ABSTRACT

ROHAL, JAMES JOSEPH. Connectivity in Semi-Algebraic Sets. (Under the direction of Hoon Hong.)

A semi-algebraic set is a subset of real space defined by polynomial equations and inequalities and is a union of finitely many maximally connected components. In this thesis we consider the problem of deciding whether two given points in a semi-algebraic set are connected; that is, whether the two points lie in the same connected component. In particular, we consider the semi-algebraic set defined by $f \neq 0$ where f is a given polynomial. The motivation comes from the observation that many important or non-trivial problems in science and engineering can be often reduced to that of connectivity. Due to its importance, there has been intense research effort on the problem. We will describe a symbolic-numeric method for solving this problem based on gradient ascent. In the first part of this thesis we will describe the symbolic part of the method. In a forthcoming second paper, we will describe the numeric part of the method. The second part of this thesis focuses on proving the partial correctness and termination of the symbolic method assuming the correctness of the numeric part. In the third part of the thesis we give an upper bound on the length of a path connecting the two input points if they lie in a same connected component. In the last part of the thesis we give experimental timing results for the symbolic-numeric method.

Connectivity in Semi-Algebraic Sets

by
James Joseph Rohal

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Applied Mathematics

Raleigh, North Carolina

2014

APPROVED BY:

Jonathan Hauenstein	Mohab Safey El Din
Erich Kaltofen	Agnes Szanto

Hoon Hong
Chair of Advisory Committee

DEDICATION

To my family, who always believed in me.

BIOGRAPHY

James J. Rohal was born in Cleveland, OH and spent his childhood in Broadview Heights, OH, where he attended Brecksville-Broadview Heights High School (BBHHS). Following in the footsteps of one of his wonderful mentors from BBHHS, James enrolled at The College of Wooster with the intention of becoming a psychologist. Motivated by friendly competition with his friends, he found himself gravitating towards the field of mathematics and away from psychology. In the summer of his sophmore year he participated in the Applied Mathematics Research Experience where he helped design a web application for the Prentke-Romich company. This experience led James to pursue web design as a hobby which he continues to do today. During his junior year, he attended the Budapest Semesters in Mathematics (BSM) program where his love for mathematics bloomed. He went on to participate in an NSF funded Research Experience for Undergraduates at the University of Akron, which gave him his first taste of mathematics research. After completing his B.A. in Mathematics at The College of Wooster in Spring 2007, he decided to pursue a M.S. in Applied Mathematics from Miami University, which he completed in Summer 2009. During his years at Miami, he started to believe that his career should be in academia. In Fall 2007, the program at Miami gave him his own class to teach. With shaking hands, he handed out his first syllabus and learned just how much fun it is to teach. James went on to North Carolina State University to ultimately receive his Ph.D in Applied Mathmatics in Summer 2014. During his years at North Carolina State University, he reflected on his career path, ultimately realizing that he still wanted to mentor and help people. He realized that all of his previous academic mentors have given him the skills to mentor other people. Naturally, James decided to pursue an academic career path by becoming an assistant professor. After a grueling job hunt, he landed a job as assistant professor at West Liberty University and eagerly awaits what comes next.

ACKNOWLEDGEMENTS

If you were to tell me eleven years ago that my academic career will culminate with me becoming an assistant professor, I would have told you that you were crazy. Yet, I somehow survived the entire ordeal. However, that would not have been possible without the help of my friends, family, and many professors along the way. You have all helped me maintain my sanity throughout this process and for that I wish to thank you from the bottom of my heart.

Let me begin with my friends. So many of you have pushed me to keep going when I faltered and helped provide the necessary distractions to live a normal life outside of academics. To Sofia, Jon, Scott, Stephen, Adam, Josh, and Ryan, you have all been my retreat; the people I go to when I need to be picked up. Without you, this thesis would not have been completed. To everyone else that I have met through the math department and over the course of my stay in North Carolina, you have made this past five years the most memorable of my life.

It goes without saying, my family members are the most amazing people in the world. There are no bigger cheerleaders than my parents. No finite number of words in this Acknowledgments section can begin to thank them for everything they have done for me. Without a doubt, they have been the most influential people in my life and have pushed me to achieve so much.

Finally, I would like to thank every professor that has ever aided me along my journey. To my committee members, thank you for taking the time to facilitate my completion of the Ph.D. It has been an honor working under your guidance, and I hope that the future will lead to many collaborations. To my advisor, Hoon Hong, the entire journey could not have been made without you. Your training has helped shape me in to the researcher that I am today. The greatest lesson you ever taught me was to become a “superman” and to “fight against yourself” to become a better person. There is no doubt that one day I will become a “superman.” This thesis is just the first step in that goal.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Let us begin by looking at Figure 1.1.

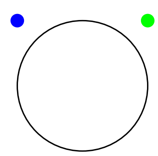


Figure 1.1 A circle and two points.

Ask yourself the following question.

> *Q1: Can I draw a continuous curve starting at the blue point and ending at the green point without crossing the black curve?*

We can immediately see that the answer is yes. Let us look at an example that is a little more complex. Ask yourself the same question after studying Figure 1.2. The answer in this instance is no. Answering the question might have been a bit more difficult because of the narrow gaps in the curves.

Let us generalize question Q1 to a higher dimension. Rather than having a curve we cannot cross, we will have a surface we cannot cross. The points in question will be points in three dimensions. In Figure 1.3, we show one surface, two points, and three different views of these three objects; the three views are the result of slight rotations counterclockwise about the vertical axis. Ask yourself the following question.

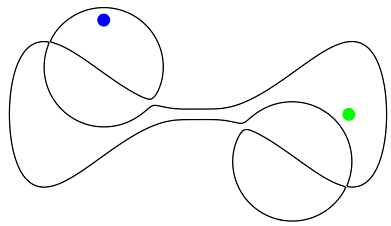


Figure 1.2 A curve and two points.

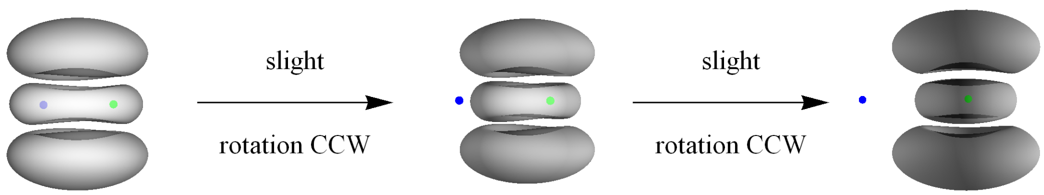


Figure 1.3 A surface and two points.

> *Q2: Can I draw a continuous curve starting at the blue point and ending at the green point without crossing the gray surface?*

The answer to the question is no. It may be difficult to tell from the pictures, but the green point lies inside one of the blobs while the blue point lies outside of all three blobs. Unfortunately, we cannot generalize this question to higher dimensions because we are unable to visualize higher dimensional surfaces very well.

Now rather than relying on pictures to describe our curves and surfaces, let us represent these objects using polynomials. We will represent a curve by a single polynomial equation in two variables and will represent a point by a tuple of numbers. We can rephrase question Q1 in the following way.

> *Q3: Can I draw a continuous curve starting at (p_1, p_2) and ending at (q_1, q_2) without crossing $f(x_1, x_2) = 0$?*

In Figure 1.1, the black circle represents the curve $f(x_1, x_2) = 0$ where $f = x_1^2 + x_2^2 - 1$. The blue and green points represent the tuples $(-1, 1)$ and $(1, 1)$, respectively. One can easily check by hand that

$$f(-1,1) > 0 \quad \text{and} \quad f(1,1) > 0,$$

and conclude that there exists a curve connecting the two given points without crossing $f = 0$. Hence the answer to question Q3 would be yes. Question Q3 becomes more difficult to answer when the polynomial f is more complicated. For example, the black curve in Figure 1.2 is the set of points where $f(x_1, x_2) = 0$ and

$$\begin{aligned} f = {} & 4096x_1^{16} - 16384x_1^{14} + 26624x_1^{12} - 22528x_1^{10} - 1024x_1^8x_2^4 + 1024x_1^8x_2^2 \\ & + 10496x_1^8 + 2048x_1^6x_2^4 - 2048x_1^6x_2^2 - 2560x_1^6 - 1280x_1^4x_2^4 + 1280x_1^4x_2^2 \\ & + 256x_1^4 + 256x_1^2x_2^4 - 256x_1^2x_2^2 - 4096x_2^{16} + 16384x_2^{14} - 26624x_2^{12} \\ & + 22528x_2^{10} - 10560x_2^8 + 2688x_2^6 - 352x_2^4 + 32x_2^2 - 1. \end{aligned} \tag{1.4}$$

Determining whether there exists such a curve connecting the two points $\left(-\frac{1}{2}, \frac{1}{2}\right)$ (blue point) and $\left(\frac{4}{5}, 0\right)$ (green point) without crossing $f = 0$ is quite difficult now. It turns out that the answer is no in this instance. It is trivial to generalize question Q3 to higher dimensions:

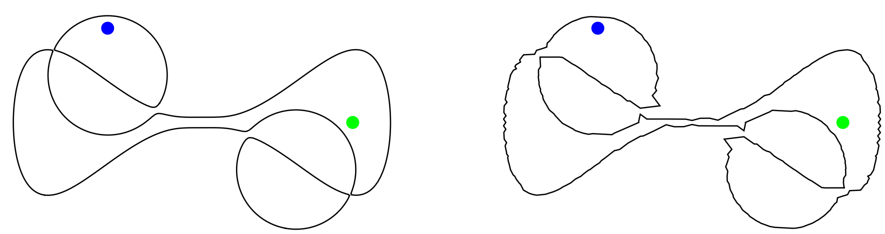
> *Q4: Can I draw a continuous curve starting at (p_1, p_2, p_3) and ending at (q_1, q_2, q_3) without crossing $f(x_1, x_2, x_3) = 0$?*

In Figure 1.3, we draw the surface $f(x_1, x_2, x_3) = 0$ where

$$\begin{aligned} f = {} & x_1^6 + 4x_1^4x_2^2 + 3x_1^4x_3^2 + 2x_1^4 + 5x_1^2x_2^4 + 8x_1^2x_2^2x_3^2 + 8x_1^2x_2^2 + 3x_1^2x_3^4 - 12x_1^2x_3^2 \\ & - 4x_1^2 + 2x_2^6 + 5x_2^4x_3^2 + 6x_2^4 + 4x_2^2x_3^4 - 24x_2^2x_3^2 + x_3^6 - 14x_3^4 + 28x_3^2 - 7 \end{aligned}$$

and draw the points $(0, 2, 0)$ (blue) and $\left(\frac{4}{5}, 0, 0\right)$ (green). Again, determining whether we can connect the given points using a curve that avoids $f = 0$ is a very difficult problem. As before, the answer is no.

In this thesis, we will focus on how to answer questions like Q3 and Q4 algorithmically. Humans seem to have a natural ability to answer questions Q1 and Q2. As a result, researchers have approached these types of question using methods from the field of computer vision. On the other hand, humans do not appear to be able to answer questions Q3 and Q4 as easily. We consider questions like Q3 and Q4 in this thesis because it is possible to answer these questions rigorously; that is, we can guarantee the correctness of our response to the questions. Representing a curve using a picture can be misleading and can lead to possibly incorrect answers. For instance, in Figure 1.5 we draw the curve $f = 0$ using the `ContourPlot` command in *Mathematica*, where f is from (1.4). In Figure 1.5a, many more sample points are used to draw the curve while far fewer sample points are used to draw the curve in Figure 1.5b. As mentioned earlier, we cannot connect the blue and green point in Figure 1.5a by a continuous curve that does not cross the black curve. However, it appears that in Figure 1.5b that we can connect the blue and green points by a continuous curve that does not cross the black curve. This discrepancy was caused by how we drew the curve $f = 0$.



(a) **(b)**

Figure 1.5 Plotting $f = 0$ using different numbers of sample points, where f is from (1.4).

Questions like Q3 and Q4 are called *connectivity queries*. The white region in which the two points lie is called a *semi-algebraic set*. We are interested in determining whether two given points in a semi-algebraic set can be connected by some continuous curve. Notice that the black curve in Figure 1.5a splits the white region into four distinct regions. These four regions are called *semi-algebraically connected components*. An alternative formulation of questions Q3 and Q4 is to determine whether the two given points lie in a same semi-algebraically connected component.

Many important or non-trivial problems in science and engineering can be reduced to the problem of deciding connectivity properties of semi-algebraic sets. The original motivation came from robot motion planning where one tries to decide collision-free motions for a robot in an environment filled with obstacles [Ito09; Lat91]. The free space in which the robot can move can be modeled as a semi-algebraic set. Then one wants to know whether a robot can move from an initial configuration (a starting point) to a final configuration (an ending point) within the free space in a continuous motion. If this is true, one must find such a continuous trajectory. Motion planning problems show up in other diverse contexts such as computational biology, virtual prototyping in manufacturing, architectural design, aerospace engineering, and computational geography [LaV06]. The field of motion planning was first introduced in scientific literature by Lozano-Perez, Wesley, and Reif [LPW79; Rei79].

In a broad sense, determining whether two points lie in a same connected component of a semi-algebraic set is part of a growing field called algorithmic real algebraic geometry [Bas03]. Real algebraic geometry is concerned with studying the properties of semi-algebraic sets. Computing properties such as the connected components of a semi-algebraic set is a fundamental problem and motivated the problem of computing the topology of semi-algebraic sets (see for example [SS83]).

In this thesis we present a robust symbolic-numeric method based on gradient ascent for deciding whether two given points in a semi-algebraic set are connected; that is, whether the two points lie in a same semi-algebraically connected component. In particular, we consider the semi-algebraic set defined by $f \neq 0$ where f is a given polynomial. In this chapter we will describe the symbolic part of the method and in a forthcoming paper, describe the numeric part of the method. The symbolic part of the method was first discussed in [Hon10]. The second part of the thesis focuses on proving the partial correctness and termination of the symbolic method assuming the correctness of the numeric subalgorithm. In the third part of the thesis we give an upper bound on the length of a path connecting the input points if they lie a same semi-algebraically connected component. In the last part of the thesis we give experimental timing results for our method.

In the first section we give a formal statement of the problem we will be studying in this thesis. In the second section we give an overview of the previous work done on this and related problems. In the third section we state the steps of the symbolic-numeric algorithm. Finally, in the fourth section we explicitly state the results in the thesis. They will be described in full detail in the chapters following.

1.1 Problem Statement

In the most general sense, the “connectivity problem” is to decide whether two given points in a given set can be connected via a continuous path within the set. In this thesis, we consider a crucial special case where the given set is a particular type of semi-algebraic set, in that it consists of the points where a given polynomial f is not equal to 0. To state the problem more precisely we first recall a few notions.

We say S is a *semi-algebraic set in* $\mathbb{R}^n$ if it is a finite union of sets of the form

$$\left\{ x \in \mathbb{R}^n \;\middle|\; P(x) = 0 \wedge \bigwedge_{Q \in \mathcal{Q}} Q(x) \neq 0 \right\}$$

where P is a polynomial in $\mathbb{R}[x_1, \ldots, x_n]$ and $\mathcal{Q}$ is a finite subset of $\mathbb{R}[x_1, \ldots, x_n]$. Let $\{f \star 0\}$ be a shorthand notation for $\{x \in \mathbb{R}^n \mid f(x) \star 0\}$ where $\star \in \{=, \neq, >, <, \geq, \leq\}$. A semi-algebraic set S is *semialgebrically connected* if S is not the disjoint union of two non-empty semi-algebraic sets that are both closed in S. A *semi-algebraically connected component* of a semi-algebraic set S is a maximal semi-algebraically connected subset of S. A semi-algebraic set has a finite number of semi-algebraically connected components. Throughout this thesis, when we use the shorthand notation of *connected component*, we mean a semi-algebraically connected component of a given semi-algebraic set.

Example 1.6. For $f \in \mathbb{R}[x_1, x_2]$ defined by

$$f = -2x_1^2 + x_1^4 - 2x_2^2 + 2x_1^2 x_2^2 + x_2^4$$

the set $\{f \neq 0\}$ is a semi-algebraic set that has two (semi-algebraically) connected components.

We now state our problem precisely.

Problem 1.7.

Input: $f \in \mathbb{Z}[x_1, \ldots, x_n]$, $n \geq 2$, $\deg f \geq 1$, squarefree, with finitely many singular points, $p, q \in \mathbb{Q}^n \cap \{f \neq 0\}$.

Output: `true`, if and only if the two points p and q lie in a same connected component of the set $\{f \neq 0\}$.

