pi_{\mathcal{C}}(\mathcal{C})$. More precisely, either $\pi_{\mathcal{C}}(q)$ is of type A_{2k+1}^- with $k \geq 0$, or there exists a non-null smooth function g defined in a neighbourhood of $0 \in \mathbb{R}$ with $\text{ord}(g) = \infty$ such that $(\pi_{\mathcal{C}}(\mathcal{C}), \pi_{\mathcal{C}}(q))$ is equivalent to the curve defined by $x^2 - g(y^2) = 0$ at the origin.*

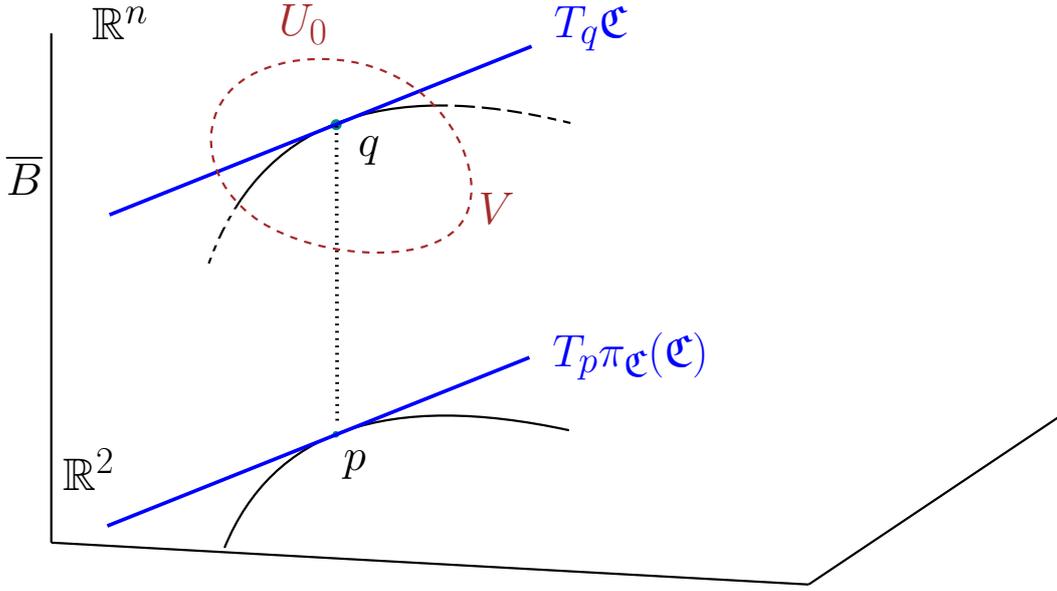


Figure 3.2: Illustration of the proof of Lemma 3.2.6.

Proof. As in Figure 3.4, let $p = \pi_{\mathcal{C}}(q)$, according to \mathcal{A}_3 , $\pi_{\mathcal{C}}^{-1}(p)$ has at most two points, and since q is in \mathcal{L}_n , it also has at least two points. Define q' such that $\pi_{\mathcal{C}}^{-1}(p) = \{q, q'\}$ and denote the plane curve $\pi_{\mathcal{C}}(\mathcal{C})$ by C . Without loss of generality, one can assume that $p = (0, 0)$. In addition, \mathcal{A}_3 also implies that the multiplicities of q and q' in the system $\{P(x) = 0 \in \mathbb{R}^{n-1}, x_1 = x_2 = 0\}$ are one. With Assumption \mathcal{A}_1 , Lemma 3.2.3 then implies that the tangents to \mathcal{C} at q and q' are not orthogonal to the (x_1, x_2) -plane. Thus there exists two neighborhoods N_q and $N_{q'}$ of q and q' in \mathbb{R}^n such that π restricted to $\mathcal{C} \cap N_q$ (resp. $\mathcal{C} \cap N_{q'}$) is an embedding. Let D_k be a sequence of open disks centered at p and of radius $\frac{1}{k}$. By contradiction, if for all k , there exists points $q_k \in \mathcal{C}$ such that q_k is not in $N_q \cup N_{q'}$ and $\pi_{\mathcal{C}}(q_k) \in D_k$, then the limit q_{∞} is a point of \mathcal{C} distinct from q and q' , and $\pi_{\mathcal{C}}(q_{\infty}) = p$. If q_{∞} is in B , it contradicts \mathcal{A}_3 and if it is in \overline{B} , it contradicts \mathcal{A}_4 . Thus for a small enough neighborhood of p , the projection of the curve is restricted to the projection of the two branches around q and q' . Finally, if for all D_k , the pre-image of $\pi^{-1}(D_k)$ contains a point in $\mathcal{L}'_n \setminus \{q, q'\}$, then this contradicts the discreteness assumption \mathcal{A}_4 . Thus there exists a neighborhood $N \subseteq \mathbb{R}^2$ of p such that $\pi_{\mathcal{C}}^{-1}(N)$ is a union of two smooth (Assumption \mathcal{A}_1) open subsets of \mathcal{C} such that q is on one branch and q' on the other, and $\pi_{\mathcal{C}}$ restricted to $\pi_{\mathcal{C}}^{-1}(N) \setminus \{p, q'\}$ is an embedding. The projection of these two smooth branches are thus two smooth curves in the plane. Let these two smooth plane branches be defined by the zero sets of the smooth functions f_1 and f_2 in $C^{\infty}(\mathbb{R}^2, \mathbb{R})$. Let u (resp. u') be a non-zero tangent vector of \mathcal{C} at q (resp. q') and v (resp. v') be its projection in \mathbb{R}^2 . We distinguish two cases:

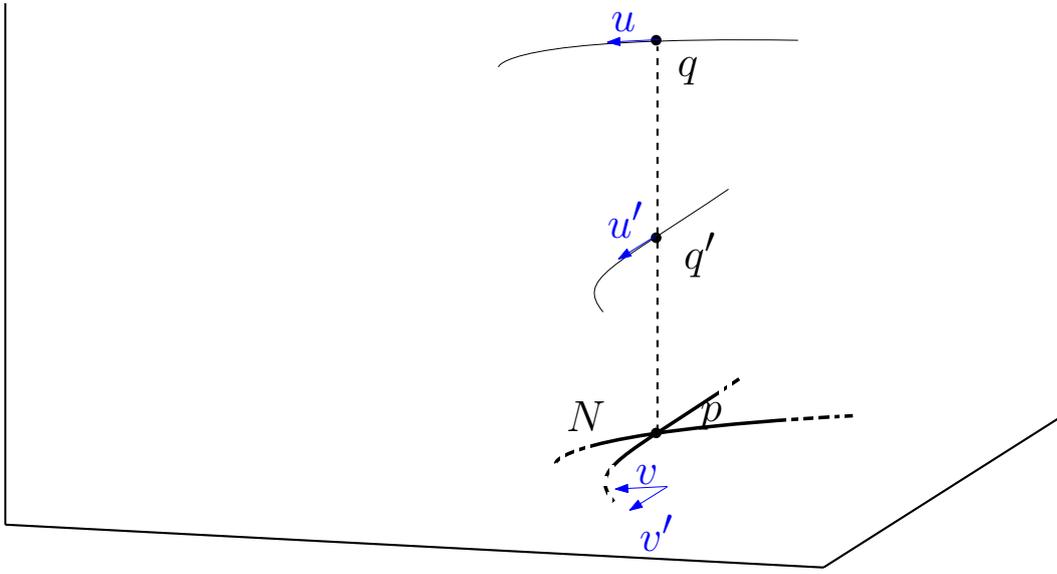


Figure 3.3: Proof of Lemma 3.2.7: The first case where v and v' are independent.

(a) The vectors v and v' are independent in \mathbb{R}^2 (see Figure 3.3). Thus, v and v' give rise to a local coordinate system (x, y) in a neighborhood of p in \mathbb{R}^2 . The vector v being tangent to the zero set of f_1 , one has $\frac{\partial f_1}{\partial x}(p) = 0$ and $\frac{\partial f_1}{\partial y}(p) \neq 0$. By the implicit function theorem [Dem2000, Corollary 2.7.3.], we deduce that there exists a real smooth function h_1 such that $y = x^2 \cdot h_1(x)$ is a local parameterization of the zero set of f_1 . Similarly, there exists a smooth function h_2 such that $x = y^2 \cdot h_2(y)$ is a local parameterization of the zero set of f_2 . Thus $(x, y) \in N$ iff $f(x, y) = f_1(x, y)f_2(x, y) = 0$ iff $(y - x^2 \cdot h_1(x))(x - y^2 \cdot h_2(y)) = 0$, equivalently, $[y - x - x^2 \cdot h_1(x) + y^2 \cdot h_2(y)]^2 - [y + x - x^2 \cdot h_1(x) - y^2 \cdot h_2(y)]^2 = 0$. The change of coordinates $X = y - x + x^2 \cdot h_1(x) + y^2 \cdot h_2(y)$ and $Y = y + x + x^2 \cdot h_1(x) - y^2 \cdot h_2(y)$ is a diffeomorphism since $\det(J_{x,y}(X, Y))_p \neq 0$. Then, the local equation of the curve C at p is of the form $X^2 - Y^2$ with these new coordinates, which means that p is a A_1^- or node singularity.

(b) The vectors v and v' are co-linear (see Figure 3.4). Then, choose $v'' \in T_p\mathbb{R}^2$ linearly independent from v , the vectors v, v'' give rise to a coordinate system (x, y) at p . In this coordinate system, we thus have $\frac{\partial f_1}{\partial x}(p) = \frac{\partial f_2}{\partial x}(p) = 0$, $\frac{\partial f_1}{\partial y}(p) \neq 0$ and $\frac{\partial f_2}{\partial y}(p) \neq 0$. By the implicit function theorem, there exist smooth functions h_1 and h_2 such that locally $f(x, y) = 0$ if and only if $(y - x^2 \cdot h_1(x))(y - x^2 \cdot h_2(x)) = 0$. The last equality is equivalent to $(2y - x^2(h_1(x) + h_2(x)))^2 - x^4(h_1(x) - h_2(x))^2 = 0$. Assumption \mathcal{A}_4 ensures that the projections of the two branches have only one common point, such that

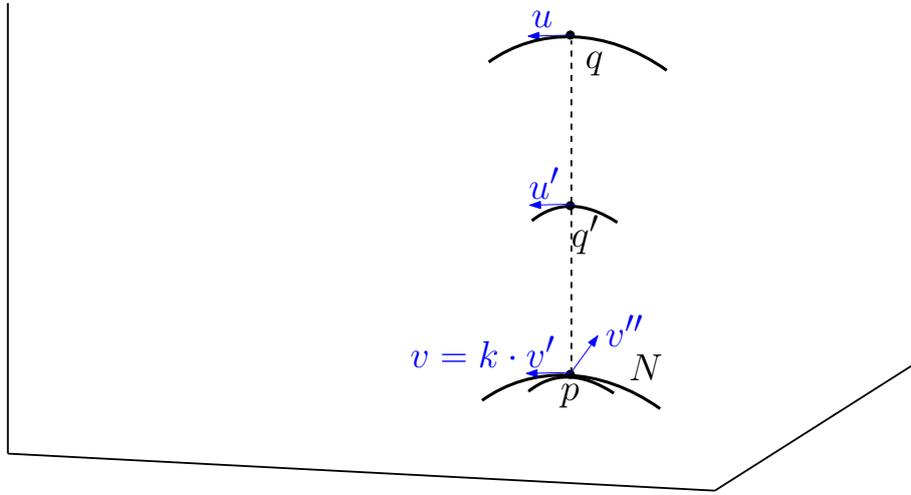


Figure 3.4: Proof of Lemma 4.3.3: The second case where v and v' are co-linear.

$h_1(x) - h_2(x)$ does not vanish identically. We distinguish two cases:

- (i) $\text{mult}(h_1(x) - h_2(x)) = k \leq \infty$, then $h_1(x) - h_2(x) = x^k \cdot u$ with $u(p) \neq 0$ and without loss of generality, assume that $u(p) > 0$. The change of coordinates $X = 2y - x^2(h_1(x) + h_2(x))$ and $Y = x \cdot u^{\frac{1}{2+k}}$ is a diffeomorphism (notice that indeed $u^{\frac{1}{2+k}}$ is a smooth function around p). Then, the local equation of the curve C at p is of the form $X^2 - Y^{(2k+3)+1}$ with these new coordinates, which means that p is a singularity of type A_{2k+3}^- .
- (ii) $\text{mult}(h_1(x) - h_2(x)) = \infty$. Since the function $x^4(h_1(x) - h_2(x))^2$ is even, by Theorem 2.2.2, there exists a smooth function g such that $x^4(h_1(x) - h_2(x))^2 = g(x^2)$. Thus, taking the diffeomorphism $X = 2y - x^2(h_1(x) + h_2(x))$ and $Y = x$, we get the second case of the claim.

□

The next definition and lemma are technical tools for proving the genericity of Assumptions \mathcal{A}_5^- and \mathcal{A}_5 . We want to translate the result of Lemma 3.2.7 to the language of algebra. The following definition is needed for this characterization:

Definition 3.2.8. Consider $P = (P_1, \dots, P_{n-1}) \in C^\infty(\mathbb{R}^n, \mathbb{R}^{n-1})$ satisfying Assumption \mathcal{A}_1 and recall that we denote by $J_P(q)$ the Jacobian matrix of P at the point q . We define the $(n-1) \times (n-2)$ sub-matrix $M_{1,2}(q)$ obtained by removing the first two columns of $J_P(q)$ and the $(n-1) \times 2$ sub-matrix $N_P(q)$ formed by the first two columns of $J_P(q)$. Let $q_1, q_2 \in \mathfrak{C}$, we define the square matrix of size

$$2n - 2, M(q_1, q_2) = \begin{pmatrix} N_P(q_1) & 0 & M_{1,2}(q_1) \\ N_P(q_2) & M_{1,2}(q_2) & 0 \end{pmatrix}.$$

Lemma 3.2.9. *Using the same assumption and notation as in Definition 3.2.8, let q_1 and q_2 be distinct points of \mathfrak{C} with $\pi_{\mathfrak{C}}(q_1) = \pi_{\mathfrak{C}}(q_2)$, then $M(q_1, q_2)$ is invertible if and only if none of q_1 or q_2 is in \mathfrak{L}_c and the plane projections of the tangent lines of \mathfrak{C} at q_1 and q_2 do not coincide.*

Proof. We prove the converse statement using

$$\begin{aligned} \det(M(q_1, q_2)) = 0 &\iff \text{There exist } \alpha \in \mathbb{R}^2 \text{ and } \beta, \gamma \in \mathbb{R}^{n-2} \text{ such that the} \\ &\text{vector } x = (\alpha, \beta, \gamma) \text{ is not trivial and satisfies the} \\ &\text{equality } M(q_1, q_2) \cdot x^T = 0. \\ &\iff (\alpha, \beta) \text{ and } (\alpha, \gamma) \text{ are in the tangent lines } T_{q_1} \mathfrak{C} \text{ and} \\ &T_{q_2} \mathfrak{C} \text{ respectively and at least one of them is not} \\ &\text{trivial.} \end{aligned}$$

The last statement can be split in two cases:

- α is not trivial which is equivalent to say that the plane projections of $T_{q_1} \mathfrak{C}$ and $T_{q_2} \mathfrak{C}$ are both generated by α and coincide.
- $\alpha = (0, 0)$ which is equivalent to β or γ is not trivial, which is equivalent to $T_{q_2} \mathfrak{C}$ or $T_{q_1} \mathfrak{C}$ projects to a point in the plane, which is equivalent to q_1 or q_2 is in \mathfrak{L}_c . \square

Now, we are ready to prove the genericity of Assumptions \mathcal{A}_5^- and \mathcal{A}_5^+ :

Corollary 3.2.10. *Assumption \mathcal{A}_5 is generic. Consequently, so is Assumption \mathcal{A}_5^- .*

Proof. Let B be an open n -box. Recall that generically \mathfrak{L}'_c (and hence \mathfrak{L}_c) is empty (Lemma 3.2.2). Hence, it is enough to prove that for a generic $P \in C^\infty(\mathbb{R}^n, \mathbb{R}^{n-1})$, the singular points of $\pi_{\mathfrak{C}}(\mathfrak{C})$ are only nodes (recall that by Lemma 3.2.6, under the generic assumptions $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$ and \mathcal{A}_4 , the points in $\mathfrak{C} \setminus (\mathfrak{L}_c \cup \mathfrak{L}_n)$ project to smooth points).

Let Γ_0 be the set of $P \in C^\infty(\mathbb{R}^n, \mathbb{R}^{n-1})$ such that P satisfies Assumptions $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$ and \mathcal{A}_4 . The previous lemmas of this section show that Γ_0 is residual in $C^\infty(\mathbb{R}^n, \mathbb{R}^{n-1})$.

Let us consider the 2-multijet of order 1 of P :

$$\begin{aligned} j_{(2)}^1 P : \Delta_{(2)}(\mathbb{R}^n) &\rightarrow J_{(2)}^1(\mathbb{R}^n, \mathbb{R}^{n-1}) \subseteq (\mathbb{R}^n \times \mathbb{R}^{n-1} \times \mathbb{R}^{(n-1) \times n})^2 \\ (x, x') &\mapsto ((x, P(x), J_P(x)), (x', P(x'), J_P(x'))) = ((x, y, z), (x', y', z')) \end{aligned}$$

Let s, s' (resp. r, r') be the sub-matrices of z, z' respectively obtained by removing the first two columns (resp. obtained by the first two columns). Define the matrix

$$M = \begin{pmatrix} r & 0 & s \\ r' & s' & 0 \end{pmatrix} \text{ and the variety}$$

$$W = \{((x, y, z), (x', y', z')) \in (\mathbb{R}^n \times \mathbb{R}^{n-1} \times \mathbb{R}^{(n-1) \times n})^2 \mid y = y' = 0, x_1 = x'_1, x_2 = x'_2, \det(M) = 0\}.$$

The variety W is a product of a determinantal variety and a linear space, thus its co-dimension is $\text{codim}(W) \geq 2n + 1 > 2n = \dim(\Delta_{(2)}(\mathbb{R}^n))$. Hence, by Corollary 2.2.16, there exists a residual subset Γ'_0 in $C^\infty(\mathbb{R}^n, \mathbb{R}^{n-1})$ such that for all $P \in \Gamma'_0$, the pre-image of W under $j_{(2)}^1 P$ is empty.

Let then P be in the residual set $\Gamma_0 \cap \Gamma'_0$. By Lemma 3.2.9 and since \mathcal{L}_c is empty, we deduce that for distinct $q_1, q_2 \in \mathcal{C}$ with $\pi_{\mathcal{C}}(q_1) = \pi_{\mathcal{C}}(q_2)$, the plane projections of the lines $T_{q_1} \mathcal{C}$ and $T_{q_2} \mathcal{C}$ intersect transversely if and only if $j_{(2)}^1((q_1, q_2)) \notin W$. Finally, by Lemma 3.2.7 (Step (a) of the proof), we deduce that $\pi_{\mathcal{C}}(q_1) = \pi_{\mathcal{C}}(q_2)$ is a node in $\pi_{\mathcal{C}}(\mathcal{C})$. \square

3.2.2 Genericity of the assumptions for the silhouette of a surface in \mathbb{R}^n

In this section, we focus on the special case of silhouette curves of surfaces in \mathbb{R}^n . For an open n -box B and \tilde{P} in $C^\infty(\mathbb{R}^n, \mathbb{R}^{n-2})$ such that $S = \tilde{P}^{-1}(0)$ is a smooth 2-sub-manifold in \mathbb{R}^n , the silhouette of \tilde{P} is the set of points q of this surface S such that the projection (with respect to a fixed direction) of the tangent plane $T_q S$ to \mathbb{R}^2 is not surjective (see Figure 3.5). We prove that Assumptions $\mathcal{A}_1, \mathcal{A}_2$ & \mathcal{A}_4 are satisfied for a generic silhouette, and we only conjecture that Assumptions \mathcal{A}_3 & \mathcal{A}_5^- also hold generically. We start by formalizing algebraically the definition of the silhouette curve.

Definition 3.2.11. For an integer $n \geq 3$, let $\tilde{P} = (P_1, \dots, P_{n-2}) \in C^\infty(\mathbb{R}^n, \mathbb{R}^{n-2})$. Define the smooth function $P_{n-1} = \det \left(\left(\frac{\partial P_i}{\partial x_j} \right)_{\substack{1 \leq i \leq n-2 \\ 3 \leq j \leq n}} \right)$ and $P = (P_1, \dots, P_{n-1})$. We define the curve \mathcal{C} (and $\bar{\mathcal{C}}$) as in Section 3.1 and call it the silhouette of \tilde{P} .

As mentioned before, we want now to prove that the silhouette of a generic surface in \mathbb{R}^n , Assumptions $\mathcal{A}_1, \mathcal{A}_2$ and \mathcal{A}_4 are satisfied. The argument used for the proofs are more involved and complicated comparing with the ones presented in Section 3.2.1. For Assumption \mathcal{A}_1 , the basic idea of the proof of Lemma 3.2.12 is Lemma 2.2.17.

Proposition 3.2.12. For a generic $\tilde{P} \in C^\infty(\mathbb{R}^n, \mathbb{R}^{n-2})$, the function P satisfies Assumption \mathcal{A}_1 .

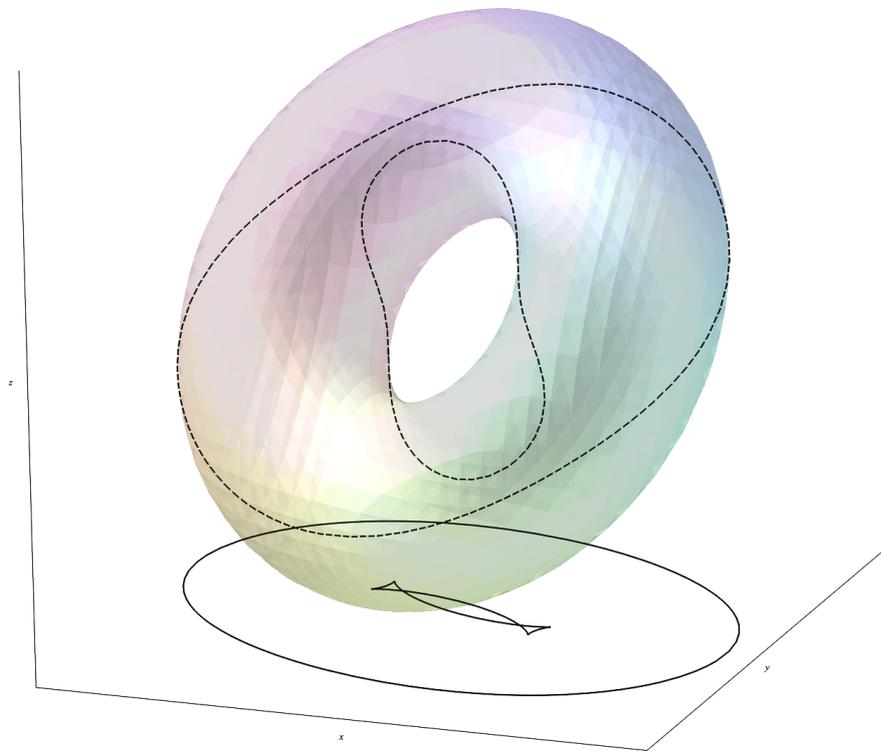


Figure 3.5: Example: The silhouette of the torus in \mathbb{R}^3 with the coordinates (x, y, z) (source: [IMP2016a]) .

Proof. Consider the jet of order 1 of \tilde{P} :

$$\begin{aligned} j^1\tilde{P} : \mathbb{R}^n &\rightarrow J^1(\mathbb{R}^n, \mathbb{R}^{n-2}) = \mathbb{R}^n \times \mathbb{R}^{n-2} \times \mathbb{R}^{(n-2)\times n} \simeq \mathbb{R}^{n^2-2} = \mathbb{R}^N \\ x &\mapsto (x, \tilde{P}(x), J_{\tilde{P}}(x)) = (x, y, z). \end{aligned}$$

We represent the jet space by the vectors $x \in \mathbb{R}^n$, $y \in \mathbb{R}^{n-2}$ and the $((n-2) \times n)$ -matrix $z \in \mathbb{R}^{(n-2)\times n}$. Let $T(z)$ denote the sub-matrix obtained by removing the first two columns of z . Define the variety $W = \{y = 0, \det(T(z)) = 0\} = \{y = 0, \text{rank}(T(z)) \leq n-3\}$ in \mathbb{R}^N . According to Proposition 2.1.4, $W = \text{Reg}(W) \cup \text{Sing}(W)$ where $\text{Reg}(W)$ (resp. $\text{Sing}(W)$) is the set of smooth (resp. singular) points in W and

$$\text{Reg}(W) = \{(x, y, z) \in \mathbb{R}^N \mid y = 0, \text{rank}(T(z)) = n-3\}$$

$$\text{Sing}(W) = \{(x, y, z) \in \mathbb{R}^N \mid y = 0, \text{rank}(T(z)) < n-3\}.$$

In addition, Proposition 2.1.4 yields that $\text{Reg}(W)$ is a manifold of co-dimension $n-1$ and $\text{Sing}(W)$ is a variety of co-dimension $n+2$. Since the co-dimension of $\text{Sing}(W)$ is larger than that of the source space, Corollary 2.2.16 implies that, generically, $(j^1\tilde{P})^{-1}(\text{Sing}(W)) = \emptyset$. One thus has $(j^1\tilde{P})^{-1}(W) = (j^1\tilde{P})^{-1}(\text{Reg}(W))$. Consider the function

$$\begin{aligned} \varphi : \mathbb{R}^n \times \mathbb{R}^{n-2} \times \mathbb{R}^{(n-2)\times n} &\rightarrow \mathbb{R}^{n-2} \times \mathbb{R} \\ \chi = (x, y, z) &\mapsto (y, \det(T(z))), \end{aligned}$$

such that $\varphi^{-1}(0) = W$. Its Jacobian matrix is $J_\varphi = \begin{pmatrix} 0_{(n-2)\times n} & I_{(n-2)\times(n-2)} & 0_{(n-2)\times(n-2)n} \\ 0_{1\times(n)} & 0_{1\times(n-2)} & v(z) \end{pmatrix}$, where $0_{k_1\times k_2}$ (resp. $I_{k_1\times k_2}$) is the zero (resp. identity) matrix of size $k_1 \times k_2$ and the vector $v(z)$ is the adjugate matrix of $T(z)$ written as the concatenation of its lines: $v(z) = (\text{Adj}^{ij}(T(z)))_{\substack{1 \leq i \leq n-2 \\ 3 \leq j \leq n}} \in \mathbb{R}^{(n-2)^2}$. Let $\chi = (x, y, z) \in \text{Reg}(W)$, then $\text{rank}(T(z)) = n-3$, thus there exists a pair (i, j) such that $\text{Adj}^{ij}(T(z)) \neq 0$. Hence, the vector $v(z)$ is non-trivial and $J_\varphi(\chi)$ has full rank $n-1$. The function φ is thus a submersion (Definition 2.2.6) on $\text{Reg}(W)$. Theorem 2.2.14 yields that, generically, $j^1\tilde{P}$ is transverse to the manifold $\text{Reg}(W)$. As illustrated in Figure 3.6, since φ is a submersion on $\text{Reg}(W)$, by Lemma 2.2.17, we imply that that $P = \varphi \circ j^1\tilde{P}$ is a submersion on $(j^1\tilde{P})^{-1}(\text{Reg}(W)) = (j^1\tilde{P})^{-1}(W) = (j^1\tilde{P})^{-1}(\varphi^{-1}(0)) = (\varphi \circ j^1\tilde{P})^{-1}(0) = P^{-1}(0) = \mathfrak{C}$. In other words, J_P has full rank $n-1$ on \mathfrak{C} , which is Assumption \mathcal{A}_1 . \square

Proposition 3.2.13. *For a generic $\tilde{P} \in C^\infty(\mathbb{R}^n, \mathbb{R}^{n-2})$, the function P satisfies Assumption \mathcal{A}_2 .*

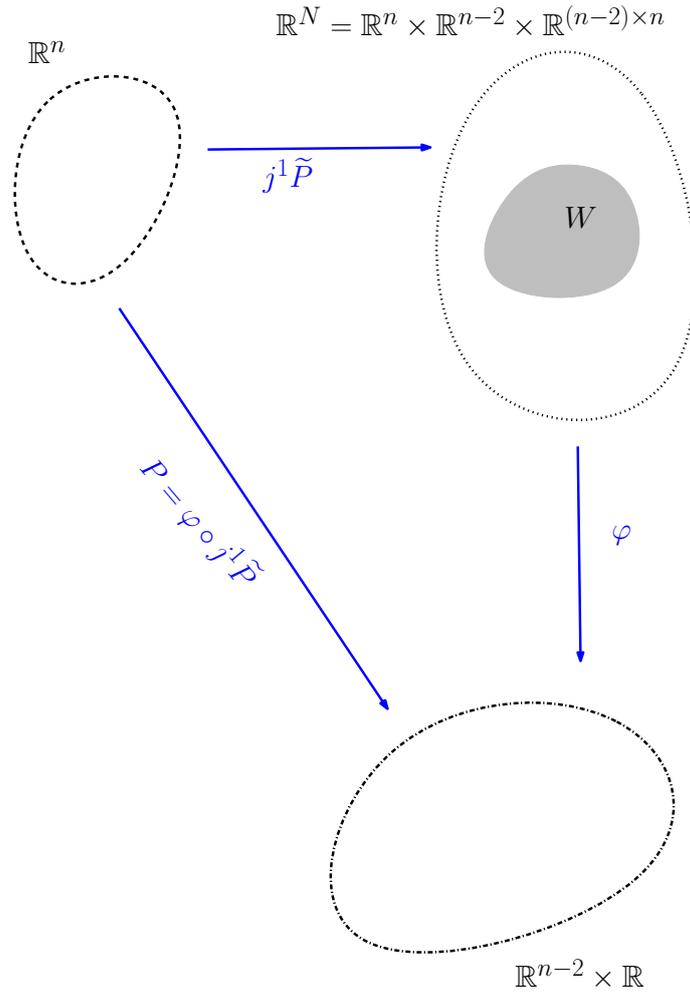


Figure 3.6: Illustration of Lemma 2.2.17 which is used in the proof of Proposition 3.2.12. Since φ is a submersion over $Reg(W)$ and since $j^1\tilde{P}$ is, generically, transversal to $Reg(W)$, the lemma implies that the composition $\tilde{P} = \varphi \circ j^1\tilde{P}$ is generically a submersion.

Proof. First we prove that, generically, \mathcal{L}'_c is discrete. For any $\tilde{P} \in C^\infty(\mathbb{R}^n, \mathbb{R}^{n-2})$ consider $j^2\tilde{P} : \mathbb{R}^n \rightarrow J^2(\mathbb{R}^n, \mathbb{R}^{n-2}) \subset \mathbb{R}^n \times \mathbb{R}^{n-2} \times \mathbb{R}^{(n-2) \times n} \times \mathbb{R}^{n^2(n-2)} = \mathbb{R}^N$. Assume that every element in \mathbb{R}^N is represented as (x, y, z, h) , where $x \in \mathbb{R}^n$, $y \in \mathbb{R}^{n-2}$, $z \in \mathbb{R}^{(n-2) \times n}$ and $h \in \mathbb{R}^{n^2(n-2)}$. With abuse of notation we can consider z as a $((n-2) \times n)$ -matrix. Let $T(z)$ denote the matrix obtained by removing the first two columns of z . The Jacobian matrix J_P is a function of the derivatives $(\frac{\partial P_i}{\partial x_j}, \frac{\partial^2 P_l}{\partial x_k \partial x_s})_{\substack{1 \leq i, l \leq n-2 \\ 1 \leq j, k, s \leq n}}$, it can thus be seen in the jet space as a function of z and h , $J_P(z, h)$. Define the matrix $T_1(z, h)$ (resp. $T_2(z, h)$) to be the submatrix of $J_P(z, h)$ obtained by removing the first (resp. second) column. Define the variety $W = \{(x, y, z, h) \mid y = 0 \in \mathbb{R}^{n-2}, \det(T(z)) = \det(T_1(z, h)) =$

$\det(T_2(z, h)) = 0\}$, so that \mathcal{L}'_c is included in the pre-image of W under $j^2\tilde{P}$. Let $W_1 = \{(x, y, z, h) \mid y = 0 \in \mathbb{R}^{n-2}, \det(T(z)) = 0\}$, we already showed in the proof of Proposition 3.2.12 that W_1 is an irreducible variety of co-dimension $n - 1$. In addition, $\det(T_1(z, h))$ does not identically vanish on W_1 , thus W is a proper sub-variety of the irreducible variety W_1 and [CLO2007, §9.4 Prop 10] implies that $\text{codim}(W) > \text{codim}(W_1) = n - 1$.

Now, write $W = \text{Reg}(W) \cup \text{Sing}(W)$, where $\text{Reg}(W)$ (resp. $\text{Sing}(W)$) is the set of smooth (resp. singular) points in W . Recall that $\text{codim}(\text{Sing}(W)) > n$ since $\text{Sing}(W)$ is a proper closed sub-variety of W [BCR1998, Proposition 3.3.14]. By Corollary 2.2.16, there exists a residual set $\Gamma' \subset C^\infty(\mathbb{R}^n, \mathbb{R}^{n-2})$ such that if $\tilde{P} \in \Gamma'$, then the pre-image of $\text{Sing}(W)$ under $j^2\tilde{P}$ is empty. Define $\Gamma = \{\tilde{P} \in C^\infty(\mathbb{R}^n, \mathbb{R}^{n-2}) \mid j^2\tilde{P} \text{ is transverse to } \text{Reg}(W)\} \cap \Gamma'$. Notice that if $\tilde{P} \in \Gamma$, then \mathcal{L}'_c is included in the pre-image of $\text{Reg}(W)$ under $j^2\tilde{P}$. Hence, since $\text{codim}(\text{Reg}(W)) = \text{codim}(W) \geq n$, we have by Proposition 2.2.15 that \mathcal{L}'_c is a sub-manifold of dimension, at most, zero. Thus, \mathcal{L}'_c is discrete for all $\tilde{P} \in \Gamma$. Using Theorem 2.2.14 we deduce that Γ is residual.

The proof that \mathcal{L}'_c does not intersect the boundary of B can be done analogously as in the proof of Lemma 3.2.5. \square

Proposition 3.2.14. *For a generic $\tilde{P} \in C^\infty(\mathbb{R}^n, \mathbb{R}^{n-2})$, the function P satisfies Assumption \mathcal{A}_4 .*

Proof. Consider the 2-multijet $j_{(2)}^1\tilde{P} : \Delta_{(2)}(\mathbb{R}^n) \rightarrow J_{(2)}^1(\mathbb{R}^n, \mathbb{R}^{n-2}) = (\mathbb{R}^n \times \mathbb{R}^{n-2} \times \mathbb{R}^{(n-2) \times n})^2$ of the function $\tilde{P} \in C^\infty(\mathbb{R}^n, \mathbb{R}^{n-2})$, where $(\mathbb{R}^n \times \mathbb{R}^{(n-2) \times n} \times \mathbb{R}^{(n-2) \times n})^2$ is described by the coordinates $x, x' \in \mathbb{R}^n, y, y' \in \mathbb{R}^{n-2}$ and $z, z' \in \mathbb{R}^{(n-2) \times n}$. With abuse of notation we can consider z and z' as $((n-2) \times n)$ -matrices. Let $T(z)$ (resp. $T(z')$) denote the matrix obtained by removing the first two columns of z (resp. z'). Define the variety W to be the solution set of the system $\{y = y' = 0, x_1 - x'_1 = x_2 - x'_2 = \det(T(z)) = \det(T(z')) = 0\}$. Denote by $\text{Reg}(W)$ the regular part of W . By Proposition 2.1.4 (a) we deduce that W is of co-dimension $2n$. Using the same argument in the proof of Proposition 3.2.12, we deduce that there exists a residual set $\Gamma \subset C^\infty(\mathbb{R}^n, \mathbb{R}^{n-2})$ such that if $\tilde{P} \in \Gamma$, then the image of $\Delta_2(\mathbb{R}^n)$ under $j_{(2)}^1\tilde{P}$ is contained in $\text{Reg}(W)$. Moreover, by Proposition 2.2.15, we have that $M_{1,2} = (j_{(2)}^1\tilde{P})^{-1}(\text{Reg}(W)) = (j_{(2)}^1\tilde{P})^{-1}(W)$ is a sub-manifold of dimension zero in $\Delta_2(\mathbb{R}^n)$. Notice that \mathcal{L}'_n is the image of $M_{1,2}$ under the projection $(x, x') \rightarrow x$. Since $M_{1,2}$ is of dimension zero, then so is \mathcal{L}'_n . Thus we have just proven that, if $\tilde{P} \in \Gamma$, then \mathcal{L}'_n is a sub-manifold of dimension zero. Hence, \mathcal{L}'_n is discrete.

The proof that \mathcal{L}'_n does not intersect the boundary of B can be done analogously as in the proof of Lemma 3.2.5. \square

Assumption \mathcal{A}_3 can be rephrased by the three following assumptions:

$\mathcal{A}_{3(a)}$ - There are no pairwise distinct $q, q', q'' \in \bar{\mathcal{C}}$ such that $\pi_{\mathcal{C}}(q) = \pi_{\mathcal{C}}(q') = \pi_{\mathcal{C}}(q'')$.