Example 1.8. We illustrate the problem using a toy example. Let

$$f = -2x_1^2 + x_1^4 - 2x_2^2 + 2x_1^2 x_2^2 + x_2^4. \tag{1.9}$$

Figure 1.10a shows the set defined by $f = 0$. The two points p, q in Figure 1.10b cannot be connected via a continuous path in $\{f \neq 0\}$ since they belong to different connected components. Hence the output should be `false`. The two points p, q in Figure 1.10c, however, can be connected via a continuous path in $\{f \neq 0\}$ since they belong to a same connected component. Hence the output should be `true`.

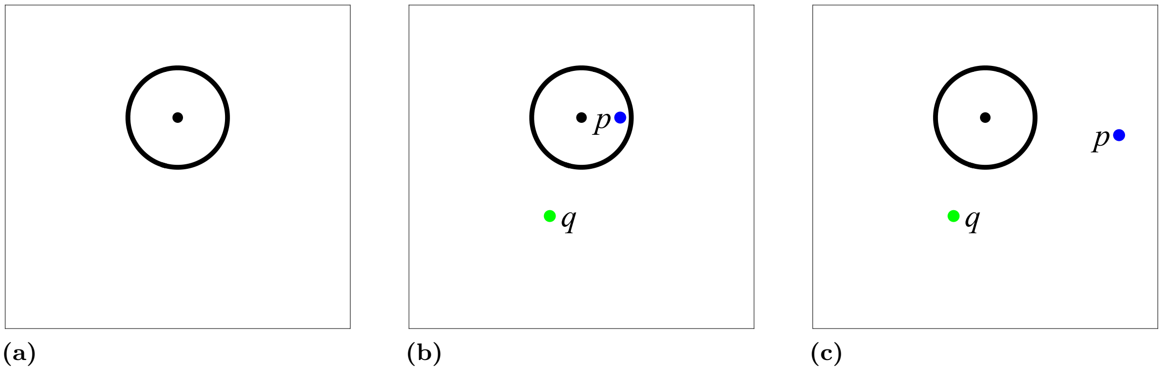


Figure 1.10 Sample inputs for Problem 1.7.

1.2 Previous Work

Initially, it was not even clear as to whether the problem of rigorously determining whether two points lie in a same connected component of a semi-algebraic set $S \subset \mathbb{R}^n$ was decidable. Evidence that the problem was decidable came in the form of the works by Tarski [Tar51] and Seidenberg [Sei54], who proved the decidability of the first order theory of real closed fields. Since then, there has been intense research effort on development of algorithms for performing quantifier elimination in the first order theory of real closed fields. A major breakthrough came in the form of the cylindrical algebraic decomposition algorithm developed by Collins [Col75], which used single variable resultants to perform quantifier elimination. Schwartz and Sharir [SS83] recognized the power of the cylindrical algebraic decomposition algorithm and used it to answer connectivity queries.

A fundamentally different strategy for solving the connectivity problem was presented by Canny [Can88; Can93]. Canny popularized the term *roadmap*, a one-dimensional semi-algebraic set that has nonempty intersection with each semi-algebraically connected component of S. Intuitively, a roadmap is a one dimensional skeleton of S. To determine whether two input points lie in a same connected component of S, one can link them to the roadmap of S and check the connectivity using the roadmap. Canny improved on the original approach by Schwartz and Sharir by using a new algebraic tool, the multivariate resultant. Canny's results spawned a movement over the past 20 years to steadily improve algorithms for specializations of the connectivity problem. The basic roadmap algorithm has been improved and extended by many researchers [Bas96; Bas00; GR93; GV92; Hei90].

In the previous papers, the construction of a roadmap of $S \subset \mathbb{R}^n$ depends on singly exponential many recursive calls to itself on several $(n-1)$ dimensional slices of S. Improving the algorithms in those papers was a notoriously difficult problem with no progress made until very recently. In the papers by Basu, Roy, Safey El Din, and Schost [Bas12; BR13; SEDS10; SEDS13], they used an improved recursive scheme to drop the dimension by more than one in each recursive call.

All of the algorithms discussed so far are based on real algebraic geometry computations, which are difficult to implement and may not be fast in practice. This inspired researchers to search for practical solutions to solving motion planning problems using heuristic or sampling based approaches. In these approaches, completeness of the method is sacrificed. One such method was based on potential fields [Kha86]. The idea was to create a scalar function, called the potential function, such that the gradient direction points away from the obstacle barrier. One can then follow the gradient field via gradient ascent (or descent) to traverse the free space. Typically the potential function is dependent on the input configurations; that is, the potential function is chosen in such a way that the goal configuration is a global minimum of the potential function.

A good middle ground between the symbolic approaches mentioned earlier and purely numeric approaches mentioned in the previous paragraph are hybrid symbolic-numeric methods. Recently, Iraji and Chitsaz [IC14] have proposed a method for computing a roadmap using a symbolic-numeric scheme. Their scheme bounds the roadmap using a chain of adjacent boxes, with each containing a slice of the roadmap. The method, called NuRA, preserves completeness of the roadmap algorithm and numerical experiments indicate it is practical.

In 2010, Hoon Hong [Hon10] published a note detailing a symbolic-numeric method for solving the problem (Problem 1.7) we are studying in this thesis. We present this method in the next section. The note did not provide a proof of partial correctness or termination. The method presented by Hong was unique in the sense that it answered connectivity queries using gradient trajectories, like in potential field methods, which typically are transcendental.

1.3 Algorithm

In this chapter, we describe a symbolic-numeric algorithm called **Connectivity** which first appeared in [Hon10]. We will describe the steps of **Connectivity** using a toy example shown earlier. We only give the input/output specification of a certified numeric subalgorithm called **Destination**. The steps will be described in a forthcoming paper. This section is divided into two subsections. The first subsection describes the steps of the algorithm **Connectivity**. The second subsection describes only the input/output specification of **Destination**.

1.3.1 Description of Algorithm Connectivity

We will illustrate the steps of **Connectivity** using the toy problem given in Example 1.8. We provide several pictures in the hope of aiding intuitive understanding of what each step does. Of course, the algorithms do not draw the pictures. We state the steps of **Connectivity** in Figure 1.11. We use the following notation. For a family $\mathcal{F} = \{f_1, \ldots, f_n\}$ of polynomials in $\mathbb{Z}[x_1, \ldots, x_n]$, we let $V(\mathcal{F})$ denote the zero-locus in $\mathbb{R}^n$ of the polynomials in $\mathcal{F}$. For a C^2 function g we let $\operatorname{Hess} g$ denote the Hessian matrix of g. For a non-zero vector v, we let $\widehat{v} = \frac{v}{\|v\|}$ where $\|\cdot\|$ is the Euclidean norm.

Example 1.12.

Input. $f = -2x_1^2 + x_1^4 - 2x_2^2 + 2x_1^2x_2^2 + x_2^4$, $p = (19/5, -1/2)\,, q = (-9/10, -14/5)$ are the blue and green points in Figure 1.10c, respectively.

- Here, $n = 2$, $\deg f = 4$, and f is a squarefree polynomial with exactly one singular point at $(0, 0)$.

1. Initially, we have

$$\begin{aligned} \gamma &= 5, \\ c &= (0, 0). \end{aligned}$$

2. In the first iteration of the loop we have

$$\begin{aligned} U &= x_1^2 + x_2^2 + 1, \\ \mathcal{F} &= \big\{-2x_1^5 - 4x_2^2x_1^3 + 20x_1^3 - 2x_2^4x_1 + 20x_2^2x_1 - 8x_1, \\ &\qquad -2x_2^5 - 4x_1^2x_2^3 + 20x_2^3 - 2x_1^4x_2 + 20x_1^2x_2 - 8x_2\big\}\,, \\ g &= \frac{\left(-2x_1^2 + x_1^4 - 2x_2^2 + 2x_1^2x_2^2 + x_2^4\right)^2}{\left(x_1^2 + x_2^2 + 1\right)^5}. \end{aligned}$$

The current $V(\mathcal{F})$ is one-dimensional. In Figure 1.13a we illustrate the contours for the current g in gray and $V(\mathcal{F})$ in red. We perturb c on the integer grid to be $c = (0, 1)$. In the

ALGORITHM: $t \leftarrow$ **Connectivity**(f, p, q)

Input : $f \in \mathbb{Z}[x_1, \dots, x_n]$, $n \geq 2$, $\deg f \geq 1$, squarefree, with finitely many singular points, $p, q \in \mathbb{Q}^n \cap \{f \neq 0\}$.

Output : t, `true` if and only if the two points p and q lie in a same semi-algebraically connected component of $\{f \neq 0\}$.

1 $\gamma \leftarrow \deg(f) + 1$
  $c \leftarrow (0, \dots, 0)$

2 **loop**
  - $U \leftarrow (x_1 - c_1)^2 + \dots + (x_n - c_n)^2 + 1$
  - $\mathcal{F} \leftarrow \left\{2 \cdot (\partial_{x_i} f) \cdot U - \gamma \cdot f \cdot (\partial_{x_i} U)\right\}_{i=1}^{n}$
  - $g \leftarrow \dfrac{f^2}{U^\gamma}$
  - **if** $\begin{pmatrix} V(\mathcal{F}) \text{ is zero-dimensional and} \\ \forall r \in V(\mathcal{F}), g(r) \neq 0 \implies \det(\operatorname{Hess} g)(r) \neq 0 \end{pmatrix}$ **then** exit **loop**
  - **else** $c \leftarrow$ perturb current c on the integer grid

3 $R \leftarrow V(\mathcal{F}) \setminus V(f)$

4 $A \leftarrow k \times k$ matrix with all entries set to 0, where k is the number of points in R

5 **foreach** $r \in R$ **do**
  - $V_r \leftarrow$ set of real algebraic orthonormal eigenvectors of $(\operatorname{Hess} g)(r)$ having positive eigenvalues
  - **foreach** $v \in V_r$ **do**
    - $j_+ \leftarrow$ **Destination**$(g, R, r_i, +v)$
    - $j_- \leftarrow$ **Destination**$(g, R, r_i, -v)$
    - $A_{ij_+} \leftarrow 1$
    - $A_{ij_-} \leftarrow 1$

6 $M \leftarrow$ the reflexive, symmetric and transitive closure of the relation represented by the matrix A

7 **if** $\nabla g(p) \neq 0$ **then** $i \leftarrow$ **Destination** $\left(g, R, p, \widehat{\nabla g(p)}\right)$
  else $i \leftarrow$ index of p in R

8 **if** $\nabla g(q) \neq 0$ **then** $j \leftarrow$ **Destination** $\left(g, R, q, \widehat{\nabla g(q)}\right)$
  else $j \leftarrow$ index of q in R

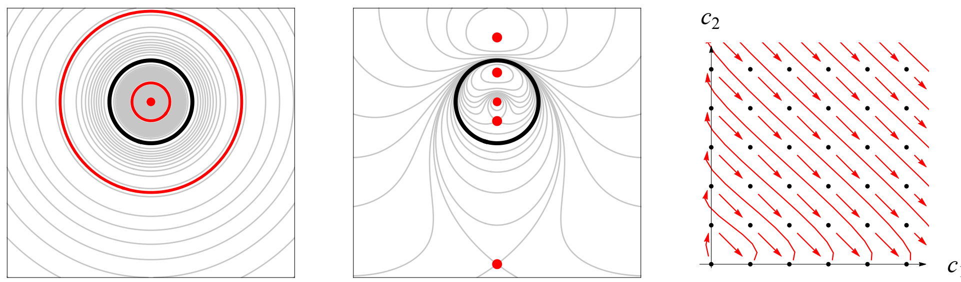
9 **return** $t \leftarrow$ `true` if and only if $M_{ij} = 1$

ALGORITHM: $i \leftarrow$ **Destination**(g, R, p, v)

Input : g, C^2 function
R, list of real algebraic points,
p, real algebraic point,
v, real algebraic unit vector,
such that there exists a unique $r \in R$ reachable from p using g and v.

Output : i, the index of the unique point r.

Figure 1.11 Connectivity and **Destination** algorithms.



(a) (b) (c)

Figure 1.13 Illustration of Step 2 of **Connectivity**.

second iteration of the loop we update U, $\mathcal{F}$ and g to be

$$\begin{aligned}
U &= x_1^2 + (x_2-1)^2 + 1,\\
\mathcal{F} &= \big\{-2x_1^5 - 4x_2^2x_1^3 - 16x_2x_1^3 + 28x_1^3 - 2x_2^4x_1\\
&\qquad - 16x_2^3x_1 + 28x_2^2x_1 + 16x_2x_1 - 16x_1,\\
&\qquad - 2x_2^5 - 6x_2^4 - 4x_1^2x_2^3 + 28x_2^3 + 4x_1^2x_2^2 - 4x_2^2\\
&\qquad -2x_1^4x_2 + 28x_1^2x_2 - 16x_2 + 10x_1^4 - 20x_1^2\big\},\\
g &= \frac{\left(-2x_1^2 + x_1^4 - 2x_2^2 + 2x_1^2x_2^2 + x_2^4\right)^2}{\left(x_1^2 + (x_2-1)^2 + 1\right)^5}.
\end{aligned}$$

The new $V(\mathcal{F})$ is zero-dimensional. We illustrate the perturbed $V(\mathcal{F})$ as the five red points in Figure 1.13b along with the contours for the new g. For all five $r \in V(\mathcal{F})$, four satisfy $g(r) \neq 0$, and $\det(\operatorname{Hess} g)(r) \neq 0$ at each of those four. Hence we exit the loop.

- One method for perturbing is using graded lexicographic order, which we visualize in Figure 1.13c. If there is an arrow having tip at α and tail at β then $x^\alpha > x^\beta$ in the graded lexicographic order. We can follow the arrows to systematically change (c_1, c_2) starting at $(0,0)$. This generalizes, of course, to any number of variables.
- One can use standard symbolic computation methods to check whether $V(\mathcal{F})$ is zero-dimensional and to compute the real algebraic points in it. Furthermore, the elements of $\operatorname{Hess} g$ are rational functions with integer coefficients, so the determinant can be computed as well.

3. We illustrate R as the four red points in Figure 1.14. Compare this to the five red points in

Figure 1.13b.

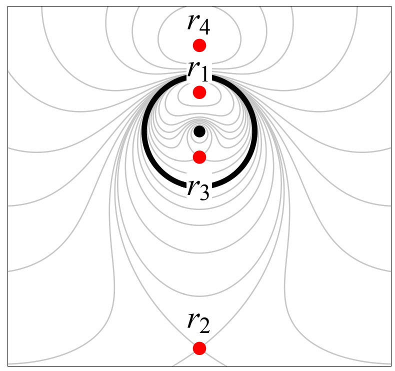


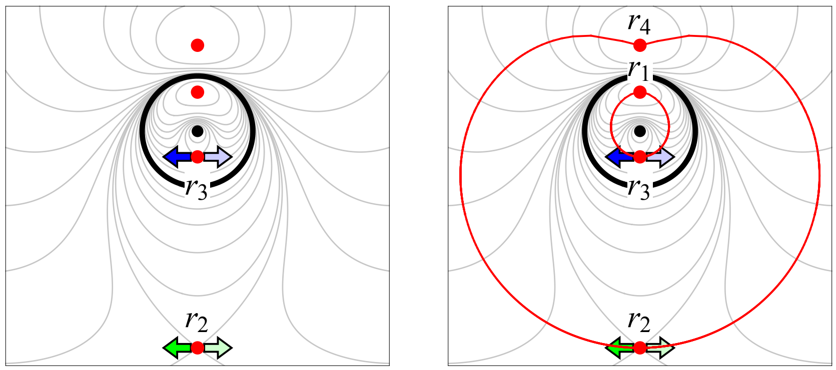
Figure 1.14 Illustration of the points in R.

- Note that each connected component of $\{f \neq 0\}$ contains at least one point from the set R.
- One may observe from the contour plot of g, that the points R are critical points of g where g is non-zero.
- Again, we can use standard symbolic computation methods to identify which of the points in $V(\mathcal{F})$ satisfy $f = 0$, and then remove them.

4. We have $A = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$ since $k = 4$.

5. Suppose $r = r_1$ or $r = r_4$. The matrix $(\operatorname{Hess} g)(r)$ has no positive eigenvalues. Hence $V_r = \emptyset$ and the body of the second **foreach** loop does not execute.

Suppose instead that $r = r_2$ or $r = r_3$, then the matrix $(\operatorname{Hess} g)(r)$ has one positive eigenvalue. For this eigenvalue, there are two corresponding real algebraic unit eigenvectors. If $r = r_2$, the two eigenvectors are $[-1\ \ 0]^T$ and $[1\ \ 0]^T$. We draw these two vectors as a dark green ( ) and light green ( ) outward pointing arrow from r_2 in Figure 1.15a. If $r = r_3$, the two eigenvectors are $[-1\ \ 0]^T$ and $[1\ \ 0]^T$. We draw these two vectors as a dark blue ( ) and light blue ( ) outward pointing arrow from r_3 in Figure 1.15a. Rather than write out an explicit expression of each of these eigenvectors in the subsequent paragraphs, we will use an arrow.