$\mathcal{A}_{3(b)}$ - $\mathcal{L}'_c \cap \mathcal{L}'_n = \emptyset$.

$\mathcal{A}_{3(c)}$ - For $q \in \mathcal{L}'_c$, the multiplicity of the system $\{P(x) = 0 \in \mathbb{R}^{n-1}, (x_1, x_2) = \pi_{\mathcal{C}}(q)\}$ at q is exactly two.

Using this rephrasing, we next show that Assumptions $\mathcal{A}_{3(a)}$ & $\mathcal{A}_{3(b)}$ generically hold and we leave the genericity of Assumption $\mathcal{A}_{3(c)}$ as a conjecture.

Proposition 3.2.15. *For a generic function $\tilde{P} \in C^\infty(\mathbb{R}^n, \mathbb{R}^{n-2})$, Assumption $\mathcal{A}_{3(a)}$ - holds.*

Proof. Consider the 3-multijet $j_{(3)}^1 \tilde{P} : \Delta_{(3)}(\mathbb{R}^n) \rightarrow J_{(3)}^1(\mathbb{R}^n, \mathbb{R}^{n-2}) = (\mathbb{R}^n \times \mathbb{R}^{n-2} \times \mathbb{R}^{(n-2) \times n})^3$. Assume that every element in $(\mathbb{R}^n \times \mathbb{R}^{n-2} \times \mathbb{R}^{(n-2) \times n})^3$ is of the form $((x, y, z), (x', y', z'), (x'', y'', z''))$, where $x, x', x'' \in \mathbb{R}^n$, $y, y', y'' \in \mathbb{R}^{n-2}$ and $z, z', z'' \in \mathbb{R}^{(n-2) \times n}$. With abuse of notation we can consider z, z' and z'' as $((n-2) \times n)$ -matrices. Let $T(z), T(z'), T(z'')$ denote the matrices obtained by removing the first two columns of z, z', z'' respectively. Consider the variety W defined by the equations: $\{x_1 = x'_1 = x''_1, x_2 = x'_2 = x''_2, y = y' = y'' = 0 \in \mathbb{R}^{n-2}, \det(T(z)) = \det(T(z')) = \det(T(z'')) = 0\}$.

Notice that $\dim(\Delta_{(3)}(\mathbb{R}^n)) = 3n < 3n + 1 = \text{codim}(W)$. Hence, by Corollary 2.2.16, we have that, generically, the pre-image of W under $j_{(3)}^1 \tilde{P}$ is empty. Hence, there are no pairwise different $q, q', q'' \in \bar{\mathcal{C}}$ such that $\pi_{\mathcal{C}}(q) = \pi_{\mathcal{C}}(q') = \pi_{\mathcal{C}}(q'')$. \square

Proposition 3.2.16. *For a generic function $\tilde{P} \in C^\infty(\mathbb{R}^n, \mathbb{R}^{n-2})$, Assumption $\mathcal{A}_{3(b)}$ - holds.*

Proof. Consider the 2-multijet $j_{(2)}^2 \tilde{P} : \Delta_{(2)}(\mathbb{R}^n) \rightarrow J_{(2)}^2(\mathbb{R}^n, \mathbb{R}^{n-2}) = (\mathbb{R}^n \times \mathbb{R}^{n-2} \times \mathbb{R}^{(n-2) \times n} \times \mathbb{R}^{n^2(n-2)})^2$ of the function $\tilde{P} \in C^\infty(\mathbb{R}^n, \mathbb{R}^{n-2})$, where $(\mathbb{R}^n \times \mathbb{R}^{(n-2) \times n} \times \mathbb{R}^{(n-2) \times n} \times \mathbb{R}^{n^2(n-2)})^2$ is described by the coordinates $x, x' \in \mathbb{R}^n$, $y, y' \in \mathbb{R}^{n-2}$, $z, z' \in \mathbb{R}^{(n-2) \times n}$ and $h, h' \in \mathbb{R}^{n^2(n-2)}$. With abuse of notation we can consider z and z' as $((n-2) \times n)$ -matrices. Let $T(z)$ (resp. $T(z')$) denote the matrix obtained by removing the first two columns of z (resp. z'). Define the matrices $T_1(z, h), T_2(z, h)$ as in the proof of Lemma 3.2.13 and the variety W to be the solution set of the system $\{y = y' = 0 \in \mathbb{R}^{n-2}, x_1 - x'_1 = x_2 - x'_2 = \det(T(z)) = \det(T(z')) = 0, \det(T_1(z, h)) = \det(T_2(z, h)) = 0\}$.

Define varieties $W' = \{(x, y, z, h) \mid y = y' = 0, \det(T(z)) = \det(T(z')) = 0, x_1 = x'_1, x_2 = x'_2\}$ and $W'' = \{(x, y, z, h) \mid y = y' = 0, \det(T_1(z, h)) = \det(T_2(z, h)) = 0\}$. Notice that $W = W' \cap W''$. Moreover, we can find a smooth silhouette curve C that is not an orthogonal line to (x_1, x_2) -plane and that contains two distinct points q, q' , with $\pi_{\mathcal{C}}(q) = \pi_{\mathcal{C}}(q')$ such that the projection of $T_q C$ (resp. $T_{q'} C$) onto \mathbb{R}^2 is injective. Notice that $j_{(2)}^2 \tilde{P}(q, q') \in W' \setminus W''$. Hence, $W' \not\subseteq W''$. Moreover, since W' is the Cartesian product of determinant varieties (which are irreducible by Proposition 2.1.4(a)) with linear spaces, we have that W' is also irreducible [BCR1998, Theorem 2.8.3 (iii)]. In other words, $W = W' \cap W''$ is a proper sub-variety of the irreducible variety W' . Hence, $\dim(W) = \dim(W' \cap W'') < \dim(W')$, equivalently, $\text{codim}(W) > \text{codim}(W') = 2n$. Hence, by Corollary 2.2.16 we have that, generically, the pre-image of W under $j_{(2)}^2 \tilde{P}$ is empty. Since, by Proposition 3.2.12, Assumption \mathcal{A}_1 (which is necessary to guarantee that $\mathcal{L}'_{\mathcal{C}}$ is well-defined) is also generic, we imply that, generically, there is no distinct pair $q, q' \in \mathcal{C}$ such that $\pi_{\mathcal{C}}(q) = \pi_{\mathcal{C}}(q')$ and $q \in \mathcal{L}'_{\mathcal{C}}$, equivalently, $\mathcal{L}'_{\mathcal{C}} \cap \mathcal{L}'_{\mathcal{C}} = \emptyset$ which proves the proposition. \square

We thus proved the following proposition that the silhouette of a generic surface in \mathbb{R}^n satisfies all assumptions except for Assumptions $\mathcal{A}_{3(c)}$ and \mathcal{A}_{5-} that we could not prove. However, based on previous results with three variables [IMP2016b], we formulate the following conjecture.

Proposition 3.2.17. *For a generic function $\tilde{P} \in C^\infty(\mathbb{R}^n, \mathbb{R}^{n-2})$, Assumptions $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_{3(a)}, \mathcal{A}_{3(b)}$ and \mathcal{A}_4 hold.*

Conjecture 3.2.18. *For a generic function $\tilde{P} \in C^\infty(\mathbb{R}^n, \mathbb{R}^{n-2})$, Assumptions $\mathcal{A}_{3(c)}$ and \mathcal{A}_{5-} hold.*

4

Modelling system

Contents

| | |
|--|----|
| 4.1 Encoding the singular points of the plane projection | 57 |
| 4.2 Singularities induced by \mathcal{L}_n | 64 |
| 4.3 Singularities induced by \mathcal{L}_c | 64 |
| 4.4 Regularity of the Ball system | 73 |

Our goal in this chapter is, under our weak assumptions (Definition 3.1.1), to encode the singularities of $\pi_{\mathcal{C}}(\mathcal{C})$ by a square and regular (see Definition 4.4.1) system, called the Ball system, so that it is solvable with certified numerical methods.

In Section 4.1, we define this system $\text{Ball}(P)$. Then, we generically study the plane projection of points in \mathcal{L}_n in Section 4.2. In Section 4.3, we then locally parametrize the curve around the points in \mathcal{L}_c (Defined in Section 3.1.1) to simplify the computation of $\text{Ball}(P)$ and its Jacobian. In Section 4.4, we determine the necessary and sufficient conditions for this system to be regular.

4.1 Encoding the singular points of the plane projection

By our Assumption \mathcal{A}_5^- , the singularities of the projected curve $\pi_{\mathcal{C}}(\mathcal{C})$ are only nodes and cusps. Intuitively, a node appears when two points of \mathcal{C} project to the same point and a cusp appears when projecting a point with a tangent line orthogonal to the projection plane (see Figure 4.1). The idea to encode the nodes is to design a system whose variables are the coordinates of two different points in \mathbb{R}^n constrained to be on \mathcal{C} and so that they have the same plane projection. To encode a cusp, we design a system whose variables are the coordinates of one point in \mathbb{R}^n constrained to be on \mathcal{C} with a tangent orthogonal to the projection plane.

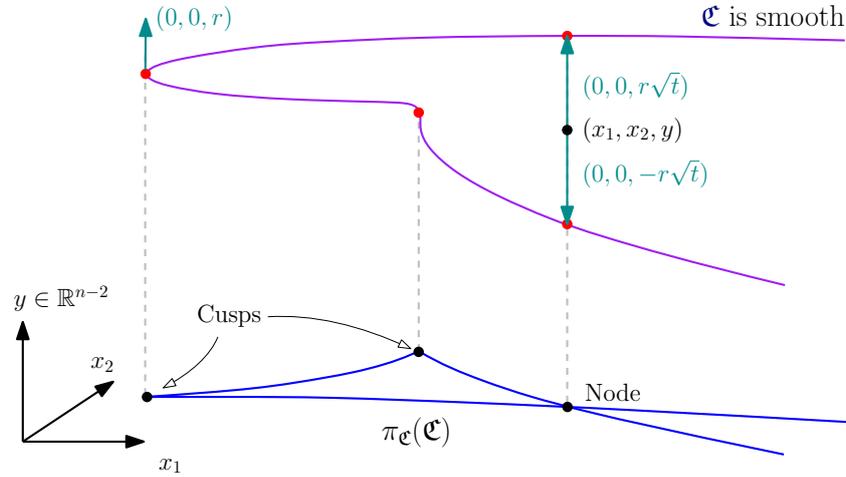


Figure 4.1: Illustration of a node and cusps in the plane projection of a smooth curve.

Furthermore, in order to apply certified numerical methods we need systems that are square and regular (Definition 4.4.1). To solve this issue and to gather the two systems into a single one, we first parameterize two different points of \mathcal{C} with the same projection by $(x_1, x_2, y + r\sqrt{t})$ and $(x_1, x_2, y - r\sqrt{t})$, with $x_1, x_2, t \in \mathbb{R}$, y, r in \mathbb{R}^{n-2} and $\|r\| = 1$, where $\|r\|$ denotes the Euclidean norm of r . Then, given any function f from \mathbb{R}^n to \mathbb{R} so that $f = 0$ is one of the $n - 1$ hypersurfaces that define \mathcal{C} , we introduce in Definition 4.1.1 two smooth functions $S \cdot f$ and $D \cdot f$. When $t > 0$, they return, roughly speaking, the arithmetic mean and difference of f at the above two points, hence they both vanish if and only if the two points are on the hypersurface $f = 0$. When $t = 0$, the two points coincide and $S \cdot f$ and $D \cdot f$ return, roughly speaking, f evaluated at this point and the gradient of f (at that point) scalar the “vertical” vector $(0, 0, r)$; hence, they both vanish if and only if the point is on the hypersurface $f = 0$ and its tangent hyperplane is normal to the plane of projection. It follows that given a curve defined by $P_1 = \dots = P_{n-1} = 0$, the solutions of the so-called Ball system of all $S \cdot P_i = D \cdot P_i = 0$ is the set of points on the curve that project to nodes and cusps (Theorem 4.1.3). Note that we consider \sqrt{t} instead of t in the parameterization $(x_1, x_2, y \pm r\sqrt{t})$ for ensuring the regularity of the Ball system when $t = 0$ (because this ensures that the linear term of the Taylor expansion of $D \cdot f$, with respect to t , does not vanish).

Definition 4.1.1. Let y, r be two variables in \mathbb{R}^{n-2} and t be a variable in \mathbb{R} . For a

smooth function $f : \bar{B} \subset \mathbb{R}^n \rightarrow \mathbb{R}$, we define the functions:

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

and

$$D \cdot f(x_1, x_2, y, r, t) = \begin{cases} \frac{1}{2\sqrt{t}}[f(x_1, x_2, y + r\sqrt{t}) - f(x_1, x_2, y - r\sqrt{t})], & \text{for } t > 0 \\ \nabla f(x_1, x_2, y) \cdot (0, 0, r) = \sum_{i=3}^n \frac{\partial f}{\partial x_i} r_i, & \text{for } t = 0. \end{cases}$$

Lemma 4.1.2. *If f is a smooth function defined on $\bar{B} \subseteq \mathbb{R}^n$, then both $S \cdot f$ and $D \cdot f$ are smooth functions on the subset*

$\bar{B}_{\text{Ball}} = \{(x_1, x_2, y, r, t) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}^{n-2} \times \mathbb{R}^{n-2} \times \mathbb{R} \mid t \geq 0, (x_1, x_2, y \pm r\sqrt{t}) \in \bar{B}, \|r\|^2 = 1\}$ of \mathbb{R}^{2n-1} , where $\|r\|$ denotes the Euclidean norm of r .

Proof. On the subset \bar{B}_{Ball} with $t > 0$, both $S \cdot f(x_1, x_2, y, r, t)$ and $D \cdot f(x_1, x_2, y, r, t)$ are the compositions of smooth functions, hence they are smooth functions.

For a point $X = (x_1, x_2, y, r, t)$ in B_{Ball} with $t = 0$, we will prove that $S \cdot f$ (resp. $D \cdot f$) is a C^s function for an arbitrarily s which implies that $S \cdot f$ (resp. $D \cdot f$) is smooth.

First define the function

$$S_0 \cdot f(x_1, x_2, y, r, t) = \begin{cases} \frac{1}{2}[f(x_1, x_2, y + rt) + f(x_1, x_2, y - rt)], & \text{for } t > 0 \\ f(x_1, x_2, y), & \text{for } t = 0. \end{cases}$$

Since $S_0 \cdot f(x_1, x_2, y, r, t)$ is an even smooth function with respect to t , the partial derivatives of $S_0 \cdot f$ with respect to t of odd orders, evaluated at X , are zero. For an integer $s > 0$, by the parametrized Taylor formula without remainder [Dem2000, Proposition 4.2.2], there exist smooth functions $a_i(x_1, x_2, y, r)$, with integers $0 \leq i < s$ such that $S_0 \cdot f(x_1, x_2, y, r, t) = \sum_{i=0}^{s-1} a_i(x_1, x_2, y, r)t^{2i} + t^{2s} \cdot \phi(x_1, x_2, y, t)$, where $\phi(x_1, x_2, y, t)$ is a smooth function. Notice that $S \cdot f(x_1, x_2, y, r, t) = \sum_{i=0}^{s-1} a_i(x_1, x_2, y, r)t^i + t^s \cdot \phi(x_1, x_2, y, \sqrt{t})$, so that a partial derivative exists up to order s at $t = 0$. Thus, $S \cdot f(x_1, x_2, y, r, t)$ is a C^{s-1} function. This holds for any arbitrarily large s , hence $S \cdot f(x_1, x_2, y, r, t)$ is a C^∞ function.

Now, we prove that $D \cdot f$ is continuous at $X = (x_1, x_2, y, r, 0)$. Let X_i be a sequence that converges to X . To prove that $D \cdot f(X_i)$ converges to $D \cdot f(X)$, it is enough to show that for a sequence t_i that converges to 0, then we have that $D \cdot f(x_1, x_2, y, r, t_n)$ converges to $D \cdot f(X)$. We can assume that $t_i \neq 0$ for all i , so that

$$\begin{aligned}
 \lim_{t_i \rightarrow 0} D \cdot f(x_1, x_2, y, r, t_i) &= \lim_{t_i \rightarrow 0} \frac{1}{2\sqrt{t_i}} [f(x_1, x_2, y + r\sqrt{t_i}) - f(x_1, x_2, y - r\sqrt{t_i})] \\
 &= \lim_{t_i \rightarrow 0} \frac{1}{2\sqrt{t_i}} [f(x_1, x_2, y + r\sqrt{t_i}) - (f(x_1, x_2, y) - \\
 &\quad f(x_1, x_2, y) - f(x_1, x_2, y - r\sqrt{t_i}))] \\
 &= \lim_{t_i \rightarrow 0} \frac{1}{2\sqrt{t_i}} [f(x_1, x_2, y + r\sqrt{t_i}) - f(x_1, x_2, y)] \\
 &\quad + \lim_{t_i \rightarrow 0} \frac{1}{2\sqrt{t_i}} [f(x_1, x_2, y) - f(x_1, x_2, y - r\sqrt{t_i})] \\
 &= \frac{1}{2} \nabla f \cdot (0, 0, r) - \frac{1}{2} \nabla f \cdot (0, 0, -r) \\
 &= \nabla f \cdot (0, 0, r).
 \end{aligned}$$

We now prove that $D \cdot f$ is smooth at X . Similarly to the proof of the case of $S \cdot f$, since the function $\frac{1}{2}[f(x_1, x_2, y + rt) - f(x_1, x_2, y - rt)]$ is odd with respect to t , there exist smooth functions $b_i(x_1, x_2, y, r)$, for $1 \leq i < s$ and $\psi(x_1, x_2, y, r, t)$ such that $\frac{1}{2}[f(x_1, x_2, y + rt) - f(x_1, x_2, y - rt)] = \sum_{i=0}^{s-1} b_i(x_1, x_2, y, r)t^{2i+1} + t^{2s+1} \cdot \psi(x_1, x_2, y, r, t)$. Notice that $D \cdot f(x_1, x_2, y, r, t) = \sum_{i=0}^{s-1} b_i(x_1, x_2, y, r)t^i + t^s \cdot \psi(x_1, x_2, y, \sqrt{t})$, so that a partial derivative exists up to order s at $t = 0$. Thus, $D \cdot f(x_1, x_2, y, r, t)$ is a C^{s-1} function. This holds for any arbitrarily large s , hence $D \cdot f(x_1, x_2, y, r, t)$ is a C^∞ function. \square

By Lemma 4.1.2, we can build our Ball system based on the smooth operators S and D . We state the main theorem of this section.

Theorem 4.1.3. Consider $P = (P_1, \dots, P_{n-1}) \in C^\infty(\mathbb{R}^n, \mathbb{R}^{n-1})$ satisfy Assumptions $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$ and \mathcal{A}_4 . Then, $X = (x_1, x_2, y, r, t) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}^{n-2} \times \mathbb{R}^{n-2} \times \mathbb{R}$ is a solution of the Ball system

$$\text{Ball}(P) = \begin{cases} S \cdot P_1(X) = \dots = S \cdot P_{n-1}(X) = 0 \\ D \cdot P_1(X) = \dots = D \cdot P_{n-1}(X) = 0 \\ \|r\|^2 - 1 = 0 \end{cases} \quad (4.1)$$

if and only if (x_1, x_2) is a singular point of $\pi_{\mathfrak{C}}(\mathfrak{C})$ (see Definition 4.1.1 for the notation $S \cdot P_i$ and $D \cdot P_i$).

We postpone the proof of Theorem 4.1.3 to the end of Section 4.3.

Remark 4.1.4. *The purpose behind the last equation $\|r\|^2 - 1 = 0$ in $\text{Ball}(P)$ is to impose the constraint that r is not trivial. Hence, the equation $\|r\|^2 - 1 = 0$ can be replaced with a different constraint. However, this condition is first a polynomial constraint. Second, it provides clear bounds of r when solving $\text{Ball}(P)$. In addition, as a special case, for $n = 3$, r then consists of one variable. Thus, the equalities $0 = \|r\|^2 - 1 = r_3^2 - 1$ imply $r_3 = \pm 1$ which reduces the complexity of solving $\text{Ball}(P)$.*

We now study a mapping from the solutions of the Ball system to pairs of points on the curve \mathfrak{C} .

Definition 4.1.5. *Let $P \in C^\infty(\mathbb{R}^n, \mathbb{R}^{n-1})$. Define $\widehat{\mathfrak{L}}_n$ to be the set of pairs (q_1, q_2) with $q_1, q_2 \in \mathfrak{C}$, $q_1 \neq q_2$ and $\pi_{\mathfrak{C}}(q_1) = \pi_{\mathfrak{C}}(q_2)$, also define $\widehat{\mathfrak{L}}_c$ to be the set of pairs (q_1, q_1) with $q_1 \in \mathfrak{L}_c$, and let $\widehat{\mathfrak{L}} = \widehat{\mathfrak{L}}_n \cup \widehat{\mathfrak{L}}_c$.*

Lemma 4.1.6. *Consider $P = (P_1, \dots, P_{n-1}) \in C^\infty(\mathbb{R}^n, \mathbb{R}^{n-1})$ and let $X = (x_1, x_2, y, r, t) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}^{n-2} \times \mathbb{R}^{n-2} \times \mathbb{R}$, with $\|r\| = 1$. Assume that P satisfies Assumption \mathcal{A}_1 . Then X is a solution of $\text{Ball}(P)$ if and only if for the points $q_1 = (x_1, x_2, y + r\sqrt{t})$ and $q_2 = (x_1, x_2, y - r\sqrt{t})$, the pair (q_1, q_2) is in $\widehat{\mathfrak{L}}_n$, or in $\widehat{\mathfrak{L}}_c$ with $(0, 0, r) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}^{n-2}$ in $T_{q_1}\mathfrak{C}$.*

Proof. Note that, by Assumption \mathcal{A}_1 , the tangent space to the curve at any of its points is well defined and is a line. First, assume that X is a solution of $\text{Ball}(P)$. We consider two cases:

- (a) If $t > 0$, then since $r \neq 0 \in \mathbb{R}^{n-2}$ we have that $q_1 \neq q_2$. Moreover, since $S \cdot P_i(X) = D \cdot P_i(X) = 0$ for all $i \in \{1, \dots, n-1\}$, we deduce that $P_i(q_1) = P_i(q_2) = 0$, thus $q_1, q_2 \in \mathfrak{C}$. Moreover, since $\pi_{\mathfrak{C}}(q_1) = \pi_{\mathfrak{C}}(q_2) = (x_1, x_2)$ we have $q_1, q_2 \in \mathfrak{L}_n$. Thus, $(q_1, q_2) \in \widehat{\mathfrak{L}}_n$.
- (b) If $t = 0$, then $q_1 = q_2$. First, $P_i(q_1) = S \cdot P_i(X) = 0$, for all indices $i \in \{1, \dots, n-1\}$, hence $q_1 \in \mathfrak{C}$. Moreover, we have that $0 = D \cdot P_i(X) = \nabla P_i(q_1) \cdot (0, 0, r)$, for all $i \in \{1, \dots, n-1\}$, equivalently, $J_P(q_1) \cdot (0, 0, r)^T = 0 \in \mathbb{R}^{n-1}$, i.e., we have $(0, 0, r) \in T_{q_1}\mathfrak{C}$. Thus, $q_1 \in \mathfrak{L}_c$ and hence, $(q_1, q_1) \in \widehat{\mathfrak{L}}_c$.

Now, let us prove the other direction:

- (a) If $(q_1, q_2) \in \widehat{\mathfrak{L}}_n$, then $q_1 \neq q_2$ and $t \neq 0$. Also, since $q_1, q_2 \in \mathfrak{C}$, we can write that $S \cdot P_i(X) = \frac{1}{2}(P_i(q_1) + P_i(q_2)) = 0$, and $D \cdot P_i(X) = \frac{1}{2\sqrt{t}}(P_i(q_1) - P_i(q_2)) = 0$, for all $i \in \{1, \dots, n-1\}$. Thus, X is a solution of $\text{Ball}(P)$.

(b) If $(q_1, q_2) \in \widehat{\mathfrak{L}}_c$ and $(0, 0, r) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}^{n-2}$ is in $T_{q_1} \mathfrak{C}$, one has $q_1 = q_2 \in \mathfrak{L}_c \subseteq \mathfrak{C}$, and $t = 0$. Moreover, for all $i \in \{1, \dots, n-1\}$ we have $S \cdot P_i(X) = P_i(q_1) = 0$ and since $(0, 0, r) \in T_{q_1} \mathfrak{C}$, we can equivalently write $D \cdot P_i(X) = \nabla P_i(q_1) \cdot (0, 0, r) = 0$. Thus, X is a solution of $\text{Ball}(P)$. \square

Definition 4.1.7. Let $\text{Sol}_{\text{Ball}(P)}$ be the solution set of $\text{Ball}(P)$. Define the function Ω_P from $\text{Sol}_{\text{Ball}(P)}$ to $\widehat{\mathfrak{L}}$ that sends $X = (x_1, x_2, y, r, t) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}^{n-2} \times \mathbb{R}^{n-2} \times \mathbb{R}$ to the ordered pair $q_1 = (x_1, x_2, y + r\sqrt{t})$ and $q_2 = (x_1, x_2, y - r\sqrt{t})$. Notice that the function Ω_P is well-defined by Lemma 4.1.6.

Lemma 4.1.8. If $P \in C^\infty(\mathbb{R}^n, \mathbb{R}^{n-1})$ satisfies Assumption \mathcal{A}_1 , then Ω_P is surjective.

Proof. For any pair $(q_1, q_2) \in \widehat{\mathfrak{L}}_n$ we have that the point $X = (\frac{1}{2}(q_1 + q_2), \frac{\Pi_{\mathfrak{C}}(q_1 - q_2)}{\|q_1 - q_2\|}, \frac{1}{4}\|q_1 - q_2\|^2) \in \mathbb{R}^n \times \mathbb{R}^{n-2} \times \mathbb{R}$ is a solution of $\text{Ball}(P)$, where $\Pi_{\mathfrak{C}}(q_1 - q_2)$ is the vector in \mathbb{R}^{n-2} obtained by omitting the first two coordinates (which are zeros) from $q_1 - q_2$. Note that $\Omega_P(X) = (q_1, q_2)$. If the pair (q_1, q_1) is in $\widehat{\mathfrak{L}}_c$, we define r in the following way, we take a unit vector $v \in T_{q_1} \mathfrak{C}$ (the first two coordinates of v are zeros since $q_1 \in \mathfrak{L}_c$). We set r to be $\Pi_{\mathfrak{C}}(v)$. Again $X = (q_1, r, 0) \in \mathbb{R}^n \times \mathbb{R}^{n-2} \times \mathbb{R}$ is a solution of $\text{Ball}(P)$, with $\Omega_P(X) = (q_1, q_1)$. Thus, Ω_P is surjective. \square

Remark 4.1.9. Notice that if $X = (x_1, x_2, y, r, t)$ is in $\text{Sol}_{\text{Ball}(P)}$, then $\Omega_P(X) \in \widehat{\mathfrak{L}}_n$ (resp. $\Omega_P(X) \in \widehat{\mathfrak{L}}_c$) if and only if $t \neq 0$ (resp. $t = 0$).

Remark 4.1.10. Preserving the notation in Lemma 4.1.6, notice that if $X = (x_1, x_2, y, r, t)$ is a solution of $\text{Ball}(P)$, then $X' = (x_1, x_2, y, -r, t)$ is another solution. Moreover, both solutions characterize the same unordered pair $\Omega_P(X) = \Omega_P(X') = (q_1, q_2)$. We call X and X' twin solutions. An alternative choice would have been to take r in a projective space instead of the sphere to identify these twin solutions.

Example 4.1.11. Let $n = 3$ and $B = \{(x_1, x_2, x_3) \in \mathbb{R}^3 \mid x_1, x_2, x_3 \in [-2, 2]\}$. Define $P_1(x_1, x_2, x_3) = x_1 - (x_3 - 1)^3$, $P_2(x_1, x_2, x_3) = x_2 - (x_3 - 1)^2$ and $P = (P_1, P_2)$.

The Jacobian matrix of P has full rank over \mathfrak{C} , thus Assumption \mathcal{A}_1 is satisfied. The set \mathfrak{L}_n is empty since $\pi_{\mathfrak{C}}$ is injective over \mathfrak{C} , hence Assumption \mathcal{A}_4 is satisfied. The only point of \mathfrak{C} with a tangent line orthogonal to the (x_1, x_2) -plane is $q_1 = (0, 0, 1)$, thus $\mathfrak{L}_c = \{q_1\}$ and Assumption \mathcal{A}_2 is satisfied. By Lemma 4.3.2, the multiplicity of the system $\{P = 0, (x_1, x_2) = \pi_{\mathfrak{C}}(q_1)\}$ at its unique solution q_1 is $\min\{\text{ord}_1((x_3 - 1)^3), \text{ord}_1((x_3 - 1)^2)\} = \min\{3, 2\} = 2$ (ord is defined in

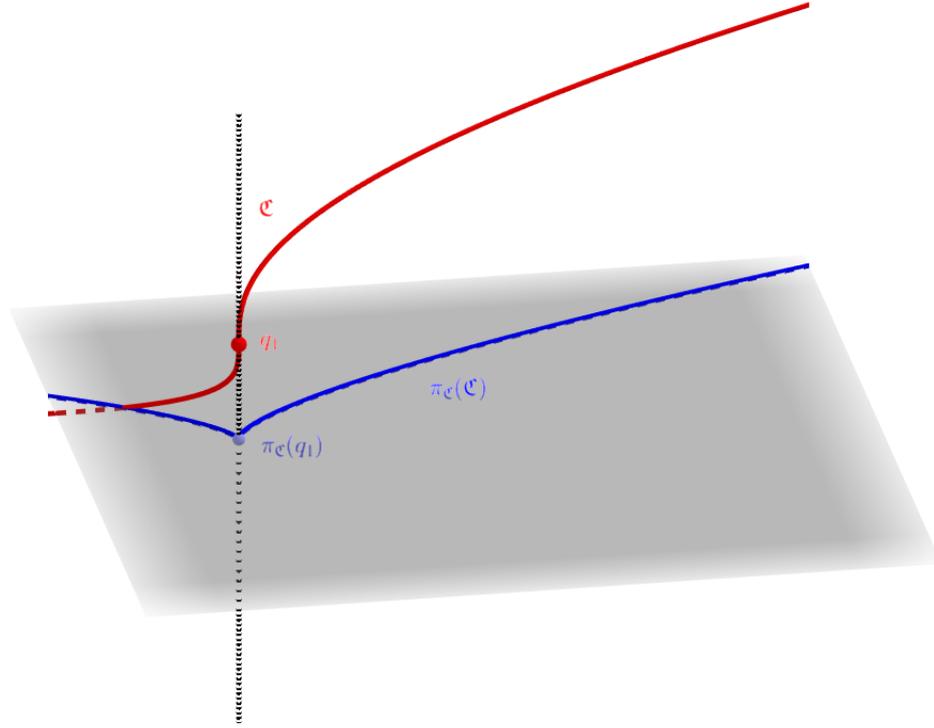


Figure 4.2: The curve \mathcal{C} (red) and its plane projection $\pi_{\mathcal{C}}(\mathcal{C})$ (blue) of Example 4.1.11 displaying a cusp singularity.

Definition 2.2.8). Moreover, for any point $q_0 \in \mathcal{C}$ different from q_1 , the multiplicity of the corresponding system at its unique solution q_0 is one, thus P satisfies Assumption \mathcal{A}_3 . The system $\text{Ball}(P)$:

$$\begin{cases} x_1 - 3r^2ty + 3r^2t - y^3 + 3y^2 - 3y + 1 = 0 \\ x_2 - r^2t - y^2 + 2y - 1 = 0 \\ -r^3t - 3ry^2 + 6ry - 3r = 0 \\ -2ry + 2r = 0 \\ r^2 - 1 = 0 \end{cases} \quad (4.2)$$

has two twin solutions $X = (0, 0, 1, 1, 0)$ and $X' = (0, 0, 1, -1, 0)$ in $B_{\text{Ball}(P)} \subset \mathbb{R}^{2 \cdot 3 - 1} = \mathbb{R}^5$ such that $\Omega_P(X) = \Omega_P(X') = (q_1, q_1) \in \widehat{\mathcal{L}}_{\mathcal{C}}$.

Example 4.1.12. Let B be defined as in Example 4.1.11. Define the functions $P_1(x_1, x_2, x_3) = x_1 - (x_3^2 - 1)$, $P_2(x_1, x_2, x_3) = x_2 - (x_3^3 - x_3)$ and $P = (P_1, P_2)$. The Jacobian matrix of P has full rank over \mathfrak{C} , thus Assumption \mathcal{A}_1 is satisfied. Moreover, the set \mathfrak{L}_c is empty and $\mathfrak{L}_n = \{q_1, q_2\}$, with $q_1 = (0, 0, 1)$, $q_2 = (0, 0, -1)$, i.e., Assumptions \mathcal{A}_2 and \mathcal{A}_4 are satisfied. The multiplicity of the system $\{P = 0 \in \mathbb{R}^{n-1}, x_1 = x_2 = 0\}$ at both q_1, q_2 is equal to one, thus Assumption \mathcal{A}_3 is also satisfied. The system $\text{Ball}(P)$:

$$\begin{cases} x_1 - r^2t - y^2 + 1 = 0 \\ x_2 - r^2ty - y^3 + y = 0 \\ -2ry = 0 \\ -r^3t - 3ry^2 + r = 0 \\ r^2 - 1 = 0 \end{cases} \quad (4.3)$$

has two twin solutions $X = (0, 0, 0, 1, 1)$ and $X' = (0, 0, 0, -1, 1)$ in \mathbb{R}^5 such that $\Omega_P(X) = \Omega_P(X') = (q_1, q_2) \in \widehat{\mathfrak{L}}_n$.

4.2 Singularities induced by \mathfrak{L}_n

Under Assumptions $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$ and \mathcal{A}_4 , we studied in Lemma 3.2.7 the singularity types of the projection of the points in \mathfrak{L}_n . More precisely, for $q \in \mathfrak{L}_n$, the point $\pi_{\mathfrak{C}}(q)$ is the intersection of two branches of $\pi_{\mathfrak{C}}(\mathfrak{C})$, that is, the curve $\pi_{\mathfrak{C}}(\mathfrak{C})$ at the point $\pi_{\mathfrak{C}}(q)$ is either an A_{2k+1} singularity with $k \geq 0$ or equivalent to a curve defined by $x_1^2 - g(x_2^2)$ at the origin, with $\text{ord}(g) = \infty$. In both cases, the point $\pi_{\mathfrak{C}}(q)$ is a singularity in $\pi_{\mathfrak{C}}(\mathfrak{C})$. Under our weak or strong assumptions, the singularity type of $\pi_{\mathfrak{C}}(q)$ is A_1 (see Definition 2.2.4). We prove in Section 4.4 that this implies the regularity of our Ball system (defined in (4.1)).

Recall that in Lemma 3.2.6 we studied, under Assumptions $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$ and \mathcal{A}_4 , the plane projection of points in $\mathfrak{C} \setminus (\mathfrak{L}_c \cup \mathfrak{L}_n)$. Hence, in order to complete the classification of the points in $\pi_{\mathfrak{C}}(\mathfrak{C})$ under the latter assumptions, we study in the next section the plane projection of the points in \mathfrak{L}_c .

4.3 Singularities induced by \mathfrak{L}_c

We now study the types of singularities of the plane curve $\pi_{\mathfrak{C}}(\mathfrak{C})$ obtained by projecting points in \mathfrak{L}_c , that is when the tangent to \mathfrak{C} is orthogonal to the projection plane. We start by locally parameterizing \mathfrak{C} around a point in \mathfrak{L}_c . This parameterization will ease the computation of $\text{Ball}(P)$ and its Jacobian in Section 4.4.

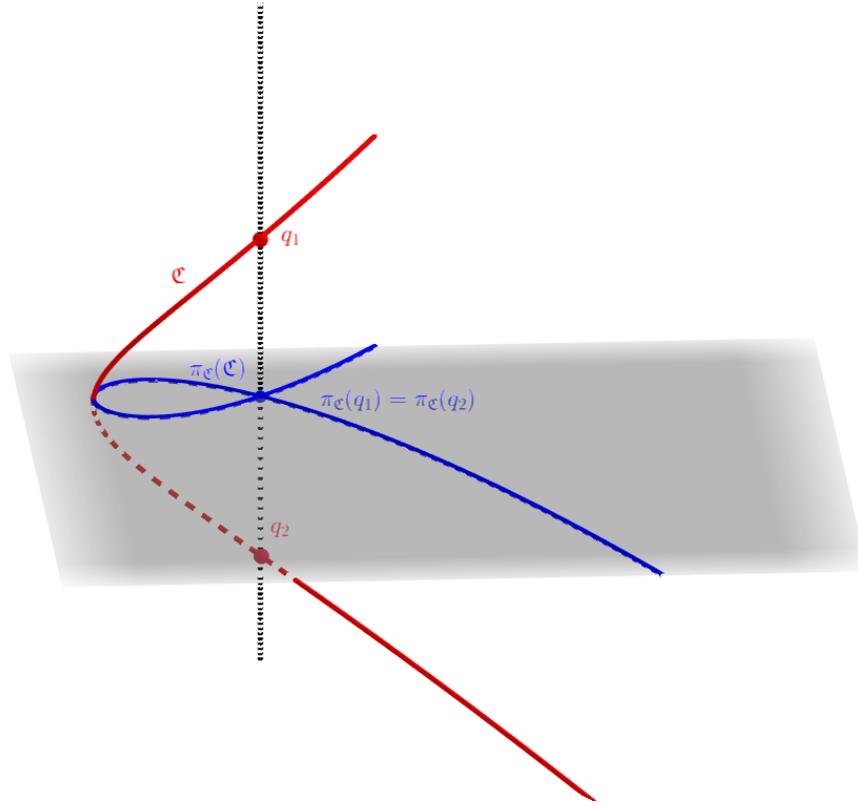


Figure 4.3: The curve \mathcal{C} (red) and its plane projection $\pi_{\mathcal{C}}(\mathcal{C})$ (blue) of Example 4.1.12 displaying a node singularity.