(a) **(b)**

Figure 1.15 Illustration of step 5 of **Connectivity**.

We let

$$V_{r_2} = \left\{ \right\} \text{ and } V_{r_3} = \left\{ \right\}.$$

Figure 1.15b shows four steepest ascent paths as red curves. Two of the red curves originate from r_2. We see that steepest ascent from r_2 in the initial direction approaches r_4. Similarly, we see that steepest ascent from r_2 in the initial direction approaches r_4. Hence when $r = r_2$, the inner **foreach** loop executes once because there is only one vector in V_{r_2} and

$$\begin{aligned} j_+ &\leftarrow \textbf{Destination}\left(g, R, r_2, \right) = 4, \\ j_- &\leftarrow \textbf{Destination}\left(g, R, r_2, \right) = 4, \\ A_{24} &\leftarrow 1, \\ A_{24} &\leftarrow 1. \end{aligned}$$

Two of the other steepest ascent paths originate from r_3. We see that steepest ascent from r_3 in the initial direction approaches r_1. Similarly, we see that steepest ascent from r_3 in the initial direction approaches r_1. Hence when $r = r_3$, the inner **foreach** loop executes once because there is only one vector in V_{r_3} and

$$\begin{aligned} j_+ &\leftarrow \textbf{Destination}\left(g, R, r_3, \Leftarrow\right) = 1, \\ j_- &\leftarrow \textbf{Destination}\left(g, R, r_3, \Rightarrow\right) = 1, \\ A_{31} &\leftarrow 1, \\ A_{31} &\leftarrow 1. \end{aligned}$$

The matrix A has the form

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

- For each $r \in R$, the Hessian $(\operatorname{Hess} g)(r)$ is a real symmetric matrix. It is a well known fact that the associated eigenvalues are all real and the eigenvectors corresponding to different eigenvalues are orthogonal. However, there is no restriction that the eigenvalues be simple, so it is possible that the geometric multiplicity of a positive eigenvalue is greater than one. In this case, finding two linearly independent eigenvectors for a given positive eigenvalue will suffice, as one can use the Gram-Schmidt process to find an orthonormal basis.
- Using standard symbolic computation techniques, we can find the eigenvalues and eigenvectors exactly because each point in R is an algebraic number and the elements of $\operatorname{Hess} g$ are rational functions with integer coefficients and the denominator is non-vanishing.
- Note that every steepest ascent path approaches a point in the set R. In fact, g was constructed to ensure that the path never spirals in a bounded region or goes forever into the infinity.
- It is crucial to observe that every two points in R can be connected if and only if they are connected via the above computed paths.

6. We have $M = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$.

- Note that we can use the matrix M to check whether two points $r_i, r_j \in R$ lie in a same connected component of $\{f \neq 0\}$ by checking the (i, j) entry of M.

- We call M a *connectivity matrix*.

7. For the input point p shown in Figure 1.16, $\nabla g(p) \neq 0$. We draw the vector $\widehat{\nabla g(p)}$ as the

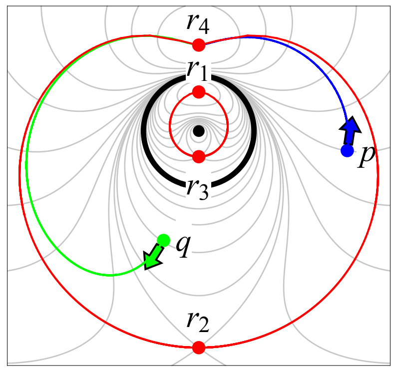


Figure 1.16 Illustration of steps 7 and 8 of **Connectivity**.

blue arrow $\left(\quad\right)$. We see that steepest ascent from p in the initial direction approaches r_4. Hence

$$i \leftarrow \textbf{Destination}\left(g, R, p, \quad\right) = 4.$$

8. For the input point q shown in Figure 1.16, $\nabla g(q) \neq 0$. We draw the vector $\widehat{\nabla g(q)}$ as the green arrow $\left(\quad\right)$. We see that steepest ascent from q in the initial direction approaches r_4. Hence

$$j \leftarrow \textbf{Destination}\left(g, R, q, \quad\right) = 4.$$

9. We note that $M_{44} = 1$ and thus the two points p, q can be connected. We set $t = \texttt{true}$.

Output. $t = \texttt{true}$.

As an overview, the algorithm **Connectivity** consists of three main stages.

1. Using f, compute "interesting" points on each connected component of $\{f \neq 0\}$. Create a function g with desirable properties, one being that $g = 0$ if and only if $f = 0$. Use g and the "interesting" points to form some vectors.

2. Connect the "interesting" points on each connected component of $\{g \neq 0\}$ using the vectors and trajectories of ∇g to create an adjacency matrix N by using **Destination**.

3. Determine the connectivity of p and q using N and trajectories of ∇g by making use of **Destination** once again.

The first and second stage are much more time-consuming than the third one. Fortunately, one needs to carry out the first and second stage only once for a given f, since it depends only on f.

1.3.2 Specification of Subalgorithm Destination

In this subsection, we will describe the input/output specification of a certified numeric subalgorithm called **Destination**, whose steps will be described a forthcoming paper. We begin by introducing some definitions.

Definition 1.17. Let $g\colon \mathbb{R}^n \to \mathbb{R}$ be a C^2 function. Let p be a point in $\mathbb{R}^n$ and v be a unit vector in $\mathbb{R}^n$. We say ϕ *is a trajectory of* ∇g if $\phi\colon I \to \mathbb{R}^n$ is a C^2 function where I is a finite union of open intervals of $\mathbb{R}$ such that

$$\phi'(t) = \nabla g\big(\phi(t)\big)$$

and $g \circ \phi\colon I \to \mathbb{R}$ is injective. The pieces of the image $\phi(I)$ are called *steepest ascent paths*. We say ϕ is a *trajectory of* ∇g *through* p *using* v if $\phi\colon (0,\infty) \to \mathbb{R}^n$ is a C^2 function and

$$\forall t > 0 \left(\phi'(t) = \nabla g\big(\phi(t)\big) \text{ and } \phi'(t) \neq 0\right) \tag{1.18}$$

and

$$\lim_{t\to 0^+} \phi(t) = p$$

and

$$\lim_{t\to 0^+} \frac{\phi'(t)}{\|\phi'(t)\|} = v$$

and $g \circ \phi\colon (0,\infty) \to \mathbb{R}$ is injective. We call the image $\phi\big((0,\infty)\big)$ a *steepest ascent path through* p *using* v and denote this as $\mathrm{SA}(g,p,v)$. We call $\mathrm{dest}(\phi)$ a *destination of* ϕ if the following limit exists:

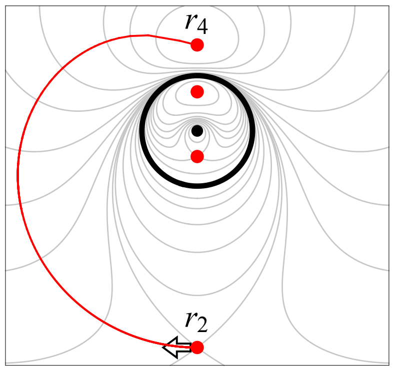
$$\mathrm{dest}(\phi) = \lim_{t\to\infty} \phi(t).$$

We say a point $q \in \mathbb{R}^n$ *is reachable from* p *using* g *and* v if there exists ϕ, a trajectory of ∇g through p using v, such that $\mathrm{dest}(\phi) = q$.

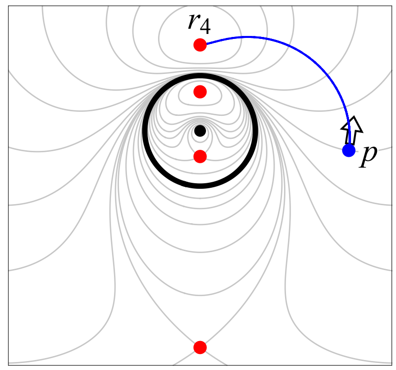
Example 1.19. Let

$$g = \frac{\left(-2x_1^2 + x_1^4 - 2x_2^2 + 2x_1^2x_2^2 + x_2^4\right)^2}{\left(x_1^2 + (x_2-1)^2 + 1\right)^5} \tag{1.20}$$

and $v = [-1\ \ 0]^T$. In Figure 1.21a we illustrate $\mathrm{SA}(g, r_2, v)$ as the red curve, where v is the arrow. We see the point r_4 is reachable from r_2 using g and v. Let $v = \widehat{\nabla g(p)}$. In Figure 1.21b we illustrate SA (g, p, v) as the blue curve, where v is the arrow. We see the point r_4 is reachable from p using g and v.



(a)



(b)

Figure 1.21 Illustration of various steepest ascent paths.

We state the specification for the algorithm **Destination** in Figure 1.11 and give a sample input and output in the following example.

Example 1.22. Let g be as in (1.20). Let and R, v be the set of points in red and the vector shown as the arrow in Figure 1.21a, respectively. Let $p = r_2$. The point r_4 is the unique point that is reachable from r_2 using g and v. Hence the output of **Destination**(g, R, p, v) would be 4.

1.4 Overview of Results

In this section we give an overview of the results in this thesis. We first give an outline and then give more precise results in the following subsections. The proofs of the results will be presented in the corresponding chapters.

The first two results in the thesis are proving the partial correctness and termination of **Connectivity** assuming the correctness of the subalgorithm **Destination**. In Chapter 2 we prove partial correctness and in Chapter 3 we prove termination. For a given polynomial f, the third result in the thesis is a bound on the length of a path connecting two points in a connected component of $\{f \neq 0\}$. Besides being an interesting question on its own, it is a possible first step toward completing a complexity analysis of **Connectivity**. We present this bound in Chapter 4. We conclude the thesis with some computational results by executing **Connectivity** for various size inputs. These results will be presented in Chapter 5.

1.4.1 Partial Correctness

We will prove the partial correctness of **Connectivity** in Theorem 2.24. The proof essentially amounts to showing that any two "interesting" points in the same connected component of $\{g \neq 0\}$ are connected by a particular set of steepest ascent paths. In order to make the claim precise, we will need to recall and introduce some notations and notions.

Definition 1.23. Let $g\colon \mathbb{R}^n \to \mathbb{R}$ be a C^2 function with $n \geq 2$. A critical point p of g is called a *routing point* of g if $g(p) \neq 0$. Let R be the set of routing points of g. We call g a *routing function* if the following conditions are satisfied:

- For all x, $g(x) \geq 0$.
- For all $\varepsilon > 0$, there exists $\delta > 0$, such that for all x, $\|x\| \geq \delta$ implies $g(x) \leq \varepsilon$.
- R is finite.
- For all $x \in R$, x is nondegenerate.
- The norms of the first and second derivatives of g are bounded.

Intuitively, the second condition in the routing function definition says that g vanishes at infinity; that is, as $\|x\| \to \infty$, $g(x) \to 0$.

Example 1.24. Let

$$g = \frac{\left(-2x_1^2 + x_1^4 - 2x_2^2 + 2x_1^2x_2^2 + x_2^4\right)^2}{\left(x_1^2 + (x_2-1)^2 + 1\right)^5}. \tag{1.25}$$

In Figure 1.26 we show the contours of g along with the routing points of g as red dots. The black curve and black dot is the set of points where $g = 0$. One may easily check that g satisfies the conditions to be called a routing function.

For the remaining examples in this section, we let g be denoted by (1.25) and let $R = \{r_1, r_2, r_3, r_4\}$ denote the set of routing points of g.

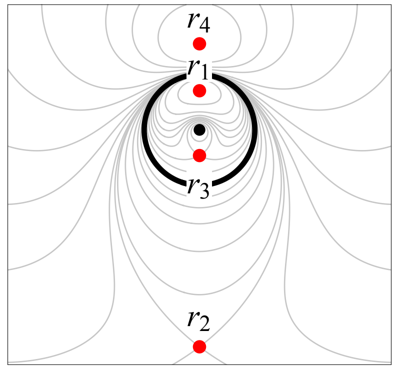


Figure 1.26 Illustration of the contours of a routing function g along with its routing points.

Definition 1.27. Let A be a real symmetric matrix and let v be a unit eigenvector of A with corresponding eigenvalue $\lambda \neq 0$. We say v is an *outgoing eigenvector* if $\lambda > 0$.

Example 1.28. In Figure 1.29, the outgoing eigenvectors of $(\operatorname{Hess} g)(r_2)$ are shown as arrows pointing outward from the point r_2. To be more precise,

$$(\operatorname{Hess} g)(r_2) \approx \begin{bmatrix} 0.000198674 & 0 \\ 0 & -0.000342484 \end{bmatrix}$$

and the vectors $[1 \;\; 0]^T$ and $[-1 \;\; 0]^T$ are outgoing eigenvectors for $(\operatorname{Hess} g)(r_2)$.

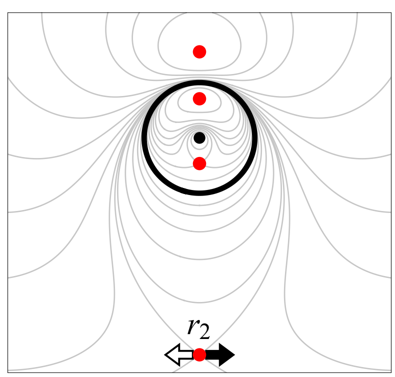


Figure 1.29 Illustration of the outgoing eigenvectors of $(\operatorname{Hess} g)(r_2)$.

Definition 1.30. Let $g\colon \mathbb{R}^n \to \mathbb{R}$ be a C^2 function with $n \geq 2$. Let $p, q \in \mathbb{R}^n$ with $p \neq q$, $g(p) > 0$, and $g(q) > 0$. We say p and q are *connected by steepest ascent paths using outgoing eigenvectors of* g if there exist functions $\phi_1, \ldots, \phi_{k+1}$ and routing points $r_1, \ldots, r_k$ such that

- if $\nabla g(p) = 0$, then $\phi_1 = p$ and $r_1 = p$, otherwise, ϕ_1 is a trajectory of ∇g through p using $\widehat{\nabla g(p)}$ and $\operatorname{dest}(\phi_1) = r_1$,
- if $\nabla g(q) = 0$, then $\phi_{k+1} = q$ and $r_k = q$, otherwise, ϕ_{k+1} is a trajectory of ∇g through q using $\widehat{\nabla g(q)}$ and $\operatorname{dest}(\phi_{k+1}) = r_k$,
- for all $2 \leq i \leq k$, there exists an outgoing eigenvector v_{i-1} of $(\operatorname{Hess} g)(r_{i-1})$ such that ϕ_i is a trajectory of ∇g through r_{i-1} using v_{i-1} and $\operatorname{dest}(\phi_i) = r_i$, or, there exists an outgoing eigenvector v_i of $(\operatorname{Hess} g)(r_i)$ such that ϕ_i is a trajectory of ∇g through r_i using v_i and $\operatorname{dest}(\phi_i) = r_{i-1}$.

Collectively, we call $r_1, \ldots, r_k$ and $\phi_1, \ldots, \phi_{k+1}$ a *connectivity path for* p *and* q.

Example 1.31. In Figure 1.32 we illustrate a connectivity for path p, q represented by r_4, r_2, ϕ_1, ϕ_2, ϕ_3. We describe what ϕ_1, ϕ_2, and ϕ_3 are now.

- Since $\nabla g(p) \neq 0$, ϕ_1 is a trajectory of ∇g through p using $\widehat{\nabla g(p)}$ where $\operatorname{dest}(\phi_1) = r_4$. The blue curve is $\operatorname{SA}\left(g, p, \widehat{\nabla g(p)}\right)$ and the light blue arrow is $\widehat{\nabla g(p)}$.
- Since $\nabla g(q) \neq 0$, ϕ_3 is a trajectory of ∇g through q using $\widehat{\nabla g(q)}$ where $\operatorname{dest}(\phi_3) = r_2$. The green curve is $\operatorname{SA}\left(g, q, \widehat{\nabla g(q)}\right)$ and the light green arrow is $\widehat{\nabla g(q)}$.
- The function ϕ_2 is a trajectory of ∇g through r_2 using v (red arrow) which is an outgoing eigenvector of $(\operatorname{Hess} g)(r_2)$. We see $\operatorname{dest}(\phi_2) = r_4$. The red curve is $\operatorname{SA}(g, r_2, v)$.

The partial correctness of the algorithm **Connectivity** relies heavily on the following theorem and is one of the main results in this thesis.