In the rest of this section and Section 4.4, the analysis is simplified by translating relevant points or assuming the curve \mathcal{C} is parameterizable by a specific variable. On the other hand, in the algorithms presented in Chapter 5, the input is not modifiable at all but every computation uses interval arithmetic. This implies that the exact coordinates of a point may not be known, instead we only compute with a box containing it and isolating it from other relevant points. The idea of our semi-algorithms is to check that some function does not vanish on such a box. This then implies that such a function does not vanish at the point this box contains. The *theoretical analysis* of this section can then be applied to the point to deduce the appropriate property without the knowledge of the exact location of that point.

Lemma 4.3.1. *Let $P \in C^\infty(\mathbb{R}^n, \mathbb{R}^{n-1})$. Let $q \in \mathcal{L}_c$ such that Assumption \mathcal{A}_1 is satisfied in a neighbourhood of q in B . Without loss of generality one can assume*

$q = 0 \in \mathbb{R}^n$. Then there exist an invertible matrix M of size $(n - 1) \times (n - 1)$ of smooth functions in a neighbourhood of q and smooth functions $f_1, f_2, f_3, \dots, f_{n-1}$ defined in a neighbourhood of $0 \in \mathbb{R}$, such that:

$$\begin{pmatrix} x_1 - f_1(x_n) \\ x_2 - f_2(x_n) \\ x_3 - f_3(x_n) \\ \dots \\ \dots \\ x_{n-1} - f_{n-1}(x_n) \end{pmatrix} = M \cdot \begin{pmatrix} P_1 \\ P_2 \\ P_3 \\ \dots \\ \dots \\ P_{n-1} \end{pmatrix}, \quad (4.1)$$

with $\min\{\text{ord}(f_1(x_n)), \text{ord}(f_2(x_n))\} > 1$ (ord is defined in Definition 2.2.8).

Proof. Since $\text{rank}(J_P(q)) = n - 1$ (Assumption \mathcal{A}_1), there exists $k \in \{1, \dots, n\}$ such that $\det(M_k(q)) \neq 0$, where M_k is the sub-matrix of J_P obtained by removing the k -th column. Notice that $k \notin \{1, 2\}$, since $q \in \mathcal{L}_c$ implies that $\det(M_1(q)) = \det(M_2(q)) = 0$. Without loss of generality, we assume that $k = n$. Using the implicit function theorem [Corollary 2.7.3][Dem2000], there exist smooth functions f_1, \dots, f_{n-1} (see Figure 4.4) of one variable such that we have that

$$P_j(f_1(x_n), \dots, f_{n-1}(x_n), x_n) = 0, j \in \{1, \dots, n - 1\}. \quad (4.2)$$

Define the function φ that maps x_i to $z_i = x_i - f_i(x_n)$, for all $i \in \{1, \dots, n - 1\}$ and x_n to $z_n = x_n$. We can see that φ is a diffeomorphism and $z = (z_1, \dots, z_n)$ is a local coordinate system around q . Hence, we can define the function $G_j(z) = P_j \circ \varphi^{-1}(z) = P_j(x)$ for all integers $1 \leq j \leq n - 1$. Using Hadamard's Lemma [Dem2000, Proposition 4.2.3] for the first $n - 1$ variables of z , we can write $G_j(z) - G_j(0, \dots, 0, z_n) = \sum_{i=1}^{n-1} z_i \cdot h_{ji}(z)$ for some smooth functions h_{ji} . Note that $\varphi^{-1}(z) = (z_1 + f_1(z_n), \dots, z_{n-1} + f_{n-1}(z_n), z_n)$. Hence, $G_j(0, \dots, 0, z_n) = P_j \circ \varphi^{-1}(0, \dots, 0, z_n) = P_j(f_1(z_n), \dots, f_{n-1}(z_n), z_n) = P_j(f_1(x_n), \dots, f_{n-1}(x_n), x_n)$. The latter function is equal to zero by (4.2). Thus, $P_j(x) = G_j(z) = \sum_{i=1}^{n-1} z_i \cdot h_{ji}(z) = \sum_{i=1}^{n-1} (x_i - f_i(x_n)) \cdot H_{ji}(x)$, with $H_{ji}(x) = h_{ji} \circ \varphi(x)$.

Defining $M_0 = \left(H_{ji} \right)_{1 \leq j, i \leq n-1}$ we get:

$$\begin{pmatrix} P_1 \\ \dots \\ \dots \\ P_{n-1} \end{pmatrix} = M_0 \cdot \begin{pmatrix} x_1 - f_1(x_n) \\ \dots \\ \dots \\ x_{n-1} - f_{n-1}(x_n) \end{pmatrix}.$$

Notice that M_0 evaluated at q is the invertible matrix $M_n(q)$. Hence, by continuity of the determinant function, there is a neighbourhood of q in which M_0 is invertible. Thus, writing M as the inverse of M_0 we get:

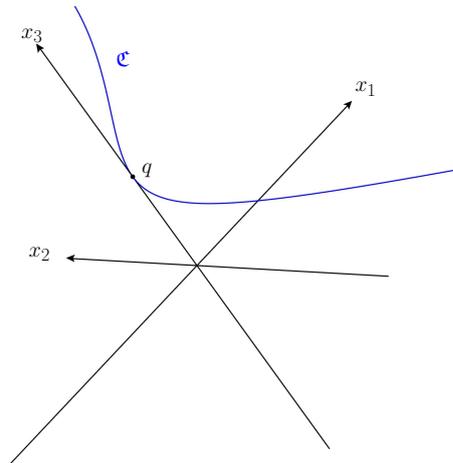
$$Q_0 = \begin{pmatrix} x_1 - f_1(x_n) \\ \cdots \\ \cdots \\ x_{n-1} - f_{n-1}(x_n) \end{pmatrix} = M \cdot \begin{pmatrix} P_1 \\ \cdots \\ \cdots \\ P_{n-1} \end{pmatrix}. \quad (4.3)$$

To prove that $\min\{\text{ord}(f_1(x_n)), \text{ord}(f_2(x_n))\} > 1$, we take the Jacobian matrices of both sides of (4.3) and we evaluate them at $q = 0$. We get the equation $J_{Q_0}(q) = M(q) \cdot J_P(q)$. By invertibility of $M(q)$ we deduce that the k -th sub-matrices, obtained by removing the k -th column, of $J_{Q_0}(q)$ and $J_P(q)$ have the same rank. Computing $J_{Q_0}(q)$ and considering the fact that $\det(M_1(q)) = \det(M_2(q)) = 0$ implies that $f'_1(0) = f'_2(0) = 0$, we thus have that $\min\{\text{ord}(f_1(x_n)), \text{ord}(f_2(x_n))\}$ is at least two. \square

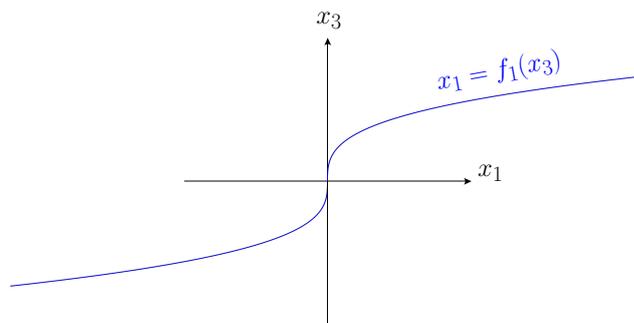
Lemma 4.3.2. *Preserving the notation and the assumptions in Lemma 4.3.1, the multiplicity m of the system $S = \{Q_0(x) = 0 \in \mathbb{R}^{n-1}, x_1 = x_2 = 0\}$ at q is equal to $d = \min\{\text{ord}(f_1(x_n)), \text{ord}(f_2(x_n))\}$.*

Proof. First, we start with the case $m < \infty$. By Proposition 2.2.11, we can assume, without loss of generality, that f_1, \dots, f_{n-1} are polynomials. Following the notation in Definition 2.1.1, let $\mathbb{R}[x]$ (resp. $\mathbb{R}[x_n]$) be the ring of polynomials with n variables (resp. one variable) and $\mathbb{R}[x]_q$ (resp. $\mathbb{R}[x_n]_0$) be its localization at q (resp. $0 \in \mathbb{R}$). Also, define I_S to be the ideal generated by the polynomials of S in $\mathbb{R}[x]_q$ (as I_G is defined in Definition 2.1.1), i.e., $I_S = \langle x_1 - f_1(x_n), x_2 - f_2(x_n), \dots, x_{n-1} - f_{n-1}(x_n), x_1, x_2 \rangle = \langle x_1 - f_1(x_n), x_2 - f_2(x_n), \dots, x_{n-1} - f_{n-1}(x_n), f_1(x_n), f_2(x_n) \rangle$. If $f_1(x_n) = f_2(x_n) = 0$, then the ideal I_S is of dimension one, hence, S has an infinite number of solutions which contradicts the assumption that $m < \infty$. Thus, $d < \infty$ which means that there exist $h_1, h_2 \in \mathbb{R}[x_n]_0$ such that $h_1(x_n)f_1(x_n) + h_2(x_n)f_2(x_n) = x_n^d$. Thus, $I_S = \langle x_1 - f_1(x_n), x_2 - f_2(x_n), \dots, x_{n-1} - f_{n-1}(x_n), x_n^d \rangle$. Note that the set $\{x_1 - f_1(x_n), x_2 - f_2(x_n), \dots, x_{n-1} - f_{n-1}(x_n), x_n^d\}$ is a Gröbner basis of I_S with respect to Local Lexicographical ordering $x_1 > \dots > x_n$. Hence, by Theorem 2.1.2, we have $\dim(\frac{\mathbb{R}[x]_q}{I_S}) = \dim(\frac{\mathbb{R}[x]_q}{LT(I_S)}) = \dim(\frac{\mathbb{R}[x]_q}{\langle x_1, x_2, \dots, x_{n-1}, x_n^d \rangle})$, where $LT(I_S)$ is the ideal generated by the leading terms of I_S . Consequently, $m = \dim(\frac{\mathbb{R}[x]_q}{I_S}) = \dim(\frac{\mathbb{R}[x_n]_0}{\langle x_n^d \rangle}) = d$.

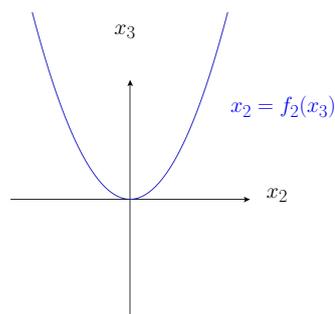
Second, assume that $m = \infty$. We prove that $d = \infty$, that is, $\frac{\partial^k f_1}{\partial x_n^k}(0) = \frac{\partial^k f_2}{\partial x_n^k}(0) = 0$ for any positive integer k . Preserving the notation in Definition 2.2.10, consider



(a) The curve \mathfrak{C} in \mathbb{R}^3 where $q \in \mathfrak{L}_c$.



(b) x_1 is locally a smooth function of x_3 at q .



(c) x_2 is locally a smooth function of x_3 at q .

Figure 4.4: An example of Lemma 4.3.1 for $n = 3$.

the dual space $D_q^k[S]$. We are going to show that for any positive integer k and any element $c \in D_q^k[S] \setminus D_q^{k-1}[S]$ (which always exists since $m = \infty$), the coefficient $c_{x_n^k}$ corresponding to $\frac{\partial^k}{\partial x_n^k}$, for c , is non-zero. We consequentially show that $\frac{\partial^k f_1}{\partial x_n^k}(0) = \frac{\partial^k f_2}{\partial x_n^k}(0) = 0$. We prove the previous statements by induction on k .

For $k = 1$, since $q \in \mathcal{L}_c$, we already showed in the proof of Lemma 3.2.3 that a non-trivial element $c = \sum_{i=1}^n v_i \frac{\partial}{\partial x_i}$ is in $D_q^1[S] \setminus D_q^0[S]$ if and only if $v = (v_1, \dots, v_n)$ is in $T_q\mathcal{C}$. On the other hand, $T_q\mathcal{C}$ is generated by the vector $(f'_1(0), \dots, f'_{n-1}(0), 1)$, thus $c_{x_n^1} = v_n \neq 0$. The function $f_1(x_n)$ is in the set of functions generated by S thus $0 = c \cdot (f_1(x_n)) = \sum_{i=1}^n v_i \frac{\partial}{\partial x_i} \cdot (f_1(x_n)) = c_{x_n^1} \frac{\partial f_1}{\partial x_n}(0)$, and hence $\frac{\partial f_1}{\partial x_n}(0) = 0$. Thus, the induction hypothesis holds for $k = 1$.

For $k > 1$, define $c' = \phi_n(c)$ and consider two cases:

- (a) $c' \in D_q^{k-1}[S] \setminus D_q^{k-2}[S]$: By the induction hypothesis, the coefficient $c'_{x_n^{k-1}}$ corresponding to $\frac{\partial^{k-1}}{\partial x_n^{k-1}}$ for c' is non-zero and $\frac{\partial^{k'} f_1}{\partial x_n^{k'}}(0) = \frac{\partial^{k'} f_2}{\partial x_n^{k'}}(0) = 0$, for all $k' < k$. Notice that by the definition of ϕ_n , we have $c_{x_n^k} = c'_{x_n^{k-1}} \neq 0$. Hence, $0 = c \cdot f_1(x_n) = \sum_{i=1}^k c_{x_n^i} \frac{\partial^i f_1}{\partial x_n^i}(0) = c_{x_n^k} \frac{\partial^k f_1}{\partial x_n^k}(0)$. Hence, $\frac{\partial^k f_1}{\partial x_n^k}(0) = 0$. Similarly, we prove that $\frac{\partial^k f_2}{\partial x_n^k}(0) = 0$. Thus in Case (a), the lemma is proved.
- (b) $c' \in D_q^{k-2}[S]$: Since $c \in D_q^k[S] \setminus D_q^{k-1}[S]$, there exists $j \in \{1, \dots, n-1\}$ such that the element $c'' = \phi_j(c)$ is in $D_q^{k-1}[S] \setminus D_q^{k-2}[S]$. By the induction hypothesis, the coefficient $c''_{x_n^{k-1}}$ corresponding to $\frac{\partial^{k-1}}{\partial x_n^{k-1}}$ for c'' , is non-zero. On the other hand, $c_{x_j x_n^{k-1}} = c''_{x_n^{k-1}} \neq 0$. Hence, since $\phi_n(c_{x_j x_n^{k-1}} \frac{\partial^{k-1}}{\partial x_j \partial x_n^{k-1}}) \in D_q^{k-1}[S] \setminus D_q^{k-2}[S]$, then so is $\phi_n(c) = c'$ which contradicts the assumption. Thus, Case (b) is impossible.

□

With the additional Assumptions \mathcal{A}_2 , \mathcal{A}_3 and \mathcal{A}_4 , one can give a more precise form of f_1 and f_2 in Equation (4.1).

Lemma 4.3.3. *Let $P \in C^\infty(\mathbb{R}^n, \mathbb{R}^{n-1})$. Let $q \in \mathcal{L}_c$ such that Assumptions \mathcal{A}_1 , \mathcal{A}_2 , \mathcal{A}_3 and \mathcal{A}_4 in \overline{B} , then there exist an invertible matrix \widetilde{M} of size $(n-1) \times (n-1)$ of smooth functions in a neighbourhood of q , a smooth diffeomorphism φ defined in an open subset of \mathbb{R}^n , with $z = (z_1, \dots, z_n) = \varphi^{-1}(x)$ and smooth functions*

f_3, \dots, f_{n-1}, g defined in a neighbourhood of $0 \in \mathbb{R}$, such that

$$Q = \begin{pmatrix} z_1 - z_n \cdot g(z_n^2) \\ z_2 - z_n^2 \\ z_3 - f_3(z_n) \\ \dots \\ \dots \\ z_{n-1} - f_{n-1}(z_n) \end{pmatrix} = \widetilde{M} \cdot \begin{pmatrix} P_1 \\ P_2 \\ P_3 \\ \dots \\ \dots \\ P_{n-1} \end{pmatrix} \circ \varphi, \quad (4.4)$$

on a neighbourhood of q . Moreover, either $\text{ord}(g(z_n)) = \infty$ or there exists an integer $k > 0$ with $g(z_n) = z_n^k$.

Proof. Step 1: Equation (4.3) implies that Q_0 and P define the same curve \mathcal{C} in a neighbourhood of q and that the function Q_0 satisfies the same assumptions as P around q . By Lemma 4.3.2, $d = \min\{\text{ord}(f_1(x_n)), \text{ord}(f_2(x_n))\}$ is the multiplicity of the system $\{Q_0(x) = 0 \in \mathbb{R}^{n-1}, x_1 = 0, x_2 = 0\}$ at q . By Assumption \mathcal{A}_3 , we have that $d = 2$.

Without loss of generality, assume that $\text{ord}(f_2(x_n)) = 2$ and $\frac{\partial^2 f_2}{\partial x_n^2}(0) = 2$. Hence, there is a smooth function v such that $f_2(x_n) = x_n^2(1 + x_n \cdot v(x_n))$. Now, consider the diffeomorphism ϕ_n that sends x_n to $z_n = x_n \sqrt{1 + x_n \cdot v(x_n)}$. We have that $x_2 - f_2(x_n) = x_2 - z_n^2$. Define $\tilde{f}_1(z_n) = f_1(\phi_n^{-1}(z_n))$ and $\tilde{f}_2(z_n) = f_2(\phi_n^{-1}(z_n)) = z_n^2$. Since $\text{ord}(\tilde{f}_1(z_n)) = \text{ord}(f_1(x_n)) \geq d = 2$, there exists a smooth function h such that $\tilde{f}_1(z_n) = z_n^2 h(z_n)$. Write $\tilde{f}_1(z_n) = z_n^2 \left[\frac{h(z_n) + h(-z_n)}{2} + \frac{h(z_n) - h(-z_n)}{2} \right]$. Since $\frac{h(z_n) + h(-z_n)}{2}$ (resp. $\frac{h(z_n) - h(-z_n)}{2}$) is even (resp. odd), then by Theorem 2.2.2 there exists a smooth function ξ_1 (resp. ξ_2) such that $\frac{h(z_n) + h(-z_n)}{2} = \xi_1(z_n^2)$ (resp. $\frac{h(z_n) - h(-z_n)}{2} = z_n \xi_2(z_n^2)$). Thus, $\tilde{f}_1(z_n) = z_n^2 (\xi_1(z_n^2) + z_n \xi_2(z_n^2))$. Notice that $\xi_2(x_n^2)$ cannot be the zero function, otherwise $\tilde{f}_1(\epsilon) = \tilde{f}_1(-\epsilon)$ and $\tilde{f}_2(\epsilon) = \tilde{f}_2(-\epsilon)$ for all small enough $\epsilon > 0$, which contradicts Assumption \mathcal{A}_4 .

Step 2: We have two cases:

Case 1: $\text{ord}(\xi_2(z_n)) = \infty$, then define the diffeomorphism ϕ which sends x_1 to $z_1 = x_1 - x_2 \xi_1(x_2)$, x_i to $z_i = x_i$ for all integers $i \in \{2, \dots, n-1\}$ and x_n to $z_n = x_n \sqrt{1 + x_n \cdot v(x_n)}$. Taking $g(z_n) = z_n \xi_2(z_n)$ and $\varphi = \phi^{-1}$ we prove the claim for the first case.

Case 2: $\text{ord}(\xi_2(z_n)) = k < \infty$, that is, $\xi_2(z_n) = z_n^k u(z_n)$, for some smooth function u , with $u(0) \neq 0$ and an integer $k \geq 0$. Hence, we can write $x_1 - \tilde{f}_1(z_n) = x_1 - z_n^2 \xi_1(z_n^2) - z_n^{2k+3} u(z_n^2) = x_1 - x_2 \xi_1(x_2) - z_n^{2k+3} u(x_2)$.

So, defining the diffeomorphism ϕ which sends x_i to $z_i = x_i$ for all integers $i \in \{2, \dots, n-1\}$, x_n to $z_n = x_n \sqrt{1 + x_n \cdot v(x_n)}$ and x_1 to $z_1 = (x_1 - x_2 \xi_1(x_2)) u^{-1}(x_2)$ (which means that $x_1 - f_1(x_n) = u(x_2)[z_1 - z_n^{2k+3}]$), we get

that:

$$\begin{pmatrix} x_1 - f_1(x_n) \\ \dots \\ \dots \\ x_{n-1} - f_{n-1}(x_n) \end{pmatrix} = \begin{pmatrix} u(x_2) & 0_{1 \times (n-2)} \\ 0_{(n-2) \times 1} & I_{n-2} \end{pmatrix} \cdot \begin{pmatrix} z_1 - z_n^{2k+3} \\ z_2 - z_n^2 \\ z_3 - f_3(z_n) \\ \dots \\ \dots \\ z_{n-1} - f_{n-1}(z_n) \end{pmatrix} \circ \phi, \quad (4.5)$$

for a small enough neighbourhood of q , where I_{n-2} is the identity matrix of size $n - 2$. Comparing with (4.1), we get:

$$M \cdot \begin{pmatrix} P_1 \\ P_2 \\ P_3 \\ \dots \\ \dots \\ P_{n-1} \end{pmatrix} = \begin{pmatrix} u(x_2) & 0_{1 \times (n-2)} \\ 0_{(n-2) \times 1} & I_{n-2} \end{pmatrix} \cdot \begin{pmatrix} z_1 - z_n^{2k+3} \\ z_2 - z_n^2 \\ z_3 - f_3(z_n) \\ \dots \\ \dots \\ z_{n-1} - f_{n-1}(z_n) \end{pmatrix} \circ \phi. \quad (4.6)$$

Hence, taking $\widetilde{M} = \begin{pmatrix} u(x_2) & 0_{1 \times (n-2)} \\ 0_{(n-2) \times 1} & I_{n-2} \end{pmatrix}^{-1} \cdot M$ and $\varphi = \phi^{-1}$ we recover (4.4). □

Following the conclusion of Lemma 4.3.3, the reader may wonder whether the projection of q in $\pi_{\mathfrak{C}}$ is always singular. This is clear when $g(x_n) = x_n^k$ for $0 < k < \infty$ since this implies $z_1^2 - z_2^{k+1} = 0$ and thus $\pi_{\mathfrak{C}}(q)$ is a singularity of the type A_{2k} . We next prove that the projection is also singular if $\text{ord}(g(z_n)) = \infty$.

Lemma 4.3.4. *Preserving the notation and the assumptions in Lemma 4.3.3, consider the function g defined in (4.4), if $\text{ord}(g(z_n)) = \infty$, then $\pi_{\mathfrak{C}}(q)$ is singular in $\pi_{\mathfrak{C}}(\mathfrak{C})$.*

Proof. Since $\text{ord}(g(z_n)) = \infty$, then **Case 1** in the proof of Lemma 4.3.3 holds. Moreover, we saw in the same proof that $\xi_2(z_n^2)$ (restricted to an open neighbourhood of $0 \in \mathbb{R}$) cannot be the zero function. This implies that neither is the function $g(z_n^2) = z_n^2 \xi(z_n^2)$, i.e., $g(z_n^2)$, restricted to an open neighbourhood of $0 \in \mathbb{R}$, is not the zero function. Assume for the sake of contradiction that $\pi_{\mathfrak{C}}(q)$ is smooth in $\pi_{\mathfrak{C}}(\mathfrak{C})$, then using the implicit function theorem, there exists a C^∞ -function defined in a neighbourhood of 0 in \mathbb{R} , with $f(0) = 0$ such that for a small neighbourhood of $\pi_{\mathfrak{C}}(q)$ in \mathbb{R}^2 , one of the following cases is satisfied:

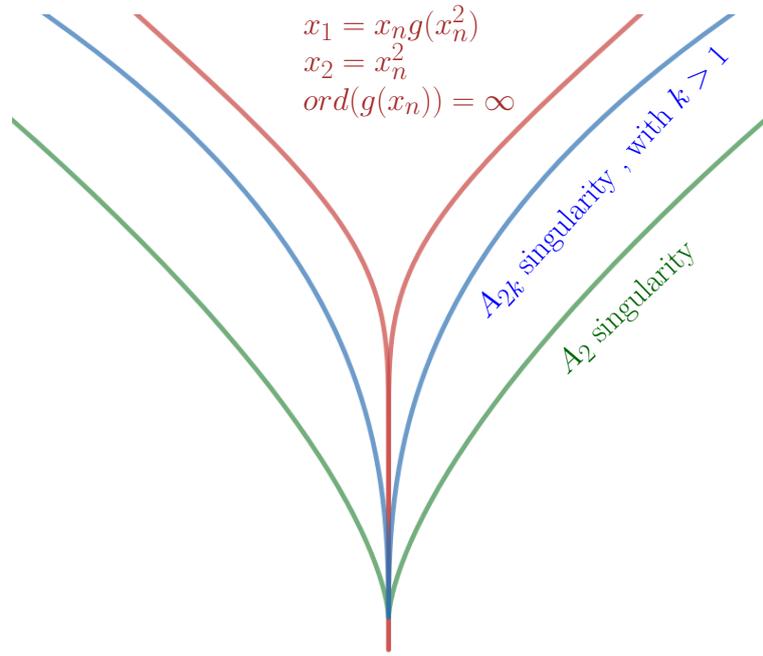


Figure 4.5: Comparing the singularities considered in Lemma 4.3.4 with A_{2k+1} types.

- (a) $f(z_1) = z_2 \iff (z_1, z_2) \in \pi_{\mathcal{C}}(\mathcal{C})$. Then, by (4.4), we have $f(z_n g(z_n^2)) = z_n^2$. Taking the second derivative of both sides with respect to z_n and then evaluating at 0 (recall that $\text{ord}(g(z_n)) = \infty$), we get the contradiction $0 = 2$.
- (b) $f(z_2) = z_1 \iff (z_1, z_2) \in \pi_{\mathcal{C}}(\mathcal{C})$. Then $f(z_n^2) = z_n g(z_n^2)$. The function $z_n g(z_n^2)$ is an odd function but not the zero function, and on the other hand $f(z_2)$ is an even function, which leads to a contradiction.

Thus, in both cases we have a contradiction, that is, f does not exist and $\pi_{\mathcal{C}}(q)$ cannot be smooth in $\pi_{\mathcal{C}}(\mathcal{C})$. \square

Returning to (4.4), notice that φ is defined in such a way that it preserves the singularity class of $\pi_{\mathcal{C}}(\mathcal{C})$ at the point $\pi_{\mathcal{C}}(q)$. In other words, if C is the plane projection of the curve defined by the Q then $(\pi_{\mathcal{C}}(\mathcal{C}), 0)$ and $(C, 0)$ are equivalent. As a corollary of Lemmas 3.2.6, 3.2.7, 4.3.3 and 4.3.4, the points of \mathcal{C} in $\mathcal{L}_c \cup \mathcal{L}_n$ are projected to the singular points of $\pi_{\mathcal{C}}(\mathcal{C})$.

Corollary 4.3.5. *If P satisfies Assumptions \mathcal{A}_1 , \mathcal{A}_2 , \mathcal{A}_3 and \mathcal{A}_4 , then a point $q \in \mathcal{C}$ projects to a singular point in $\pi_{\mathcal{C}}(\mathcal{C})$ if and only if $q \in \mathcal{L}_c \cup \mathcal{L}_n$.*

Proof. If $q \in \mathcal{L}_c \cup \mathcal{L}_n$, then by Lemmas 3.2.7, 4.3.3, and 4.3.4, $\pi_{\mathcal{C}}(q)$ is singular in $\pi_{\mathcal{C}}(\mathcal{C})$. If $q \notin \mathcal{L}_c \cup \mathcal{L}_n$, then by Lemma 3.2.6, $\pi_{\mathcal{C}}(q)$ is smooth in $\pi_{\mathcal{C}}(\mathcal{C})$. \square

Finally, we prove that the solutions of the Ball system project to the singular points of $\pi_{\mathcal{C}}(\mathcal{C})$.

Proof of Theorem 4.1.3: By Corollary 4.3.5, if (x_1, x_2) is singular in $\pi_{\mathcal{C}}(\mathcal{C})$, then there exists a point $q_1 \in \mathcal{L}_c \cup \mathcal{L}_n$, with $\pi_{\mathcal{C}}(q_1) = (x_1, x_2)$. If $q_1 \in \mathcal{L}_c$, let $q_2 = q_1$ and otherwise let q_2 be the unique (by Assumption \mathcal{A}_3) point in \mathcal{L}_n , distinct from q_1 , that projects onto (x_1, x_2) , i.e. $\pi_{\mathcal{C}}(q_1) = \pi_{\mathcal{C}}(q_2) = (x_1, x_2)$. Hence, (q_1, q_2) is in $\widehat{\mathcal{L}}$. Since Ω_P is surjective (Lemma 4.1.8), there exists $X = (x_1, x_2, y, r, t) \in \text{Sol}_{\text{Ball}(P)}$ with $\Omega_P(X) = (q_1, q_2)$.

On the other hand, if X is a solution of $\text{Ball}(P)$, then by Lemma 4.1.6 the pair $(q_1, q_2) = \Omega_P(X)$ is in $\widehat{\mathcal{L}}$. Hence, $q_1 = (x_1, x_2, y + r\sqrt{t}) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}^{n-2}$ is in $\mathcal{L}_c \cup \mathcal{L}_n$. Hence, by Corollary 4.3.5 the point (x_1, x_2) is singular in $\pi_{\mathcal{C}}(\mathcal{C})$. \square

4.4 Regularity of the Ball system

In this section, our goal is to prove Theorem 4.4.2 determining, under generic assumptions, necessary and sufficient conditions for $\text{Ball}(P)$ to be regular. We first recall the definition of a regular system.

Definition 4.4.1. For some integer $m \leq n$, let $F = (f_1, \dots, f_m)$ be a vector of smooth real-valued functions that are defined in \mathbb{R}^n and let $a \in \mathbb{R}^n$ be a solution of the system $\{F = 0\}$. We say that the latter system is regular at $a \in \mathbb{R}^n$ if the rank of its Jacobian matrix, evaluated at a , equals to m . We call $\{F = 0\}$ regular if it is regular at all of its solutions.

Theorem 4.4.2. Let $P \in C^\infty(\mathbb{R}^n, \mathbb{R}^{n-1})$ that satisfies Assumptions $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$ and \mathcal{A}_4 , then P satisfies Assumption \mathcal{A}_5 if and only if $\text{Ball}(P)$ is regular in B_{Ball} .

In order to prove Theorem 4.4.2, we are going to show that the Jacobian matrices of $\text{Ball}(P)$ and $\text{Ball}(Q)$ evaluated at X have the same rank, where Q is defined in Equation (4.4). Recall that Equation (4.4) implies that P and Q define the same curve around q . Notice also that if $X = (q, r, 0) \in \mathbb{R}^n \times \mathbb{R}^{n-2} \times \mathbb{R}$ is in $\Omega_P^{-1}((q, q))$, then $X \in \Omega_Q^{-1}((q, q))$.

Lemma 4.4.3. Let P and Q be as defined in (4.4). Under Assumption \mathcal{A}_1 , let $(q, r, 0) \in \mathbb{R}^n \times \mathbb{R}^{n-2} \times \mathbb{R}$ be a solution of the system $\text{Ball}(P)$ in B_{Ball} , then $\text{Ball}(P)$ is regular at $(q, r, 0)$ if and only if $\text{Ball}(Q)$ is regular at the point $(0, r, 0) \in \mathbb{R}^n \times \mathbb{R}^{n-2} \times \mathbb{R}$ (recall that for simplicity, we assume in Lemma 4.3.3 that $q = 0 \in \mathbb{R}^n$).

Proof. Let us write $X = (q, r, 0)$. We are going to prove that the Jacobian matrices of $\text{Ball}(P)$ and $\text{Ball}(Q)$ evaluated at X have the same rank. By Remark 4.1.9 we have that $\Omega_P(X) = (q, q) \in \widehat{\mathfrak{L}}_c$ (see Definitions 4.1.7 and 4.1.5), and hence, $q \in \mathfrak{L}_c$. By Lemma 4.1.6 we have that $(0, 0, r) \in T_q \mathfrak{C}$. We prove the claim in three steps:

Step 1: Let $\widetilde{M} = (f_{ij})_{1 \leq i, j \leq n-1}$ be as defined in the Equality (4.4). We define $S \cdot \widetilde{M}$ (resp. $D \cdot \widetilde{M}$) to be the matrix $(S \cdot f_{ij})_{1 \leq i, j \leq n-1}$ (resp. $(D \cdot f_{ij})_{1 \leq i, j \leq n-1}$). Using the identity $\frac{1}{2}(ab + cd) = \frac{1}{4}(a + c)(b + d) + \frac{1}{4}(a - c)(b - d)$, one deduces the properties for any $f, g \in C^\infty(\mathbb{R}^n, \mathbb{R})$:

$$S \cdot fg = (S \cdot f)(S \cdot g) + t(D \cdot f)(D \cdot g) \quad (4.1)$$

$$D \cdot fg = (D \cdot f)(S \cdot g) + (S \cdot f)(D \cdot g) \quad (4.2)$$

These identities applied to Equation (4.4) yield

$$\begin{pmatrix} S \cdot Q_1 \\ \cdots \\ \cdots \\ S \cdot Q_{n-1} \end{pmatrix} = \begin{pmatrix} S \cdot \widetilde{M} & tD \cdot \widetilde{M} \end{pmatrix} \cdot \begin{pmatrix} S \cdot (P_1 \circ \varphi) \\ \cdots \\ \cdots \\ S \cdot (P_{n-1} \circ \varphi) \\ D \cdot (P_1 \circ \varphi) \\ \cdots \\ \cdots \\ D \cdot (P_{n-1} \circ \varphi) \end{pmatrix}$$

and

$$\begin{pmatrix} D \cdot Q_1 \\ \cdots \\ \cdots \\ D \cdot Q_{n-1} \end{pmatrix} = \begin{pmatrix} D \cdot \widetilde{M} & S \cdot \widetilde{M} \end{pmatrix} \cdot \begin{pmatrix} S \cdot (P_1 \circ \varphi) \\ \cdots \\ \cdots \\ S \cdot (P_{n-1} \circ \varphi) \\ D \cdot (P_1 \circ \varphi) \\ \cdots \\ \cdots \\ D \cdot (P_{n-1} \circ \varphi) \end{pmatrix}$$

Combining the last two equalities:

$$\begin{pmatrix} S \cdot Q_1 \\ \dots \\ \dots \\ S \cdot Q_{n-1} \\ D \cdot Q_1 \\ \dots \\ \dots \\ D \cdot Q_{n-1} \end{pmatrix} = \begin{pmatrix} S \cdot \widetilde{M} & tD \cdot \widetilde{M} \\ D \cdot \widetilde{M} & S \cdot \widetilde{M} \end{pmatrix} \cdot \begin{pmatrix} S \cdot (P_1 \circ \varphi) \\ \dots \\ \dots \\ S \cdot (P_{n-1} \circ \varphi) \\ D \cdot (P_1 \circ \varphi) \\ \dots \\ \dots \\ D \cdot (P_{n-1} \circ \varphi) \end{pmatrix} \quad (4.3)$$

Notice that $\begin{pmatrix} S \cdot \widetilde{M} & tD \cdot \widetilde{M} \\ D \cdot \widetilde{M} & S \cdot \widetilde{M} \end{pmatrix}_X = \begin{pmatrix} \widetilde{M}(q) & 0 \\ D \cdot \widetilde{M}(X) & \widetilde{M}(q) \end{pmatrix}$ (recall that in our case we have $S \cdot \widetilde{M}(X) = \widetilde{M}(q)$) and that the latter matrix has an inverse (recall that, by Lemma 4.3.3, $\widetilde{M}(q)$ is an invertible matrix of size $n - 1$), namely, $\begin{pmatrix} \widetilde{M}(q)^{-1} & 0 \\ -\widetilde{M}(q)^{-1} \cdot (D \cdot \widetilde{M})(X) \cdot \widetilde{M}(q)^{-1} & \widetilde{M}(q)^{-1} \end{pmatrix}$ which implies (by continuity of the determinant function) that $\begin{pmatrix} S \cdot \widetilde{M} & tD \cdot \widetilde{M} \\ D \cdot \widetilde{M} & S \cdot \widetilde{M} \end{pmatrix}$ is invertible in a neighbourhood of X .