Theorem 1.33. *If* g *is a routing function then any two points in a same connected component of* $\{g \neq 0\}$ *are connected by steepest ascent paths using outgoing eigenvectors of* g.

Intuitively, this theorem says that any two points in a connected component of $\{g \neq 0\}$ can be connected using a particular set of steepest ascent paths. These paths exist are because of the nice properties a routing function has. The proof Theorem 1.33 uses non-trivial results from Morse theory [BH04; Mat02; Nic11]. We use Morse theory to derive information about the shape of a connected component of $\{g \neq 0\}$ by studying the routing points of g. With the proof of Theorem 1.33 in hand, we can easily prove the partial correctness of **Connectivity** in Theorem 2.24.

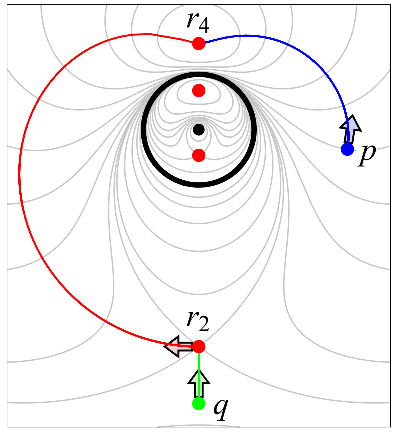


Figure 1.32 Illustration of a connectivity path connecting p and q.

1.4.2 Termination

The termination of **Connectivity** relies heavily on the following theorem, which is another one of the main results in this thesis.

Theorem 1.34. *For all nonzero* $f \in \mathbb{R}[x_1, \ldots, x_n]$ *there exists a semi-algebraic set* $S \subset \mathbb{R}^n$ *such that* $\dim(\mathbb{R}^n \setminus S) < n$ *and for all* $(c_1, \ldots, c_n) \in S$ *the mapping* $g \colon \mathbb{R}^n \to \mathbb{R}$ *defined by*

$$g = \frac{f^2}{\left((x_1 - c_1)^2 + \cdots + (x_n - c_n)^2 + 1\right)^{\deg(f)+1}} \tag{1.35}$$

is a routing function.

Intuitively, this theorem says there is a set of "bad" choices $(\mathbb{R}^n \setminus S)$ for $(c_1, \ldots, c_n)$ which is "small." By choosing $(c_1, \ldots, c_n)$ outside of this "bad" set, the function g in (1.35) is a routing function. The proof of this theorem uses non-trivial results from semi-algebraic geometry such as Sard's Theorem and the Constant Rank Theorem. We will use Theorem 1.34 to prove the termination of **Connectivity** in Theorem 3.13.

1.4.3 Length Bound

The next problem we tackle is in Chapter 4. For a given polynomial f, we will compute an upper bound on the length of a connectivity path for any two points p, q in a connected component of

$\{f \neq 0\}$. To do so, we will bound the length of individual steepest ascent paths in a given ball. Before we can state the bound precisely, we introduce some notions.

Definition 1.36. Let $g\colon \mathbb{R}^n \to \mathbb{R}$ be a C^2 function with $n \geq 2$. Suppose $p, q \in \mathbb{Q}^n$ are connected by steepest ascent paths using outgoing eigenvectors of g and denote by $r_1, \ldots, r_k, \phi_1, \ldots, \phi_{k+1}$ a connectivity path P for p and q. We define the *length of the connectivity path* P to be

$$\sum_{i=1}^{k+1} \mathrm{Length}(\phi_i).$$

Example 1.37. Let

$$\frac{\left(10x_1^3 - 10x_1^2 + 10x_2^2 - 1\right)^2}{\left(x_1^2 + x_2^2 + 1\right)^4}. \tag{1.38}$$

In Figure 1.39 we visualize a connectivity path P given by $r_3, r_5, r_8, r_4, r_2, \phi_1, \phi_2, \phi_3, \phi_4, \phi_5, \phi_6$ for points p and q as the union of red steepest ascent paths and red routing points. The white arrows $v_4, -v_4, v_5, -v_5$ represent the outgoing eigenvectors of $(\mathrm{Hess}\, g)(r_4)$, $(\mathrm{Hess}\, g)(r_5)$, respectively, that appear in the definition of P. We give approximations of the lengths of six steepest ascent paths connecting p and q in Table 1.40. We approximate the length of the connectivty path P to be

$$2.97553 + 2 \cdot 1.3696 + 2 \cdot 1.96328 + 0.964633 = 10.6059.$$

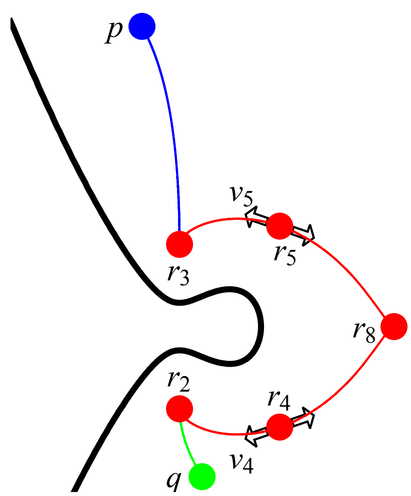


Figure 1.39 A sample connectivity path connecting p and q.

Table 1.40 Approximate steepest ascent path lengths for a sample connectivity path connecting p and q.

ϕ_i	Image of ϕ_i	Adjacent Points	Approximate Length(ϕ_i)
ϕ_1	$\mathrm{SA}\left(g, p, \widehat{\nabla g(p)}\right)$	p, r_2	2.97553
ϕ_2	$\mathrm{SA}\left(g, r_5, v_5\right)$	r_3, r_5	1.3696
ϕ_3	$\mathrm{SA}\left(g, r_5, -v_5\right)$	r_5, r_8	1.96328
ϕ_4	$\mathrm{SA}\left(g, r_4, -v_4\right)$	r_4, r_8	1.96328
ϕ_5	$\mathrm{SA}\left(g, r_4, v_4\right)$	r_4, r_2	1.3696
ϕ_6	$\mathrm{SA}\left(g, q, \widehat{\nabla g(q)}\right)$	r_1, q	0.964633

Definition 1.41. For a polynomial $P \in \mathbb{Z}[x_1, \ldots, x_n]$ of the form

$$P = \sum_E a_E x^E,$$

where $E = (E_1, \ldots, E_n)$ runs over n-tuples of nonnegative integers and $x^E = x_1^{E_1} \cdots x_n^{E_n}$, we define the *height of* P to be

$$\mathrm{hgt}(P) = \max_E |a_E|.$$

If $\mathcal{P} = \{P_1, \ldots, P_r\}$ is a subset of $\mathbb{Z}[x_1, \ldots, x_n]$, then we define the *height of* $\mathcal{P}$ to be

$$\mathrm{hgt}(\mathcal{P}) = \max\bigl\{\mathrm{hgt}(P_1), \ldots, \mathrm{hgt}(P_r)\bigr\}.$$

Example 1.42. Let

$$P_1 = x_1^2 + 4x_1x_2 - 20x_2 + 3,$$
$$P_2 = 2x_1^2 - 4x_1x_2 + 3x_1 - 1,$$

then

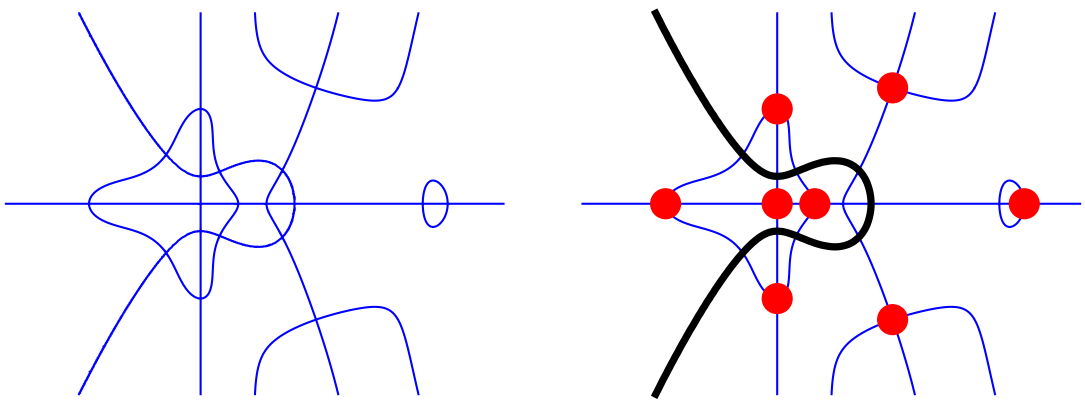
$$\mathrm{hgt}(P_1) = \max\bigl\{|1|, |4|, |-20|, |3|\bigr\} = 20,$$
$$\mathrm{hgt}(P_2) = \max\bigl\{|2|, |-4|, |3|, |-1|\bigr\} = 4.$$

If $\mathcal{P} = \{P_1, P_2\}$, then $\mathrm{hgt}(\mathcal{P}) = \max\{20, 4\} = 20$.

Definition 1.43. [Hof86] For a C^2 function $g\colon \mathbb{R}^n \to \mathbb{R}$ we define the *gradient extremal of* g as

$$\Theta(g) = \{x \in \mathbb{R}^n \mid \exists \lambda \in \mathbb{R},\ (\operatorname{Hess} g)(x) \cdot \nabla g(x) = \lambda \nabla g(x)\}\,. \tag{1.44}$$

Example 1.45. We illustrate $\Theta(g)$ for (1.38) as the blue curve in Figure 1.46a. In Figure 1.46b, we can observe that the set $\{g = 0\}$ (black curve) and the routing points of g (red points) are contained in $\Theta(g)$.



(a) **(b)**

Figure 1.46 Illustration of the ridge and valley set of g.

An example of when $\Theta(g)$ is not a curve can be seen when

$$g = \frac{\left(x_1^2 + x_2^2\right)^2}{\left(x_1^2 + x_2^2 + 1\right)^3}.$$

Here, $\Theta(g) = \mathbb{R}^2$ and g is not a routing function because the set of points $\left\{(x_1, x_2) \;\middle|\; x_1^2 + x_2^2 = 2\right\}$ are routing points of g that are degenerate.

We now state the length bound result. A detailed proof will be given in Chapter 4.

Theorem 1.47. *Let $f \in \mathbb{Z}[x_1, \ldots, x_n]$, $n \geq 2$, degree $d \geq 2$ with no singular points. Suppose $(c_1, \ldots, c_n) \in \mathbb{Z}^n$ such that*

$$g = \frac{f^2}{\left((x_1 - c_1)^2 + \cdots + (x_n - c_n)^2 + 1\right)^{d+1}}$$

is a routing function. Let $H = \mathrm{hgt}(f)$*. Let* $\Theta(g)$ *be the gradient extremal of* g*. Let* D *be a connected component of* $\{f \neq 0\}$ *and* $p, q \in \mathbb{Q}^n \cap D$*. Let* B *be a ball of radius*

$$r = n\big(120A_1A_2Hd(c_1^2 + \cdots + c_n^2 + 1)\big)^{4n^3(6d)^{3n}}$$

where

$$\frac{A_1}{A_2} = \min\left\{ g(p), g(q), \frac{1}{\Big(2dH\big(c_1^2 + \cdots + c_n^2 + 2\big)\Big)^{104n^3(5d)^{5n}}} \right\}$$

is an irreducible fraction with $A_1, A_2 > 0$*. Suppose* $\Theta(g) \cap \overline{B}$ *is a compact rectifiable curve. Then* p *and* q *can be connected in* D *by a connectivity path of length bounded by*

$$4nr(6d+4)^{n-1}.$$

The proof idea was motivated by the work of D'Acunto and Kurdyka [DK04]. The basic idea being that we can bound the length of a trajectory of ∇g in a ball by bounding the length of $\Theta(g)$ in a ball. To find an appropriate sized ball, we use the simple observation that a connectivity path for p and q must be contained in $\{g \geq \varepsilon\}$ where

$$\varepsilon = \min\big\{g(p), g(q), M\big\}$$

and M is the minimum value of $g(r)$ over all routing points r of g. For instance, for the g, p, and q given in Example 1.37, we visualize $\{g \geq g(p)\}, \{g \geq g(q)\}$, and $\{g \geq M\}$ as the blue, green, and red regions, respectively, in Figure 1.48. We see that

$$\{g \geq g(q)\} \subset \{g \geq M\} \subset \{g \geq g(p)\}$$

and any connectivity path P for p and q must be contained in $\{g \geq \varepsilon\} = \{g \geq g(p)\}$ (the blue region).

1.4.4 Experimental Results

The length bound we gave in Theorem 1.47 is a sort of intrinsic complexity result for the **Connectivity** algorithm. Presently, we have not carried out a full complexity analysis of the **Connectivity** algorithm. This is because the steps for the subalgorithm **Destination** have not been fully detailed. What we present in Chapter 5 are computational results to show that the method **Connectivity** is fast in practice. We accomplish this by executing the **Connectivity**

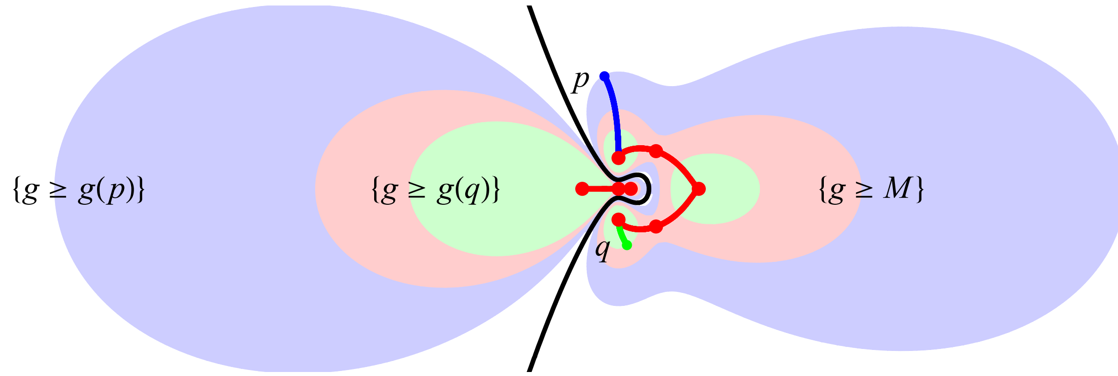


Figure 1.48 Illustration of several superlevel sets of g.

algorithm for several non-trivial inputs. We then uniformly generate points on a grid and show how quickly we can answer connectivity queries. We also give evidence that the **Connectivity** method runs faster on sparse polynomial input as opposed to dense polynomial input.

Chapter 2

Partial Correctness

In this chapter, we will prove the partial correctness of the algorithm **Connectivity** in the form of Theorem 2.24. It essentially amounts to showing Theorem 1.33 is true. We assume throughout this section that $g\colon \mathbb{R}^n \to \mathbb{R}$ is a C^2 function with $n \geq 2$. The examples in this section will assume g takes the form

$$g = \frac{\left(-2x_1^2 + x_1^4 - 2x_2^2 + 2x_1^2x_2^2 + x_2^4\right)^2}{\left(x_1^2 + (x_2 - 1)^2 + 1\right)^5}.$$

In the first section we give some preliminary notions and lemmas necessary for proving Theorem 1.33. We then prove Theorem 1.33 and the correctness of the algorithm **Connectivity** in the second section.

2.1 Preliminaries

To prove Theorem 1.33, we will use results motivated from the field of Morse theory. In Morse theory, one analyzes the topology of a manifold by studying differentiable functions on that manifold. In our case, we will be studying the manifold $\mathbb{R}^n$ and decomposing a region into sets of similar behavior based on trajectories.

Definition 2.1. If $p \in \mathbb{R}^n$ is a nondegenerate critical point of g, then the *stable manifold of* p is defined to be

$$W^s(p) = \{x \in \mathbb{R}^n \mid \operatorname{dest}(\phi_x) = p\} \cup \{p\}.$$

where ϕ_x is the trajectory of ∇g through x using $\widehat{\nabla g(x)}$.

Notice that the stable manifold of p contains p.

Example 2.2. Figure 2.3 illustrates the stable manifolds for the routing points of g. The stable manifolds for r_1 and r_4 are the blue and green regions, respectively. The stable manifold for r_2 is the blue line while the stable manifold for r_3 is the green line.

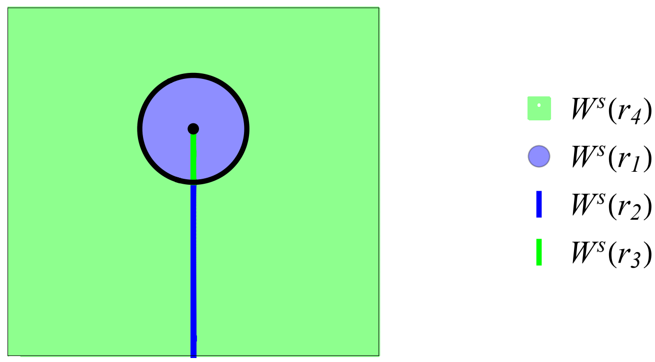


Figure 2.3 Illustration of the stable manifolds for the routing points of g.

According to Figure 2.3, it appears we can decompose each connected component of $\{g \neq 0\}$ into a disjoint union of stable manifolds. We will use the following lemmas to show that if g is a routing function, we can in fact decompose a connected component into a disjoint union of stable manifolds. First, we observe the simple fact that g strictly increases along a steepest ascent path.

Lemma 2.4. *Let $p \in \mathbb{R}^n$. If p is not a critical point of g then g increases along a trajectory of ∇g through p using $\widehat{\nabla g(p)}$.*

Proof. Let $p \in \mathbb{R}^n$ with $\nabla g(p) \neq 0$. Let ϕ denote a trajectory of ∇g through p using $\widehat{\nabla g(p)}$. We have

$$\frac{d}{dt} g\big(\phi(t)\big) = \big\langle \nabla g\big(\phi(t)\big), \phi'(t) \big\rangle = \big\langle \nabla g\big(\phi(t)\big), \nabla g\big(\phi(t)\big) \big\rangle = \big\| \nabla g\big(\phi(t)\big) \big\|^2 . \tag{2.5}$$

Since p is not a critical point of g, $\big\| \nabla g\big(\phi(t)\big) \big\|^2 > 0$ for all $t > 0$. It follows from (2.5) that

$$\frac{d}{dt} g\big(\phi(t)\big) > 0$$

for all $t > 0$. Hence g strictly increases along ϕ. $\square$

For the rest of the section we assume g is a routing function and D is a connected component of $\{g \neq 0\}$. First, we state some simple facts about g.

Lemma 2.6. *g is a bounded.*

Proof. The first property in the definition of routing function guarantees g is bounded below by 0. Suppose for a contradiction that g is not bounded above. Then for all M, there exists $x \in \mathbb{R}^n$ such that $|g(x)| > M$. In particular, for every $n \in \mathbb{N}$, there exists $x_n \in \mathbb{R}^n$ for which $|g(x_n)| > n$. Fix such a sequence $\{x_n\}_{n=1}^{\infty}$. Certainly, $g(x_n) \geq 0$. Let

$$L = \min_{n \in \mathbb{N}} \{g(x_n) \mid g(x_n) > 0\},$$
$$S = \{x \in \mathbb{R}^n \mid g(x) \geq L\}.$$

Let k be the index such that $g(x_k) = L$. The second property in the definition of a routing function guarantees S is bounded by letting $\varepsilon = L > 0$. Since the tail $\{x_n\}_{n=k}^{\infty}$ is contained in S, the Bolzano-Weierstrass theorem implies there exists a subsequence $\{x_{n_j}\}_{j=1}^{\infty}$ which converges to some limit M. Since g is continuous everywhere,

$$\lim_{j \to \infty} g(x_{n_j}) = g(M).$$

In particular, the sequence $\left\{g(x_{n_j})\right\}_{j=1}^{\infty}$ is convergent, hence bounded. However, by construction, $\left|g(x_{n_j})\right| > n_j \geq j$ for all $j \in \mathbb{N}$, and hence this sequence is not bounded, a contradiction. Thus g is bounded above. Therefore g is bounded. □

Lemma 2.7. *For all $L > 0$, $\{x \in D \mid g(x) \geq L\}$ is compact.*

Proof. Let $L > 0$ and $K = \{x \in D \mid g(x) \geq L\}$. Recall that g is bounded (Lemma 2.6), hence there exists M, such that for all $x \in \mathbb{R}^n$, $|g(x)| \leq M$. The set

$$S = g^{-1}\big([L, M]\big) = \{x \in \mathbb{R}^n \mid g(x) \geq L\}$$

is closed because it is the preimage of a closed set under a continuous function and bounded due to the second property of g being a routing function (letting $\varepsilon = L > 0$). Therefore, S is compact. The semi-algebraic set $\{g > 0\}$ is a disjoint union of open semi-algebraic connected components $D_1, \ldots, D_k$ where $D_1 = D$ (without loss of generality). Since $L > 0$, S is contained in $\{g > 0\}$ and hence the disjoint union of $D_1, \ldots, D_k$. It follows that $K = D \cap S$ is compact. □

We show next that the trajectories are unique assuming a certain condition.

Lemma 2.8. *Let $p \in D$. If $\nabla g(p) \neq 0$, then there exists a unique trajectory ϕ of ∇g through p using $\widehat{\nabla g(p)}$*

Proof. Let $p \in D$. The component D is an open subset of $\mathbb{R}^n$ containing p and $g \in C^2(D)$. According to the Fundamental Existence-Uniqueness Theorem [Per01, Section 2.2, pp. 74], there exists $a > 0$ such that

$$\begin{aligned} \phi'(t) &= \nabla g\big(\phi(t)\big) \\ \phi(0) &= p \end{aligned} \tag{2.9}$$

has a unique solution $\phi(t)$ on the interval $[-a, a]$. Let $[0, \beta)$ be the right maximal interval of existence of $\phi(t)$.

Because g is bounded (Lemma 2.6), the trajectory ϕ is bounded. It follows from [Per01, Theorem 3, Section 2.4, pp. 91] that $\beta = \infty$. Certainly $\lim_{t \to 0^+} \phi(t) = p$ and

$$\lim_{t \to 0^+} \frac{\phi'(t)}{\|\phi'(t)\|} = \widehat{\nabla g(p)}.$$

Hence ϕ is the trajectory of ∇g through p using $\widehat{\nabla g(p)}$. $\square$

Remark 2.10. A similar argument to the one above shows that if $p \in D$ and $\nabla g(p) \neq 0$ then there exists a unique C^2 function $\phi\colon (-\infty, 0] \to \mathbb{R}^n$ satisfying

$$\begin{aligned} \phi'(t) &= -\nabla g\big(\phi(t)\big) \\ \phi(0) &= p. \end{aligned}$$

Combined with the argument above, this means that there exists a unique C^2 function $\phi\colon \mathbb{R} \to \mathbb{R}^n$ satisfying (2.9). When $\nabla g(p) = 0$, $\phi = p$ is the unique solution to (2.9), which exists for all $t \in \mathbb{R}$. We can conclude that the gradient vector field ∇g is complete.

Next, we have the important observation that the destination of every steepest ascent path is a routing point of g.

Lemma 2.11. *Let $p \in D$ with $\nabla g(p) \neq 0$ and ϕ be the trajectory of ∇g through p using $\widehat{\nabla g(p)}$. Then* $\operatorname{dest}(\phi)$ *exists and is a routing point of g in D.*

Proof. Let $p \in D$ with $\nabla g(p) \neq 0$ and ϕ be the trajectory of ∇g through p using $\widehat{\nabla g(p)}$, whose existence is guaranteed by Lemma 2.8. Let $K = \{x \in D \mid g(x) \geq g(p)\}$. Lemma 2.7 implies K is compact. Let $\{t_n\} \subset \mathbb{R}_+$ be a sequence with $\lim_{n\to\infty} t_n = \infty$. Let $\big\{\widetilde{t_n}\big\}$ denote the tail of $\{t_n\}$

so that $\{\phi(\widetilde{t}_n)\} \subseteq K$ for all n. The sequence $\{\phi(\widetilde{t}_n)\}$ is an infinite set of points in a compact set, so it has an accumulation point q.

First, we show q is a critical point of g. It suffices to show $\nabla g(\phi(t)) \to 0$ as $t \to \infty$. Differentiating $\phi'(t)$ we find

$$\begin{aligned}\phi''(t) &= \left(\nabla_{\frac{\partial}{\partial \phi}} \nabla g(\phi(t))\right) \phi'(t) \\ &= \left(\nabla_{\frac{\partial}{\partial \phi}} \nabla g(\phi(t))\right) \nabla g(\phi(t))\end{aligned} \tag{2.12}$$

holds for all $t > 0$. The first and second derivatives of g are bounded because g is a routing function, hence we may deduce from (2.12) that ϕ' is uniformly Lipschitz continuous for $t > 0$.

Since g is bounded (Lemma 2.6), $g_\infty := \lim_{t\to\infty} g(\phi(t)) < \infty$, and for $0 < t < \infty$

$$g_\infty \geq g(\phi(t)) > g(p),$$

so from (2.5)

$$\int_0^\infty \|\phi'(t)\|^2 \, dt = \int_0^\infty \frac{d}{dt} g(\phi(t)) \, dt = g_\infty - g(p) < \infty. \tag{2.13}$$

Since ϕ' is uniformly Lipschitz continuous, (2.13) implies

$$\lim_{t\to\infty} \nabla g(\phi(t)) = \lim_{t\to\infty} \phi'(t) = 0$$

as desired.

We claim $\mathrm{dest}(\phi) = \lim_{t\to\infty} \phi(t) = q$. Since nondegenerate critical points are isolated [BH04, Lemma 3.2, Section 3.1, pp. 47], we can pick a closed neighborhood U of q where q is the only critical point of U. Suppose for a contradiction $\lim_{t\to\infty} \phi(t) \neq q$, then there is an open neighborhood $V \subset U$ of q and a sequence $\{s_n\} \subset \mathbb{R}_+$ with $\lim_{n\to\infty} s_n = \infty$ and $\phi(s_n) \in U \setminus V \subseteq \overline{U \setminus V}$. Thus, the sequence $\{\phi(s_n)\}$ has an accumulation point in the compact set $\overline{U \setminus V}$ which, as above, must be a critical point of g. This contradicts the choice of U, and therefore, $\mathrm{dest}(\phi) = q$.

Finally, we show q is a routing point in D. We find $g(q) > g(p) > 0$ because g increases along ϕ as $t \to \infty$ (Lemma 2.4). Hence, $q \in D$ is a routing point. □

We now show that the connected components of $\{g \neq 0\}$ can be decomposed in to a disjoint union of stable manifolds.

Lemma 2.14. *The component D is a disjoint union of stable manifolds corresponding to the* *routing points contained in D; that is,*

$$D = \coprod_{p \in R_D} W^s(p).$$

where R_D is the set of routing points of g in D.

Proof. Let R_D be the set of routing points of g in D. Let $q \in D$ be arbitrary. Certainly $q \in W^s(q)$, so we may assume $\nabla g(q) \neq 0$. Let ϕ denote the trajectory of ∇g through q using $\widehat{\nabla g(q)}$, whose existence is guaranteed by Lemma 2.8. It follows from Lemma 2.11 that there exists a routing point $r \in R_D$ such that $\operatorname{dest}(\phi) = r$. Hence $q \in W^s(r)$. This shows D is a union of stable manifolds. It is a disjoint union due to the uniqueness of ϕ. $\square$

Now that we have a decomposition, the next natural question to ask is whether we can determine the dimension of each stable manifold. The definition of a stable manifold relies on a critical point, so one may believe that the dimension relies on the index of the critical point. To see this, we use the Stable Manifold Theorem, a fundamental result in the field of dynamical systems.

Lemma 2.15. *If $p \in D$ is a routing point of g with index k, then $W^s(p)$ is a smooth k-dimensional manifold.*

Proof. Let p be a routing point of index k of g contained in D. The result in [BH04, Theorem 4.2, Section 4.1, pp. 94] has the same conclusion but the assumptions are that g is a Morse function defined on a finite dimensional compact smooth Riemannian manifold. The function g restricted to D is Morse because g is a routing function. The connected component D of $\{g \neq 0\}$ is a finite dimensional smooth Riemannian manifold, but it is not compact. The compactness assumption is used in several spots throughout the proof of the cited theorem.

(1) There exist finitely many critical points of g on the given manifold [BH04, Corollary 3.3, Section 3.1, pp. 47].
(2) The gradient vector field ∇g generates a unique 1-parameter group of diffeomorphisms defined on $\mathbb{R} \times D$ [BH04, Section 4.1, pp. 94].
(3) The destination of a trajectory is a critical point [BH04, Corollary 3.19, Section 3.2, pp. 59].

All of these issues can be addressed though.

(1) The manifold D contains finitely many routing points because g is a routing function.
(2) This follows from the fact that the gradient vector field ∇g is complete (Remark 2.10).
(3) This is exactly Lemma 2.11.

□

We expect all the routing points in a connected component to be connected via steepest ascent paths, so we expect each component to have a "peak" to ascend to; that is, we expect each component to have a local maximum. The simple observation follows from the routing function properties.

Lemma 2.16. *The component D contains a routing point of g having index n.*

Proof. Take $x_0 \in D$. Then $g(x_0) > 0$. Let $K = \{x \in D \mid g(x) \geq g(x_0)\}$. The set K is compact (Lemma 2.7), hence g has a maximum z on K. The maximum must occur on the interior of K. If the interior is non-empty, then there exists an open ball B around z such that $g(z) \geq g(x)$ for all $x \in B$. Hence z is a local maximum of g; that is, z is a routing point having index n. If the interior is empty, choose $x_1 \in D$ such that $g(x_1) < g(x_0)$, which is possible due to the second property of g being a routing function. Let $\widetilde{K} = \{x \in D \mid g(x) \geq g(x_1)\}$. Again, the set $\widetilde{K}$ is compact so g has a maximum $\widetilde{z}$ on K. The interior of $\widetilde{K}$ is non-empty, so as argued before, $\widetilde{z}$ is a local maximum of g; that is $\widetilde{z}$ is a routing point having index n. □

Throughout this section we will use the notation ∂W to denote the boundary of a stable manifold W.

Lemma 2.17. *If p is a routing point of g of index n, then $\partial W^s(p)$ contains no routing points of index n.*

Proof. Let p be a routing point of g of index n. Assume for a contradiction that $\partial W^s(p)$ contains a routing point q of index n. Hence q is a local maximum of g. Any neighborhood U of q must contain a point $y \in W^s(p)$ where $g(y) > g(q)$, contradicting the fact that q is a local maximum. Hence, $\partial W^s(p)$ contains no routing points of index n. □

Lemma 2.18. *Let $r \in D$ be a routing point of g of index n. Let $p \in \partial W^s(r) \cap D$ with $\nabla g(p) \neq 0$ and ϕ be the trajectory of ∇g through p using $\widehat{\nabla g(p)}$. Then there exists a routing point $q \in \partial W^s(r) \cap D$ such that* $\operatorname{dest}(\phi) = q$.

Proof. Let $r \in D$ be a routing point of g of index n. Let $p \in \partial W^s(r) \cap D$ with $\nabla g(p) \neq 0$ and ϕ be the trajectory of ∇g through p using $\widehat{\nabla g(p)}$, whose existence is guaranteed by Lemma 2.8. According to Lemma 2.11, there exists a routing point $q \in D$ such that $\operatorname{dest}(\phi) = q$. Hence

$p \in W^s(q)$. In fact, all the points along $\mathrm{SA}\left(g, p, \widehat{\nabla g(p)}\right)$ are in $W^s(q)$. Since ϕ is continuous and D is a disjoint union of stable manifolds (Lemma 2.14), we find that $q \in \partial W^s(r)$. Thus $q \in \partial W^s(r) \cap D$ as desired. □

Lemma 2.19. *Let $r \in D$ be a routing point of g of index n. Let $p \in \partial W^s(r) \cap D$ be a routing point of g of index strictly less than n. If v is a outgoing eigenvector of $(\operatorname{Hess} g)(p)$ tangent to $\partial W^s(r)$, then there exists a routing point $q \in \partial W^s(r) \cap D$ that is reachable from p using v.*

Proof. Let $r \in D$ be a routing point of g of index n. Let $p \in \partial W^s(r) \cap D$ be a routing point of g of index strictly less than n. Let v be a outgoing eigenvector of $(\operatorname{Hess} g)(p)$ tangent to $\partial W^s(r)$. As argued in the proof of Lemma 2.15, we may use the conclusions of the Stable Manifold Theorem [BH04, Theorem 4.2, Section 4.1, pp. 94]. This theorem guarantees the existence of the unstable manifold

$$W^u(p) = \{x \in \mathbb{R}^n \mid \operatorname{dest}(\phi_x) = p\} \cup \{p\}.$$

where ϕ_x is the trajectory of $-\nabla g$ through x using $-\widehat{\nabla g(x)}$. There exists a submanifold of $W^u(p)$ that is tangent to the eigenspace spanned by outgoing eigenvectors of $\operatorname{Hess} g(p)$. In particular, this submanifold corresponds to $\mathrm{SA}(g, p, v)$. For each s in $\mathrm{SA}(g, p, v)$, $\nabla g(s) \neq 0$. We can argue using Lemma 2.11 that there exists a routing point $q \in D$ such that for each s in $\mathrm{SA}(g, p, v)$, $\operatorname{dest}(\phi_s) = q$ where ϕ_s is the trajectory of ∇g through s using $\widehat{\nabla g(s)}$. In particular, q is reachable from p using g and v. Certainly $q \in D$. Since $\mathrm{SA}(g, p, v)$ is a continuous curve and D is a disjoint union of stable manifolds (Lemma 2.14), we find that $q \in \partial W^s(r)$. Thus $q \in \partial W^s(r) \cap D$ as desired. □

Lemma 2.20. *Let $r \in D$ be a routing point of g of index n. Let q be a routing point of g on $\partial W^s(r) \cap D$. Then q is connected to r by steepest ascent paths using outgoing eigenvectors of g.*

Proof. Let $r \in D$ be a routing point of g of index n. Let q be a routing point of g on $\partial W^s(r) \cap D$. According to Lemma 2.17, q must be a routing point of index strictly less than n. Hence, $(\operatorname{Hess} g)(q)$ has at least one outgoing eigenvector, call it v.