Step 2: Writing $y = (y_3, \dots, y_n)$ and $r = (r_3, \dots, r_n)$, consider the diffeomorphism φ defined in Lemma 4.3.3 and define the smooth function ψ over an open subset of \mathbb{R}^{2n-1} containing X which maps the point (x_1, x_2, y, r, t) to $(\varphi_1, \varphi_2, S \cdot \varphi_3, \dots, S \cdot \varphi_n, D \cdot \varphi_3, \dots, D \cdot \varphi_n, t)$. Notice that we have:

$$S \cdot (P_j \circ \varphi) = (S \cdot P) \circ \psi \text{ and } D \cdot (P_j \circ \varphi) = (D \cdot P) \circ \psi, \text{ for } 1 \leq j \leq n-1, \quad (4.4)$$

since $\varphi_i(x_1, x_2, y \pm r\sqrt{t}) = \psi_i \pm \psi_{n+i-2}\sqrt{\psi_{2n-1}}$ for all $i \in \{3, \dots, n\}$. In fact, using the last two equalities we can also see that ψ^{-1} exists and is smooth. Thus, ψ is a diffeomorphism.

Step 3: Now, comparing (4.3) with (4.4) we get:

$$SD \cdot Q := \begin{pmatrix} S \cdot Q_1 \\ \dots \\ \dots \\ S \cdot Q_{n-1} \\ D \cdot Q_1 \\ \dots \\ \dots \\ D \cdot Q_{n-1} \end{pmatrix} = \begin{pmatrix} S \cdot \widetilde{M} & tD \cdot \widetilde{M} \\ D \cdot \widetilde{M} & S \cdot \widetilde{M} \end{pmatrix} \cdot \begin{pmatrix} S \cdot P_1 \\ \dots \\ \dots \\ S \cdot P_{n-1} \\ D \cdot P_1 \\ \dots \\ \dots \\ D \cdot P_{n-1} \end{pmatrix} \circ \psi.$$

Consider the vector $SD \cdot P = (S \cdot P_1, \dots, S \cdot P_{n-1}, D \cdot P_1, \dots, D \cdot P_{n-1})^T$ and let $J_{SD \cdot P}$, $J_{SD \cdot Q}$ and J_ψ be the Jacobian matrices of $SD \cdot P$, $SD \cdot Q$ and ψ respectively. Taking the Jacobian matrix of both sides of the last equality yields:

$$J_{SD \cdot Q} = \begin{pmatrix} S \cdot \widetilde{M} & tD \cdot \widetilde{M} \\ D \cdot \widetilde{M} & S \cdot \widetilde{M} \end{pmatrix} \cdot J_{SD \cdot P} \cdot J_\psi + \text{Jacobian} \left(\begin{pmatrix} S \cdot \widetilde{M} & tD \cdot \widetilde{M} \\ D \cdot \widetilde{M} & S \cdot \widetilde{M} \end{pmatrix} \right) \cdot \begin{pmatrix} S \cdot P_1 \\ \dots \\ \dots \\ S \cdot P_{n-1} \\ D \cdot P_1 \\ \dots \\ \dots \\ D \cdot P_{n-1} \end{pmatrix} \circ \psi.$$

Evaluating the last equality at $X = (0, r, 0)$ and using the fact that $\psi(X) = \psi(0, r, 0) = (0, r, 0) = X$, we note that the second term of the right-hand side is zero. One thus has:

$$J_{SD \cdot Q}(X) = \begin{pmatrix} S \cdot \widetilde{M} & tD \cdot \widetilde{M} \\ D \cdot \widetilde{M} & S \cdot \widetilde{M} \end{pmatrix}_X \cdot J_{SD \cdot P}(X) \cdot J_\psi(X). \quad (4.5)$$

Computing $J_\psi(X)$, we get $J_\psi(X) = \left(\begin{array}{c|c} \frac{\partial \varphi_1}{\partial z_1}(0) & \frac{\partial \varphi_1}{\partial z_2}(0) \quad 0_{1 \times (2n-3)} \\ \hline 0_{(2n-2) \times 1} & I_{2n-2} \end{array} \right)$, with $\frac{\partial \varphi_1}{\partial z_1}(0) \neq 0$ according to the formula in Lemma 4.3.3.

Hence by Equation (4.5), it is straightforward to check that:

$$\begin{aligned} J_{\text{Ball}(Q)} &= \begin{pmatrix} J_{SD \cdot Q}(X) \\ 2X \end{pmatrix} = \begin{pmatrix} S \cdot \widetilde{M} & tD \cdot \widetilde{M} & 0 \\ D \cdot \widetilde{M} & S \cdot \widetilde{M} & 0 \\ 0_{1 \times (n-1)} & 0_{1 \times (n-1)} & 1 \end{pmatrix}_X \cdot \begin{pmatrix} J_{SD \cdot P}(X) \\ 2X \end{pmatrix} \cdot J_\psi(X) \\ &= \begin{pmatrix} S \cdot \widetilde{M} & tD \cdot \widetilde{M} & 0 \\ D \cdot \widetilde{M} & S \cdot \widetilde{M} & 0 \\ 0_{1 \times (n-1)} & 0_{1 \times (n-1)} & 1 \end{pmatrix}_X \cdot J_{\text{Ball}(P)}(X) \cdot J_\psi(X). \end{aligned} \quad (4.6)$$

Recalling that $J_\psi(X)$ and $\begin{pmatrix} S \cdot \widetilde{M} & tD \cdot \widetilde{M} & 0 \\ D \cdot \widetilde{M} & S \cdot \widetilde{M} & 0 \\ 0_{1 \times (n-1)} & 0_{1 \times (n-1)} & 1 \end{pmatrix}_X$ are invertible matrices, the proof of the lemma follows. \square

Now, we are ready to prove Theorem 4.4.2 which characterizes the regularity of the solutions of $\text{Ball}(P)$ under generic assumptions. We split the proof in two

Lemmas 4.4.6 and 4.4.7. Before that, we introduce a new assumption that helps to simplify the proof.

Definition 4.4.4. Let $(q_1, q_2) \in \widehat{\mathfrak{L}}$. We say that (q_1, q_2) satisfies Assumption $\mathcal{A}_5^{-'}$ if q_1 and q_2 are isolated in $\mathfrak{L}_n \cup \mathfrak{L}_c$ and any of the following conditions is satisfied:

- (a) If $(q_1, q_2) \in \widehat{\mathfrak{L}}_n$, then the plane projections of the tangent lines of q_1 and q_2 to \mathfrak{C} are linearly independent.
- (b) If $(q_1, q_2) \in \widehat{\mathfrak{L}}_c$, then the plane projection of a small enough neighbourhood of q_1 in \mathfrak{C} is an ordinary cusp at $\pi_{\mathfrak{C}}(q_1)$ and the multiplicity of the system $\{P(x) = 0, (x_1, x_2) = \pi_{\mathfrak{C}}(q_1)\}$ at q_1 is two.

Remark 4.4.5. Assumption $\mathcal{A}_5^{-'}$ can be seen as a “local version” of Assumption \mathcal{A}_5^- . We are going to prove that if Assumptions \mathcal{A}_1 , \mathcal{A}_2 , \mathcal{A}_3 and \mathcal{A}_4 are satisfied, then Assumption \mathcal{A}_5^- is equivalent to the condition that Assumption $\mathcal{A}_5^{-'}$ is satisfied for all $\widehat{\mathfrak{L}}$.

The main reason behind introducing Assumption $\mathcal{A}_5^{-'}$, is that we are going to prove in Lemma 4.4.6 that, under Assumption \mathcal{A}_1 , a pair $(q_1, q_2) \in \widehat{\mathfrak{L}}$ satisfies Assumption $\mathcal{A}_5^{-'}$ if and only if every X in $\Omega_P^{-1}((q_1, q_2))$ is a regular solution of $\text{Ball}(P)$, whereas Assumption \mathcal{A}_5^- is, in general, not sufficient for the regularity of the solutions of $\text{Ball}(P)$. For example, take $n = 3$ and $P = (x_1 - (x_3 - 1)^6, x_2 - (x_3 - 1)^9)$ (Figure 4.6). We can see that P satisfies Assumption \mathcal{A}_1 , the set \mathfrak{L}_c consists of a unique point $q = (0, 0, 1)$ and the set \mathfrak{L}_n is empty. The plane projection of \mathfrak{C} is the curve given by the equation $x_1^3 - x_2^2 = 0$. Hence, Assumption \mathcal{A}_5^- is satisfied. However, the multiplicity of the system $S = \{P(x_1, x_2, x_3) = 0 \in \mathbb{R}^2, x_1 = x_2 = 0\}$ at the point q equals 6 (Lemma 4.3.2). Hence, Assumption $\mathcal{A}_5^{-'}$ is not satisfied and one can also check that $\text{Ball}(P)$ is not regular.

Lemma 4.4.6. Let $P \in C^\infty(\mathbb{R}^n, \mathbb{R}^{n-1})$ that satisfies Assumption \mathcal{A}_1 . Let X be a solution of $\text{Ball}(P)$ and $(q_1, q_2) = \Omega_P(X)$ (Definition 4.1.7), then X is a regular solution of $\text{Ball}(P)$ if and only if (q_1, q_2) satisfies Assumption $\mathcal{A}_5^{-'}$.

Proof. Let $X = (x_1, x_2, y, r, t) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}^{n-2} \times \mathbb{R}^{n-2} \times \mathbb{R}$ be a solution of $\text{Ball}(P)$. We consider two cases:

Case (a): $t \neq 0$, i.e., $q_1 \neq q_2$.

It is easy to see that $\frac{\partial(S \cdot P_i)}{\partial x_j}, \frac{\partial(D \cdot P_i)}{\partial x_j}, \frac{\partial(S \cdot P_i)}{\partial r_k}, \frac{\partial(D \cdot P_i)}{\partial r_k}, \frac{\partial(S \cdot P_i)}{\partial t}, \frac{\partial(D \cdot P_i)}{\partial t}$ are, respectively, equal to: $S \cdot \frac{\partial(P_i)}{\partial x_j}, D \cdot \frac{\partial(P_i)}{\partial x_j}, t \cdot D \cdot \frac{\partial(P_i)}{\partial x_k}, S \cdot \frac{\partial(P_i)}{\partial x_k}, \frac{1}{2} \sum_{m=3}^n D \cdot \left(\frac{\partial P_i}{\partial x_m}\right) \cdot r_m, \frac{1}{2t} \left[\sum_{m=3}^n S \cdot \left(\frac{\partial P_i}{\partial x_m}\right) \cdot r_m - D \cdot P_i \right]$. Hence, by computing the Jacobian matrix of the $\text{Ball}(P)$ we get the matrix:

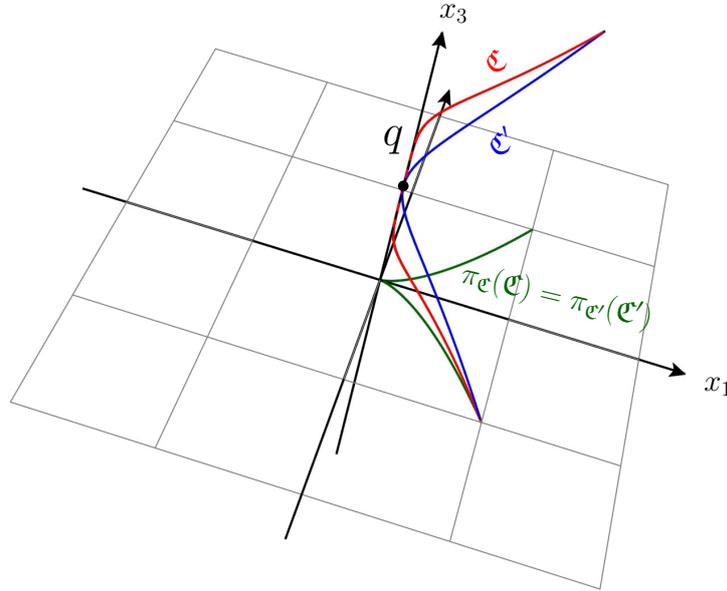


Figure 4.6: Comparing the curve \mathcal{C} defined in Remark 4.4.5 with \mathcal{C}' that is defined by $x_1 - (x_3 - 1)^2 = x_2 - (x_3 - 1)^3 = 0$ where both of them have the same plane projection which contains an ordinary cusp; however, the multiplicities of S (defined in the same remark) at q differ. In this example, the pair (q, q) satisfies Assumption \mathcal{A}_5^- for \mathcal{C}' , however, it does not for \mathcal{C} .

$$\begin{pmatrix} S \cdot \frac{\partial(P_1)}{\partial x_1} & \dots & S \cdot \frac{\partial P_1}{\partial x_n} & t \cdot D \cdot \frac{\partial(P_1)}{\partial x_3} \dots & t \cdot D \cdot \frac{\partial(P_1)}{\partial x_n} & \frac{1}{2} \sum_{m=3}^n D \cdot \left(\frac{\partial P_1}{\partial x_m}\right) \cdot r_m \\ \dots & \dots & \dots & \dots & \dots & \dots \\ S \cdot \frac{\partial(P_{n-1})}{\partial x_1} & \dots & S \cdot \frac{\partial(P_{n-1})}{\partial x_n} & t \cdot D \cdot \frac{\partial(P_{n-1})}{\partial x_3} \dots & t \cdot D \cdot \frac{\partial(P_{n-1})}{\partial x_n} & \frac{1}{2} \sum_{m=3}^n D \cdot \left(\frac{\partial P_{n-1}}{\partial x_m}\right) \cdot r_m \\ D \cdot \frac{\partial(P_1)}{\partial x_1} & \dots & D \cdot \frac{\partial(P_1)}{\partial x_n} & S \cdot \frac{\partial(P_1)}{\partial x_3} \dots & S \cdot \frac{\partial(P_1)}{\partial x_n} & \frac{1}{2t} \left[\sum_{m=3}^n S \cdot \left(\frac{\partial P_1}{\partial x_m}\right) \cdot r_m - D \cdot P_1 \right] \\ \dots & \dots & \dots & \dots & \dots & \dots \\ D \cdot \frac{\partial(P_{n-1})}{\partial x_1} & \dots & D \cdot \frac{\partial(P_{n-1})}{\partial x_n} & S \cdot \frac{\partial(P_{n-1})}{\partial x_3} \dots & S \cdot \frac{\partial(P_{n-1})}{\partial x_n} & \frac{1}{2t} \left[\sum_{m=3}^n S \cdot \left(\frac{\partial P_{n-1}}{\partial x_m}\right) \cdot r_m - D \cdot P_{n-1} \right] \\ 0 & \dots & 0 & 2r_3 \dots & 2r_n & 0 \end{pmatrix}.$$

We denote by C_i (resp. L_i) the i -th column (resp. line) of the latter matrix. Replace the last column C_{2n-1} with $\sum_{m=1}^{n-2} \frac{r_{m+2}}{2t} C_{n+m} + C_{2n-1}$, also for all integers $1 \leq k \leq n-1$ we replace the line L_k with $L_k + \sqrt{t} \cdot L_{k+n-1}$ and then the line L_{k+n-1} with

$L_k - 2\sqrt{t}L_{k+n-1}$. The resulting matrix is:

$$\begin{pmatrix} \frac{\partial(P_1)}{\partial x_1}(q_1) & \cdots & \frac{\partial P_1}{\partial x_n}(q_1) & \sqrt{t} \cdot \frac{\partial(P_1)}{\partial x_3}(q_1) & \cdots & \sqrt{t} \frac{\partial(P_1)}{\partial x_n}(q_1) & 0 \\ & & \cdots & \cdots & \cdots & \cdots & \cdots \\ & & \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\partial(P_{n-1})}{\partial x_1}(q_1) & \cdots & \frac{\partial P_{n-1}}{\partial x_n}(q_1) & \sqrt{t} \cdot \frac{\partial(P_{n-1})}{\partial x_3}(q_1) & \cdots & \sqrt{t} \frac{\partial(P_{n-1})}{\partial x_n}(q_1) & 0 \\ \frac{\partial(P_1)}{\partial x_1}(q_2) & \cdots & \frac{\partial(P_1)}{\partial x_n}(q_2) & -\sqrt{t} \frac{\partial(P_1)}{\partial x_3}(q_2) & \cdots & -\sqrt{t} \frac{\partial(P_1)}{\partial x_n}(q_2) & 0 \\ \cdots & & \cdots & \cdots & \cdots & \cdots & \cdots \\ & & \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\partial(P_{n-1})}{\partial x_1}(q_2) & \cdots & \frac{\partial(P_{n-1})}{\partial x_n}(q_2) & -\sqrt{t} \frac{\partial(P_{n-1})}{\partial x_3}(q_2) & \cdots & -\sqrt{t} \frac{\partial(P_{n-1})}{\partial x_n}(q_2) & 0 \\ 0 \cdots & & 0 & 2r_3 & \cdots & 2r_n & \frac{1}{2t} \end{pmatrix}.$$

The determinant of the latter matrix is zero if and only if the determinant of the following matrix is zero:

$$M_0 = \begin{pmatrix} N_P(q_1) & M_P(q_1) & M_P(q_1) \\ N_P(q_2) & M_P(q_2) & -M_P(q_2) \end{pmatrix}, \text{ where } M_P(q_1), M_P(q_2) \text{ are the}$$

sub-matrices that are obtained, respectively, by removing the first two columns from $J_P(q_1), J_P(q_2)$ and $N_P(q_1), N_P(q_2)$ are the matrices formed by the first two columns of $J_P(q_1), J_P(q_2)$, respectively. By linear operations on M_0 , we can see that M_0 has same rank as the matrix $M(q_1, q_2)$ (see Definition 3.2.8). Thus, X is regular for $\text{Ball}(P)$ if and only if $M(q_1, q_2)$ is invertible. By Lemma 3.2.9 we have that $M(q_1, q_2)$ is invertible if and only if none of q_1, q_2 is in \mathfrak{L}_c (and hence none of the plane projections of $T_{q_1}\mathfrak{C}, T_{q_2}\mathfrak{C}$ is trivial) and the plane projection of their tangent spaces are different. Equivalently, the pair (q_1, q_2) is in $\widehat{\mathfrak{L}}_n$ and satisfies Assumption \mathcal{A}_5^- .

Case (b): $t = 0$, i.e., $q_1 = q_2$.

Let us write $q = q_1$. We prove the claim in three steps:

Step 1: We first simplify P . Without loss of generality and by Lemma 4.3.1 we can assume that $q = 0$ and P_1, \dots, P_{n-1} are, respectively, equal to $x_1 - f_1(x_n), x_2 - f_2(x_n), \dots, x_{n-1} - f_{n-1}(x_n)$ with the property that $\min\{\text{mult}(f_1), \text{mult}(f_2)\} \geq 2$. For all $i \in \{3, \dots, n-1\}$, using Taylor's theorem, we can write $f_i(x_n) = \sum_{j=1}^3 a_{i,j} x_n^j + x_n^4 h_i(x_n)$, for some $a_{i,j} \in \mathbb{R}$ and smooth functions $h_i(x_n)$. Since

$\min\{\text{mult}(f_1), \text{mult}(f_2)\} \geq 2$, we can write $f_1(x_n) = \sum_{j=2}^3 \alpha_j x_n^j + x_n^4 h_1(x_n)$ and

$f_2(x_n) = \sum_{j=2}^3 \beta_j x_n^j + x_n^4 h_2(x_n)$. Notice that

$$(f_1(x_n), f_2(x_n), f_3(x_n), \dots, f_{n-1}(x_n), x_n)$$

is a local parameterization system of \mathfrak{C} around q . Since $\dim(T_q\mathfrak{C}) = 1$ (Assump-

tion \mathcal{A}_1), there exists $\lambda \in \mathbb{R}^*$ with $(a_{3,1}, \dots, a_{n-1,1}, 1) = \lambda r$ (because the vectors $(0, 0, r) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}^{n-2}$ and $(0, 0, a_{1,3}, \dots, a_{1,n-1}, 1)$ are in $T_q \mathcal{C} \setminus \{0\}$). In particular, $r_n \neq 0$.

Step 2: Now, we compute $J_{\text{Ball}(P)}$ at $X = (x_1, x_2, y, r, 0)$ by first computing it for $X_t = (x_1, x_2, y, r, t)$ with $t \neq 0$, and then taking the limit when t goes to 0. Since

the operator S is linear, we write $S(x_i - f_i(x_n)) = S(x_i - \sum_{j=1}^3 a_{i,j} x_n^j) - S(x_n^4 h_i(x_n))$.

On the other hand, using the identity (4.1) we deduce that $S(x_n^4 h_i(x_n)) = S(x_n^4) \cdot S(h_i(x_n)) + tD(x_n^4) \cdot D(h_i(x_n))$, for all $i \in \{1, \dots, n-1\}$. It is straightforward to see that $S(x_n^4) = r_n^4 t^2 + 6r_n^2 t x_n^2 + x_n^4$ and $tD(x_n^4) = 4r_n^3 x_n t^2 + 4r_n x_n^3 t$ with $r = (r_3, \dots, r_n)$. Hence, all of the first-order partial derivatives of $S(x_n^4 h_i(x_n))$, evaluated at X_t , converge to zero when t goes to 0. Hence, the partial derivatives of

the functions $S(x_i - f_i(x_n))$ and $S(x_i - \sum_{j=1}^3 a_{i,j} x_n^j)$ evaluated at X are equal. Using an analogous argument, we deduce that the evaluation of the partial derivatives of

the functions $D(x_i - f_i(x_n))$ and $D(x_i - \sum_{j=1}^3 a_{i,j} x_n^j)$, at X are also equal. Thus,

$J_{\text{Ball}(P)}(X_t)$ and $J_{\text{Ball}(\bar{P})}(X_t)$ converge to the same limit $J_{\text{Ball}(P)}(X)$, where \bar{P} is the function obtained by truncating P beyond degree 3 with respect to the variable x_n .

Computing $J_{\text{Ball}(P)}(X) = \lim_{t \rightarrow 0} J_{\text{Ball}(\bar{P})}(X_t)$, we get:

$$\begin{pmatrix} 1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & -\alpha_2 r_n^2 \\ 0 & 1 & \dots & 0 & 0 & 0 & \dots & 0 & -\beta_2 r_n^2 \\ 0 & 0 & \dots & \dots & -a_{3,1} & 0 & \dots & 0 & -a_{3,2} r_n^2 \\ \dots & \dots \\ \dots & \dots \\ 0 & 0 & \dots & 1 & -a_{n-1,1} & 0 & \dots & 0 & -a_{n-1,2} r_n^2 \\ 0 & 0 & \dots & \dots & -2\alpha_2 r_n & 0 & \dots & 0 & -\alpha_3 r_n^3 \\ 0 & 0 & \dots & \dots & -2\beta_2 r_n & 0 & \dots & 0 & -\beta_3 r_n^3 \\ 0 & 0 & \dots & \dots & -2a_{3,2} r_n & 1 & \dots & -a_{3,1} & -a_{3,3} r_n^3 \\ \dots & \dots \\ \dots & \dots \\ 0 & 0 & \dots & \dots & -2a_{n-1,2} r_n & 0 & \dots & 1 & -a_{n-1,1} & -a_{n-1,3} r_n^3 \\ 0 & 0 & \dots & \dots & 0 & 2r_3 & \dots & 2r_{n-1} & 2r_n & 0 \end{pmatrix}.$$

Hence, observing that the matrix is block diagonal, its determinant is zero if and

only if the determinant of the following one is:

$$\begin{pmatrix} -2\alpha_2 r_n & 0 & \dots & 0 & 0 & -\alpha_3 r_n^3 \\ -2\beta_2 r_n & 0 & \dots & 0 & 0 & -\beta_3 r_n^3 \\ -2a_{3,2} r_n & 1 & 0 \dots & 0 & -a_{3,1} & -a_{3,3} r_n^3 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ -2a_{n-1,2} r_n & 0 & 0 \dots & 1 & -a_{n-1,1} & -a_{n-1,3} r_n^3 \\ 0 & 2r_3 & \dots & 2r_{n-1} & 2r_n & 0 \end{pmatrix}.$$

Shifting the columns of the last matrix we get:

$$\begin{pmatrix} -\alpha_3 r_n^3 & -2\alpha_2 r_n & 0 & \dots & 0 & 0 \\ -\beta_3 r_n^3 & -2\beta_2 r_n & 0 & \dots & 0 & 0 \\ -a_{3,3} r_n^3 & -2a_{3,2} r_n & 1 & 0 \dots & 0 & -a_{3,1} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ -a_{n-1,3} r_n^3 & -2a_{n-1,2} r_n & 0 & 0 \dots & 1 & -a_{n-1,1} \\ 0 & 0 & 2r_3 & \dots & 2r_{n-1} & 2r_n \end{pmatrix}.$$

To compute the determinant of the second block, we expand it about the last row. Hence, the determinant of the last matrix is zero if and only if $r_n(\alpha_2\beta_3 - \alpha_3\beta_2)(r_n + \sum_{i=3}^{n-1} a_{i,1}r_i) = 0$. Notice that, by Step 1, we have that $r_n \neq 0$ and the third factor $(r_n + \sum_{i=3}^{n-1} a_{i,1}r_i)$ is never zero since it is equal to λ . Thus, $J_{\text{Ball}(P)}(X)$ is invertible

iff $\alpha_2\beta_3 - \alpha_3\beta_2 \neq 0$, equivalently, the matrix $A = \begin{pmatrix} \alpha_2 & \alpha_3 \\ \beta_2 & \beta_3 \end{pmatrix}$ is invertible.

Step 3: We now show that the invertibility of A is equivalent to the condition that (q, q) satisfies Assumption \mathcal{A}_5^{-1} .

First assume that A is invertible. It follows that either $\alpha_2 \neq 0$ or $\beta_2 \neq 0$ and this yields that the minimum of the multiplicities of f_1 and f_2 is 2. By Lemma 4.3.2, the multiplicity of the system $\{P(x_1, x_2, y) = 0 \in \mathbb{R}^{n-1}, (x_1, x_2) = \pi_{\mathfrak{e}}(q)\}$ at q is equal to 2, thus Assumption \mathcal{A}_5^{-1} (b) is satisfied. Using the same notation as in the proof of Lemma 4.3.3, one can write $\tilde{f}_1(z_n) = z_n^2(\xi_1(z_n^2) + z_n\xi_2(z_n^2))$. Notice that $\xi_2(x_n^2)$ cannot be the zero function, otherwise $\tilde{f}_1(\epsilon) = \tilde{f}_1(-\epsilon)$ and $\tilde{f}_2(\epsilon) = \tilde{f}_2(-\epsilon)$ for all small enough $\epsilon > 0$, which means that X would be the limit of solutions X_ϵ of $\text{Ball}(P)$ with $\Omega_P(X_\epsilon) \in \hat{\mathcal{L}}_n$. X would then be a non-isolated solution and thus a non-regular solution of $\text{Ball}(P)$ which contradicts the assumption. We then have two cases as in Lemma 4.3.3. The first one is when $\text{mult}(\xi_2(z_n)) = \infty$, that would imply that $\alpha_2 = \alpha_3 = 0$ and contradicts the invertibility of A . We then must satisfy the second case $\text{mult}(\xi_2(z_n)) = k < \infty$ and, after a change of variables, the first equation of the system becomes equivalent to $z_1 - z_n^{2k+3} = 0$. The invertibility

of A implies that $k = 0$. The projection of the curve in the plane is thus locally parameterized by (z_n^3, z_n^2) and is an ordinary cusp, Assumption $\mathcal{A}_5^{-'}$ (a) is satisfied.

Second, assume that Assumption $\mathcal{A}_5^{-'}$ is satisfied. By Lemma 4.3.2 and Assumption $\mathcal{A}_5^{-'}$ (b), the minimum of the multiplicities of f_1 and f_2 is 2. Using the proof of Lemma 4.3.3 once again, one can assume that $f_2(z_n) = z_n^2$ and $f_1(z_n) = z_n g(z_n^2)$ or $f_1(z_n) = z_n^{2k+3}$. By Assumption $\mathcal{A}_5^{-'}$ (a), the projection is an ordinary cusp and thus has a parameterization of the form (z_n^2, z_n^3) , that is $f_1(z_n) = z_n^3$. This implies that A is equivalent to $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ and hence is invertible. \square

Lemma 4.4.7. *If Assumptions \mathcal{A}_1 , \mathcal{A}_2 , \mathcal{A}_3 and \mathcal{A}_4 are satisfied, then Assumption \mathcal{A}_5^- is satisfied if and only if Assumption $\mathcal{A}_5^{-'}$ is satisfied for all $(q_1, q_2) \in \widehat{\mathfrak{L}} \subset B \times B$.*

Proof. Assume that Assumption \mathcal{A}_5^- is satisfied and $(q_1, q_2) \in \widehat{\mathfrak{L}}$. If $(q_1, q_2) \in \widehat{\mathfrak{L}}_c$, then by Lemma 4.3.3 and Assumption \mathcal{A}_5^- we must have that the plane projection of a small enough neighborhood of q_1 in \mathfrak{C} is an ordinary cusp at $\pi_{\mathfrak{C}}(q_1)$. By Assumption \mathcal{A}_3 and Lemma 3.2.3, the multiplicity of the mentioned system at $q_1 = q_2$ is two. Thus, (q_1, q_2) satisfies Assumption $\mathcal{A}_5^{-'}$. If $(q_1, q_2) \in \widehat{\mathfrak{L}}_n$, then by Lemma 3.2.7 and Assumption \mathcal{A}_5^- , we have that $\pi_{\mathfrak{C}}(q_1)$ is a node in $\pi_{\mathfrak{C}}(\mathfrak{C})$. Thus, we have that $\pi_{\mathfrak{C}}(q_1)$ is a transverse intersection of two smooth branches of $\pi_{\mathfrak{C}}(\mathfrak{C})$. Those branches are the plane projections of two disjoint branches of \mathfrak{C} each of which contains either q_1 or q_2 . Hence, the plane projections of the tangent spaces of q_1 and q_2 to \mathfrak{C} are linearly independent. Thus, (q_1, q_2) satisfies Assumption $\mathcal{A}_5^{-'}$.

Assume conversely that $\mathcal{A}_5^{-'}$ is satisfied for all $(q_1, q_2) \in \widehat{\mathfrak{L}}$. By Corollary 4.3.5, any singular point of $\pi_{\mathfrak{C}}(\mathfrak{C})$ is the plane projection of a point $q_1 \in \mathfrak{L}_c \cup \mathfrak{L}_n$. For some $q_2 \in \mathfrak{C}$, the pair (q_1, q_2) is in $\widehat{\mathfrak{L}}$ (which satisfies Assumption $\mathcal{A}_5^{-'}$). Hence, if (q_1, q_2) is in $\widehat{\mathfrak{L}}_n$ (resp. in $\widehat{\mathfrak{L}}_c$) the plane projection of q_1 is a node (resp. an ordinary cusp) by Lemma 3.2.7 (resp. Lemma 4.3.3). \square

Lemmas 4.4.6 and 4.4.7 then imply Theorem 4.4.2.

In Chapter 7, we present examples that show how the Ball system encodes the singularities of $\pi_{\mathfrak{C}}(\mathfrak{C})$. The next chapter is dedicated to provide a method to examine whether a given curve satisfies the weak or strong assumptions.

5

Semi-algorithms to check the assumptions and isolate singularities

Contents

| | | |
|------------|---|-----------|
| 5.1 | Interval arithmetic | 84 |
| 5.2 | Semi-algorithms to check assumptions and isolate singularities | 85 |
| 5.3 | Semi-algorithms improvements | 92 |
| 5.3.1 | Subdivision-based solver on a high-dimensional system | 92 |
| 5.3.2 | Evaluating the operator D | 98 |
| 5.3.3 | Improvement integration | 99 |

In this chapter we present Semi-algorithm 3 that checks the weak assumptions of Section 3.1.2 and, if it terminates, outputs a superset of isolating boxes of the singularities of $\pi_{\mathcal{C}}(\mathcal{C})$. We also present Semi-algorithm 4 that checks the strong assumptions of Section 3.1.2 and, if it terminates, outputs a set of isolating boxes of the singularities of $\pi_{\mathcal{C}}(\mathcal{C})$. The main idea of these semi-algorithms comes from Theorems 4.1.3 and 4.4.2: Theorem 4.1.3 states that, under Assumptions \mathcal{A}_{1-4} , the singularities of $\pi_{\mathcal{C}}(\mathcal{C})$ are the plane projections of the solutions of $\text{Ball}(P)$. Theorem 4.4.2 states that, under the further Assumption \mathcal{A}_5^- , $\text{Ball}(P)$ is regular, so we can use certified numerical methods such as interval Newton methods [MKC2009] to solve $\text{Ball}(P)$. In addition, in order to verify these assumptions, we use subdivision approaches based on interval arithmetic in a semi-algorithm framework.

These semi-algorithms are theoretically relevant but they are not very practical. We thus present in Section 5.3 practical improvements and Semi-algorithm 6.

We present in Section 5.1 the basics of interval arithmetic with the notation and definitions by Lin and Yap [LY2011] and Semi-algorithm 3 and 4 in Section 5.2.

5.1 Interval arithmetic

Recall that for some positive integer k , by a closed (resp. open) k -box \mathfrak{B} , we mean the Cartesian product of k closed (resp. open) intervals. The width of a box \mathfrak{B} , denoted by $w(\mathfrak{B})$, is the maximal length of the intervals of that product. For a subset $A \subset \mathbb{R}^k$, the set IA is the set of all closed k -boxes that are contained in A .

Functions over boxes. ([MKC2009, §2.3]) closed n -boxes can be seen as an extension of \mathbb{R}^n . Hence, it would be interesting to speak about extending a function f defined on \mathbb{R}^n to a function $\square f$ over to $I\mathbb{R}^n$, preserving the “behavior” of a f . More formally:

Definition 5.1.1. For the positive integer m , $A \subseteq \mathbb{R}^k$, and a function $f : A \rightarrow \mathbb{R}^m$, the function $\square f : IA \rightarrow I\mathbb{R}^m$ is called an inclusion of f if the set $f(\mathfrak{B}) = \{f(x) \mid x \in \mathfrak{B}\}$ is contained in $\square f(\mathfrak{B})$, for all $\mathfrak{B} \in IA$. An inclusion $\square f$ of f is called a box function, if for any descending sequence of closed k -boxes $\mathfrak{B}_1 \supseteq \mathfrak{B}_2 \supseteq \dots$ that converges to a point $q \in \mathbb{R}^k$ (that is, $\bigcap_i \mathfrak{B}_i = \{q\}$), the sequence $\square f(\mathfrak{B}_1) \supseteq \square f(\mathfrak{B}_2) \supseteq \dots$ converges to $f(q)$.

Example 5.1.2. For the case of intervals, the first examples that come to mind are inclusions of arithmetic operations. For instance, the addition $+$, can be seen as a function over \mathbb{R}^2 to \mathbb{R} . Hence, the addition of two intervals $\mathfrak{B}_1 = [a, b]$ and $\mathfrak{B}_2 = [c, d]$ can be defined to be the interval $\mathfrak{B}_1 + \mathfrak{B}_2 = [a + c, b + d]$. This inclusion is a box function. The same can be said about multiplication and subtraction. Of course, one must be careful when defining the division since the dominator has to be zero-free. Addition and subtraction can be generalized over higher-dimensional boxes in a natural way. Moreover, outward rounding of an interval ([MKC2009, §3.2]) means to move the left endpoint to the left and the right endpoint to the right at the last digit carried. Outward rounding, when evaluating a box function, guarantees that these enclosures are rigorous when using fixed-point arithmetic (representation of real numbers approximation with a fixed number of digits). We are going to assume that outward rounding is used for any box evaluation performed.

In the rest of this chapter, we assume that we are given a box function $\square f$ for any function f we consider. The command *subdivide* is applied to a closed k -box $\overline{\mathfrak{B}}$, and it returns the set of boxes obtained by bisecting $\overline{\mathfrak{B}}$ in all dimensions.

An interval matrix $\square M$ is a matrix whose coefficients are intervals. It can also be seen as the set of all matrices whose (i, j) -th coefficients belong to the (i, j) -th interval. The rank of an interval matrix $\square M$, denoted by $\text{rank}(\square M)$, is the minimum of the ranks of all the matrices in this set.

5.2 Semi-algorithms to check assumptions and isolate singularities

This section is dedicated to prove the following theorem. Recall that the weak and strong assumptions are defined in Definition 3.1.1.

Theorem 5.2.1. *For an open n -box B and a smooth function P from \overline{B} to \mathbb{R}^{n-1} , Semi-algorithm 3 stops if and only if P satisfies the weak assumptions in \overline{B} and then it returns a set of isolating boxes of all the singularities of $\pi_{\mathcal{C}}(\mathcal{C})$, plus possibly other spurious disjoint boxes. Semi-algorithm 4 stops if and only if P satisfies the strong assumptions in \overline{B} and then it returns a set of isolating boxes of all the singularities of $\pi_{\mathcal{C}}(\mathcal{C})$.*

To check whether a given function P satisfies the weak assumptions (\mathcal{A}_1 , \mathcal{A}_2 , \mathcal{A}_3 , \mathcal{A}_4 and \mathcal{A}_5^-) in \overline{B} , we use their relation to the solutions of $\text{Ball}(P)$ studied in the previous sections. Recall that for any subset $A \subseteq \mathbb{R}^n$, we defined $A_{\text{Ball}} = \{(x_1, x_2, y, r, t) \mid t \geq 0, (x_1, x_2, y + r\sqrt{t}), (x_1, x_2, y - r\sqrt{t}) \in A, \|r\|^2 = 1\}$. Let B be an open n -box and P be a smooth function from \overline{B} to \mathbb{R}^{n-1} that satisfies Assumption \mathcal{A}_1 in \overline{B} . Consider the following assumptions:

- \aleph_1 - All solutions of $\text{Ball}(P)$ in $\overline{B}_{\text{Ball}}$ are regular.
- \aleph_2 - For every solution X of $\text{Ball}(P)$ in $\overline{B}_{\text{Ball}}$, none of the points of the pair $\Omega_P(X)$ (Definition 4.1.7) is in the boundary of B .
- \aleph_3 - No two distinct solutions of $\text{Ball}(P)$ in $\overline{B}_{\text{Ball}}$, except the twin solutions (Remark 4.1.10), have the same plane projection.