If v is not tangent to $\partial W^s(r)$, then $\mathrm{SA}(g, q, v)$ or $\mathrm{SA}(g, q, -v)$ lies in the stable manifold $W^s(r)$ because D is a disjoint union of stable manifolds (Lemma 2.14). Hence r is reachable from q using v (or $-v$). We see q is connected to r by steepest ascent paths using outgoing eigenvectors of g.

If v is tangent to $\partial W^s(r)$, according to Lemma 2.19 there exists another routing point q_2 that is reachable from $q = q_1$ using v. The routing point q_2 has index strictly less than n, so as before, there exists a routing point q_3 that is reachable from q_2 using v. We repeat this process.

The function g is bounded (Lemma 2.6) and there are finitely many routing points, so eventually the process will terminate, and we will find a routing point q_k, $k \geq 1$, where $(\operatorname{Hess} g)(q_k)$ has an outgoing eigenvector v_k that is not tangent to $\partial W^s(r)$. The point r is reachable from q_k using g and v_k as before. We have found a sequence of routing points $q_1 \dots, q_k$, $k \geq 2$ such that q_i is reachable from q_{i-1} using g and an outgoing eigenvector of $(\operatorname{Hess} g)(q_{i-1})$. Thus the point q is connected to r by steepest ascent paths using outgoing eigenvectors of g by the connectivity path $q_1, \dots, q_k, r$ and the corresponding trajectories connecting the routing points $q_1, \dots, q_k, r$. □

Definition 2.21. Let $p, q \in D$, $p \neq q$, be routing points of g of index n. We say $W^s(p)$ is *adjacent* to $W^s(q)$ if $D \cap \partial W^s(p) \cap \partial W^s(q)$ is non-empty.

Lemma 2.22. *Let $p, q \in D$, $p \neq q$, be routing points of g of index n. If $W^s(p)$ is adjacent to $W^s(q)$, then $D \cap \partial W^s(p) \cap \partial W^s(q)$ must contain a routing point of g.*

Proof. Let $p, q \in D$, $p \neq q$, be routing points of g of index n. Assume $Z = D \cap \partial W^s(p) \cap \partial W^s(q)$ is non-empty. Suppose Z does not contain a routing point of g. As Z is non-empty, there exists a point $x \in Z$ that is not a routing point of g. According to Lemma 2.18, there exists a routing point $q \in Z$. However, this contradicts our assumption. Hence, Z contains a routing point of g. □

2.2 Proof of Main Result

In this section we will prove Theorem 1.33 and prove the partial correctness of **Connectivity** in the form of Theorem 2.24.

Proof of Theorem 1.33. Let R denote the set of routing points of g in D and $p, q \in D$, $p \neq q$ be arbitrary. We will show p and q are connected by steepest ascent paths using outgoing eigenvectors of g. We may assume without loss of generality that p and q are routing points of g, otherwise we can always ascend to one using Lemma 2.11 if $\nabla g(p) \neq 0$ or $\nabla g(q) \neq 0$. Let $m_1, \dots, m_\ell$ denote the routing points in R having index n. We see $\ell \geq 1$ due to Lemma 2.16. We see that $|R| > 1$ because p and q are both distinct routing points of g.

Suppose first that $\ell = 1$. According to Lemma 2.15, $W^s(m_1)$ is n-dimensional and the stable manifolds for the points in $R \setminus \{m_1\}$ have dimension strictly less than n. As D is a disjoint union of stable manifolds of the routing points in R (Lemma 2.14), it follows that the points in $R \setminus \{m_1\}$ lie on $\partial W^s(m_1)$. We see for all $r \in R \setminus \{m_1\}$, r is connected to m_1 by steepest ascent paths using outgoing eigenvectors (Lemma 2.20), hence any two routing points in D can be connected using steepest ascent paths using outgoing eigenvectors of g.

Now suppose $\ell > 1$. According to Lemma 2.15, for all i, $W^s(m_i)$ is n-dimensional and the stable manifolds for the points in $R \setminus \{m_1, \ldots, m_\ell\}$ have dimension strictly less than n. If p (or q) is a routing point with index strictly less than n, then it must lie on the boundary of some stable manifold $W^s(m_i)$. According to Lemma 2.20, we can connect p (or q) to m_i by steepest ascent paths using outgoing eigenvectors. Hence, we may assume without loss of generality that p and q are routing points having index n. We will connect p and q by looking at a sequence of adjacent stable manifolds of dimension n as seen in Figure 2.23a.

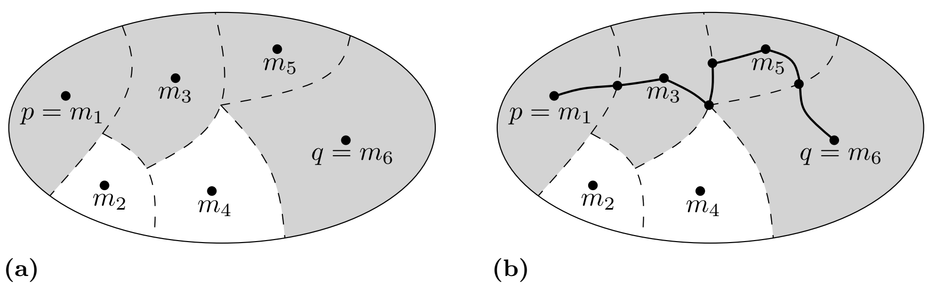


Figure 2.23 A decomposition of a connected component of g.

It suffices to show that we can connect any two m_i, m_j whose stable manifolds $W^s(m_i)$ and $W^s(m_j)$ are adjacent because D is a disjoint union of stable manifolds of the routing points in R (Lemma 2.14). From Lemma 2.22, we know two adjacent manifolds have a routing point in common in their boundary. According to Lemma 2.20, we can connect this common routing point to both m_i and m_j by steepest ascent paths using outgoing eigenvectors. Hence we can connect m_i and m_j by steepest ascent paths using outgoing eigenvectors. We illustrate this in Figure 2.23b. This completes the proof of Theorem 1.33. □

Theorem 2.24. *Algorithm* ***Connectivity*** *is correct.*

Proof. Let f, p, q be the inputs to **Connectivity** satisfying the specification. Suppose **Connectivity** terminated with output t. Let

$$g = \frac{f^2}{U^\gamma} \text{ where } U = (x_1 - c_1)^2 + \cdots + (x_n - c_n)^2 + 1, \gamma = \deg(f) + 1$$

be the function formed in step 2. First, we claim that the set R formed in step 3 is the set of routing points of g. We observe that

$$\nabla g = \frac{f}{U^{\gamma+1}}(2\nabla f U - \gamma f \nabla U) \tag{2.25}$$

so

$$\begin{aligned} R &= \{x \in \mathbb{R}^n \mid \nabla g(x) = 0, g(x) \neq 0\} \\ &= \{x \in \mathbb{R}^n \mid 2\nabla f(x)U(x) - \gamma f(x)\nabla U(x) = 0, f(x) \neq 0\}. \end{aligned}$$

Let $\mathcal{F} = 2\nabla f U - \gamma f \nabla U$. We see that the $V(\mathcal{F})$ contains exactly the routing points of g and the singular points of f because U is non-zero. In step 3, we remove the finitely many singular points of f from $V(\mathcal{F})$, leaving us with the correct set of routing points. The set of routing points is finite because $V(\mathcal{F})$ is zero-dimensional.

We now claim that g is a routing function. The function g is C^2 because it is a rational function where the denominator is nonnegative. According to step 2, the finitely many routing points of g are all nondegenerate because $\det(\operatorname{Hess} g)(r) \neq 0$ for all $r \in R$. The choice of $\gamma = \deg(f) + 1$ guarantees the property that g vanishes at infinity (property two) because the degree of the numerator is smaller than the degree of the denominator. Certainly the function g is nonnegative. To understand why the first derivative of g is bounded, we observe in (2.25) that each component of ∇g is a rational function where the degree of the numerator is smaller than the degree of the denominator, which is nonnegative. A similar argument holds for each component of $\operatorname{Hess} g$. Hence g satisfies the properties in the definition of a routing function.

Observe that $g = 0$ if and only if $f = 0$. Due to Theorem 1.33, we know the routing points of g on a connected component of $\{f \neq 0\}$ are connected by steepest ascent paths using outgoing eigenvectors of g. It is important to observe that these steepest ascent paths do not cross $f = 0$ due to Lemma 2.4. In steps 5 and 6, we use the certified **Destination** algorithm to determine which routing points are adjacent to one another via steepest ascent paths using outgoing eigenvectors. The matrix A is the adjacency matrix for the graph whose vertices are the routing points and whose edges are the steepest ascent paths connecting them. Hence, the matrix M, the reflexive, symmetric, transitive closure of A, satisfies the condition that $M_{ij} = 1$ if and only if $r_i, r_j \in R$ lie in a same connected component of $\{f \neq 0\}$.

We claim that the point p can be connected to a routing point r_i lying in the same connected component of $\{f \neq 0\}$. If $\nabla g(p) = 0$ then p is a routing point of g because $f(p) > 0$ implies $g(p) > 0$; that is, there exists i such that $r_i = p$. Otherwise, if $\nabla g(p) \neq 0$, let ϕ_p be the trajectory of ∇g through p using $\widehat{\nabla g(p)}$. According to Lemma 2.11, there exists i such that the destination of ϕ_p is a routing point r_i. The index i in this case can be determined using the **Destination** algorithm (step 7). A similar arugment holds for q; the point q can be connected to a routing point r_j lying in the same connected component of $\{f \neq 0\}$, with this index being determined in step 8. We use the connectivity matrix M in step 9 to determine if r_i and r_j lie in a same connected component of $\{f \neq 0\}$ to conclude whether p and q lie in a same connected component. $\square$

Chapter 3

Termination

In this chapter, we will prove that the termination of the algorithm **Connectivity** in the form of Theorem 3.13. For this, we must show that the perturbation step completes after a finite number of iterations. We will show in Theorem 1.34 that there is only a small (measure zero) set of parameters for which the function g formed in **Connectivity** is not a routing function. Hence we are guaranteed to find a routing function by finitely many perturbation of these parameters on the integer grid.

In the first section we state some preliminary notions and a lemma used in the proof of Theorem 1.34. In the second section we prove Theorem 1.34 and show the algorithm **Connectivity** terminates.

3.1 Preliminaries

We begin by recalling defintions from semi-algebraic geometry [Bas03]. Let $A \subset \mathbb{R}^m$ and $B \subset \mathbb{R}^n$ be two semi-algebraic sets. A function $f\colon A \to B$ is *semi-algebraic* if its graph is a semi-algebraic subset of $\mathbb{R}^{m+n}$. For open A, the set of semi-algebraic functions from A to B for which all partial derivatives up to order ℓ exist and are continuous is denoted $\mathcal{S}^\ell(A, B)$. The class $\mathcal{S}^\infty(A, B)$ is the intersection of $\mathcal{S}^\ell(A, B)$ for all finite ℓ. A $\mathcal{S}^\infty$*-diffeomorphism* ϕ from a semi-algebraic open $U \subset \mathbb{R}^n$ to a semi-algebraic open $V \subset \mathbb{R}^n$ is a bijection from U to V such that $\phi \in \mathcal{S}^\infty(U, V)$ and $\phi^{-1} \in \mathcal{S}^\infty(V, U)$.

Let $\ell \geq 0$. A semi-algebraic $A \subset \mathbb{R}^n$ is a $\mathcal{S}^\infty$*-submanifold of* $\mathbb{R}^n$ *of dimension* ℓ if for every $x \in A$ there exists a semi-algebraic open U of $\mathbb{R}^n$ and an $\mathcal{S}^\infty$-diffeomorphism ϕ from U to a semi-algebraic open neighborhood V of x in $\mathbb{R}^n$ such that $\phi(0) = x$ and

$$\phi\Big(U \cap \big(\mathbb{R}^\ell \times \{0\}\big)\Big) = A \cap V,$$

where $\mathbb{R}^\ell \times \{0\} = \big\{(a_1, \ldots, a_\ell, 0, \ldots, 0) \in \mathbb{R}^n \bigm| (a_1, \ldots, a_\ell) \in \mathbb{R}^\ell\big\}$.

Lemma 3.1. *Let A be an open $\mathcal{S}^\infty$ manifold and $f \in \mathcal{S}^\infty(A, \mathbb{R}^m)$. Then there exists a semi-algebraic set $S \subseteq \mathbb{R}^m$ and semi-algebraic open set $U \subseteq A$ such that for all $y^0 \in S$, $\dim \big\{x \in U \bigm| f(x) - y^0 = 0\big\} = \dim A - m$. Furthermore, $\dim(\mathbb{R}^m \setminus S) < m$.*

Proof. Let A be an open $\mathcal{S}^\infty$ manifold and $f \in \mathcal{S}^\infty(A, \mathbb{R}^m)$. By the semi-algebraic version of Sard's Theorem [Bas03, Theorem 5.56, Section 9, pp. 192], the set C of critical values of f is a semi-algebraic set in $\mathbb{R}^m$ and $\dim(\mathbb{R}^m \setminus S) < m$. Let $S = \mathbb{R}^m \setminus C$ be its complement (which is a semi-algebraic set). For any $y^0 \in S$ there exists $x^0 \in A$ where $y^0 = f(x^0)$. Let $g\colon A \to \mathbb{R}^m$ be defined by $g(x) = f(x) - y^0$. Since $y^0 \notin C$, $\operatorname{rank} \mathrm{d}\, g(x^0) = m$ because f has full rank on a neighborhood of x^0. By the Constant Rank Theorem [Bas03, Theorem 5.57, Section 9, pp. 192] there exists a semi-algebraic open neighborhood U of x^0 in A where $\dim \big\{x \in U \bigm| f(x) - y^0 = 0\big\} = \dim \ker g = \dim A - \operatorname{rank} g = \dim A - m$. □

3.2 Proof of Main Result

We now have the machinery to present the proof of Theorem 1.34.

Proof of Theorem 1.34. Assume $f \in \mathbb{R}[x_1, \ldots, x_n]$ is non-zero. For notational purposes let $x = (x_1, \ldots, x_n)$. We will find a set S so that g is a routing function in the following manner. First, let $p = (p_1, \ldots, p_n)$ be the mapping where $p_i\colon A \subset \mathbb{R}^{n+1} \to \mathbb{R}$ is defined by

$$p_i(x, t) = -\partial_i f(x) t + x_i \tag{3.2}$$

and $A = \{(x, t) \in \mathbb{R}^n \times \mathbb{R} \mid t \neq 0 \text{ and } f(x) \neq 0\}$. Observe that A is an open $\mathcal{S}^\infty$ manifold of dimension $n + 1$ and $p \in \mathcal{S}^\infty(A, \mathbb{R}^n)$. By Lemma 3.1 there exists a semi-algebraic set $S_1 \subseteq \mathbb{R}^n$ and semi-algebraic open set $U_1 \subseteq A \subseteq \mathbb{R}^n \times \mathbb{R}$ such that for all $y \in S_1$, $\dim V_1 = \dim A - n = (n + 1) - n = 1$ where $V_1 = \{(x, t) \in U_1 \mid p(x, t) - y = 0\}$.