The next lemma shows the relation between these new assumptions and those of Section 3.1.2. The motivation of these alternative assumptions is that they are stated in terms of $\text{Ball}(P)$, which makes them easier to verify in our semi-algorithms.

Lemma 5.2.2. *Let B be an open n -box and P be a smooth function from \overline{B} to \mathbb{R}^{n-1} that satisfies Assumption \mathcal{A}_1 in \overline{B} . Then, Assumptions \aleph_1 , \aleph_2 and \aleph_3 are satisfied if and only if Assumptions \mathcal{A}_2 , \mathcal{A}_3 , \mathcal{A}_4 and \mathcal{A}_5^- are satisfied in \overline{B} .*

Proof. If Assumptions \mathcal{A}_2 , \mathcal{A}_3 , \mathcal{A}_4 and \mathcal{A}_5^- are satisfied in \overline{B} , then by Theorem 4.4.2 we have Assumption \aleph_1 is satisfied. Moreover, by Assumptions \mathcal{A}_2 and \mathcal{A}_4 we have that none of \mathcal{L}'_n , \mathcal{L}'_c intersects ∂B . By Definition 4.1.7, for any solution X of $\text{Ball}(P)$, we have that the points of the pair $\Omega_P(X)$ are in $\mathcal{L}'_n \cup \mathcal{L}'_c$ and hence are not in ∂B which implies that Assumption \aleph_2 is satisfied. Assume

that Assumption \aleph_3 is not satisfied, that is, there exist two distinct non-twin solutions X, X' that have the same plane projection $p \in \mathbb{R}^2$. Let $(q_1, q_2) = \Omega_P(X)$ and $(q'_1, q'_2) = \Omega_P(X')$. By Lemma 4.1.6, the pairs $(q_1, q_2), (q'_1, q'_2)$ are distinct and the points q_1, q_2, q'_1, q'_2 have the same plane projection p . By Assumption \aleph_3 , we cannot have three pairwise distinct points among q_1, q_2, q'_1, q'_2 . Moreover, if the multiplicity at all of the points q_1, q_2, q'_1, q'_2 is one, then $(q_1, q_2), (q'_1, q'_2)$ are in $\widehat{\mathcal{L}}_n$ and not distinct. Hence, at least a point say q_1 has multiplicity larger than one, i.e., $q_1 \in \mathcal{L}_c$ (Lemma 3.2.3). Hence, the number of solutions counted with multiplicity is at least three which contradicts Assumption \aleph_3 . Hence, Assumption \aleph_3 is satisfied.

Now, assume that Assumptions \aleph_1, \aleph_2 and \aleph_3 are satisfied. Since, by Assumption \aleph_1 , $\text{Ball}(P)$ is a regular square system, its solution set is a zero-dimensional manifold in the compact set $\overline{B}_{\text{Ball}(P)}$ (regular value theorem). Hence, $\text{Ball}(P)$ has a finite number of solutions in $\overline{B}_{\text{Ball}}$. Since Ω_P (Definition 4.1.7) is surjective (Lemma 4.1.8), the set $\widehat{\mathcal{L}}$ (Definition 4.1.5) is also finite. Hence, the set $\mathcal{L}_c \cup \mathcal{L}_n$ is finite (since $\mathcal{L}_c \cup \mathcal{L}_n$ is the image of $\widehat{\mathcal{L}}$ under the surjective function $(q_1, q_2) \rightarrow q_1$). Moreover, by Assumption \aleph_2 , the set $\mathcal{L}'_n \cup \mathcal{L}'_c$ does not intersect the boundary of B . Hence, Assumptions \aleph_2 and \aleph_4 are satisfied in \overline{B} . To prove that Assumption \aleph_3 is satisfied, let $p = (\alpha, \beta) \in \pi_{\mathcal{C}}(\mathcal{C})$ and $|\pi^{-1}(p)| \geq 3$. For pairwise distinct points $q_1, q_2, q_3 \in \pi^{-1}(p)$, by Lemma 4.1.6, we have that there exist two distinct non-twin solutions X, X' of $\text{Ball}(P)$, with $\Omega_P(X) = (q_1, q_2)$ and $\Omega_P(X') = (q_1, q_3)$ such that we have the same plane projection p which contradicts Assumption \aleph_3 . Hence, $\pi_{\mathcal{C}}^{-1}(p)$ consists of at most two distinct points. We consider two cases:

- (a) $\pi_{\mathcal{C}}^{-1}(p)$ has two distinct elements, say q_1, q_2 . By Lemma 4.1.6, the pair (q_1, q_2) is in $\widehat{\mathcal{L}}_n$, and hence, there exists a solution $X = (\alpha, \beta, y, r, t) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}^{n-2} \times \mathbb{R}^{n-2} \times \mathbb{R}$ of $\text{Ball}(P)$, with $t \neq 0$ and $\Omega_P(X) = (q_1, q_2)$. Since X is a regular solution (Assumption \aleph_1), by Lemma 4.4.6 we have that none of q_1, q_2 is in \mathcal{L}_c . Hence, by Lemma 3.2.3, the multiplicity of $\{P(x_1, x_2, y) = 0 \in \mathbb{R}^{n-1}, x_1 - \alpha = x_2 - \beta = 0\}$ at q_1 (resp. q_2) is one. Thus, the number of solutions counted with multiplicity is two.
- (b) $\pi_{\mathcal{C}}^{-1}(p)$ has a unique point q . Let m denote the multiplicity of the system $\{P(x_1, x_2, y) = 0 \in \mathbb{R}^{n-1}, x_1 - \alpha = x_2 - \beta = 0\}$ at q . If $m = 1$, then we are done. If $m > 1$, then by Lemma 3.2.3 we have that $q \in \mathcal{L}_c$. Hence, there exists a solution of $\text{Ball}(P)$ of the form $X = (\alpha, \beta, y, r, 0) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}^{n-2} \times \mathbb{R}^{n-2} \times \mathbb{R}$ such that $\Omega_P(X) = (q, q)$ (Lemma 4.1.8). Since X is regular (Assumption \aleph_1), by Lemma 4.4.6 we have that (q, q) satisfies assumption \aleph'_3 . In particular, the multiplicity m is equal to two.

Thus, for all $p \in \pi_{\mathcal{C}}(\mathcal{C})$ the sum of the multiplicities of the solutions in the system $\{P(x) = 0 \in \mathbb{R}^{n-1}, x_1 - \alpha = x_2 - \beta = 0\}$ is at most two, i.e., Assumption \mathcal{A}_3 is satisfied. Now, Since Assumptions $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3$ and \mathcal{A}_4 are satisfied and since all solutions of $\text{Ball}(P)$ are regular, by Theorem 4.4.2, we have that Assumption \mathcal{A}_5^- is also satisfied. \square

Using Lemma 5.2.2, we are ready to check Assumptions $\mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4$ and \mathcal{A}_5^- using \aleph_1, \aleph_2 and \aleph_3 . Since Lemma 5.2.2 requires Assumption \mathcal{A}_1 , we start by checking that assumption with Semi-algorithm 1 that is based on subdivision.

Semi-algorithm 1: Checking Assumption \mathcal{A}_1

Input: An open n -box B and a function P from \overline{B} to \mathbb{R}^{n-1} .

Termination: If and only if P satisfies Assumption \mathcal{A}_1 in \overline{B} .

Output: True

- 1: $L := \{\overline{B}\}$
 - 2: **while** $L \neq \emptyset$ **do**
 - 3: $\mathfrak{B} := \text{pop}(L)$
 - 4: **if** $0 \in \square P(\mathfrak{B})$ and $\text{rank}(\square J_P(\mathfrak{B})) < n - 1$ **then**
 - 5: Subdivide \mathfrak{B} and add its children to L .
 - 6: **return** True.
-

Lemma 5.2.3. *Semi-algorithm 1 stops if and only if P satisfies Assumption \mathcal{A}_1 in \overline{B} .*

Proof. If Semi-algorithm 1 stops, by the conditions in Step (4), the box \overline{B} is partitioned into two sets of boxes. A set of boxes that are disjoint with $\overline{\mathcal{C}}$ and the other one is a set of boxes that contain parts of $\overline{\mathcal{C}}$ that satisfy Assumption \mathcal{A}_1 . Thus, Assumption \mathcal{A}_1 is satisfied in \overline{B} . On the other hand, assume that P satisfies Assumption \mathcal{A}_1 in \overline{B} and Semi-algorithm 1 does not stop, then, for every positive real ϵ there exists a closed box $\overline{\mathfrak{B}}_\epsilon \subset \overline{B}$, with $w(\overline{\mathfrak{B}}_\epsilon) < \epsilon$ such that the conditions in Step (4) are satisfied in $\overline{\mathfrak{B}}_\epsilon$. Consider the infinite chain $\overline{\mathfrak{B}}_{\frac{1}{1}}, \overline{\mathfrak{B}}_{\frac{1}{2}}, \overline{\mathfrak{B}}_{\frac{1}{3}} \dots$ and take $q_k \in \overline{\mathfrak{B}}_{\frac{1}{k}}$, with $q_k \neq q_{k'}$ for $k \neq k'$. Since \overline{B} is compact, then there exists a subsequence of q_k that converges to a point on \overline{B} say q . Since $\square P$ and $\square J_P$ are box function we must have that $P(q) = 0$ and $\text{rank}(J_P(q)) < n - 1$. Thus, q is a point in $\overline{\mathcal{C}}$ that does not satisfy Assumption \mathcal{A}_1 which is a contradiction. Hence, Semi-algorithm 1 stops. \square

The next step is to check Assumptions \aleph_1 and \aleph_2 . For this goal, we want to find a finite set of pairwise disjoint boxes in $\overline{B}_{\text{Ball}}$ such that every box contains at

most one solution of $\text{Ball}(P)$ and the union of these boxes contains all solutions of $\text{Ball}(P)$ in $\overline{B}_{\text{Ball}}$. Notice that, by the definition of box functions, for a closed $(2n-1)$ -box $\overline{\mathfrak{U}}$, if $0 \notin \square \text{Ball}(P)(\overline{\mathfrak{U}})$, then $\overline{\mathfrak{U}}$ does not contain a solution of $\text{Ball}(P)$, whereas the condition $0 \in \square \text{Ball}(P)(\overline{\mathfrak{U}})$ does not necessarily imply that a solution is in $\overline{\mathfrak{U}}$. This is why the set we are going to find might have unnecessary boxes. However, we will see later that this is enough for our purpose. Before introducing Semi-algorithm 2, we define the following functions.

Definition 5.2.4. Consider the set $\mathbb{R}_{t \geq 0}^{2n-1} = \{(x_1, x_2, y, r, t) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}^{n-2} \times \mathbb{R}^{n-2} \times \mathbb{R} \mid t \geq 0\}$ and define

$$f_{\text{Ball}}^{\pm} : \mathbb{R}_{t \geq 0}^{2n-1} \rightarrow \mathbb{R}^n$$

$$(x_1, x_2, y, r, t) \mapsto (x_1, x_2, y \pm r\sqrt{t})$$

Define the function $f_{\text{Ball}} : \mathbb{R}_{t \geq 0}^{2n-1} \rightarrow \mathbb{R}^n \times \mathbb{R}^n$ that maps X to $(f_{\text{Ball}}^+(X), f_{\text{Ball}}^-(X))$. Notice that f_{Ball} is an extension of Ω_P (Definition 4.1.7). By abuse of notation, for a set $S \subset \mathbb{R}^{2n-1}$, we define $f_{\text{Ball}}(S)$ as $f_{\text{Ball}}(S \cap \mathbb{R}_{t \geq 0}^{2n-1})$.

Semi-algorithm 2: Isolating the solutions of $\text{Ball}(P)$ (under Assumption \mathcal{A}_1)

Input: An open n -box B , a function P from \overline{B} to \mathbb{R}^{n-1} such that P satisfies Assumption \mathcal{A}_1 in \overline{B} and a $(2n-1)$ -open box \mathfrak{U}_0 that contains $\overline{B}_{\text{Ball}}$ (see Remark 5.2.7).

Termination: If and only if $\text{Ball}(P)$ satisfies Assumptions \mathfrak{N}_1 and \mathfrak{N}_2 in $\overline{B}_{\text{Ball}}$.

Output: A list of pairwise disjoint isolating boxes of the solutions of $\text{Ball}(P)$ in \mathfrak{U}_0 such that their images by f_{Ball} lies in $B \times B$.

- 1: $Solutions = \emptyset$.
- 2: $L := \{\mathfrak{U}_0\}$.
- 3: **while** $L \neq \emptyset$ **do**
- 4: $\mathfrak{U} := pop(L)$.
- 5: **if** $0 \notin \square \text{Ball}(P)(\overline{\mathfrak{U}})$ or $(\square f_{\text{Ball}}(\overline{\mathfrak{U}})) \cap (\overline{B} \times \overline{B}) = \emptyset$ **then**
- 6: Do nothing (\mathfrak{U} is simply removed from L).
- 7: **else if** $\text{rank}(\square J_{\text{Ball}(P)}(\overline{\mathfrak{U}})) = 2n - 1$ and $\square f_{\text{Ball}}(\epsilon\text{-inflation}(\overline{\mathfrak{U}}))^6 \subset B \times B$ **then**
- 8: **if** $\epsilon\text{-inflation}(\mathfrak{U})$ contains a solution of $\text{Ball}(P)$ (see Remark 5.2.5) **then**
- 9: Add $\epsilon\text{-inflation}(\mathfrak{U})$ to $Solutions$.
- 10: **else**

⁶For a box B and $\epsilon > 0$, $\epsilon\text{-inflation}(B)$ is the box that has the same center as B and its width is that of B multiplied by $(1 + \epsilon)$. The box $\epsilon\text{-inflation}(B)$ thus contains B , the exact value of ϵ is not important for the algorithm and it is usually set to 0.1 in subdivision algorithms [Rum2010].

-
- 11: Subdivide \mathfrak{U} and add its children to L .
 - 12: Remove duplicates in *Solutions* (see Remark 5.2.5).
 - 13: **return** *Solutions*
-

Remark 5.2.5. *Steps (8) and (12) are not detailed because they are standard in subdivision algorithms to handle the issue of solutions on or near box boundaries and ensuring that solution boxes are pairwise disjoint. We only sketch below how these steps are done and refer to [Sta1995, §5.9.1 ; Kea1997 ; XY2019] for details. In Step (8), an existence test is performed by evaluating an interval Newton operator on an ϵ -inflation of the box \mathfrak{U} . The inflated box ϵ -inflation(\mathfrak{U}) is certified to contain a solution if its image by the interval Newton operator is contained in the interior of ϵ -inflation(\mathfrak{U}). When the existence test is positive, the solution may be on the boundary or even outside \mathfrak{U} , but still in the interior of ϵ -inflation(\mathfrak{U}). The side effect is that the same solution may be reported several times when it is on or near a boundary of the subdivision. This issue is then solved in Step (12) as follows. When two boxes in the set *Solutions* intersect, they must report the same solution, and in addition, this solution is in the intersection of the two boxes. In Step (12), we thus compute intersections between boxes and replace intersecting ones by their intersection box. The boxes in the output set *Solutions* are thus pairwise disjoint.*

Lemma 5.2.6. *Under Assumption \mathcal{A}_1 in \overline{B} , if Semi-algorithm 2 stops, it returns a list of pairwise disjoint isolating boxes of the solutions of $\text{Ball}(P)$ in \mathfrak{U}_0 such that their images by f_{Ball} lies in $B \times B$. Moreover, Semi-algorithm 2 stops if and only if $\text{Ball}(P)$ satisfies Assumptions $\mathfrak{N}_1, \mathfrak{N}_2$ in $\overline{B}_{\text{Ball}}$.*

Proof. We first prove the correctness of the Semi-algorithm 2 assuming that it terminates. Since Step (5) is the only time the algorithm discards boxes, it never discards a box that contains a solution of $\text{Ball}(P)$ in \mathfrak{U}_0 such that its image by f_{Ball} lies in $B \times B$. Hence, all such solutions of $\text{Ball}(P)$ lie in output boxes. The rank condition in Step (7) guarantees that each output box contains at most one solution of $\text{Ball}(P)$ [Sny1992, Theorem A.1]. The fact that every output box contains at least one solution is ensured by a standard algorithm in Step (8) (see e.g., [Neu1991, Theorem 5.6.2 ; XY2019] and Remark 5.2.5). Finally, by Step (12), the output boxes are pairwise disjoint, hence the algorithm outputs isolating boxes of the solutions of $\text{Ball}(P)$ in \mathfrak{U}_0 such that their images by f_{Ball} lie in $B \times B$.

To prove the equivalence for the termination, first assume that Semi-algorithm 2 stops and returns *Solutions*. According to the correctness proof, every solution X of $\text{Ball}(P)$ in $\overline{B}_{\text{Ball}}$ is regular and satisfies $\Omega_P(X) \in B \times B$. Thus, Assumptions \mathfrak{N}_1 and \mathfrak{N}_2 are satisfied in $\overline{B}_{\text{Ball}}$.

On the other hand, assume that \aleph_1 and \aleph_2 hold in $\overline{B}_{\text{Ball}}$. We prove that Semi-algorithm 2 terminates. By Assumption \aleph_1 all solutions in $\overline{B}_{\text{Ball}}$ of the square system $\text{Ball}(P)$ are regular. Hence, they form a zero dimensional manifold in the compact space $\overline{B}_{\text{Ball}}$. Thus, the solution set is finite. We now prove that for any box $\overline{\mathcal{U}} \in L$ with a small enough width, one of the conditions in Step (5) or the conditions in Steps (7-8) are satisfied. Thus, in both cases $\overline{\mathcal{U}}$ will be removed from L , and hence, Semi-algorithm 2 stops after a finite number of iterations. Due to Assumption \aleph_2 , after a finite number of iterations, no box \mathcal{U} in L intersects the boundary of $B \times B$. Moreover, due to the convergence of the box evaluations, we can also assume that either $\square f_{\text{Ball}}(\epsilon\text{-inflation}(\overline{\mathcal{U}})) \subset B \times B$, which is the second condition of Step (7), or $(\square f_{\text{Ball}}(\overline{\mathcal{U}})) \cap (\overline{B} \times \overline{B}) = \emptyset$ which is the second condition of Step (5).

If \mathcal{U} does not contain a solution of $\text{Ball}(P)$, then due to convergence of the box function evaluation of $\text{Ball}(P)$, after a finite number of iterations, every children \mathcal{U}' of \mathcal{U} satisfies $0 \notin \square \text{Ball}(P)(\overline{\mathcal{U}'})$, that is, it is discarded in Step (5).

If \mathcal{U} contains a solution of $\text{Ball}(P)$ in $\overline{B}_{\text{Ball}}$, according to Assumption \aleph_1 , it is a regular solution. Due to the convergence of the box evaluation $\det(J_{\text{Ball}(P)}(\overline{\mathcal{U}}))$ will eventually be non zero and thus $\text{rank}(\square J_{\text{Ball}(P)}(\overline{\mathcal{U}}))$ will eventually be $2n - 1$ after a finite number of iterations, which is the first condition of Step (7). Due to the convergence of the interval Newton existence test, the condition of Step (8) will also be eventually satisfied (see Remark 5.2.5). The refined box will then eventually be added in the Solutions list.

Thus, for any box in L with a small enough width, one of the conditions of Step (5) is satisfied or all of the conditions in Step (7-8) are satisfied, thus it is either discarded or added to the output. Hence, Semi-algorithm 2 terminates. \square

Remark 5.2.7. *Semi-algorithm 2 requires a closed $(2n - 1)$ -box $\overline{\mathcal{U}}_0$ that contains $\overline{B}_{\text{Ball}}$. For instance the following set could be used: $\{(q, r, t) \in \mathbb{R}^{2n-1} \mid q \in \overline{B}, -1 \leq r_i \leq 1 \text{ for } i \in \{3, \dots, n\}, 0 \leq t \leq \frac{\xi^2}{4}\}$ with $\xi = \max \{\|q - q'\| \mid q, q' \in \overline{B}\}$.*

Finally, using Lemma 5.2.2, Semi-algorithm 3 checks whether P satisfies Assumptions $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4$ and \mathcal{A}_5^- in \overline{B} and outputs a superset of isolating boxes of the singularities of $\pi_{\mathcal{C}}(\mathcal{C})$.

Semi-algorithm 3: Checking the weak assumptions and computing a superset of the singularities of $\pi_{\mathcal{C}}(\mathcal{C})$

Input: An open n -box B and a smooth function P from \overline{B} to \mathbb{R}^{n-1} .

Termination: If and only if P satisfies Assumptions $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4$ and \mathcal{A}_5^- in \overline{B} .

Output: N , a list of certified node singularities: a list of boxes in \mathbb{R}^{2n-1} whose projections in \mathbb{R}^2 contain each a single node of $\pi_{\mathcal{C}}(\mathcal{C})$.

U , a list of uncertified singularities: a list of boxes in \mathbb{R}^{2n-1} whose projections in \mathbb{R}^2 contain each at most one node or one cusp of $\pi_{\mathcal{C}}(\mathcal{C})$.

The union of all these projected 2D boxes contains all the singularities of $\pi_{\mathcal{C}}(\mathcal{C})$.

- 1: Check Assumption \mathcal{A}_1 (Semi-algorithm 1).
- 2: Compute a closed $(2n - 1)$ -box $\bar{\mathcal{U}}_0$ that contains \bar{B}_{Ball} (Remark 5.2.7).
- 3: $L :=$ output of Semi-algorithm 2.
- 4: Keep refining all boxes $\bar{\mathcal{U}} \in L$ (see Remark 5.2.8) until no triplets of boxes overlap in projection. Then remove from L one box from every pair (see Remark 4.1.10). This ensures Assumption \mathcal{N}_3 .
- 5: $N :=$ boxes of L that lie in the halfspace $t > 0$.
- 6: $U :=$ boxes of L that intersect the hyperplane $t = 0$.
- 7: **return** N and U .

Remark 5.2.8. *The refinement of an isolating box of a solution is performed by iterative evaluation of an interval Newton operator; we refer to [Neu1991, Theorem 5.6.2] for details.*

To identify the possible cusp points in the set U returned by Semi-algorithm 3, one may wish to solve independently the Ball system with the additional constraint $t = 0$ (by Remark 4.1.9). Unfortunately, in this case we have an over-determined system and thus we cannot certify its solutions numerically. In the special case of a silhouette curve, it is possible to identify cusp points with numerical algorithms in the case $n = 3$ [IMP2016a, Lemmas 9 & 10], but we leave as a conjecture its generalization for $n > 3$.

On the other hand, for curves that satisfy the strong assumptions, \mathcal{A}_5 ensures that there are no cusps in the projection, which is equivalent to $\widehat{\mathcal{L}}_{\mathcal{C}}$ being empty and $\text{Ball}(P)$ having no solutions on the hyperplane $t = 0$ (by Remark 4.1.9). Hence, if Assumptions \mathcal{A}_{1-5} hold, we can refine the boxes output by Semi-Algorithm 3 until no box intersects $t = 0$. Boxes in the half-space $t < 0$ correspond to imaginary points in \mathbb{C}^n (Definition 4.1.7). Then the boxes satisfying $t > 0$ are all the isolating boxes of the nodes of $\pi_{\mathcal{C}}(\mathcal{C})$ by Lemmas 5.2.6 and 4.1.6, Remark 4.1.9 and Corollary 4.3.5.

Semi-Algorithm 4: Checking Assumptions \mathcal{A}_{1-5} and isolating the singularities of $\pi_{\mathcal{C}}(\mathcal{C})$

Input: An open n -box B and a smooth function P from \bar{B} to \mathbb{R}^{n-1} .

Termination: If and only if P satisfies Assumptions \mathcal{A}_{1-5} in \overline{B} .

Output: A list of boxes in \mathbb{R}^{2n-1} whose projections in \mathbb{R}^2 are isolating boxes of the singularities $\pi_{\mathcal{C}}(\mathcal{C})$ (all singularities are in some boxes and each box contains a unique singularity).

- 1: $N, U :=$ output of Semi-Algorithm 3.
 - 2: **for** all $\overline{\mathcal{U}} \in U$ **do**
 - 3: Keep refining $\overline{\mathcal{U}}$ (see Remark 5.2.8) until it does not intersect the hyperplane $t = 0$ and discard it if it lies in the half-space $t < 0$.
 - 4: **return** $N \cup U$.
-

5.3 Semi-algorithms improvements

Semi-algorithm 3 (resp. 4) of Section 5.2 takes as input an open n -box B and a curve \mathcal{C} defined by a smooth function P from \overline{B} to \mathbb{R}^{n-1} . It terminates if and only if P satisfies our weak (resp. strong) assumptions of Definition 3.1.1. Upon termination, it outputs a superset of the singularities (resp. the singularities) of the curve $\pi_{\mathcal{C}}(\mathcal{C})$. This section is dedicated to describe the (semi) algorithm we implemented which is a variant of Semi-algorithms 3 and 4.

In the following, we present the challenges we faced and how we overcame them for implementing Semi-algorithms 3 and 4.

5.3.1 Subdivision-based solver on a high-dimensional system

The first challenge we faced is solving a high-dimensional system such as the Ball system which is of dimension $2n - 1$. The solutions of the Ball system are detected in Semi-algorithm 3 using a method based on the subdivision of the space. Such methods usually suffer from the curse of dimensionality in practice. In the following, we describe the way we overcame this challenge.

For the rest of Section 5.3.1, we assume that Assumption \mathcal{A}_1 is always satisfied. Moreover, recall that M_i is the sub-matrix of J_P obtained by removing the i -th column. Over a closed n -box $\overline{\mathfrak{B}}$, we call P parameterizable in x_i if M_i is full rank at every point of $\overline{\mathfrak{B}}$. This notion has a geometric interpretation: the function P being parameterizable in x_i implies that the tangent space at any point, of \mathcal{C} , is not orthogonal to the x_i axis.

The domain B_{Ball} where the Ball system is solved is refined by first enclosing the curve \mathcal{C} in the smaller space \mathbb{R}^n . Our approach follows the observation that if the curve is enclosed by a set *enclosing_curve* of boxes in \mathbb{R}^n such that:⁷

⁷Regarding the following properties, Items (a) and (b) are to remove redundancy which, of course, improves the perfor-

- (a) every box \mathfrak{B} in *enclosing_curve* contains points of \mathcal{C} ,
- (b) the intersection of two distinct $\mathfrak{B}, \mathfrak{B}'$ in *enclosing_curve* lives in their boundaries, moreover, the set $\mathfrak{B} \setminus \mathfrak{B}'$ still contains points of \mathcal{C} ,
- (c) for non-disjoint \mathfrak{B} and \mathfrak{B}' in *enclosing_curve*, the curve \mathcal{C} is connected in $\mathfrak{B} \cup \mathfrak{B}'$,

then every node in $\pi_{\mathcal{C}}(\mathcal{C})$ is contained in at least one of the following 2-boxes:

- (i) The intersection of the plane projections of two boxes in *enclosing_curve*. These 2-boxes contain the nodes induced by the projections of two points in \mathcal{C} , with the same projection, that do not belong to one and the same box of *enclosing_curve*.
- (ii) The projection of a box \mathfrak{B} in *enclosing_curve* such that the curve \mathcal{C} in this box is not parameterizable in x_1 and not parameterizable in x_2 . Such a box can be detected using the condition $0 \in \det(\square M_1(\mathfrak{B})) \cap \det(\square M_2(\mathfrak{B}))$, where $\square M_1$ and $\square M_2$ are box functions of M_1 and M_2 respectively. These 2-boxes contain the nodes induced by the projections of two points, with the same projection, that belong to one and the same box \mathfrak{B} of *enclosing_curve*. Figure 5.1 illustrates an example of this case where \mathfrak{B} contains a part of \mathcal{C} that projects to a loop. In this figure, to give a geometric intuition, we presume that q_1 and q_2 are in \mathcal{C} .

Moreover, every cusp of $\pi_{\mathcal{C}}(\mathcal{C})$ lies in the projection of a box of *enclosing_curve* containing a point of \mathcal{C} with a tangent orthogonal to the (x_1, x_2) -plane, that is, P is parameterizable neither in x_1 nor in x_2 at the same point in \mathcal{C} . These boxes are contained in the set of boxes that satisfy the constraint (ii) above.

The above observation helps to cover the set $\mathcal{L}_c \cup \mathcal{L}_n$ using *enclosing_curve*. This paves the way to cover all Ball solutions using Lemma 4.1.6. We are going to consider this approach to prune B_{Ball} which makes Semi-algorithms 3 and 4 more efficient in practice. However, the reader might notice that any two neighbor boxes in \mathbb{R}^n satisfy the condition in (i). Hence, a non-ignorable set of spurious pairs satisfy (i). This makes the previous condition a bad criteria to detect points of \mathcal{L}_n . In the following lemma, we develop the previous observation to obtain the criteria by which we cover $\mathcal{L}_c \cup \mathcal{L}_n$ with less redundancy than the above observation.

Lemma 5.3.1. *Let \mathcal{C} be a curve defined by P satisfying Assumption \mathcal{A}_1 in a closed n -box \overline{B} . Then, every point of $\mathcal{L}_c \cup \mathcal{L}_n$ is in at least one of the following:*

mance of the algorithm. Item (c) is to preserve the connectivity of the components of \mathcal{C} . This will be used in Lemma 5.3.1 where the connectivity of \mathcal{C} in a box plays a role in detecting points of \mathcal{L}_n in our approach.

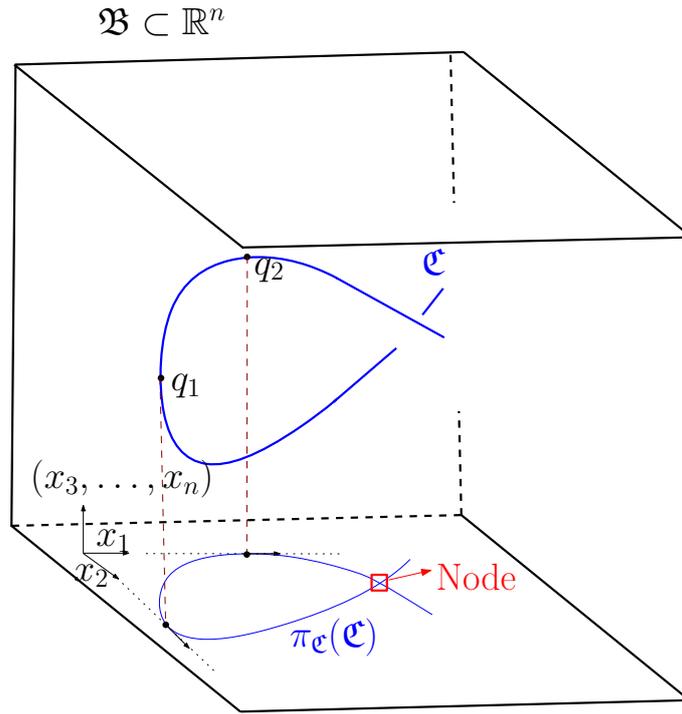


Figure 5.1: An example of Case (ii) for a curve \mathcal{C} in \mathbb{R}^n . The box \mathcal{B} contains a smooth part of \mathcal{C} that projects to a loop with a node. This implies that two points q_1 and q_2 are in \mathcal{B} such that $\det(M_1(q_1)) = 0$ and $\det(M_2(q_2)) = 0$. Hence, the tangent space plane projection of q_1 (resp. q_2) is parallel to x_2 -axis (resp. x_1 -axis).

- (1) The union of disjoint boxes \mathcal{B} and \mathcal{B}' in enclosing_curve such that their plane projections intersect.
- (2) The union of two non-disjoint boxes \mathcal{B} and \mathcal{B}' in enclosing_curve such that \mathcal{C} has at least one x_1 -critical point and at least one x_2 -critical point in $\mathcal{B} \cup \mathcal{B}'$.

Figures 5.2 and 5.3 show two examples of (2) where $\mathcal{B} \neq \mathcal{B}'$.

Proof. For a points $q \in \mathcal{L}_{\mathcal{C}}$, since $T_q\mathcal{C}$ is orthogonal on \mathbb{R}^2 , then q is x_1 -critical and x_2 -critical. Hence any box \mathcal{B} in enclosing_curve, containing q , satisfies (2) (taking $\mathcal{B}' = \mathcal{B}$). On the other hand, let q in \mathcal{L}_n and q' be a point in \mathcal{L}_n with $\pi_{\mathcal{C}}(q) = \pi_{\mathcal{C}}(q')$. Suppose that q and q' are in (possibly equal) boxes \mathcal{B} and \mathcal{B}' respectively. We distinguish the following cases:

- (a) $\mathcal{B} \cap \mathcal{B}' = \emptyset$: Then, Case (1) applies; recall that the projections of \mathcal{B} and \mathcal{B}' intersect since both contain $\pi_{\mathcal{C}}(q) = \pi_{\mathcal{C}}(q')$.

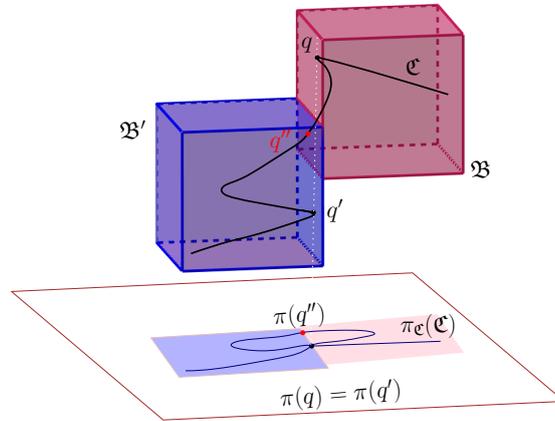


Figure 5.2: An example of \mathfrak{B} and \mathfrak{B}' satisfying (2) in Lemma 5.3.1. Their projections have a node in the boundary. The curve \mathcal{C} is tangent to the boundaries of both \mathfrak{B} and \mathfrak{B}' at q and q' respectively. The curve and the boxes intersect at q'' such that q'' does not have the same plane projection as q and q' .

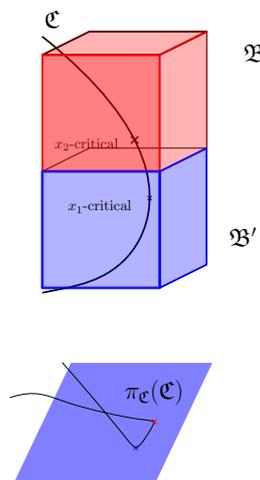


Figure 5.3: An example of \mathfrak{B} and \mathfrak{B}' satisfying (2) in Lemma 5.3.1. The projection has a loop. This implies that there are a x_1 -critical point and a x_2 -critical point in the union of \mathfrak{B} and \mathfrak{B}' . In this example, these points are different and the projection has a node.

- (b) If $\mathfrak{B} \cap \mathfrak{B}' \neq \emptyset$: We are going to prove that \mathfrak{B} and \mathfrak{B}' satisfy (2) using Lagrange multiplier theorem (LMT). We consider x_1 as a function from \mathbb{R}^n to \mathbb{R} and P to be the constrains function. Since \mathfrak{C} is connected in $\mathfrak{B} \cup \mathfrak{B}'$ and since $\pi_{\mathfrak{C}}(q) = \pi_{\mathfrak{C}}(q')$, there must be an x_1 -critical point q in the compact subset \mathfrak{C} in $\mathfrak{B} \cup \mathfrak{B}'$.⁸ By LMT, there exists a non-trivial $\lambda \in \mathbb{R}^{n-1}$ such that $J_{x_1}(q) = \lambda^t J_P(q)$. This implies that $\lambda^t M_1(q) = 0 \in \mathbb{R}^{n-1}$. Thus, $\det(M_1(q)) = 0$. In other words, q is a x_1 -critical point. In the same way, we prove the existence of q' in \mathfrak{C} restricted to $\mathfrak{B} \cup \mathfrak{B}'$ such that $\det(M_2(q')) = 0$. This proves that q' is a x_2 -critical point. Thus, \mathfrak{B} and \mathfrak{B}' satisfy (2).

□

Remark 5.3.2. From the proof of Lemma 5.3.1, we can obtain a more precise statement: Every point of $\mathfrak{L}_{\mathfrak{C}}$ is in a box \mathfrak{B} of enclosing_curve that satisfies (2), considering $\mathfrak{B}' = \mathfrak{B}$. Let q and q' be in \mathfrak{C} with $\pi_{\mathfrak{C}}(q) = \pi_{\mathfrak{C}}(q')$, in particular, q and q' are in $\mathfrak{L}_{\mathfrak{n}}$. Assume that q and q' are in the (possibly equal) boxes \mathfrak{B} and \mathfrak{B}' respectively. If \mathfrak{B} and \mathfrak{B}' are disjoint, then they satisfy (1). Otherwise, they satisfy (2).

An interesting question is how to detect boxes satisfying (1) or (2). Detecting boxes of enclosing_curve satisfying (1) can be done using any black-box tool that reports the pairs of intersected boxes of a given set of boxes. See Remark 6.1.1 for information about the tool we chose. Boxes $\mathfrak{B}, \mathfrak{B}'$ satisfying (2) can be detected using the same tool in addition to checking the condition:

$$0 \in \square M_1(\mathfrak{B}) \cup \square M_1(\mathfrak{B}') \text{ and } 0 \in \square M_2(\mathfrak{B}) \cup \square M_2(\mathfrak{B}').$$

The next step is to cover the Ball solutions. This is be done by the following proposition which is a corollary of Lemma 5.3.1.

Proposition 5.3.3. Let \mathfrak{C} be covered by enclosing_curve as above. Let $X \in \overline{B}_{\text{Ball}}$ be a solution $\text{Ball}(P)$. Then, at least one of the following holds:

- (a) There exists \mathfrak{B} in enclosing_curve which is the cross product of boxes $\overline{B}_{(x_1, x_2)}$ in \mathbb{R}^2 and B_y in \mathbb{R}^{n-2} satisfying (2) and X lives in $f_{\text{Ball}}^{-1}(\mathfrak{B} \times \mathfrak{B})$. Hence, X is in $(x_1, x_2, y, r, t) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}^{n-2} \times \mathbb{R}^{n-2} \times \mathbb{R}$, where $(x_1, x_2, y) \in \mathfrak{B}$, $r \in [-1, 1]^{n-2}$ and $t = [0, \max(\frac{\|B_y - B_y\|^2}{4})]$.
- (b) There exist (possibly equal) boxes $\mathfrak{B} = \mathfrak{B}_{(x_1, x_2)} \times \mathfrak{B}_y$ and $\mathfrak{B}' = \mathfrak{B}'_{(x_1, x_2)} \times \mathfrak{B}'_y$ in enclosing_curve satisfying (1) or (2) such that X is in $f_{\text{Ball}}^{-1}(\mathfrak{B} \times \mathfrak{B}')$. Hence, X lives in (x_1, x_2, y, r, t) , with $(x_1, x_2) = \mathfrak{B}_{(x_1, x_2)} \cap \mathfrak{B}'_{(x_1, x_2)}$, $y =$

⁸Notice that the x_1 -values of q , q and q' might be equal. This case occurs when \mathfrak{C} is a segment orthogonal to \mathbb{R}^2 in $\mathfrak{B} \cup \mathfrak{B}'$.

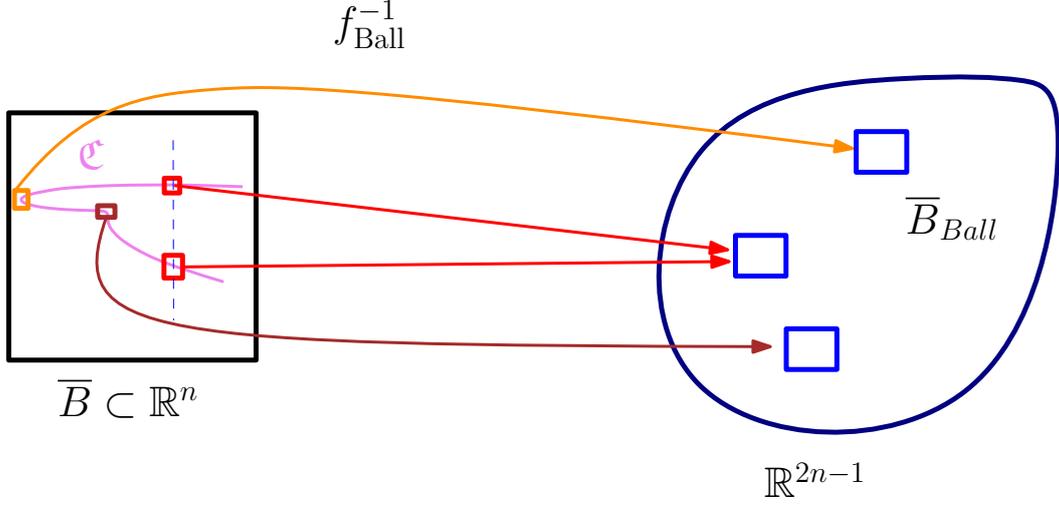


Figure 5.4: A simplified illustration of pruning $\overline{B}_{\text{Ball}}$ as described in Proposition 5.3.3. The red boxes satisfy (1). On the other hand, the brown (resp. orange) box satisfies (2). Hence, we solve the Ball system over f_{Ball} pre-images (in blue) of the red boxes, the brown box and the orange box.

$$\frac{1}{2}(\mathfrak{B}_y + \mathfrak{B}'_y), \mathbf{r} = [-1, 1]^{n-2} \text{ if } \mathfrak{B} \cap \mathfrak{B}' \neq \emptyset, \text{ otherwise, } \mathbf{r} = \frac{\mathfrak{B}_y - \mathfrak{B}'_y}{\|\mathfrak{B}_y - \mathfrak{B}'_y\|}, \mathbf{t} = \frac{\|\mathfrak{B}_y - \mathfrak{B}'_y\|^2}{4}.$$

The arithmetic operations mentioned in Proposition 5.3.3 are the natural box function of the usual arithmetic operations defined on \mathbb{R}^n (see Example 5.1.2). Figure 5.4 illustrates a global picture of the steps above.

Proof. Let $X = (x_1, x_2, y, r, t) \in \mathbb{R} \times \mathbb{R} \times \mathbb{R}^{n-2} \times \mathbb{R}^{n-2} \times \mathbb{R}$ be a solution of $\text{Ball}(P)$ in $\overline{B}_{\text{Ball}}$. Let (q, q') be a pair of $\widehat{\mathcal{L}}$ (Definition 4.1.5) such that $f_{\text{Ball}}(X) = (q, q')$. Recall that q and q' are in $\mathcal{L}_c \cup \mathcal{L}_n$. Hence, by Remark 5.3.2, q and q' are respectively in two boxes \mathfrak{B} and \mathfrak{B}' of *enclosing_curve* satisfying (1) or (2). In both cases, we imply that $X \in f_{\text{Ball}}^{-1}(\mathfrak{B} \times \mathfrak{B}')$ which proves the proposition. \square

We now reformulate the results of Lemma 5.3.1 and Proposition 5.3.3 in Algorithm 5. This algorithm is used to find a set of $(2n - 1)$ -boxes that cover the Ball solutions instead of solving it in the whole $\overline{B}_{\text{Ball}}$. It turns out in practice, that computing this set and solving $\text{Ball}(P)$ over its boxes is faster than solving the Ball system over $\overline{B}_{\text{Ball}}$.

Algorithm 5: Pruning $\overline{B}_{\text{Ball}}$.

Input: An open n -box B and a curve \mathfrak{C} defined by $P \in C^\infty(\mathbb{R}^n, \mathbb{R}^{n-1})$, with $\mathfrak{C} \subset B$ such that Assumption \mathcal{A}_1 is satisfied.

Output: A finite set *Solving_domains* which is a set of $(2n - 1)$ -boxes contained in B_{Ball} that contains all the solutions of $\text{Ball}(P)$ in B_{Ball} .

- 1: Compute a set *enclosing_curve* of closed n -boxes that enclose \mathfrak{C} (using a black-box solver) with Properties (a), (b) and (c).
 - 2: Find the boxes of *enclosing_curve* satisfying (1) or (2) (Remark 5.3.2)
 - 3: *Solving_domains* := $\{\}$
 - 4: **for** a box $\mathfrak{B}' = \mathfrak{B}$ in *enclosing_curve* satisfying (2) **do**
 - 5: $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}) = \mathfrak{B}$, $\mathbf{r} = [-1, 1]^{n-2}$ and $\mathbf{t} = [0, \max(\frac{\|B_y - B'_y\|^2}{4})]$
 - 6: Add $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}, \mathbf{r}, \mathbf{t})$ to *Solving_domains*.
 - 7: **for** distinct $\mathfrak{B} = \mathfrak{B}_{(x_1, x_2)} \times \mathfrak{B}_y$ and $\mathfrak{B}' = \mathfrak{B}'_{(x_1, x_2)} \times \mathfrak{B}'_y$ in *enclosing_curve* satisfying (1) or (2) **do**
 - 8: $(\mathbf{x}_1, \mathbf{x}_2) = \mathfrak{B}_{(x_1, x_2)} \cap \mathfrak{B}'_{(x_1, x_2)}$ and $\mathbf{y} = \frac{1}{2}(\mathfrak{B}_y + \mathfrak{B}'_y)$
 - 9: $\mathbf{r} = [-1, 1]^{n-2}$ if $\mathfrak{B} \cap \mathfrak{B}' \neq \emptyset$, otherwise, $\mathbf{r} = \frac{\mathfrak{B}_y - \mathfrak{B}'_y}{\|\mathfrak{B}_y - \mathfrak{B}'_y\|}$ and $\mathbf{t} \in \frac{\|\mathfrak{B}_y - \mathfrak{B}'_y\|^2}{4}$
 - 10: Add $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{y}, \mathbf{r}, \mathbf{t})$ to *Solving_domains*.
 - 11: **return** *Solving_domains*
-

As a part of the improvements, the output of Algorithm 5 replaces L in Semi-Algorithm 2, Step (2). For more details, Section 5.3.3 is dedicated to provide a global vision of the improvements.

5.3.2 Evaluating the operator D

To solve the Ball system we need box functions for the operators S and D (Definition 4.1.1). We first note that if $P(x_1, x_2, y) \in C^\infty(\mathbb{R}^n, \mathbb{R})$ is a polynomial function, $S \cdot P$ (resp. $D \cdot P$) is also a polynomial function and can be computed from the terms of $P(x_1, x_2, y + rt)$ that have even (resp. odd) exponents in the variable t , see [IMP2016b, Lemma 6] for details.

If P is a more general C^∞ function for which we have a box function, using Definition 4.1.1, computing $S \cdot P$ on any $(2n - 1)$ -box $\bar{\mathfrak{U}}$ or computing $D \cdot P$ on $\bar{\mathfrak{U}}$ such that its t -interval does not contain 0, is implemented from the box function of P and interval arithmetic. On the other hand, when the t -interval contains 0 or is close to 0, the division by \sqrt{t} in the formula for $D \cdot P$ makes the computation undefined or unstable. In such non-polynomial cases, we use a Taylor form at order 3 [Ral1983], that is, we compute a Taylor expansion with remainder at $t = 0$ of $D \cdot P$ and evaluate by interval the third order derivative. We define a threshold δ_{Taylor} (which we set to 10^{-2}) such that this Taylor form is used when the t -interval

has values smaller than δ_{Taylor} .

5.3.3 Improvement integration

In this section, we discuss how we modify Semi-algorithms 2, 3 and 4 in view of the improvements we discussed in Sections 5.3.1 and 5.3.2. We also add a new parameter eps_{\min} in addition to changes in several steps in Semi-algorithms 2, 3 and 4, preparing the semi-algorithms to the implementation stage.

Parameter eps_{\min} . We start with adding the parameter eps_{\min} to Semi-algorithm 4 and any (semi) algorithm called by it. The parameter eps_{\min} is a non-negative real that guarantees the termination in Semi-algorithm 4 even if the termination criteria is not satisfied. More precisely, if $\text{eps}_{\min} = 0$, then the parameter does not affect Semi-algorithm 4. Otherwise, this parameter forces the termination of subdivision or refinement steps in Semi-algorithm 4, that is, the semi-algorithm does not subdivide (or refine) further if the width of the processed box (which is a child of B) is less than eps_{\min} . In this case, the remaining boxes (where eps_{\min} forces the termination of subdivision or refinement) are labeled as “unknown-status”. These boxes are left aside and not considered for the further steps. For example, in Semi-algorithm 2 (which is called by Semi-algorithm 4), Step (11), the parameter eps_{\min} prohibits the subdivision when the width of \mathfrak{U} is less than eps_{\min} . From a practical point of view, the parameter eps_{\min} allows the use of fixed precision arithmetic.⁹ In addition, it is useful if we want to ensure the termination even when not all assumptions \mathcal{A}_{1-5} are satisfied, certifying $\pi_{\mathcal{C}}(\mathcal{C})$ singularities only on the part for which we verify the strong assumptions. On the other hand, an “unknown-status” label of a box prohibits the use of the Ball system to certify the singularities of its plane projection (if any). This is, of course, a drawback of setting $\text{eps}_{\min} > 0$.

Changes in Semi-algorithm 4. We now describe the changes in Semi-algorithm 4 and all (semi) algorithms called by it.

- i. Assumption \mathcal{A}_1 is not checked by Semi-algorithm 1 anymore. It is rather checked using a black-box solver (see Black-box solver 7 in Section 6.1), considering the parameter eps_{\min} mentioned above. Such a solver does not only check Assumption \mathcal{A}_1 , but also computes a set *enclosing_curve* of n -boxes that enclose \mathcal{C} with Properties (a), (b) and (c) in Section 5.3.1. More precisely, for $\text{eps}_{\min} > 0$, the solver returns possibly-empty two sets of n -boxes. The first *enclosing_curve* encloses a part of \mathcal{C} such that the solver succeeds to verify Assumption \mathcal{A}_1 . The second set contains boxes of width less than eps_{\min} and encloses the rest of \mathcal{C} .

⁹In fact, a black-box tool in our implementation uses fixed precision arithmetic; see Section 6.1 for more details.

- ii. The set *enclosing_curve* is required for Algorithm 5. Furthermore, the output of Algorithm 5 is used instead of \mathcal{L}_0 in Step (2) of Semi-algorithm 2. Consequently, Step (2) of Semi-algorithm 3 is omitted. In practice, solving the Ball system over Algorithm 5 output is more efficient than solving over \mathcal{L}_0 , despite the fact that computing \mathcal{L}_0 is simpler.
- iii. Recall that every box \mathcal{L} of L (after the change in Item i) is computed in such a way that $f_{\text{Ball}}(\overline{\mathcal{L}})$ is a pair of boxes in *enclosing_curve*. Hence, no need to check the condition $(\square f_{\text{Ball}}(\overline{\mathcal{L}})) \cap (\overline{B} \times \overline{B}) = \emptyset$ in Step (5) of Semi-algorithm 2 since it is never satisfied.
- iv. As mentioned in Section 5.3.2, when solving the Ball system (Semi-algorithm 2, Step (8)) over a box ϵ -inflation(\mathcal{L}), we consider the two different approaches to evaluate the operator D . More precisely, if the t -interval $[t_{\min}, t_{\max}]$ of ϵ -inflation(\mathcal{L}) contains zero,¹⁰ we use Taylor form over the sub-box of ϵ -inflation(\mathcal{L}) satisfying $t_{\min} \leq t \leq \min(\delta_{\text{Taylor}}, t_{\max})$. If $t_{\max} > \delta_{\text{Taylor}}$, we solve the Ball system over the complimentary sub-box (i.e., with $\delta_{\text{Taylor}} \leq t \leq t_{\max}$) as defined in (4.1).

Considering the parameter eps_{\min} and the changes above, we reformulate the specifications of Semi-algorithm 4:

Semi-Algorithm 6: Variant of Semi-algorithm 4

Input: An open n -box B , a smooth function P from \overline{B} to \mathbb{R}^{n-1} and two non-negative reals δ_{Taylor} and eps_{\min} .

\mathcal{A}_{1-5} in \overline{B} or $\text{eps}_{\min} > 0$.

Output: N , a list of pairwise disjoint $(2n-1)$ -boxes whose projections in \mathbb{R}^2 are isolating boxes of node singularities in $\pi_{\mathcal{C}}(\mathcal{C})$.

U , a list of n -boxes and $(2n-1)$ -boxes such that Assumption \mathcal{A}_1 is not verified (n -boxes), or Assumptions $\mathcal{N}_1, \mathcal{N}_2$ or \mathcal{N}_3 are not satisfied ($(2n-1)$ -boxes). The width of every box is less than eps_{\min} .

Moreover, every singularity is in the plane projection of a box of N or U .

In the next chapter, we provide more details about the implementation and how to use the software we develop based on Semi-algorithm 6 and the improvements explained in this section.

¹⁰Notice that, in this case, t_{\min} is in $(-\infty, 0]$ by ϵ -inflation.

Part II

Software, experiments and application in robotics

This part details our implementations of the (semi) algorithms presented in Chapter 5 and reports on our experimentations. Based on the theoretical arguments in Chapters 3 and 4 and the (semi) algorithms presented in Chapter 5, we developed the software *Isolating_singularities*¹¹ written in Python. Given a curve \mathcal{C} in \mathbb{R}^n , our software finds the singularity of its plane projection. In addition, the software provides a tool to visualize the projected curve and its singularities. Chapter 6 is dedicated to describe *Isolating_singularities* in more details.

In chapter 7, we examine the efficiency of our software performing four experiments that show different cases: \mathcal{C} is an analytic curve in \mathbb{R}^3 or \mathbb{R}^4 (Sections 7.1 and 7.2), \mathcal{C} is a high-degree algebraic curve defined by randomly-generated polynomials (Section 7.3) and \mathcal{C} is a curve that contains two very close branches whose projections define a node (Section 7.4).

Finally, in Chapter 8, we revisit a classical problem in robotics design and analysis. This problem consists of determining the configurations in which a robot is allowed to move without damaging it. More formally, it consists of numerically computing the generalized aspects of that robot (Definition 8.1.1 (g)). For a robot of two degrees of freedom, the boundaries of the generalized aspects is generically a space curve in \mathbb{R}^4 . We use our software on the fully parallel robot RRRRR and we present our experiments in Section 8.3.

¹¹The software is accessible at https://github.com/gkrait/Isolating_singularities.

6

Implementation

Contents

| | |
|--|-----|
| 6.1 Third-party libraries | 106 |
| 6.2 The Python software <i>Isolating_singularities</i> | 107 |

We present in this chapter a brief description of our software *Isolating_singularities*¹² which is an implementation of Semi-algorithm 6.

Our software consists of three main functions: *enclosing_curve*, *enclosing_singularities* and *ploting_boxes*. The first one encloses the curve \mathcal{C} in \mathbb{R}^n and checks Assumption \mathcal{A}_1 . More precisely, it returns (a) a set of n -dimensional boxes that cover a smooth part of the curve \mathcal{C} , and (b) a set of n -dimensional boxes that cover an unknown part of \mathcal{C} , where the solver fails to certify the smoothness.

The goal of the second function *enclosing_singularities* is to isolate the Ball system solutions. Consequentially, the plane projections of the boxes output by *enclosing_singularities* are, in the best case,¹³ isolating boxes of the singularities of the projected curve, and in the worst case, they enclose the singularities. More precisely, the projection of the boxes returned by *enclosing_singularities* consists of:

- a set of 2-dimensional boxes each of which contains a unique singular point of $\pi_{\mathcal{C}}(\mathcal{C})$, and
- a set of 2-dimensional boxes where the behavior of $\pi_{\mathcal{C}}(\mathcal{C})$ is unknown, but such that every singular point in $\pi_{\mathcal{C}}(\mathcal{C})$ is contained in a box of one of these two sets.

The third function *ploting_boxes* performs a visualization task. Using the Python

¹²The software is accessible at https://github.com/gkrait/Isolating_singularities.

¹³That is, when our software succeeds to verify the weak assumptions (Section 3.1.2).

Package [Matplotlib](#) [Hea2007], we provide a tool to $2D$ -visualize the results of the first and second functions.

In Section 6.1, we present the third-party libraries that we used in our software. The second part of this chapter (Section 6.2) documents the software *Isolating_singularities*.

6.1 Third-party libraries

Our software is based on interval arithmetic, interval evaluations of analytic functions and an interval solver. We use the following libraries, [Python-FLINT](#) [Joh2012b] and [Ibexsolve](#) [Nin2015] for these tasks.

[Python-FLINT](#) is a Python extension module wrapping [FLINT](#) (Fast Library for Number Theory) and [Arb](#) (arbitrary-precision ball arithmetic) [Joh2012a], which offers a toolbox for interval arithmetic and evaluation of analytic functions.

[Ibexsolve](#) is a C++ end-user program that solves systems of non-linear equations rigorously, that is, it does not lose any solution and returns each solution under the form of a small box enclosing the true value. It implements a classical branch-and-prune algorithm that interleaves contractions and branching (bisections) to enclose the solutions of a system at any given desired precision. However, as opposed to [Arb](#), [Ibexsolve](#) has a fixed precision, hence when several solutions are closer to each other than this precision, it will correctly return an enclosing box for these solutions but it will fail at isolating them. In our software, we use the default precision which is 10^{-7} . [Ibexsolve](#), and thus also our software *Isolating_singularities*, use a parameter *eps_max* that defines a maximum width for the isolating boxes output by the solver (the box bisections are forced until all output boxes are not larger than *eps_max*). We use [Ibexsolve](#) for solving the Ball system (Semi-algorithm 2) and also in a variant of Semi-algorithm 1 to check the smoothness of the curve \mathcal{C} and at the same time enclosing \mathcal{C} in a set of boxes of \mathbb{R}^n . The following is a specification of [Ibexsolve](#):

Black-box solver 7: [Ibexsolve](#)

Input: A system of equations, the ranges of the variables and non-negative reals

$$\text{eps}_{\min} \leq \text{eps}_{\max}.$$
¹⁴

Output: The following sets of boxes whose union contains all solutions of the given system:

- (a) A set of certified boxes, of width at most eps_{\min} , where every box contains a regular solution (resp. full-ranked-Jacobian part of the solution

¹⁴Recall that eps_{\min} is the parameter used in Semi-algorithm 6.

set) if the system is zero-dimensional (resp. of positive dimension).

- (b) A set of boxes, of width at most eps_{\max} , that intersect the boundary of the input box where the variables are defined and contain solutions of the given system. `Ibexsolve` provides an option to verify the regularity of the solutions inside these boxes.
- (c) A set of unknown-status boxes of width at least eps_{\min} .

Remark 6.1.1. *Detecting pairs of boxes in `enclosing_curve` satisfying (1) or (2) in Lemma 5.3.1 can be done using a black-box tool. It is a software that reports the pairs of intersected boxes of a given set of boxes. In our implementation, we use the SciPy class `KDTree` [VGO⁺2020].¹⁵ This class provides a tool for quick nearest-neighbor lookup for points in \mathbb{R}^n . Considering the center of every box in `enclosing_curve`, we can find the centers of the other boxes that intersect it. Although `KDTrees` tools offer a fast way to detect the neighbored boxes, it might return in addition spurious boxes which we simply remove by checking whether a pair of boxes returned by `KDTree` intersect.*

In addition to the previously-mentioned tools, other libraries are used such as `SymPy` [MSP⁺2017], `NumPy` [HMvdW⁺2020] and `Matplotlib` [Hea2007].

6.2 The Python software *Isolating_singularities*

The following is a short specification of the software *Isolating_singularities*¹⁶ which is a Python implementation of Semi-algorithm 6. As we mentioned in the introduction of this chapter, two main tasks can be performed using our software: computing the output of Semi-algorithm 6 (the function `enclosing_singularities`) and a visualization tool (`plotting_boxes`). There is an additional function `enclosing_curve` that performs a sub-task, namely, computing a set of n -boxes that enclose the curve \mathcal{C} . The output of this function is a part of the previous functions input. The following is a more detailed description of how to use these functions. By an interval we mean a list $[a, b]$ of floats a, b that represent the lower and upper bounds respectively. By a box we mean a list of intervals.

- `enclosing_curve(system, Box, X, eps_min, eps_max)`:

¹⁵This reference is related to SciPy which is a collection of open source software for scientific computing in Python where `KDTree` is a class of this collection; see <https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.KDTree.html>.

¹⁶The software is accessible at https://github.com/gkrait/Isolating_singularities.

This function returns a list of two lists: The first one is a list of boxes in \mathbb{R}^n that cover a smooth part of the curve (more precisely, where the used solver **IbexSolve** succeeds to certify the smoothness). The second one contains boxes where **IbexSolve** fails to certify the smoothness of the curve.

Parameters.

- *system*: a string that is the name of the txt file which contains the equations that define the curve
 - *Box*: a box in \mathbb{R}^n
 - *X*: a list of the **SymPy** symbols that appear in the equations of system
 - *eps_max* : the maximal width of the output boxes, that is, this is a criterion to force bisection: the width of output box will not be larger than *eps_max*. By default, *eps_max* = 0.1.
 - *eps_min* : the minimal width of output boxes. This is a criterion to stop bisection: for a box where **IbexSolve** fails to certify the regularity of the solutions inside that box, the width of this box will not be larger than *eps_min*. By default, *eps_min* = 10^{-7} .
- *enclosing_singularities* (*system*, *enclosing_curve*, *Box*, *X*, *eps_min*, *eps_max*, *delta_Taylor*)

The output of this function consists of the following two lists such that every solution of $\text{Ball}(P)$ (no matter it is regular or not) lives in the union of these sets of boxes:

- (a) A set of enclosing boxes, of width at most eps_{\max} , each of which contains one certified (regular) solution of $\text{Ball}(P)$. Thus, the plane projection of every box contains a unique singularity, namely, a node.
- (b) A set of boxes of widths at most eps_{\min} such that, in their projections, the behavior of the plane curve is unknown.

Parameters.

- *system*, *Box*, *X*, *eps_min* and *eps_max*: are as in *enclosing_curve*.
- *enclosing_curve*: is the output of function *enclosing_curve*.
- *threshold*: Represents the parameter δ_{Taylor} introduced in Section 5.3.2. When evaluating the Ball system, the parameter controls when to use the system in (4.1) or the Taylor form. The default value of this parameter is 10^{-2} .

- `plotting_boxes(certified_boxes, uncer_boxes, var, Box, nodes)`

Based on `NumPy` [HMvdW⁺2020] and `Matplotlib` [Hea2007], this function plots a plane projection of `enclosing_curve` output and `enclosing_singularities` output.

Parameters.

- `certified_boxes` : A list of boxes that cover a smooth part of the curve certified by `IbexSolve`. The function plots the plane projection of this list in green.
- `uncer_boxes` : A list of boxes that cover parts of the space curve with unknown status. The function plots the plane projection of this list in red.
- `var` (optional): A list of two integers that determines the variables for which the plane projection is considered. By default, the function considers the first two variables.
- `Box` (optional): determines the domain of the graph. By default, it is set to be $[-20, 20]^2$.
- `nodes` (optional): A set of boxes each of which contains a node.

7

Experiments

Contents

| | |
|---|---------------------|
| 7.1 Experiment 1: Analytic curve in \mathbb{R}^3 generating one node and one ordinary cusp | 112 |
| 7.2 Experiment 2: Analytic curve in \mathbb{R}^4 with many nodes | 113 |
| 7.3 Experiment 3: High degree algebraic curve in \mathbb{R}^4 | 114 |
| 7.4 Experiment 4: Two close lines in \mathbb{R}^3 generating a node | 115 |

In this section, we present four experiments performed with our software *Isolating_singularities*. The first example is pedagogical and considers a simple analytic curve in \mathbb{R}^3 that induces only one node and one ordinary cusp in \mathbb{R}^2 . The second example considers a smooth analytic curve in \mathbb{R}^4 that induces many nodes in \mathbb{R}^2 . The third one considers sparse but reasonably-high-degree algebraic equations in \mathbb{R}^4 . It should be stressed that, up to our knowledge, the two latter examples are out of reach by other methods: indeed, no other certified algorithm can handle non-algebraic curves in dimension higher than 3 and, for reasonably-high-degree algebraic equations in \mathbb{R}^4 , the bivariate equation defining their 2D projection often has a very high degree (see Section 7.3 for details).

Finally, in the fourth example, we exhibit the behavior of our software when a node in \mathbb{R}^2 is induced by a pair of points (on the space curve) that are very close. Indeed, when the equations defining the space curve are not algebraic, the Ball system contains a division by \sqrt{t} (due to the formula of $D \cdot P$), which may cause instability since t tends to zero when the distance between the pair of points tend to zero. For that purpose, we consider two very close skew lines defined analytically (and not algebraically).

We report the running times and other relevant parameters in Tables 7.1 and 7.2. Running times are in seconds on a sequential Intel(R) Core(TM) i7-7600U CPU @ 2.80GHz machine with Linux. We emphasize that the experiments are done

| | | Enclosing curve \mathcal{C} | | | Solving Ball system | | Output | |
|--------------|------------------|-------------------------------|--------------|------|---------------------|------|------------|-------------------|
| Experiments | Boxes max. width | Tree size | Output boxes | Time | Tree size | Time | Total time | Singularity boxes |
| Experiment 1 | 0.1 | 535 | 134 | 0.1 | 70 | 3.6 | 3.7 | 2 |
| | 0.03 | 1835 | 456 | 0.3 | 90 | 3.8 | 4.1 | |
| | 0.01 | 5427 | 1354 | 0.7 | 188 | 4.4 | 5.1 | |
| Experiment 2 | 0.1 | 2243 | 520 | 1.1 | 6098 | 52 | 53.1 | 43 |
| | 0.03 | 6759 | 1639 | 3.4 | 1078 | 35.4 | 38.8 | |
| | 0.01 | 19583 | 4847 | 10.2 | 372 | 35.8 | 45.6 | |
| Experiment 3 | 0.1 | 1151 | 203 | 1.0 | 655 | 4.2 | 5.2 | 7 |
| | 0.03 | 2503 | 523 | 1.8 | 272 | 3.5 | 5.3 | |
| | 0.01 | 6347 | 1482 | 4.5 | 163 | 5.7 | 10.2 | |

Table 7.1: Timings (in seconds) and numbers of boxes in Experiments 1 to 3.

with a prototype implementation that is under ongoing development. The *tree size* columns reports the total number of boxes created during the subdivision algorithm either for enclosing the curve \mathcal{C} in \mathbb{R}^n or for solving the Ball system in \mathbb{R}^{2n-1} . For the enclosing part, the column *output boxes* is the number of boxes in the set *enclosing_curve*. For each experiment, we provide a visualization of the plane projected curve $\pi_{\mathcal{C}}(\mathcal{C})$ with its singularities. On each figure, the green boxes are the plane projections of the boxes in *enclosing_curve* that enclose \mathcal{C} , hence these green boxes enclose $\pi_{\mathcal{C}}(\mathcal{C})$. The black boxes are the projections of the Ball system solution boxes identifying nodes of the plane curve $\pi_{\mathcal{C}}(\mathcal{C})$.

For each experiment, we consider three values of *eps_max* and it can be observed (see Table 7.1) that the smaller the value of *eps_max*, the larger the set *enclosing_curve*, and the longer it is to compute. As expected, even with the improvement to reduce the Ball system domains to be solved in, the subdivision in the high dimensional space \mathbb{R}^{2n-1} is the dominant step of the algorithm.

7.1 Experiment 1: Analytic curve in \mathbb{R}^3 generating one node and one ordinary cusp

We start with a pedagogical example pictured in Figure 7.1. Running times are given in Table 7.1. The curve \mathcal{C} is defined in the box $B = (-1, 4) \times (-1, 4) \times (-4.8, -1.4)$ by

$$P(x_1, x_2, x_3) = [x_1 - \cos(x_3)(3 + \sin^4(x_3)) + 3, \quad x_2 - \sin^2(x_3)(3 + \sin(2x_3))].$$

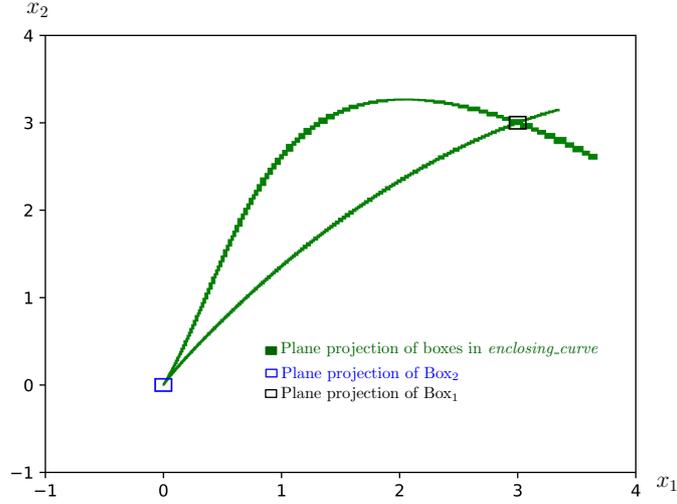


Figure 7.1: Experiment 1: Plane projection of an analytic curve in \mathbb{R}^3 with one node and one ordinary cusp.

Our improved Semi-algorithm 3 outputs the following solutions for the Ball system in \mathbb{R}^5 :

$$\begin{cases} N = \{\text{Box}_1 = [3, 3] \times [3, 3] \times [-3.15, -3.14] \times [1, 1] \times [2.4673, 2.4675] \\ U = \{\text{Box}_2 = [-0.06, 0.04] \times [-0.04, 0.07] \times [-3.15, -3.14] \times [1, 1] \times [-0.01, 0.01]\}. \end{cases}$$

Box_1 in the set N thus projects to a node of $\pi_{\mathcal{C}}(\mathcal{C})$. Box_2 being in the set U , one cannot decide whether its projection in the plane contains a node, a cusp or no singularity at all. On the other hand, one can notice on the equation $P = 0$ that the curve is parametrizable by the variable x_3 . It is thus an easy computation to check that for the value $x_3 = -\pi$, the point $q = (0, 0, -\pi)$ is on the curve \mathcal{C} and its tangent line at q is generated by the vector $(0, 0, 1)$ which is orthogonal to the projection plane. It is then clear that the projection of q generates a cusp that is witnessed by Box_2 .

7.2 Experiment 2: Analytic curve in \mathbb{R}^4 with many nodes

Figure 7.2 illustrates the output of our improved Semi-algorithm 4 for the curve defined by

$$\begin{aligned} P = [& x_1 + 2 \sin(x_1) - \cos(x_4) - (3 \cos(x_3) - \cos(2.8571x_3)), \\ & x_2 + 0.2 \cos(x_2) + (3 \sin(x_3) - \sin(2.8571x_3)) + \sin(x_4), \\ & x_4^2 - \sin(x_2)] \end{aligned}$$

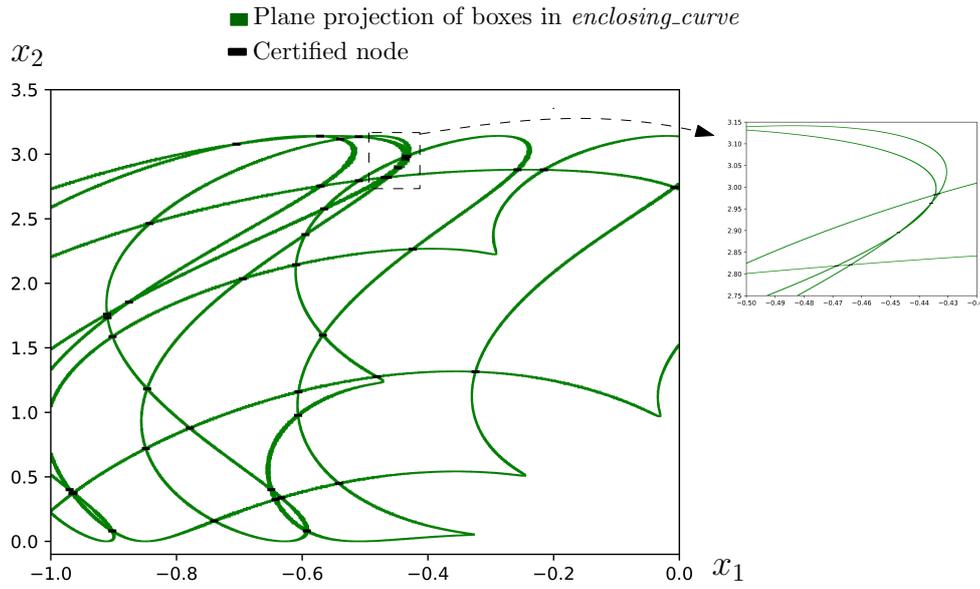


Figure 7.2: Experiment 2: Plane projection of an analytical curve \mathcal{C} in \mathbb{R}^4 . Each of the 43 black boxes contains a node of $\pi_{\mathcal{C}}(\mathcal{C})$ and is the projection of a box in \mathbb{R}^7 containing one solution of $\text{Ball}(P)$.

over the box $B = (-1, 0) \times (-0.1, 3.5) \times (-20, 20) \times (-10, 10)$. This curve has many nodes, some of them very close to each other. Running times are given in Table 7.1.

7.3 Experiment 3: High degree algebraic curve in \mathbb{R}^4

The goal of this experiment is to emphasize the genericity of the assumptions and the efficiency of our software in the sparse polynomial case. The curve \mathcal{C} is defined in the 4-box $B = (-1, 0.2) \times (-0.2, 1.4) \times (-10, 10)^2$ and is the zero set of three polynomials of degrees 17, 15 and 13, respectively that have a unique monomial of highest degree (which is monic) and 9 other random monomials of degrees at most 2 with integer coefficients in $(-25, 25)$.

$$\begin{aligned}
 P = [& x_1^{17} - 14x_1^2 - 7x_1x_3 - 7x_2^2 - 22x_2x_4 - x_3x_4 - 19x_4^2 + 8x_1 - 14x_3 + 9, \\
 & x_2^{15} + 8x_1x_3 - 14x_1x_4 - 15x_2^2 + 16x_2x_3 + 8x_2x_4 + 2x_3^2 + 13x_4^2 + 11x_1 + 11x_2, \\
 & x_3^{13} + 17x_1^2 - 15x_1x_2 + 4x_1x_3 - 20x_1x_4 + 2x_2^2 - 10x_2x_3 + 4x_2x_4 + 20x_4^2 - 23x_2].
 \end{aligned}$$

Figure 7.3 illustrates the 7 nodes of the projection of \mathcal{C} and running times are given in Table 7.1.