Let $y = (y_1, \ldots, y_n) \in S_1$. Define $q\colon B \subset \mathbb{R}^{n+1} \to \mathbb{R}$ to be

$$q(x, t) = \frac{(x_1 - y_1)^2 + \cdots + (x_n - y_n)^2 + 1}{t f(x)} \tag{3.3}$$

where $B = A \cap V_1$. Observe B is an open $\mathcal{S}^\infty$ manifold of dimension 1 and $q \in \mathcal{S}^\infty(B, \mathbb{R})$. From Lemma 3.1 we find a semi-algebraic set $S_{2,y} \subseteq \mathbb{R}$ and semi-algebraic open set $U_2 \subseteq B \subseteq \mathbb{R}^n \times \mathbb{R}$ such that for all $\tilde{y} \in S_{2,y}$, $\dim V_{2,y} = \dim B - 1 = 1 - 1 = 0$ where $V_{2,y} = \{(x,t) \in U_2 \mid q(x,t) - \tilde{y} = 0\}$.

We claim $S_{2,y} = \mathbb{R}$. From Lemma 3.1 we know

$$\mathbb{R} \setminus S_{2,y} = \{\text{critical values of } q\}.$$

For notational purposes let

$$W(x) = (x_1 - y_1)^2 + \cdots + (x_n - y_n)^2 + 1.$$

so $q(x,t) = \frac{W(x)}{tf(x)}$. Consider the system $\nabla q(x,t) = 0$:

$$\begin{bmatrix} \partial_{x_1} q(x,t) \\ \vdots \\ \partial_{x_n} q(x,t) \\ -\frac{W(x)}{f(x)t^2} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}.$$

For all $(x,t) \in B$, we have $f(x) \neq 0$, $t \neq 0$, and $W(x) \neq 0$, which leads us to conclude

$$-\frac{W(x)}{f(x)t^2} = 0$$

is not true. Hence, the mapping q has no critical points. Since the set of critical values of q is empty, $S_{2,y} = \mathbb{R}$.

Let $S = S_1$. Clearly $S \subset \mathbb{R}^n$ and S is semi-algebraic. The fact $\dim(\mathbb{R}^n \setminus S) < n$ follows directly from Lemma 3.1.

Let $c = (c_1, \ldots, c_n) \in S$, $\gamma \in S_{2,c} \setminus \{0\} = \mathbb{R} \setminus \{0\}$,

$$U(x) = (x_1 - c_1)^2 + \cdots + (x_n - c_n)^2 + 1$$

and

$$g(x) = \frac{f(x)^2}{U(x)^\gamma}.$$

Let $R = \{x \in \mathbb{R}^n \mid \nabla g(x) = 0 \text{ and } f(x) \neq 0\}$ denote the set of routing points of g. We claim R

is finite. Observe

$$\nabla g(x) = \frac{2f(x)\nabla f(x)U(x)^{\gamma} - \gamma f(x)^2 U(x)^{\gamma-1}\nabla U(x)}{U(x)^{2\gamma}} \tag{3.4}$$

$$= \frac{f(x)U(x)^{\gamma-1}\big[2\nabla f(x)U(x) - \gamma f(x)\nabla U(x)\big]}{U(x)^{2\gamma}} \tag{3.5}$$

$$= \frac{f(x)}{U^{\gamma+1}}\big[2\nabla f(x)U(x) - \gamma f(x)\nabla U(x)\big]. \tag{3.6}$$

Let

$$P(x) = \frac{f(x)}{U(x)^{\gamma+1}}$$
$$Q(x) = 2\nabla f(x)U(x) - \gamma f(x)\nabla U(x).$$

so $\nabla g(x) = P(x)Q(x)$. For all x, $P(x) \neq 0$, so $x \in R$ if and only if $Q(x) = 0$ and $f(x) \neq 0$. Let us rewrite $Q(x) = 0$ in the following way:

$$0 = 2\nabla f(x)U(x) - \gamma f(x)\nabla U(x)$$
$$\begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} 2\partial_{x_1} f(x)U(x) - 2\gamma f(x)(x_1 - c_1) \\ \vdots \\ 2\partial_{x_n} f(x)U(x) - 2\gamma f(x)(x_n - c_n) \end{bmatrix}$$
$$\begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} -\partial_{x_1} f(x)\frac{U(x)}{\gamma f(x)} + x_1 \\ \vdots \\ -\partial_{x_n} f(x)\frac{U(x)}{\gamma f(x)} + x_n \end{bmatrix}.$$

Let $t = \frac{U(x)}{\gamma f(x)}$ so

$$\begin{aligned} c_1 &= -\partial_{x_1} f(x)t + x_1 \\ &\vdots \\ c_n &= -\partial_{x_n} f(x)t + x_n \\ \gamma &= \frac{U(x)}{tf(x)} \end{aligned} \tag{3.7}$$

Since $\gamma \neq 0$, $x \in R$ if and only if x satisfies (3.7) and $f(x) \neq 0$. Using our previous notation, rewrite (3.7) as

$$\begin{aligned} 0 &= p_1(x,t) - c_1 \\ &\vdots \\ 0 &= p_n(x,t) - c_n \\ 0 &= q(x,t) - \gamma. \end{aligned} \tag{3.8}$$

Thus, when $(c_1, \dots, c_n) \in S$ and $\gamma \neq 0$, $x \in R$ if and only if x satisfies (3.8) and $f(x) \neq 0$. Suppose now that $x \in R$. It follows that $t = \frac{U(x)}{\gamma f(x)} \neq 0$ and $q(x_1, \dots, x_n, t) - \gamma = 0$, implying $(x_1, \dots, x_n, t) \in V_{2,c}$. As shown earlier, $\dim V_{2,c} = 0$. Combining this with the fact that $R \times (t \neq 0) \subset V_{2,c}$ implies $\dim R = 0$. The set R is finite because R is semi-algebraic and has dimension zero.

We now show the routing points of g are nondegenerate. From (3.6) we see

$$(\operatorname{Hess} g)(x) = JP(x)Q(x) + P(x)JQ(x)$$

where JP is the jacobian of P. When we evaluate $\operatorname{Hess} g$ at a point $x \in R$,

$$(\operatorname{Hess} g)(x) = JP(x)Q(x) + P(x)JQ(x) = P(x)JQ(x).$$

Hence

$$\det(\operatorname{Hess} g)(x) = \det\big(P(x)JQ(x)\big) = P(x)^n \det JQ(x).$$

Clearly $P(x) \neq 0$. When $(c_1, \dots, c_n) \in S$ then $(c_1, \dots, c_n)$ is not a critical value of p. Also γ is not a critical value of q. Thus $\det JQ(x) \neq 0$. It follows $\det(\operatorname{Hess} g)(x) \neq 0$ as desired.

What we have shown so far is that if $(c_1, \dots, c_n) \in S$ and $\gamma \neq 0$, then the function

$$g = \frac{f^2}{\big((x_1 - c_1)^2 + \cdots + (x_n - c_n)^2 + 1\big)^\gamma}$$

has finitely many routing points that are all nondegenerate. The choice of $\gamma = \deg(f) + 1$ guarantees the function g vanishes at infinity (property two) because the degree of the numerator is smaller than the degree of the denominator. Certainly the function g is nonnegative. To understand why the first derivative of g is bounded, we observe in (2.25) that each component of ∇g is a rational function where the degree of the numerator is smaller than the degree of the denominator, which is nonnegative. A similar argument holds for each component of $\operatorname{Hess} g$.

Hence the function g is a routing function, as desired. $\square$

Before we present the termination proof, we make a small remark. In a careful reading of the previous proof, one will observe that the set S is explicitly found. For a given polynomial f, S was chosen as

$$S = \mathbb{R}^n \setminus \{\text{critical values of } p\} \tag{3.9}$$

where $p = (p_1, \ldots, p_n)$, $p_i \colon \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$, $p_i(x,t) = -\partial_{x_i} f(x)t + x_i$ and each of critical points (x,t) of p must satisfy $f(x) \neq 0$ and $t \neq 0$. This explicit construction allows us to visualize the "bad" set of parameters for which g in (1.35) may not be a routing function. We illustrate this idea in the following example.

Example 3.10. Let us suppose f takes the form (1.9) in our toy example from Example 1.8. In this particular example, $p = (p_1, p_2)$ and

$$p_1 = x_1\Big(1 - t\left(x_1^2 + x_2^2 - 1\right)\Big),$$
$$p_2 = x_2\Big(1 - t\left(x_1^2 + x_2^2 - 1\right)\Big).$$

One can compute the critical points of p to be

$$\left\{(x,t) \in \mathbb{R}^n \times \mathbb{R} \;\middle|\; \left(x_1^2 + x_2^2 = 1 \wedge t = \frac{1}{2}\right) \vee \left(x_1^2 + x_2^2 = 1 \wedge t = \frac{1}{x_1^2 + x_2^2 - 1} \wedge f \neq 0\right)\right\}.$$

We visualize the set of critical points in Figure 3.12a as the red surface and red curve excluding the black dashed curve and black point. The critical values of p are shown in Figure 3.12b as the red curve and red point. Hence the white region in Figure 3.12b is the set S. By choosing a (c_1, c_2) value outside of the red in Figure 3.12b; that is, by choosing $(c_1, c_2) \in S$, Theorem 1.34 guarantees that

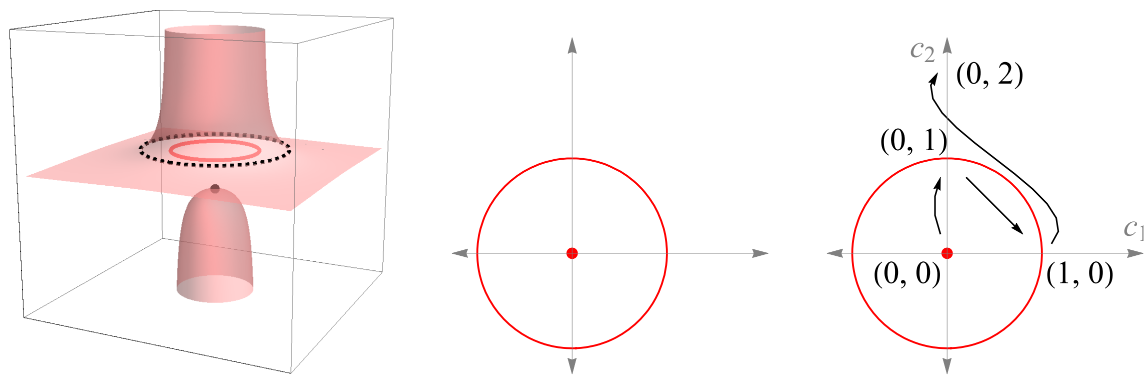
$$g = \frac{f^2}{\left((x_1 - c_1)^2 + (x_2 - c_2)^2 + 1\right)^5} \tag{3.11}$$

is a routing function. In Figure 3.12c we see that after three perturbations using graded lexicographic order, we arrive at a $(c_1, c_2) = (0, 2) \in S$. Therefore, we can safely assume

$$g = \frac{f^2}{\left(x_1^2 + (x_2 - 2)^2 + 1\right)^5}$$

is a routing function.

In Chapter 1.3 we ran **Connectivity** with f as input in Example 1.12. For that run, only



(a) (b) (c)

Figure 3.12 Illustration of how to avoid the "bad" set of parameters.

one perturbation was necessary to find a routing function g. The choice of $(c_1, c_2) = (0, 1)$ was sufficient because the function

$$g = \frac{f^2}{\left(x_1^2 + (x_2 - 1)^2 + 1\right)^5}$$

was a routing function. Interestingly, $(c_1, c_2) = (0, 1) \in \mathbb{R}^n \setminus S$. This example seems to indicate that the set of "bad" parameters may be even "smaller" than what we determine in Theorem 1.34.

We now present the termination proof for **Connectivity**.

Theorem 3.13. *Algorithm **Connectivity** terminates.*

Proof. Let f, p, q be the inputs to **Connectivity** satisfying the specification. To show Algorithm **Connectivity** terminates, first we must show that the loop in step 2 terminates in a finite number of iterations. Let S be the semi-algebraic set from Theorem 1.34 for the given f. According to Theorem 1.34 the set of choices for $(c_1, \ldots, c_n)$ for which

$$g = \frac{f^2}{\left((x_1 - c_1)^2 + \cdots + (x_n - c_n)^2 + 1\right)^{\deg(f)+1}}$$

is not a routing function is "small" since $\dim(\mathbb{R}^n \setminus S) < n$; that is, $\mathbb{R}^n \setminus S$ is contained in a Zariski closed set of dimension strictly less than n. Hence, after a finite number of perturbations on the integer grid, we are guaranteed to find a parameter $(c_1, \ldots, c_n) \in S$ which will guarantee g is a routing function.

Let $(c_1, \ldots, c_n) \in S$ and

$$\begin{aligned} \gamma &= \deg(f) + 1, \\ U &= (x_1 - c_1)^2 + \cdots + (x_n - c_n)^2 + 1, \\ V(\mathcal{F}) &= \left\{2 \cdot (\partial_{x_i} f) \cdot U - \gamma \cdot f \cdot (\partial_{x_i} U)\right\}_{i=1}^{n}. \end{aligned}$$

We claim $V(\mathcal{F})$ is zero-dimensional. As mentioned previously, $V(\mathcal{F})$ is the union of the set of routing points of g along with the singular points of f. Since g is a routing function, it has finitely many routing points. Combined with the fact that f has finitely many singular points by assumption, then $V(\mathcal{F})$ must be zero-dimensional. We see that the loop terminates because each of the finitely many routing points of g, the set of points $r \in V(\mathcal{F})$ where $f(r) \neq 0$, are nondegenerate.

The rest of the algorithm terminates because there are finitely many routing points, the Hessian at each of these routing points has finitely many outgoing eigenvectors, and the algorithm **Destination** terminates. □

Chapter 4

Length Bound

For a routing function g, we give an upper bound on the length of a connectivity path connecting any two points in a semi-algebraically connected component of $\{g \neq 0\}$ in the form of Theorem 1.47. The proof relies on several preliminary lemmas which we give in the first section. A first step in the proof of our upper bound is to show the existence of an upper bound on the length of a single trajectory of ∇g. Such an argument is given in the first subsection for the case when the trajectory is contained in a unit ball. We then extend this result to bound the length of a trajectory contained in any ball. A second step in the proof, given in the second subsection, is to identify a ball containing the connectivity path for any two given points. In the second section of this chapter we prove Theorem 1.47.

4.1 Preliminaries

In this section we present several notions and preliminary lemmas used in the proof of Theorem 1.47. The first subsection gives a bound on the length of trajectories in a given ball. The second subsection gives a bound on the radius of a ball containing a connectivity path.

4.1.1 Bound on Trajectory Length in a Ball

In this subsection we will give an upper bound on the length of a single trajectory restricted to a ball. We must restrict the trajectory to a ball, otherwise we could have trajectories of infinite length. We use the idea of D'Acunto and Kurdyka [DK04] of comparing the length of a trajectory to a length of the "thalweg" of this function — the locus of points where the level sets are the most far apart — which has the advantage of being semi-algebraic. See [CM12; DK05; DK06] for other applications of this idea. One can show that this thalweg is contained in an algebraic set having dimension 1. The length of this algebraic curve can be estimated via the Cauchy-Crofton formula, by counting intersection points with a generic hyperplane. This length then can be used to give the upper bound we desire. Once a bound on the trajectory length is found in a unit ball, an easy translation can be made to that of any ball of radius r.

Suppose g is a C^1 function and consider a C^1 curve Ω having the following property: for all $x \in \Omega$ and for all $y \in g^{-1}\big(g(x)\big)$ we have $\|\nabla g(x)\| \leq \|\nabla g(y)\|$. For all $x \in \mathbb{R}^n$, the fiber $g^{-1}\big(g(x)\big)$ is a level set of g; that is,

$$g^{-1}\big(g(x)\big) = \{p \in \mathbb{R}^n \mid g(p) = g(x)\}\,.$$

The curve Ω is the set of points where the gradient norm is smallest along the contour. In the picture below we see that the curve Ω travels between level sets that are furthest apart because that is when the slope is the shallowest.

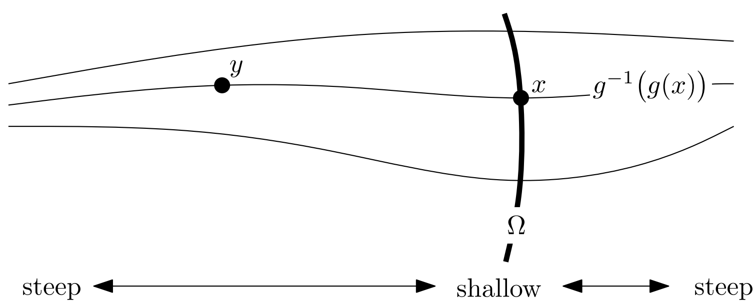


Figure 4.1 Illustration of Ω curve.

We give a specific name to the curve Ω.