Note that since P is polynomial, the implicit equation of $\pi_{\mathcal{C}}(\mathcal{C})$ can be computed

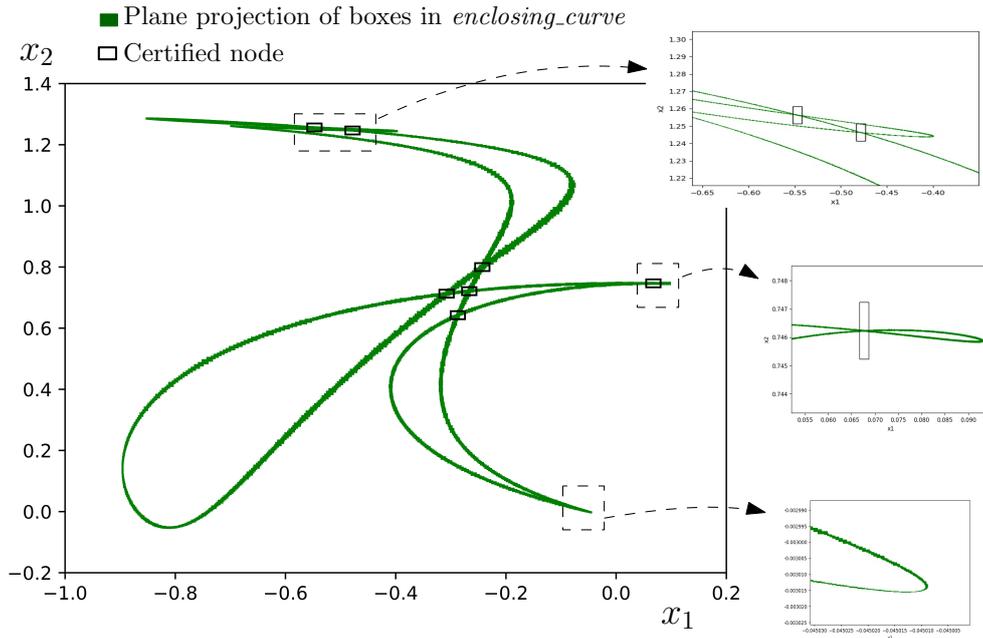


Figure 7.3: Experiment 3: High degree algebraic curve in \mathbb{R}^3 generating 7 nodes.

using elimination theory and its singularities can then be isolated using algebraic solvers. However, the implicit equation we obtained for $\pi_{\mathcal{C}}(\mathcal{C})$ is defined by an irreducible bivariate polynomial of degree 442 with 51074 monomials. Isolating the singularities of such a high-degree polynomial is then a real challenge.

Note also that our class of examples is rather specific and our software does not work that well if our defining polynomials are dense or even if they have non-monic high-degree monomials. However, it should be stressed that our software is a prototype and that it would be more efficient to use an interval solver specialized for the algebraic case than the versatile Ibex solver we used. Such a specialized approach has been proved successfully for generic curves in the 3-dimensional case by Imbach et al. [IMP2018].

7.4 Experiment 4: Two close lines in \mathbb{R}^3 generating a node

As mentioned in the preamble of Chapter 7, the purpose of this experiment is to study the behavior of our software when a node in \mathbb{R}^2 is induced by a pair of points (on the space curve) that are very close; namely when a node $(x_1, x_2) \in \mathbb{R}^2$ is induced by the pair of points $(x_1, x_2, y \pm r\sqrt{t}) \in \mathbb{R}^n$ with t that tends to zero. Indeed, when the equations defining the space curve are not algebraic, the Ball system contains a division by \sqrt{t} (due to the formula of $D \cdot P$), which may cause instability since t tends to zero when the distance between the pair of points tend

| ϵ -values | > 1 | 1 | 0.99 | 0.9 | 10^{-1} | 10^{-2} | 10^{-3} | 10^{-4} | 10^{-5} | 10^{-6} | 10^{-7} |
|---|-------------------------|----------------------------|---------------------------------------|-----|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| Time ($\delta_{\text{Taylor}} = 10^{-2}$) | \mathfrak{C} is empty | \mathfrak{C} is singular | Taylor forms are not triggered 0.1 | | | 0.8 | | | | | |
| Time without Taylor forms | | | | | | 0.1 | | | 0.1 | | 0.3 |

Table 7.2: Experiment 4: Performances for different values of ϵ .

to zero.

The simplest example to consider is the two skew lines $x_2 = x_1$ in the plane $x_3 = \epsilon$ and $x_2 = -x_1$ in the plane $x_3 = -\epsilon$, whose projection in the (x_1, x_2) -plane has a node at the origin, and to make ϵ vary towards 0. The pair of lines is thus defined by $[\epsilon x_2 - x_1 x_3, (x_3 - \epsilon)(x_3 + \epsilon)]$ but, in order to have non-algebraic equations, we replace x_3 by $\sin x_3$ and define our two lines by $P = [\epsilon x_2 - x_1 \sin x_3, \sin^2 x_3 - \epsilon^2]$ in the box $B = (-1, 1)^3$.

The goal of this experiment is to illustrate the stability of our software when ϵ varies towards 0. Recall from Section 5.3 that the D operator is evaluated on a box in two different ways depending on how close to zero is the t -interval of that box. The Ball system is thus solved either with Equation (4.1) (involving a division by \sqrt{t}) when the values of the t -interval are larger than a parameter δ_{Taylor} (set to 10^{-2}) or using a Taylor expansion otherwise.

To illustrate the stability of our software, we compared in Table 7.2 its running times when ϵ varies towards 0 with what it would be without using Taylor expansions. It shows that if we do not use Taylor expansions, the solution is not certified (by the Ibexsolver; see Section 6.1) when $\epsilon \leq 10^{-5}$. On the other hand, our software is stable, although its running time increases from 0.1 to 0.8 seconds when ϵ gets smaller than or equal to 10^{-2} , which is when the D operator starts to be evaluated using a Taylor expansion.

8

Applications in robotic mechanisms

Contents

| | |
|---|------------|
| 8.1 Preliminaries in robotics | 118 |
| 8.2 Description of <u>RRRRR</u> robot | 122 |
| 8.3 Results | 123 |
| 8.3.1 The parallel singularity $S_{parallel}$ | 123 |
| 8.3.2 The serial singularity S_{serial} | 128 |

In this section we use our software *Isolating_singularities* to deal with a classical problem in robotics design and analysis. This problem is about determining the configurations in which a robot is allowed to move without damaging it; see for example [BL1986, Cha2010, CCG⁺2012]. Caro et al. [CCG⁺2012] provided an approach, based on branch and prune algorithm, to compute a certified enclosure of the generalized aspects (see Definition 8.1.1 (g)) which fully automatically applies to arbitrary robot kinematic model under numerical constrains [CCG⁺2012, §4.1]. However, regions that include the serial and parallel singularities (i.e., the boundaries of the generalized aspects) are left undecided since the numerical constrains are not satisfied. The results we present in section 8.3 can be considered as a continuation of the above-mentioned work in [CCG⁺2012]. More precisely, we consider the manipulator RRRRR which we describe in Section 8.2. We certify the location of the geometric singularities (defined in Section 2.2.1) appearing in the reachable workspace and reachable joint-space (Definitions 8.1.1 (d)) as long as our strong assumptions are satisfied. Before that, we start with recalling preliminaries in Section 8.1.

8.1 Preliminaries in robotics

Quoted from [LP2017, §2], a robot is mechanically constructed by connecting a set of bodies, called links, to each other using various types of joints. Actuators, deliver forces or torques that cause the robot's links to move. The positions that the robot attains are called configurations of the robot. The number of degrees of freedom of a robot is, roughly speaking, the smallest number of real-valued coordinates needed to represent the robot's configurations.

Robots can be classified into serial and parallel. Serial robots [Mer2006, §1.1] consist of series of links that are connected by joints equipped with motors. A parallel robot [Mer2006, §1.4.3] is made up of an end-effector with k degrees of freedom, and of a fixed base, linked together by at least two independent kinematic chains. Actuation takes place through k simple actuators.

Since the example we consider in our experiment is a parallel robot, we start with defining related vocabulary for modeling parallel robots:

Definitions 8.1.1. *Consider the following notions.*

- (a) ([Mer2006, §1.4.3]) *A fully parallel robot is a parallel robot for which the number of chains is equal to the number of degrees of freedom of the end-effector.*
- (b) [CCG⁺2012, §3.2] *The pose-command space is the product of the pose and command spaces which are ambient spaces of all poses and all commands respectively.*
- (c) [LP2017, Definition 2.1] *For a robot with k degrees of freedom, the k -dimensional variety Σ containing all possible configurations of the robot is called the configuration space (or C-space). The C-space Σ is embedded in the pose-command space. The C-space Σ of a robot is presented as a system of equations $F(x, q) = 0$ that characterizes the relation between the commands q (which represent the command variables) and the corresponding poses x .*
- (d) [CCG⁺2012, §3.2] *The projection of Σ on the pose space (resp. command space) is called the reachable workspace Σ_x (resp. reachable joint-space Σ_q).*
- (e) [GA1990, Page 282] *Let J_F be the Jacobian matrix of F . Define the direct kinematic matrix A (resp. the inverse-kinematics B) to be the sub-matrix of J_F obtained by removing the columns corresponding to q (resp. x).*
- (f) [GA1990, Page 282] *The parallel (resp. serial) singularity set S_{parallel} (resp. S_{serial}) is the set of points (x, q) in Σ such that $\det(A(x, q)) = 0$ (resp.*

$\det(B(x, q)) = 0$). The serial (resp. parallel) singularities are also called first (resp. second) kinds of singularities.

- (g) [CW1998, Definition 5] A generalized aspect \mathbb{A} is defined as a maximal subset in Σ such that \mathbb{A} is connected and for all $(x_0, q_0) \in \mathbb{A}$ the matrices $A(x_0, q_0)$ and $B(x_0, q_0)$ are full-rank.

In this section, we assume that the dimensions of the pose and joint spaces are both equal to $\text{corank}(J_F(x, q))$ for a generic $(x, q) \in \Sigma$.

Remark 8.1.1. It is important to emphasize that the notions of serial and parallel singularities differ from the singularity defined on Section 2.2.1. In the rest of this section, by geometric singularity we mean the one defined in Section 2.2.1.

To illustrate the objects that are defined above, we start with a simple example presented by Caro et al. [CCG⁺2012]. Consider the planar robot \underline{PRRP} depicted in Figure 8.1. A demo for simulating the motion of this robot is accessible in [Kra2021b] The robot consists of two prismatic joints sliding along two perpendicular axes. The positions of the prismatic joints are denoted by x (along x-axis) and q (along q-axis). These prismatic joints are connected through rigid links of lengths a, l and b respectively, with l is strictly larger than $\max(a, b)$. The three links are connected to each other by two revolute joints each of which is rotatable 360 degrees.

As illustrated in Figure 8.2, the pose-command space is hence \mathbb{R}^2 and the configuration space Σ is the solution set of the equation $(x - a)^2 + (q - b)^2 = l^2$. The pose space (resp. joint space) is $[a - l, l + a]$ (resp. $[b - l, l + b]$). The serial singularity S_{serial} is the set $\{(a + l, b), (a - l, b)\}$. Figure 8.4 illustrates the positions of the robot corresponding to the serial singularities. In these two configurations, the robot motion is restricted but not damaged. However, when the robot passes through a point in the parallel singularity $S_{\text{parallel}} = \{(a, l + b), (a, b - l)\}$ (the corresponding positions are depicted in Figure 8.3), a possibility of robot damage exists. Hence, in the design and the path planning processes of the robot, it is important to determine the location of the serial and parallel singularities in order to avoid any dangerous motion.

The green curve in Figure 8.2 is the union of all generalized aspects (Definition 8.1.1 (g)) of the robot. The robot is safely allowed to move from and to two points that are in the same generalized aspect (along that aspect). Notice that for every point (x_0, q_0) in a generalized aspect, there exists a local bijection between the commands and the poses near that point. Equivalently, both of the matrices $A(x_0, q_0)$ and $B(x_0, q_0)$ are invertible.

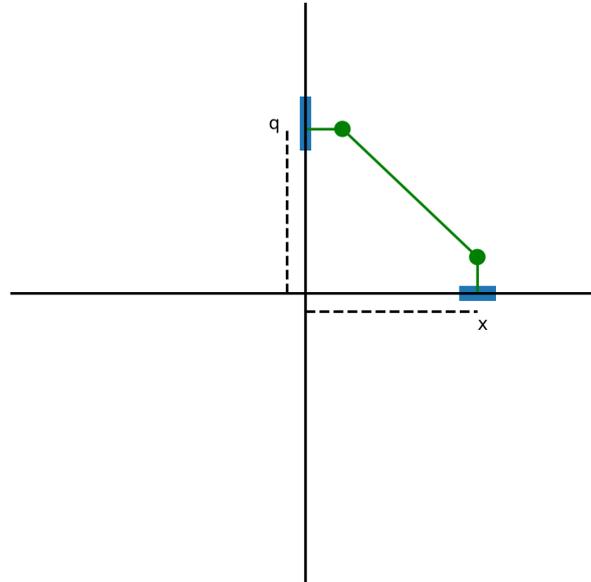


Figure 8.1: A non-singular configuration of \underline{PRRP} where there is a local bijection between poses and commands.

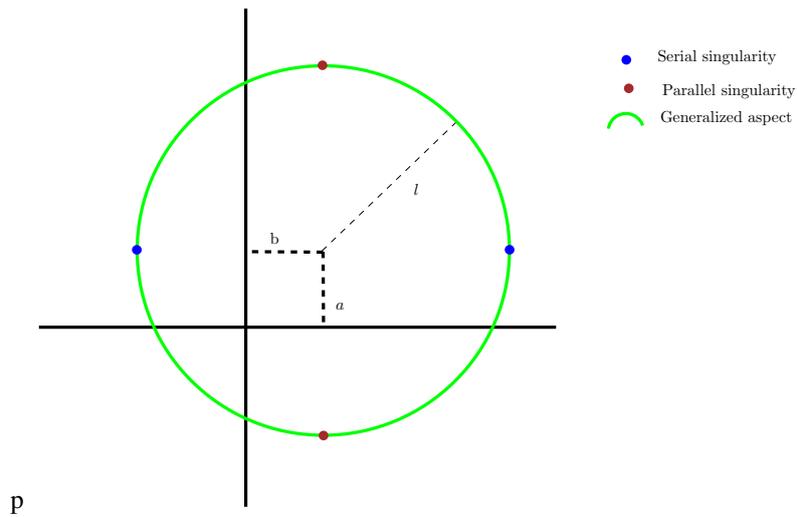


Figure 8.2: Geometric modelling of \underline{PRRP} motion.



Figure 8.3: The parallel singular poses of \underline{PRRP} where the pose x admits two commands.

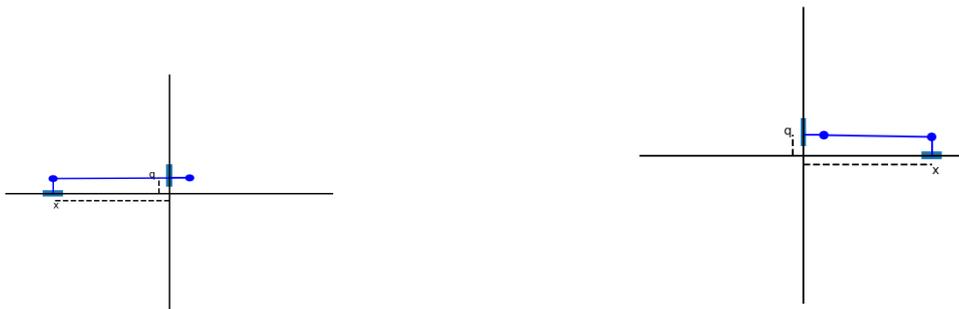


Figure 8.4: The serial singular poses of \underline{PRRP} where the command q admits two poses.

In the following section we consider a more challenging example, namely, \underline{RRRRR} robot where the serial and parallel singularities are more complicated than in the above example.

8.2 Description of \underline{RRRRR} robot

Consider the robot \underline{RRRRR} (Figure 8.5) that consists of five links AP' , $P'P$, PP'' , $P''B$ and BA connected by five joints A, P', P, P'' and B . The links' lengths are respectively 8, 5, 8, 5 and 9. The angles BAP' and ABP'' are the commands q_1, q_2 respectively that vary in the interval $(-\pi, \pi)$. The pose variables x_1, x_2 are the coordinates of the point P and vary in the interval $[-15, 15]$. A simulation of this robot motion is accessible in [Kra2021a]. The kinematic equations are:

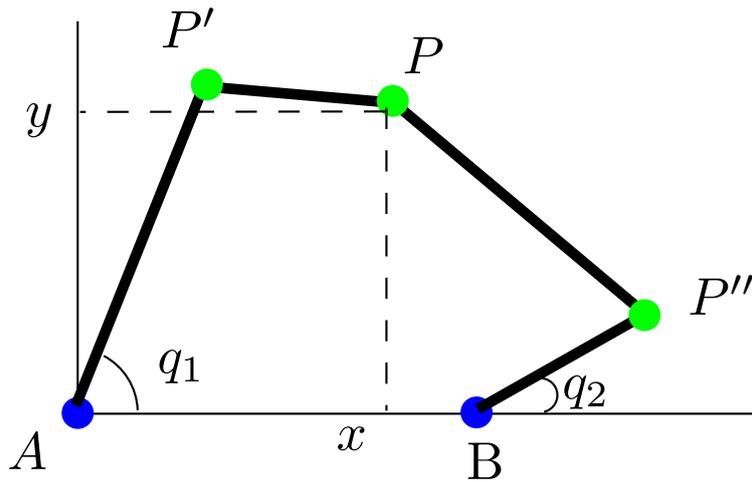


Figure 8.5: \underline{RRRRR} robot in a non-singular configuration.

$$\begin{cases} (x_1 - 8 \cos(q_1))^2 + (x_2 - 8 \sin(q_1))^2 - 25 = 0, \\ (x_1 - 9 - 5 \cos(q_2))^2 + (x_2 - 5 \sin(q_2))^2 - 64 = 0. \end{cases} \quad (8.1)$$

Thus, the configuration space is a surface in the four-dimensional vector space represented by the variables x_1, x_2, q_1 and q_2 . Using the notation of Definitions 8.1.1 (e,f), the parallel (resp. serial) singularity is the set of points (x_1, x_2, q_1, q_2) in the configuration space (i.e., satisfying (8.1)) that satisfy the equation $\det(A) = 0$ (resp. $\det(B) = 0$). In other words, the functions P_1 and P_2 that define the parallel

and serial singularities are respectively:

$$\begin{aligned}
P_1 = & \\
& [(x_1 - 8 \cos(q_1))^2 + (x_2 - 8 \sin(q_1))^2 - 25, \\
& (x_1 - 9 - 5 \cos(q_2))^2 + (x_2 - 5 \sin(q_2))^2 - 64, \\
& \det(A) = (2x_1 - 16 \cos(q_1))(2x_2 - 10 \sin(q_2)) - (2x_2 - 16 \sin(q_1)) \\
& \times (2x_1 - 10 \cos(q_2) - 18)].
\end{aligned} \tag{8.2}$$

and

$$\begin{aligned}
P_2 = & \\
& [(x_1 - 8 \cos(q_1))^2 + (x_2 - 8 \sin(q_1))^2 - 25, \\
& (x_1 - 9 - 5 \cos(q_2))^2 + (x_2 - 5 \sin(q_2))^2 - 64, \\
& \det(B) = (16(x_1 - 8 \cos(q_1)) \sin(q_1) - 16(x_2 - 8 \sin(q_1)) \cos(q_1)) \\
& \times (-10(x_2 - 5 \sin(q_2)) \cos(q_2) + 10(x_1 - 5 \cos(q_2) - 9) \sin(q_2))]
\end{aligned} \tag{8.3}$$

Figures 8.6 and 8.7 illustrate the projections of the parallel and serial singularities on different variable sub-spaces. These figures are plots of the boxes returned by *enclosing_curve*.

8.3 Results

As mentioned in the preamble of this chapter, our goal is to certify the nodes of the projections of the parallel and serial singularities with respect to the command and pose spaces. This is a continuation of classical work in robotics such as [CCG⁺2012]. Applying the functions *enclosing_curve* and *enclosing_singularities* in software *Isolating_singularities* we get the following results:

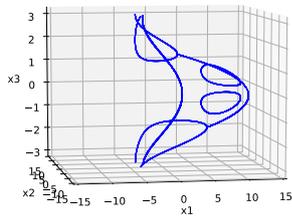
8.3.1 The parallel singularity $S_{parallel}$

All boxes that *enclosing_curve* returns are certified. Hence, P_1 satisfies Assumption \mathcal{A}_1 . Taking the plane projections of $S_{parallel}$ with respect to (x_1, x_2) and (q_1, q_2) and solving $\text{Ball}(P_1)$ accordingly we get:

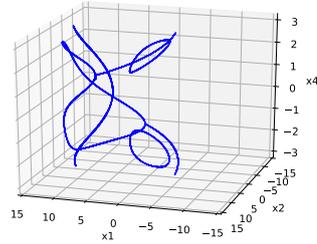
- The plane projection of $S_{parallel}$ with respect to x_1 and x_2 :

The output of *Isolating_singularities* consists of:

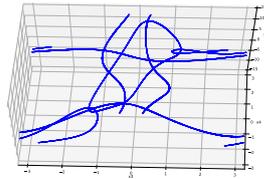
- (a) an empty set of uncertified solutions of $\text{Ball}(P_1)$. Hence, P_1 satisfies Assumption \mathfrak{N}_1 in $\overline{B}_{\text{Ball}}$,



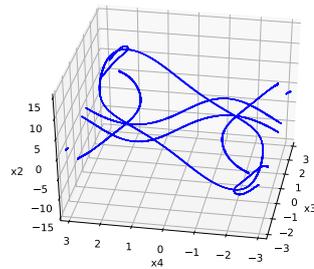
(a) The projection of S_{serial} on \mathbb{R}^3 with respect to (x_1, x_2, q_1) .



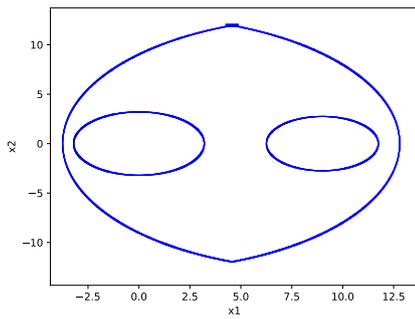
(b) The projection of S_{serial} on \mathbb{R}^3 with respect to (x_1, x_2, q_2) .



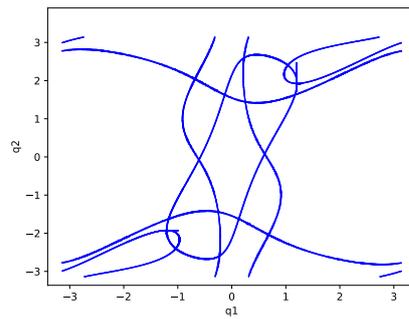
(c) The projection of S_{serial} on \mathbb{R}^3 with respect to (q_1, q_2, x_1) .



(d) The projection of S_{serial} on \mathbb{R}^3 with respect to (q_1, q_2, x_2) .

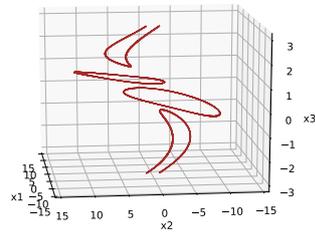


(e) The plane projection of S_{serial} on \mathbb{R}^3 with respect to (x_1, x_2) .

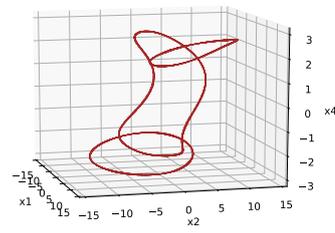


(f) The plane projection of S_{serial} on \mathbb{R}^3 with respect to (q_1, q_2) .

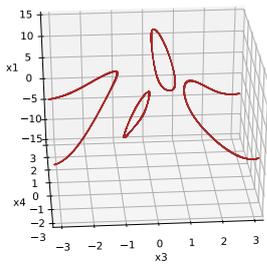
Figure 8.6: Illustration of the projections of S_{serial} on different variable sub-spaces.



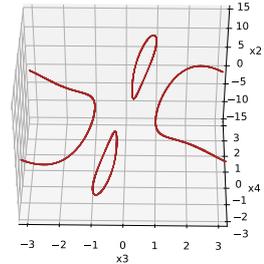
(a) The projection of $S_{parallel}$ on \mathbb{R}^3 with respect to (x_1, x_2, q_1) .



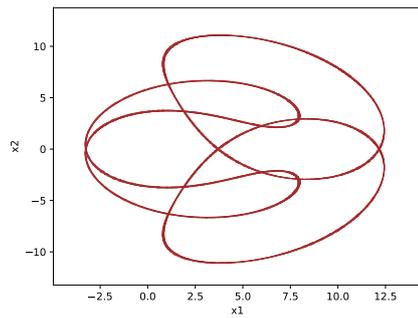
(b) The projection of $S_{parallel}$ on \mathbb{R}^3 with respect to (x_1, x_2, q_2) .



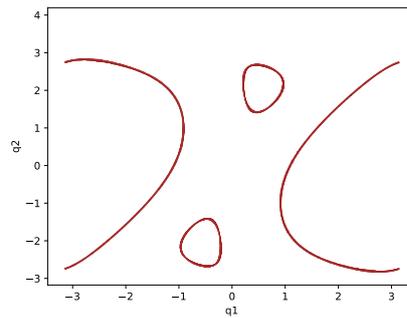
(c) The projection of $S_{parallel}$ on \mathbb{R}^3 with respect to (q_1, q_2, x_1) .



(d) The projection of $S_{parallel}$ on \mathbb{R}^3 with respect to (q_1, q_2, x_2) .



(e) The plane projection of $S_{parallel}$ on \mathbb{R}^3 with respect to (x_1, x_2) .



(f) The plane projection of $S_{parallel}$ on \mathbb{R}^3 with respect to (q_1, q_2) .

Figure 8.7: Illustration of the projections of $S_{parallel}$ on different variable subspaces.

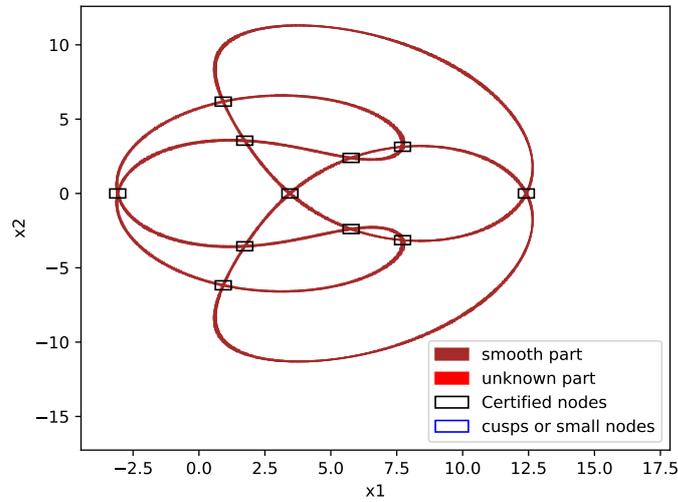


Figure 8.8: Enclosing the plane projection of $S_{parallel}$ with respect to x_1 and x_2 .

- (b) an empty set of solution enclosures $\mathbf{X} = (x_1, x_2, y, r, t)$, with $0 \in t$, and
- (c) a list of eleven $\text{Ball}(P_1)$ solutions that are corresponding to nodes. As shown in Table 8.1, every row represents an enclosing box of a solution of $\text{Ball}(P_1)$. Notice that the boxes returned by *enclosing_singularities* enclose only one solution of every pair of twin solutions. Moreover, recall that *enclosing_singularities* checks Assumptions \aleph_2 and \aleph_3 which is straightforward to verify by hand from Table 8.1. Hence, By Lemma 5.2.2, P_1 satisfies Assumptions $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4$ and \mathcal{A}_5 .

Figure 8.8 shows the plane projection of $S_{parallel}$ with respect to x_1 and x_2 (brown boxes) in addition to the boxes enclosing the solutions of $\text{Ball}(P_1)$ (black boxes). In fact, the black boxes are inflations of the output of *Isolating_singularities* since the width of the latter ones are not large enough to be visualized.

- The plane projection with respect to q_1 and q_2 :

In this case, *enclosing_singularities* returns an empty set of solutions (certified and uncertified) of $\text{Ball}(P_1)$. Hence, Assumptions $\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3, \mathcal{A}_4$ and \mathcal{A}_5 are satisfied. By Theorem 4.1.3, the plane projection of $S_{parallel}$ with respect to q_1 and q_2 is smooth; see Figure 8.9.

| x1 | x2 | x3 | x4 | z | r4 | t |
|---|---|---|--|---|---|---|
| [3.082612383354, -3.0826123833399] | [4.32953244422151e-14, 4.74276298769349e-14] | [1.00005123941644e-14, 9.31060578936555e-15] | [4.82757003508379e-15, 4.2642170390076e-15] | [0.743897901776617, 0.743897901776618] | [0.668293282722748, -0.668293282722746] | [1.5.8228036397883, 1.5.8228036397884] |
| [12.4143983309575, 12.4143983309576] | [2.19643732065757e-14, 2.16843986868023e-14] | [1.76557550892842e-15, 1.7608276928627e-15] | [3.26320463863822e-15, 3.493881940039759e-15] | [0.09400663393605286, 0.09400663393605299] | [0.995565931417558, 0.995565931417558] | [6.3027493738415, 6.30274937384154] |
| [7.72198469285079, 7.72198469285081] | [3.14146695654038, 3.1414669565404] | [0.386372842093721, 0.386372842093724] | [1.18442348470118, -1.18442348470117] | [0.704545623272464, 0.704545623272469] | [0.709658695943063, 0.709658695943068] | [0.776365783937367, 0.776365783937377] |
| [5.76855440419258, 5.7685544041926] | [2.39244228475219, 2.3924422847522] | [0.393147161987983, 0.393147161987985] | [1.6041358360562, -1.6041358360562] | [1.0, 1.0] | [1.35072900290922e-15, 1.3524309601545e-15] | [0.455802586765906, 0.455802586765907] |
| [3.44858372485897, 3.44858372485899] | [7.82218870513019e-15, 7.05250886308966e-15] | [1.00674811863852e-15, 9.00186606496201e-16] | [1.88040204779071e-15, 2.0639656946144e-15] | [0.268547393470069, 0.26854739347007] | [0.963264472716886, 0.963264472716886] | [2.18193106992156, 2.18193106992156] |
| [1.73739024480918, 1.7373902448092] | [3.55218064581808, 3.552180644581809] | [1.115902433489, 1.11590243348901] | [0.454893893305893, -0.454893893305891] | [0.267600107931582, -0.267600107931581] | [0.96353006296379, 0.963530062963791] | [3.90057257003159, 3.90057257003161] |
| [7.7219846928508, 7.72198469285081] | [3.14146695654039, -3.14146695654038] | [0.386372842093724, -0.386372842093722] | [1.18442348470117, 1.18442348470118] | [0.704545623272468, -0.704545623272464] | [0.709658695943067, -0.709658695943064] | [0.776365783937368, 0.776365783937376] |
| [5.76855440419258, 5.7685544041926] | [2.39244228475219, -2.39244228475219] | [0.393147161987983, -0.393147161987981] | [1.6041358360562, 1.6041358360562] | [1.0, 1.0] | [1.12614686367309e-15, 1.57701309939061e-15] | [0.455802586765905, 0.455802586765907] |
| [1.73739024480918, 1.7373902448092] | [3.55218064581809, -3.55218064581808] | [1.11590243348901, -1.115902433489] | [0.454893893305893, 0.454893893305893] | [0.267600107931581, 0.267600107931582] | [0.963530062963791, -0.96353006296379] | [3.90057257003159, 3.90057257003161] |
| [0.920765994156912, 0.920765994156939] | [6.17674590722489, 6.17674590722491] | [1.4228162805306, 1.4228162805306] | [1.6041358360562, 1.6041358360562] | [1.0, 1.0] | [4.3576092934693e-16, 6.34230694600428e-16] | [0.455802586765906, 0.455802586765907] |
| [0.920765994156913, 0.920765994156938] | [6.17674590722489, -6.17674590722489] | [1.4228162805306, -1.4228162805306] | [1.6041358360562, -1.6041358360562] | [1.0, 1.0] | [5.98565367784164e-16, 4.71428056750956e-16] | [0.455802586765906, 0.455802586765907] |

Table 8.1: The list of certified nodes enclosure that *enclosing_singularities* returns, where every row represents an enclosing box of a solution of $\text{Ball}(P_1)$. Notice that *enclosing_singularities* returns exactly one solution of every pair of twin solutions.

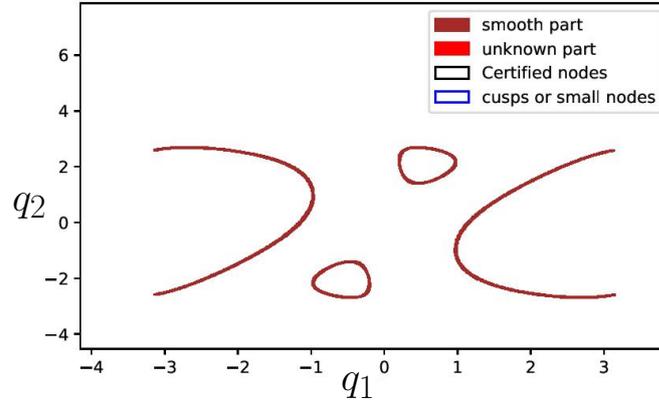


Figure 8.9: The plane plotting of *Isolating_singularities* output for $S_{parallel}$ with respect to q_1 and q_2 .

8.3.2 The serial singularity S_{serial}

Applying *enclosing_curve* to P_2 , both sets of certified and uncertified boxes are not empty (for $\epsilon_{\min} > 0$). Hence, *enclosing_curve* does not certify the smoothness of S_{serial} . Indeed, using a computer algebra system, we checked that the points $(\frac{9}{2}, \frac{\sqrt{595}}{2})$ and $(\frac{9}{2}, -\frac{\sqrt{595}}{2})$ are both the plane projections (with respect to x_1, x_2) of singular points of S_{serial} . Projecting to the planes (x_1, x_2) and (q_1, q_2) and solving $\text{Ball}(P_2)$ accordingly, we get:

- The plane projection with respect to x_1 and x_2 ; Figure 8.10:

For all values of the parameter ϵ_{\min} that we chose, *Isolating_singularities* returns an empty set of certified Ball solutions. Moreover, the plane projections of uncertified boxes cover the projection of S_{serial} . To understand this result, see Figure 8.11, where two different configurations have the same position $P = (x_1, x_2)$. This holds for uncountable number of configurations. Thus, Assumption \mathcal{A}_2 is not satisfied (i.e., \mathcal{L}_n is uncountable). In fact, using a similar argument, we can see that \mathcal{L}_n is equal to S_{serial} except the singular points.

- The plane projection with respect to q_1 and q_2 ; Figure 8.12:

In this case, *enclosing_singularities* returns an empty set uncertified $\text{Ball}(P_2)$ solutions in addition to 14 boxes each of which encloses a certified node (Table 8.2). In Figure 8.12 we illustrate the projection of S_{serial} with respect to q_1 and q_2 together with certified nodes which are inflated in the figure so they can be seen. The projection of unknown boxes (which contain geometric singularities of S_{serial}) are plotted in red.

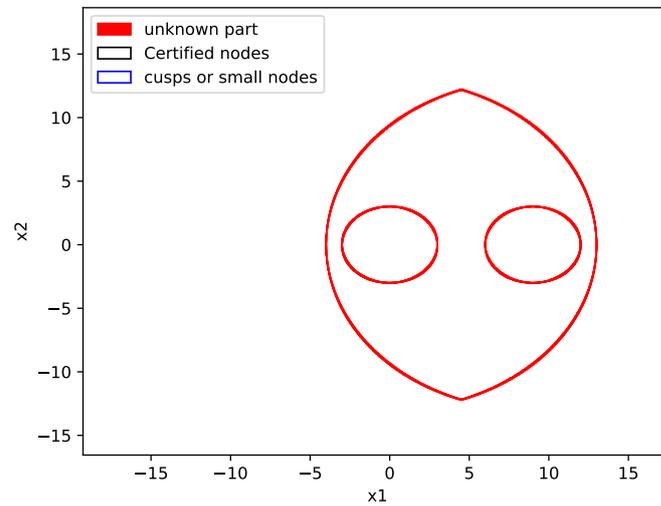


Figure 8.10: Plotting the result of applying *Isolating_singularities* to S_{serial} with respect to x_1, x_2 , where it plots only red boxes since neither Assumption \mathcal{A}_2 nor Assumption \mathcal{A}_1 is satisfied.

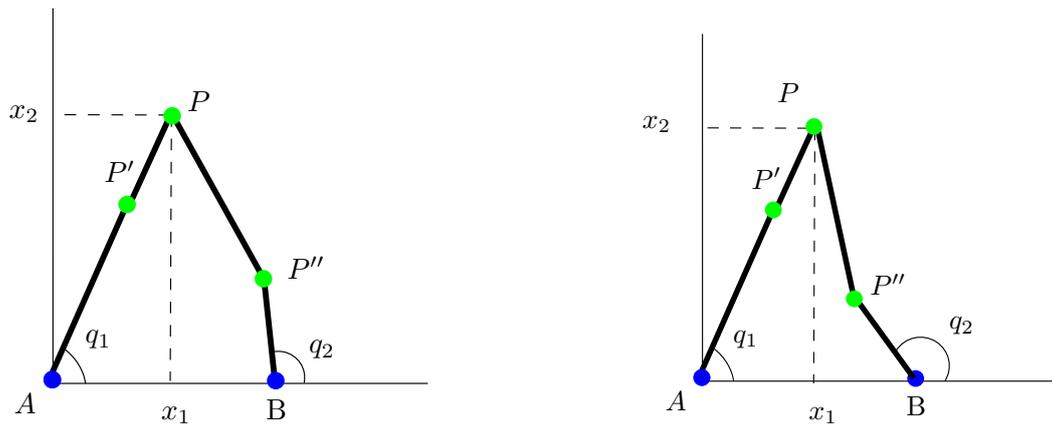


Figure 8.11: An example of two different (serial) singular configurations of \underline{RRRRR} that define the same pose (x_1, x_2) .

| x1 | x2 | x3 | x4 | r3 | r4 | t |
|--|---|---|--|--|---|---|
| [1.045652017194436, 1.04565201719444] | [2.408973115811948, 2.408973115811954] | [2.926433957572022, 2.926433957572062] | [9.97146124008183, 9.97146124008188] | [0.9422126706172985, 0.9421216706173009] | [0.3350153478994864, 0.3350153478994923] | [1.45252250070272, 1.45252250070274] |
| [1.17764916480691, 1.177649164806915] | [1.604135836056195, 1.604135836056203] | [3.06477892914073, 3.064778929140758] | [7.38966373440301, 7.38966373440316] | [0.3830973661425916, 0.3830973661425948] | [0.9237079668675628, 0.9237079668675643] | [2.499999999999996, 2.500000000000005] |
| [0.2207608354396288, 0.2207608354396333] | [2.609778324737173, 2.609778324737185] | [1.213508501616866, 1.213508501616868] | [0.6626714857892166, 0.6626714857892166] | [0.2439661633826514, 0.2439661633826514] | [0.9697832413770097, 0.9697832413770097] | [5.071566226733524, 5.071566226733524] |
| [0.7246545121089883, 0.7246545121089882] | [2.007831411346909, -2.0078314113469] | [6.257975350818948, 6.257975350818971] | [2.353331047150218, 2.353331047150238] | [0.9958933752895709, -0.9958933752895703] | [0.09053388898278258, -0.0905338889827812] | [1.62274647679033, 1.62274647679048] |
| [0.1479800462642971, 0.1479800462642971] | [1.604135836056195, 1.604135836056195] | [7.912567349176403, 7.912567349176411] | [1.179524457036588, 1.179524457036588] | [0.9890709186470505, 0.9890709186470513] | [0.1474405571295711, 0.1474405571295743] | [2.499999999999997, 2.500000000000003] |
| [0.6296139402821168, 0.6296139402821204] | [0.08968188640931876, 1.604135836056202] | [8.259684894168872, 7.912567349176411] | [3.693070408800386, 3.693070408800381] | [0.4934502112526666, 0.9890709186470513] | [0.8697740448039905, 0.8697740448039933] | [2.074734200111457, 2.07473420011147] |
| [0.1479800462642971, 0.1479800462642943] | [1.604135836056195, 1.604135836056202] | [7.912567349176403, 7.912567349176411] | [1.179524457036588, -1.179524457036574] | [0.9890709186470505, 0.9890709186470513] | [0.1474405571295742, -0.1474405571295711] | [2.499999999999997, 2.500000000000003] |
| [0.7246545121089883, 0.7246545121089882] | [2.007831411346909, 2.007831411346909] | [6.257975350818947, 6.257975350818973] | [2.353331047150239, -2.353331047150217] | [0.9958933752895711, -0.9958933752895702] | [0.09053388898277859, 0.09053388898278265] | [1.62274647679031, 1.62274647679048] |
| [0.6296139402821204, 0.6296139402821161] | [0.08968188640932837, -0.0896818864093167] | [8.25968489416887, 8.259684894168895] | [3.693070408800383, -3.693070408800354] | [0.4934502112526664, 0.4934502112526716] | [0.8697740448039935, -0.8697740448039902] | [2.074734200111456, 2.07473420011147] |
| [0.2478711625342984, -0.2478711625342982] | [1.428558711394765, -1.428558711394758] | [5.74151763387491, 5.741517633874925] | [1.116840795820682, 1.116840795820698] | [0.8369179815425689, 0.8369179815425714] | [0.547328322098088, 0.5473283220980916] | [1.146016277422995, 1.146016277423008] |
| [1.177649164806914, -1.177649164806911] | [1.604135836056201, -1.604135836056196] | [3.064778929140735, 3.064778929140753] | [7.3896637344040515, -7.3896637344040504] | [0.3830973661425941, -0.3830973661425922] | [0.9237079668675641, 0.9237079668675641] | [2.499999999999997, 2.500000000000004] |
| [1.04565201719444, -1.045652017194438] | [2.408973115811954, -2.408973115811948] | [2.926433957572022, 2.926433957572065] | [9.971461240081863, -9.97146124008183] | [0.9422126706172984, 0.9422126706173011] | [0.3350153478994925, -0.3350153478994859] | [1.452522500702719, 1.452522500702741] |
| [0.2478711625342984, 0.2478711625343013] | [1.428558711394758, 1.428558711394765] | [5.741517633874912, 5.741517633874925] | [1.116840795820698, -1.116840795820683] | [0.836917981542569, 0.8369179815425712] | [0.5473283220980913, -0.5473283220980882] | [1.146016277423007, 1.146016277423007] |
| [0.2207608354396332, -0.2207608354396283] | [2.609778324737186, -2.609778324737173] | [1.213508501616868, 1.213508501616868] | [0.6626714857892184, -0.6626714857891866] | [0.2439661633826517, 0.2439661633826517] | [0.9697832413770098, -0.9697832413770076] | [5.071566226733526, 5.071566226733526] |

Table 8.2: The list of certified nodes enclosure that *enclosing_singularities* returns, where every row represents an enclosing box of a solution of $\text{Ball}(P_2)$. Notice that *enclosing_singularities* returns exactly one solution of every pair of twin solutions.

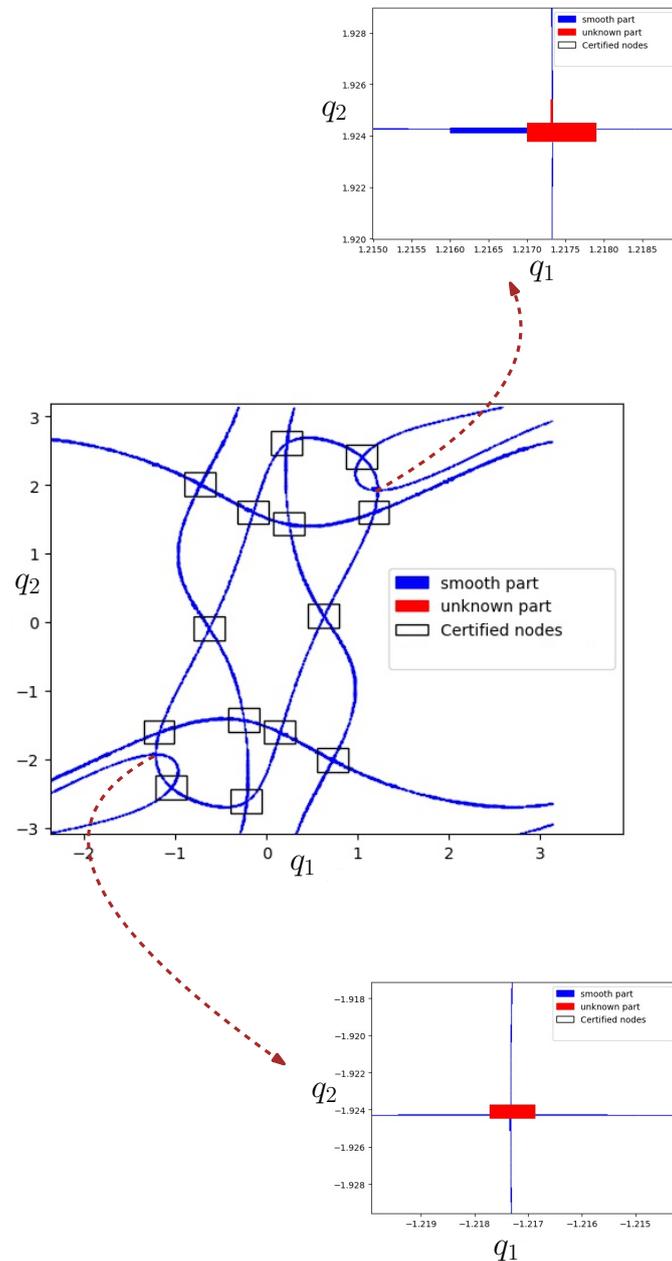


Figure 8.12: An illustration of `enclosing_singularities` output for the serial singularity S_{serial} of the robot \underline{RRRRR} . The projection is with respect to the (q_1, q_1) -plane (the command variables). The certification of the geometric singularities is done only where our strong assumptions are satisfied. The red boxes contain the projections of the (geometric) singular points of S_{serial} .

Conclusion

9

Conclusion

We addressed in this thesis the problem of isolating the singularities of plane curves using *certified* and *numerical* methods. This is motivated by the fundamental problem of approximating (drawing) plane curves with the correct topology. We specifically considered the case of curves that are given as the plane projection of generic implicit curves in \mathbb{R}^n . The study of such curves was motivated by applications in robotics design. Our contribution in this thesis has two parts:

Theoretical contributions. Given a generic curve \mathcal{C} contained in an n -dimensional closed box B , defined as the zero locus of a C^∞ -function from \mathbb{R}^n to \mathbb{R}^{n-1} , we identified in Chapter 4 a regular square system, called the Ball system that encodes the singularities of the plane projection $\pi_{\mathcal{C}}(\mathcal{C})$ of \mathcal{C} . The idea of the Ball system is based on a theoretical analysis of generic curves in \mathbb{R}^n that classifies the points of \mathcal{C} depending on the types of their projections. Thanks to its (square and regular) structure, this system is solvable using efficient certified numerical methods such as Interval Newton [Neu1991]. In Section 3.2, we precisely defined the conditions that \mathcal{C} must satisfy in order to guarantee the correctness of our results. Furthermore, in the same section, we proved the genericity of these conditions via transversality theory. In the same way, we studied whether the silhouette of a generic surface in \mathbb{R}^n satisfies our conditions. In this case, we only proved the genericity of some of the assumptions and we stated the genericity of the missing part as Conjecture 3.2.18.

Algorithmic contributions and applications. We presented semi-algorithms that terminate if and only if the above-mentioned generic conditions are satisfied (Chapter 5). Furthermore, under the same assumptions, the semi-algorithms provide a set (or a superset) of isolating boxes of $\pi_{\mathcal{C}}(\mathcal{C})$ singularities. We also presented an implementation of our semi-algorithms which we modified (by adding a threshold parameter eps_{\min}) to ensure its termination even when our assumptions are not

satisfied. Hence, given any curve \mathcal{C} in \mathbb{R}^n , our software provides two sets of $2D$ -boxes: (i) a set of boxes each of which contains exactly one singularity, namely, a node of $\pi_{\mathcal{C}}(\mathcal{C})$ and (ii) a set of boxes of widths smaller than eps_{\min} and above which our software is unable to verify the assumptions, and where, the behavior of $\pi_{\mathcal{C}}(\mathcal{C})$ inside these boxes is thus unknown. Furthermore, any singularity of $\pi_{\mathcal{C}}(\mathcal{C})$ is ensured to be in the union of these two sets of boxes even when our assumptions are not satisfied. We also discuss the performance of our software in Chapter 7 on non-algebraic instances for which no certified methods used to exist and algebraic instances that used to be out of reach by known certified methods. Finally, we finished this thesis with an application in robotics in Chapters 8.

Strengths, weaknesses and open problems

Our approach considers not only algebraic curves, but also those that are defined by C^∞ -functions as long as interval evaluations of the functions and their derivatives are provided. Moreover, our approach is the first certified numerical method that considers the curve \mathcal{C} to be in \mathbb{R}^n for an arbitrarily large n .

Our Ball-system-based approach is certified and appears to be efficient in several examples (see Chapter 7). Furthermore, it allows to isolate $\pi_{\mathcal{C}}(\mathcal{C})$ singularities without computing an implicit equation of $\pi_{\mathcal{C}}(\mathcal{C})$ in the plane for which the available approaches are usually restricted to the algebraic case and might increase the complexity of the method used.

Although, for technical reasons, we restricted \mathcal{C} to live in a closed box in \mathbb{R}^n , we believe that our theoretical results in Chapters 3 & 4 still hold for any compact subset of \mathbb{R}^n . We also expect that the semi-algorithms in Chapter 5 can be developed to handle curves over compact sets as long as the membership problem¹⁷ over these sets is solvable. This generalization is, of course, helpful to tackle some visualization or robotics problems (that are similar to the ones that motivated us) where the region of interest is as general as compact sets.

It might also be interesting to investigate the possibility of generalizing the considered type of functions to the class C^k , with k large enough, where C^k is the class of functions that are differentiable k times with continuous derivatives. This generalization is expected to produce the same types of singularities in the projection. In fact, our motivation behind this suggestion is purely mathematical.

Another interesting open problem is to investigate the possibility of computing a certified approximation of singular plane curves by first lifting the input curve to a

¹⁷The membership problem asks whether a given point is in a given subset in \mathbb{R}^n .

smooth curve in \mathbb{R}^n , for some n , and then applying our approach. Even though it seems unintuitive to first lift for then to project, this approach is similar in spirit to blow-up techniques [JP2000, §5.3], although our lifting problem is global instead of local. However, if possible, it might allow to use certified numerical methods for computing the topology of implicit non-algebraic plane curves.

A similar idea of our approach has been already used by Diatta et al. [DMP2019]. The authors encoded the singularity of a surface in \mathbb{R}^3 that is the projection of a generic surface in \mathbb{R}^4 . This motivates the question of generalizing the idea of the Ball system to higher-dimensional objects. In particular, to find a regular system that encodes the singularities of the projection of a generic surface in \mathbb{R}^n to \mathbb{R}^3 , with $n > 4$. If possible, this would help to address some problems raised when designing robots of degrees of freedom larger than two (see Section 1.1).

The genericity of the assumptions implies that a randomly chosen curve in \mathbb{R}^n satisfies our assumption. In this regards, it would be interesting to continue the investigation on whether the silhouette of a generic surface in \mathbb{R}^n satisfies our weak assumptions. We have already proved a part and we left the rest as a conjecture (Conjecture 3.2.18). Such silhouette curves are relevant because they naturally appear in parametric systems since they partition the parametric space with respect to the number of solutions of the system.

Recall that we added a parameter eps_{\min} in our semi-algorithm to enforce its termination. From a practical point of view, eps_{\min} also makes the algorithm practical with fixed-precision arithmetic since we can stop the refinements before we go beyond the limit of the fixed precision. In addition, it is useful if we want to ensure the termination even when not all assumptions are satisfied, certifying $\pi_{\mathcal{C}}(\mathcal{C})$ singularities only on the part above which \mathcal{C} verifies the assumptions. In particular, we can handle that way the projections of non-generic singular curves in \mathbb{R}^n .

The software we provided does not only offer singularity detection but also a visualization tool for $\pi_{\mathcal{C}}(\mathcal{C})$. Figure 8.12 illustrates an example of how the output of our software look likes. Although the provided visualization does not guarantee the topology, the certified isolation of the singularities of $\pi_{\mathcal{C}}(\mathcal{C})$ paves the way towards drawing $\pi_{\mathcal{C}}(\mathcal{C})$ with the correct topology. Indeed, note that, for a given precision, if the drawing shows two branches that *appear* to intersect but without any reported box that contains the intersection, it is ensured that the two branches do not actually intersect; however, in order to determine the local topology (see Figure 9.1), further refinements are needed in our approach. We expect that an analogous approach to [IMP2018] might be found.

Our software handles only the functions that are implemented in [the Minibex](#)

■ Boxes enclosing smooth parts of the projected curve

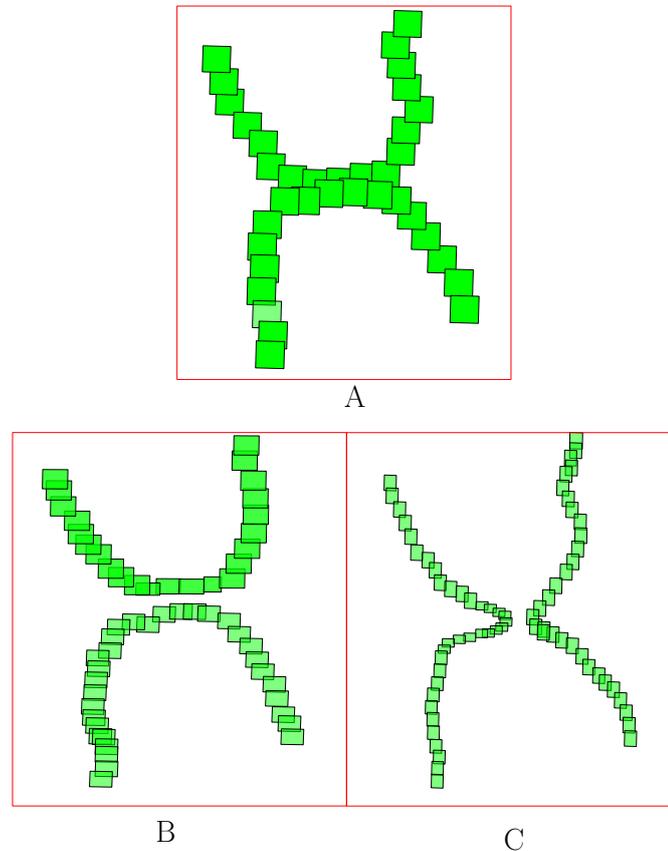


Figure 9.1: An example where our visualization tool does not provide a correct topology for some precision. A: Our software returns a set of boxes that encloses the projected curve without reporting any node (or unknown-status boxes). B & C: Two different scenarios of the plot in A after enough refinements.

language¹⁸ (e.g., \sin , \cos , $\exp \dots$) which is the language used in our black-box solver. In fact, we also developed another version of our software where the user can implement interval evaluations of his/her own functions and their derivatives. However, this version is too slow in practice. An interesting project would be to investigate the possibilities of improving this version. For this goal, Python-FLINT [Joh2012b] offers an interesting toolbox for interval arithmetic.

Another natural open question regarding the semi-algorithms is its complexity. It is worth noticing that our semi-algorithms use a subdivision approach with *diameter distance tests* as defined by Burr et al. [BGT2020]. As such, it should be possible to study our complexities using the method of *continuous amortization*. This

¹⁸<http://www.ibex-lib.org/doc/minibex.html>.

should yield explicit bounds for the case of polynomial input either in the worst case (as in [BGT2020]), or in a smoothed complexity setting (as in [CETC2019]).

Bibliography

- [ABRW1996] M.-E. Alonso, E. Becker, M. F. Roy, and T. Wörmann, *Zeros, multiplicities, and idempotents for zero-dimensional systems*, Algorithms in algebraic geometry and applications, 1996, pp. 1–15.
- [AGZV2012] Vladimir Igorevich Arnold, Sabir Medgidovich Gusein-Zade, and Alexander Nikolaevich Varchenko, *Singularities of differentiable maps. Volume 1*, Modern Birkhäuser Classics, Birkhäuser/Springer, New York, 2012. MR2896292
- [Bôc1964] Maxime Bôcher, *Introduction to higher algebra*, Dover Publications, Inc., New York, 1964.
- [BCR1998] Jacek Bochnak, Michel Coste, and Marie-Francoise Roy, *Real algebraic geometry*, Springer, 1998.
- [BES2011] E. Beberich, P. Emeliyanenko, and M. Sagraloff, *An elimination method for solving bivariate polynomial systems: eliminating the usual drawbacks.*, SIAM, Philadelphia (2011), 35–47. International Symposium on Symbolic and Algebraic Computation.
- [BGT2020] Michael Burr, Shuhong Gao, and Elias Tsigaridas, *The complexity of subdivision for diameter-distance tests*, Journal of Symbolic Computation **101** (2020), 1–27.
- [BL1986] P. Borrel and A. Liegeois, *A study of multiple manipulator inverse kinematic solutions with applications to trajectory planning and workspace determination*, Proceedings. 1986 IEEE International Conference on Robotics and Automation, 1986, pp. 1180–1185.
- [BL2013] Carlos Beltrán and Anton Leykin, *Robust certified numerical homotopy tracking*, Foundations of Computational Mathematics **13** (2013), no. 2, 253–295.
- [BLM⁺2016] Yacine Bouzidi, Sylvain Lazard, Guillaume Moroz, Marc Pouget, Fabrice Rouillier, and Michael Sagraloff, *Solving bivariate systems using rational univariate representations*, Journal of Complexity **37** (2016), 34–75.
- [Bou2014] Yacine Bouzidi, *Solving bivariate algebraic systems and topology of plane curves*, Theses, 2014.
- [BSS2001] Alin Bostan, Bruno Salvy, and Éric Schost, *Fast Algorithms for Zero-Dimensional Polynomial Systems Using Duality*, Technical Report RR-4291, INRIA, 2001.
- [Buc1965] B. Buchberger, *Ein Algorithmus zum Auffinden der Basiselemente des Restklassenringes nach einem nulldimensionalen Polynomideal*, 1965.
- [BV1988] Winfried Bruns and Udo Vetter, *Determinantal rings*, Lecture Notes in Mathematics, vol. 1327, Springer-Verlag, Berlin, 1988.
- [CCG⁺2012] Stéphane Caro, Damien Chablat, Alexandre Goldsztejn, Daisuke Ishii, and Christophe Jermann, *A Branch and Prune Algorithm for the Computation of Generalized Aspects of Parallel Robots*, 18th international conference on Principles and Practice of Constraint Programming, 2012, pp. 867–882.
- [CETC2019] Felipe Cucker, Alperen A. Ergür, and Josue Tonelli-Cueto, *Plantinga-veger algorithm takes average polynomial time*, Proceedings of the 2019 on international symposium on symbolic and algebraic computation, 2019, pp. 114–121.
- [CGL2009] Jin-San Cheng, X. Gao, and J. Li, *Root isolation for bivariate polynomial systems with local generic position method*, Issac '09, 2009.
- [Cha2010] Damien Chablat, *Joint space and workspace analysis of a two-DOF closed-chain manipulator*, ROMANSY 18 Robot Design, Dynamics and Control, 2010, pp. 81–90.
- [CLO2005] David A. Cox, John Little, and Donal O’Shea, *Using algebraic geometry*, Second, Graduate Texts in Mathematics, vol. 185, Springer, New York, 2005. MR2122859

BIBLIOGRAPHY

- [CLO2007] ———, *Ideals, varieties, and algorithms: An introduction to computational algebraic geometry and commutative algebra*, Undergraduate texts in mathematics. Third Edition., Springer-Verlag New York-Berlin-Paris, 2007.
- [CLP⁺2010] Jinsan Cheng, Sylvain Lazard, Luis Peñaranda, Marc Pouget, Fabrice Rouillier, and Elias Tsigaridas, *On the topology of real algebraic plane curves*, Mathematics in Computer Science **4** (2010Nov), no. 1, 113–137.
- [CW1998] Damien Chablat and Philippe Wenger, *Working Modes and Aspects in Fully-Parallel Manipulator*, International Conference on Robotics and Automation, May 1998, pp. 1964–1969.
- [Dem2000] Michel Demazure, *Bifurcations and catastrophes*, Universitext, Springer-Verlag, Berlin, 2000. Geometry of solutions to nonlinear problems, Translated from the 1989 French original by David Chillingworth.
- [DET2009] Dimitrios I. Diochnos, Ioannis Z. Emiris, and Elias P. Tsigaridas, *On the asymptotic and practical complexity of solving bivariate systems over the reals*, Journal of Symbolic Computation **44** (2009), no. 7, 818–835. International Symposium on Symbolic and Algebraic Computation.
- [DL2014] Nicolas Delanoue and Sébastien Lagrange, *A numerical approach to compute the topology of the apparent contour of a smooth mapping from R^2 to R^2* , Journal of Computational and Applied Mathematics **271** (2014), 267–284.
- [DLZ2011] Barry Dayton, Tien-Yien Li, and Zhonggang Zeng, *Multiple zeros of nonlinear systems*, Math. Comp. **80** (2011), no. 276, 2143–2168.
- [DMP2019] Sény Diatta, Guillaume Moroz, and Marc Pouget, *Reliable Computation of the Singularities of the Projection in R^3 of a Generic Surface of R^4* , MACIS 2019 - Mathematical Aspects of Computer and Information Sciences, November 2019.
- [EH2004] G. William Walster Eldon Hansen, *Global optimization using interval analysis*, 2nd ed., revised and expanded, Monographs and textbooks in pure and applied mathematics 264, Marcel Dekker, 2004.
- [Epp2013] J.F. Epperson, *An introduction to numerical methods and analysis*, Wiley, 2013.
- [FM2007] Dominique Faudot and Dominique Michelucci, *A new robust algorithm to trace curves*, Reliable Computing **13** (2007Aug), no. 4, 309–324.
- [GA1990] C. Gosselin and J. Angeles, *Singularity analysis of closed-loop kinematic chains*, IEEE Transactions on Robotics and Automation **6** (1990), no. 3, 281–290.
- [GG1973] Marty Golubitsky and Victor Guillemin, *Stable mappings and their singularities*, Springer-Verlag, New York-Heidelberg, 1973. Graduate Texts in Mathematics, Vol. 14.
- [GM1989] Patrizia Gianni and Teo Mora, *Algebraic solution of systems of polynomial equations using gröbner bases*, Applied algebra, algebraic algorithms and error correcting codes, proceedings of aaecc-5, volume 356 of Incs, 1989, pp. 247–257.
- [GS2001a] Jürgen Garloff and Andrew P. Smith, *Solution of systems of polynomial equations by using bernstein expansion*, Symbolic algebraic methods and verification methods, 2001, pp. 87–97.
- [GS2001b] Jürgen Garloff and A.P. Smith, *Investigation of a subdivision based algorithm for solving systems of polynomial equations*, Nonlinear Analysis: Theory, Methods & Applications **47** (2001), no. 1, 167–178. Proceedings of the Third World Congress of Nonlinear Analysts.
- [GVJ⁺2009] A. Gomes, I. Voiculescu, J. Jorge, B. Wyvill, and C. Galbraith, *Implicit curves and surfaces: Mathematics, data structures and algorithms*, Springer London, 2009.
- [GVK1996] Laureano González-Vega and M’ammed El Kahoui, *An improved upper complexity bound for the topology computation of a real algebraic plane curve*, Journal of Complexity **12** (1996), no. 4, 527–544.
- [Hea2007] J. D. Hunter et al., *Matplotlib: A 2d graphics environment*, Computing in Science & Engineering **9** (2007), no. 3, 90–95.
- [HMS2015] Jonathan D. Hauenstein, Bernard Mourrain, and Agnes Szanto, *Certifying isolated singular points and their multiplicity structure*, ISSAC’15, July 2015, pp. 213–220.
- [HMvdW⁺2020] Charles R. Harris, K. Jarrod Millman, St’efan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fern’andez del R’io, Mark Wiebe, Pearu Peterson, Pierre G’erard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant, *Array programming with NumPy*, Nature **585** (September 2020), no. 7825, 357–362.

- [IMP2016a] Rémi Imbach, Guillaume Moroz, and Marc Pouget, *A certified numerical algorithm for the topology of resultant and discriminant curves*, Journal of Symbolic Computation **80**, Part 2 (2016), 285–306.
- [IMP2016b] Rémi Imbach, Guillaume Moroz, and Marc Pouget, *Numeric and certified isolation of the singularities of the projection of a smooth space curve*, Mathematical aspects of computer and information sciences, 2016, pp. 78–92.
- [IMP2018] Rémi Imbach, Guillaume Moroz, and Marc Pouget, *Reliable location with respect to the projection of a smooth space curve*, Reliab. Comput. **26** (2018), 13–55. MR3804276
- [Joh2012a] Fredrik Johansson, *Arb*, 2012. <https://arblib.org/>.
- [Joh2012b] ———, *Python-flint*, 2012. <http://fredrikj.net/python-flint/>.
- [JP2000] T. D. Jong and G. Pfister, *Local analytic geometry - basic theory and applications*, Advanced lectures in mathematics, 2000.
- [Kal1993] M. Kalkbrener, *A generalized euclidean algorithm for computing triangular representations of algebraic varieties*, J. Symb. Comput. **15** (1993), 143–167.
- [Kea1997] Ralph Baker Kearfott, *Empirical evaluation of innovations in interval branch and bound algorithms for nonlinear systems*, SIAM Journal on Scientific Computing **18** (1997), no. 2, 574–594.
- [Kea2013] Baker Kearfott, *Rigorous global search: continuous problems*, Vol. 13, Springer Science Business Media, 2013.
- [Kra2021a] George Krait, *Demonstration of RRRRR robot*, Zenodo, 2021. <https://doi.org/10.5281/zenodo.4451292>.
- [Kra2021b] ———, *planar robot PRRP*, Zenodo, 2021. <https://doi.org/10.5281/zenodo.4451257>.
- [Kul1997] Wladyslaw Kulpa, *The poincaré-miranda theorem*, The American Mathematical Monthly **104** (1997), no. 6, 545–550, available at <https://doi.org/10.1080/00029890.1997.11990676>.
- [KX1994] R. Baker Kearfott and Zhaoyun Xing, *An interval step control for continuation methods*, SIAM Journal on Numerical Analysis **31** (1994), no. 3, 892–914.
- [Laz1992] D. Lazard, *Solving zero-dimensional algebraic systems*, Journal of Symbolic Computation **13** (1992), no. 2, 117–131.
- [LOD2001] H. Lopes, J. B. Oliveira, and L. H. De Figueiredo, *Robust adaptive approximation of implicit curves*, Proceedings xiv brazilian symposium on computer graphics and image processing, 2001, pp. 10–17.
- [LP2017] K. Lynch and F. Park, *Modern robotics: Mechanics, planning, and control*, 2017.
- [LVZ2006] Anton Leykin, Jan Verschelde, and Ailing Zhao, *Newton’s method with deflation for isolated singularities of polynomial systems*, Theoretical Computer Science **359** (2006), no. 1, 111–122.
- [LY2011] Long Lin and Chee Yap, *Adaptive isotopic approximation of nonsingular curves: the parameterizability and nonlocal isotopy approach*, Discrete & Computational Geometry **45** (2011Jun), no. 4, 760–795.
- [Mer2006] Merlet, *Parallel robots*, Springer Netherlands, 2006.
- [MGGJ2013] B. Martin, A. Goldsztejn, L. Granvilliers, and C. Jermann, *Certified parallelotope continuation for one-manifolds*, SIAM J. Numer. Anal. **51** (2013), 3373–3401.
- [MKC2009] Ramon E Moore, R Baker Kearfott, and Michael J Cloud, *Introduction to interval analysis*, Siam, 2009.
- [MM2011] Angelos Mantzaflaris and Bernard Mourrain, *Deflation and Certified Isolation of Singular Zeros of Polynomial Systems*, International Symposium on Symbolic and Algebraic Computation (ISSAC), June 2011, pp. 249–256.
- [MP2009] B. Mourrain and J.P. Pavone, *Subdivision methods for solving polynomial equations*, Journal of Symbolic Computation **44** (2009), no. 3, 292–306. Polynomial System Solving in honor of Daniel Lazard.
- [MPS⁺2006] Bernard Mourrain, Sylvain Pion, Susanne Schmitt, Jean-Pierre Técourt, Elias Tsigaridas, and Nicola Wolpert, *Algebraic issues in computational geometry* (Jean-Daniel Boissonnat and Monique Teillaud, eds.), Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.
- [MSP⁺2017] Aaron Meurer, Christopher P. Smith, Mateusz Paprocki, Ondřej Čertík, Sergey B. Kirpichev, Matthew Rocklin, AMiT Kumar, Sergiu Ivanov, Jason K. Moore, Sartaj Singh, Thilina Rathnayake, Sean Vig, Brian E. Granger, Richard P. Muller, Francesco Bonazzi, Harsh Gupta, Shivam Vats, Fredrik Johansson, Fabian Pedregosa, Matthew J. Curry, Andy R. Terrel, Štěpán Roučka, Ashutosh Saboo, Isuru Fernando, Sumith Kulal, Robert Cimrman, and Anthony Scopatz, *Sympy: symbolic computing in python*, PeerJ Computer Science **3** (January 2017), e103.

BIBLIOGRAPHY

- [Neu1991] Arnold Neumaier, *Interval methods for systems of equations*, Encyclopedia of Mathematics and its Applications, Cambridge University Press, 1991.
- [Nin2015] Jordan Ninin, *Global Optimization based on Contractor Programming: an Overview of the IBEX library*, MACIS, November 2015.
- [OWM1983] Takeo Ojika, Satoshi Watanabe, and Taketomo Mitsui, *Deflation algorithm for the multiple roots of a system of nonlinear equations*, Journal of Mathematical Analysis and Applications **96** (1983), no. 2, 463–479.
- [PV2004] Simon Plantinga and Gert Vegter, *Isotopic Approximation of Implicit Curves and Surfaces*, Symposium on geometry processing, 2004.
- [PV2006] _____, *Computing contour generators of evolving implicit surfaces*, ACM Trans. Graph. **25** (October 2006), no. 4, 1243–1280.
- [PVHK1997] David McAllester Pascal Van Hentenryck and Deepak Kapur, *Solving polynomial systems using a branch and prune approach*, SIAM Journal on Numerical Analysis **34** (1997).
- [Ral1983] L. B. Rall, *Mean value and taylor forms in interval analysis*, SIAM Journal on Mathematical Analysis **14** (1983), no. 2, 223–238, available at <https://doi.org/10.1137/0514019>.
- [Rou1998] Fabrice Rouillier, *Solving Zero-dimensional Polynomial Systems through the Rational Univariate Representation*, Technical Report RR-3426, INRIA, 1998.
- [Rum2010] Siegfried M. Rump, *Verification methods: Rigorous results using floating-point arithmetic*, Acta Numerica **19** (2010), 287–449.
- [RZ2009] Greg Reid and Lihong Zhi, *Solving polynomial systems via symbolic-numeric reduction to geometric involutive form*, Journal of Symbolic Computation **44** (2009), no. 3, 280–291. Polynomial System Solving in honor of Daniel Lazard.
- [Sny1992] John M. Snyder, *Interval analysis for computer graphics*, SIGGRAPH Comput. Graph. **26** (July 1992), no. 2, 121–130.
- [SP1993] Evan C. Sherbrooke and Nicholas M. Patrikalakis, *Computation of the solutions of nonlinear polynomial systems*, Computer Aided Geometric Design **10** (1993), no. 5, 379–405.
- [Sta1995] Volker Stahl, *Interval methods for bounding the range of polynomials and solving systems of nonlinear equations*, Ph.D. Thesis, 1995.
- [Sva2014] Jules Svartz, *Solving zero-dimensional structured polynomial systems*, Theses, 2014.
- [SW2005] R. Seidel and N. Wolpert, *On the exact computation of the topology of real algebraic curves*, Proceedings of the twenty-first annual symposium on Computational geometry (2005).
- [TBV2019] Simon Telen, M. Barel, and J. Verschelde, *A robust numerical path tracking algorithm for polynomial homotopy continuation*, ArXiv [abs/1909.04984](https://arxiv.org/abs/1909.04984) (2019).
- [VDH2015] Joris Van Der Hoeven, *Reliable homotopy continuation*, LIX, Ecole polytechnique, 2015.
- [vdHL2018] Joris van der Hoeven and Robin Larrieu, *Fast reduction of bivariate polynomials with respect to sufficiently regular gröbner bases*, Proceedings of the 2018 acm international symposium on symbolic and algebraic computation, 2018, pp. 199–206.
- [VGO⁺2020] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, António H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors, *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python*, Nature Methods **17** (2020), 261–272.
- [Vil2018] Gilles Villard, *On computing the resultant of generic bivariate polynomials*, Proceedings of the 2018 acm international symposium on symbolic and algebraic computation, 2018, pp. 391–398.
- [WCB2001] D. Wang, G.E. Collins, and B. Buchberger, *Elimination methods*, Texts & Monographs in Symbolic Computation, Springer, 2001.
- [Whi1943] Hassler Whitney, *Differentiable even functions*, Duke Mathematical Journal **10** (1943), 159–160.

- [Whi1955] ———, *On singularities of mappings of euclidean spaces. i. mappings of the plane into the plane*, *Annals of Mathematics* **62** (1955), no. 3, 374–410.
- [Whi1965] ———, *Local properties of analytic varieties*, *Differential and Combinatorial Topology (A Symposium in Honor of Marston Morse)*, 1965, pp. 205–244. MR0188486
- [XY2019] Juan Xu and Chee Yap, *Effective subdivision algorithm for isolating zeros of real systems of equations, with complexity analysis*, *Proceedings of the 2019 international symposium on symbolic and algebraic computation*, 2019, pp. 355–362.
- [ZSP1993] Jingfang Zhou, Evan C. Sherbrooke, and Nicholas M. Patrikalakis, *Computation of stationary points of distance functions*, *Engineering with Computers* **9** (1993Dec), no. 4, 231–246.