Definition 4.2. [DK05] For a function $g\colon \mathbb{R}^n \to \mathbb{R}$, we say that a point $x \in \mathbb{R}^n$ belongs to the *ridge and valley set of* g if the function $\|\nabla g\|^2$ restricted to $g^{-1}\big(g(x)\big)$ has a local minimum at x. We denote by $\Omega(g)$ the ridge and valley set of g.

The terminology "ridge and valley lines" used here are motivated by its analogy with the geographic thalweg, the line of lowest elevation within a valley, and the ridges and valleys that appear in the Earth's landscape. Under certain mild assumptions, if the ridge and valley set is a curve, then it is longer than a given trajectory. Let D be an open subset of $\mathbb{R}^n$ and let $g\colon \mathbb{R}^n \to \mathbb{R}$ be a C^2 function in some neighborhood of $\overline{D}$. Suppose $\Omega(g) \subset \overline{D}$ is a C^1 curve. We assume that for each $t \in g(D)$, the set $g^{-1}(t) \cap \Omega(g)$ consists of exactly one point and that, for all but finitely many $t \in f(D)$, the curve $\Omega(g)$ is transverse to $g^{-1}(t)$.

Lemma 4.3. *[DK04, Lemma 7.9] Let $X \subset D$ be the image of a trajectory of ∇g, then* $\operatorname{Length}(X) \leq \operatorname{Length}\big(\Omega(g)\big)$.

A careful reading of the proof of Lemma 4.3 gives a more precise result.

Lemma 4.4. *[DK04] Let $a, b \in \mathbb{R}$ with $a < b$. Let D_i, $i \in I$ be all connected components of $g^{-1}\big((a,b)\big)$ and let $\lambda_i \colon [\alpha_i, \beta_i] \to D_i$ be a trajectory of ∇g in D_i and X_i be its image. Then*

$$\operatorname{Length}(X_i) \leq \operatorname{Length}\big(\Omega(g) \cap D_i\big).$$

In particular,

$$\sum_{i \in I} \operatorname{Length}(X_i) \leq \operatorname{Length}\Big(\Omega(g) \cap g^{-1}\big((a,b)\big)\Big).$$

We wish to estimate the length of the curve $\Omega(g)$. The approach taken by Kurdyka and D'Acunto [DK04] uses the observation that $\Omega(g)$ is contained in the ridge and valley set of g.

Remark 4.5. Observe that $\nabla\left(\|\nabla g\|^2\right) = 2(\operatorname{Hess} g) \cdot \nabla g$. Hence $\Theta(g)$ is the set of points where the function $\|\nabla g\|^2$ restricted to $g^{-1}\big(g(x)\big)$ has a critical point at x. Thus we can study $\Theta(g)$ by looking at 2×2 minors of the matrix

$$\begin{bmatrix} 2(\operatorname{Hess} g)(x) \cdot \nabla g(x) \\ \nabla g(x) \end{bmatrix}.$$

From the previous remark, we see $\Omega(g)$ is contained in $\Theta(g)$. The dimension of $\Theta(g)$ is not always equal to 1 as we saw in Example 1.45. However, we will be assuming that $\Theta(g)$ is a compact rectifiable curve when restricted to a closed ball. Knowing that $\Theta(g)$ is a curve and that it contains $\Omega(g)$ will allow us to write the bounds in Lemmas 4.3 and 4.4 in terms of $\Theta(g)$. Furthermore, as $\Theta(g)$ is algebraic, we can estimate its length using the Cauchy-Crofton formula.

Lemma 4.6 (Cauchy-Crofton formula [CM12; Cro68; DK04; Don96; Fed96; San04])**.** *Let Θ be a compact rectifiable curve, and let $\mathcal{H}$ be the set of affine hyperplanes in $\mathbb{R}^n$. Let $i(\Theta, H)$ denote the cardinality of $\Theta \cap H$. There exists a normalization $\mathrm{d}\,\mu$ of the canonical measure $\mathrm{d}\,\widetilde{\mu}$ on $\mathcal{H}$ such that the length of Θ can be expressed by the following formula:*

$$\operatorname{Length}(\Theta) = \int_{\mathcal{H}} i(\Theta, H)\ \mathrm{d}\,\mu.$$

Let us denote the open n-ball centered at x with radius r using the notation

$$B_n(x,r) = \left\{ y \in \mathbb{R}^n \;\middle|\; \|y-x\| < r^2 \right\}.$$

Let $\mathbb{B}^n = B_n(0,1)$.

Remark 4.7. According to [Don96], the set of hyperplanes which meet the closed ball $\overline{\mathbb{B}^n}$ is compact, so has finite volume V, say. If Θ is the intersection of $\overline{\mathbb{B}^n}$ with a real algebraic curve of degree δ, then the intersection number $i(\Theta, H)$ is at most δ (almost everywhere) and it follows that the length of Θ is at most $V\delta$ where

$$V = \int_{\mathcal{H}_1} \mathrm{d}\,\mu,$$

and $\mathcal{H}_1$ is the set of affine hyperplanes that cut the unit ball. One can compute V [DK04; Fed96], the μ-volume of the set of affine hyperplanes having a non-empty intersection with the closed unit ball, to be

$$V := \nu(n) = 2\Gamma\left(\frac{1}{2}\right)\Gamma\left(\frac{n+1}{2}\right)\Gamma\left(\frac{n}{2}\right)^{-1} \leq 2n \tag{4.8}$$

where Γ is the Euler gamma function.

Lemma 4.9. *Suppose $f \in \mathbb{Z}[x_1,\dots,x_n]$ with $n \geq 2$ and degree $d \geq 2$ and $(c_1,\dots,c_n) \in \mathbb{Z}^n$ such that*

$$g = \frac{f^2}{\big((x_1-c_1)^2 + \cdots + (x_n-c_n)^2 + 1\big)^{d+1}}$$

is a routing function. Suppose $\Theta(g) \cap \overline{\mathbb{B}^n}$ is a compact rectifiable curve. The length of any trajectory of ∇g in $\mathbb{B}^n$ is bounded by

$$2n(6d+4)^{n-1}.$$

Proof. Suppose $f \in \mathbb{Z}[x_1,\dots,x_n]$ with $n \geq 2$ and degree $d \geq 2$ and $(c_1,\dots,c_n) \in \mathbb{Z}^n$ such that

$$g = \frac{f^2}{\big((x_1-c_1)^2 + \cdots + (x_n-c_n)^2 + 1\big)^{d+1}}$$

is a routing function. We will estimate the length of a trajectory of ∇g in the closed unit ball by estimating the length of the ridge and valley set of g restricted to the closed unit ball; that is

$$\widetilde{\Omega}(g) = \left\{ x \in \overline{\mathbb{B}^n} \;\middle|\; \|\nabla g\|^2 \text{ has a local minimum at } x \in g^{-1}\big(g(x)\big) \cap \overline{\mathbb{B}^n} \right\}.$$

Observe that the fibers $g^{-1}(t)$ are compact for each $t > 0$ because they are closed and they are bounded (since g is bounded by Lemma 2.6). Hence for any $t > 0$, the minimum of $\|\nabla g\|^2$ restricted to the hypersurface $g^{-1}(t) \cap \overline{\mathbb{B}^n}$ is reached inside $\overline{\mathbb{B}^n}$. It follows that $\Omega(g)$ restricted to the closed unit ball is contained in the gradient extremal of g restricted to the closed unit ball; that is,

$$\widetilde{\Omega}(g) \subseteq \widetilde{\Theta}(g) = \left\{x \in \overline{\mathbb{B}^n} \;\middle|\; \exists \lambda \in \mathbb{R},\ (\operatorname{Hess} g)(x) \cdot \nabla g(x) = \lambda \nabla g(x)\right\}.$$

We will compare the length of a trajectory of ∇g in the closed unit ball to the length of $\widetilde{\Theta}(g)$. To bound the length of $\widetilde{\Theta}(g)$, we will use the Cauchy-Crofton formula. To do so, we must calculate the the number of points of intersection of a generic affine hyperplane with $\widetilde{\Theta}(g)$.

First, we claim that for a generic affine hyperplane H, the set $H \cap \widetilde{\Theta}(g)$ has at most $(6d+4)^{n-1}$ points. According to Remark 4.5, we need only look at the 2×2 minors of

$$\begin{bmatrix} 2(\operatorname{Hess} g)(x) \cdot \nabla g(x) \\ \nabla g(x) \end{bmatrix}. \tag{4.10}$$

Write $g(x) = \frac{f(x)^2}{U(x)^\gamma}$. Then

$$\nabla g(x) = \frac{f(x)\left[2\nabla f(x) U(x) - \gamma f(x) \nabla U(x)\right]}{U(x)^{\gamma+1}} = \frac{f(x)P(x)}{Q(x)}$$

and each component is a rational function whose numerator $f(x)P_i(x)$ has degree at most $2d+1$ where $P = (P_1, \ldots, P_n)$. Furthermore, if $H = (\operatorname{Hess} g)(x)$ then

$$H_{ij} = \frac{\left[\partial_j f(x) P_i(x) + f(x)\partial_j P_i(x)\right] Q(x) - \left[f(x)P_i(x)\right]\partial_j Q(x)}{U(x)^{2\gamma+2}}$$

where $\partial_j f(x)$ is the partial derivative of f with respect to x_j. We see the numerator of H_{ij} is a polynomial of degree at most $2d+2$.

The zero set of the first $n-1$ minors of (4.10) define $\widetilde{\Theta}(g)$, which is a compact rectifiable curve. This is equivalent to a system of $n-1$ polynomial equations, each having degree at most $(2d+1) + (2d+2) + (2d+1) = 6d+4$.

Bezout's Theorem states that if an algebraic curve is contained in $\mathbb{R}^n$ is given by $n-1$ polynomial equations $p_1 = \cdots = p_n = 0$ where p_i is a polynomial of degree d_i, then the number of points of intersection with a generic affine hyperplane of $\mathbb{R}^n$ is bounded by $d_1 \ldots d_{n-1}$. Applying this result to $\widetilde{\Theta}(g)$, the maximum number of points of intersection is bounded by $(6d+4)^{n-1}$ as desired.

Let ϕ be a trajectory of ∇g whose image X is contained in $\overline{\mathbb{B}^n}$. According to Lemma 4.3, $\operatorname{Length}(X) \leq \operatorname{Length}\left(\widetilde{\Omega}(g)\right)$. As $\widetilde{\Theta}(g)$ contains $\widetilde{\Omega}(g)$, it suffices to find a bound on the length of $\widetilde{\Theta}(g)$ to bound the length of X. According to Remark 4.7, we may apply the Cauchy-Crofton formula to $\widetilde{\Theta}(g)$ to find

$$\operatorname{Length}(X) \leq \nu(n)(6d+4)^{n-1} \leq 2n(6d+4)^{n-1}$$

as desired. □

We can extend the results to any ball of radius r in the following way. Let B denote a ball centered at x_0 of radius r. Suppose g is a routing function and ϕ is a trajectory of ∇g whose image X is in B. Define the mapping $T\colon \mathbb{B}^n \to B$ by $T(X) = x_0 + rX$ and define the function $h = g \circ T$. There exists a trajectory α of ∇h whose image Y is contained in $\mathbb{B}^n$ and $\alpha(t) = T^{-1}\big(\phi(t)\big)$. We observe that $\phi'(t) = T'\big(\alpha(t)\big)\alpha'(t) = r\alpha'(t)$. Hence $\operatorname{Length}(X) = r\operatorname{Length}(Y)$. This allows us to rewrite the previous lemmas like so.

Lemma 4.11. *Suppose $f \in \mathbb{Z}[x_1, \ldots, x_n]$ with $n \geq 2$ and degree $d \geq 2$ and $(c_1, \ldots, c_n) \in \mathbb{Z}^n$ such that*

$$g = \frac{f^2}{\big((x_1 - c_1)^2 + \cdots + (x_n - c_n)^2 + 1\big)^{d+1}}$$

is a routing function. Let B be a closed n-ball of radius r. Suppose $\Theta(g) \cap B$ is a compact rectifiable curve. The length of any trajectory of ∇g in B is bounded by

$$2nr(6d+4)^{n-1}.$$

Lemma 4.12. *Suppose $f \in \mathbb{Z}[x_1, \ldots, x_n]$ with $n \geq 2$ and degree $d \geq 2$ and $(c_1, \ldots, c_n) \in \mathbb{Z}^n$ such that*

$$g = \frac{f^2}{\big((x_1 - c_1)^2 + \cdots + (x_n - c_n)^2 + 1\big)^{d+1}}$$

is a routing function. Let B be a closed n-ball of radius r. Suppose $\Theta(g) \cap B$ is a compact rectifiable curve. Let $a, b \in \mathbb{R}$ with $a < b$. Let D_i, $i \in I$ be all connected components of $g^{-1}\big((a,b)\big) \cap B$ and let $\lambda_i\colon [\alpha_i, \beta_i] \to D_i$ be a trajectory of ∇g in D_i and X_i be its image. Then

$$\operatorname{Length}(X_i) \leq \operatorname{Length}\big(\Omega(g) \cap D_i\big).$$

In particular,

$$\sum_{i\in I} \mathrm{Length}(X_i) \leq \mathrm{Length}\Big(\Omega(g) \cap B \cap g^{-1}\big((t,s)\big)\Big) \leq \mathrm{Length}\big(\Omega(g) \cap B\big) \leq 2nr(6d+4)^{n-1}.$$

4.1.2 Ball Enclosing Connectivity Path

The second step to bounding the length of a connectivity path between two points is to identify a ball containing the connectivity path. Such a bound will be found by bounding the level sets of g. To calculate our bounds, we use bounds on polynomial heights. A careful reading of the proofs of [HS00, Appendix B, Proposition B.7.2, pp. 226] give the following result on the height of sums and products of polynomials, which we state for completeness.

Lemma 4.13. *If $P_1, \dots, P_r \in \mathbb{Z}[x_1, \dots, x_n]$, then*

$$\begin{aligned} \mathrm{hgt}(P_1 + \cdots + P_r) &\leq r \max\big\{\mathrm{hgt}(P_1), \dots, \mathrm{hgt}(P_r)\big\} \\ \mathrm{hgt}(P_1 \cdots P_r) &\leq 2^{\deg(P_1\cdots P_r)+n(r-1)}\, \mathrm{hgt}(P_1)\cdots \mathrm{hgt}(P_r). \end{aligned}$$

Throughout this subsection we let $f \in \mathbb{Z}[x_1, \dots, x_n]$, $n \geq 2$, degree $d \geq 2$ with no singular points and suppose $(c_1, \dots, c_n) \in \mathbb{Z}^n$ such that

$$g = \frac{f^2}{\big((x_1-c_1)^2 + \cdots + (x_n-c_n)^2 + 1\big)^{d+1}}$$

is a routing function. We also assume $H = \mathrm{hgt}(f)$.

Lemma 4.14. *Suppose $\varepsilon \in \mathbb{Q}$ is given as an irreducible fraction $\varepsilon = A_1/A_2$ with $A_1, A_2 > 0$. There exists a ball, centered at the origin, of radius*

$$n\big(120A_1A_2Hd(c_1^2 + \cdots + c_n^2 + 1)\big)^{4n^3(6d)^{3n}}$$

containing $\{g = \varepsilon\}$.

Proof. Suppose $\varepsilon \in \mathbb{Q}$ is given as an irreducible fraction $\varepsilon = A_1/A_2$ with $A_1, A_2 > 0$. Let $Q(x) = A_2 f(x)^2 - A_1 U(x)^{d+1}$ where $U(x) = (x_1-c_1)^2 + \cdots + (x_n-c_n)^2 + 1$. Observe that

$$\{x \in \mathbb{R}^n \mid Q(x) = 0\} = \left\{x \in \mathbb{R}^n \;\middle|\; \frac{f(x)^2}{U(x)^{d+1}} = \frac{A_1}{A_2}\right\} = \{x \in \mathbb{R}^n \mid g(x) = \varepsilon\}.$$

The level set $\{g = \varepsilon\}$ is bounded (Lemma 2.7) and $Q \in \mathbb{Z}[x_1, \dots, x_n]$ has degree $2d+2$, so according to [BR10, Section 2.2, Theorem 1, pp. 1272] there exists a ball, centered at the origin, of radius

$$R = n^{1/2}(N+1)2^{ND(\beta+\mathrm{bit}(N)+\mathrm{bit}(2d+3)+3)} \tag{4.15}$$

containing $\{g = \varepsilon\}$, where

$$\begin{aligned} N &= (2d+3)(2d+2)^{n-1}, \\ D &= n(2d+1)+2, \end{aligned}$$

and β is an upper bound on the bitsizes of the coefficients of Q. We wish to simplify our radius bound in (4.15). Observe that for all $x > 0$, $\mathrm{bit}(x) = \lceil \log_2 x \rceil \le 1 + \log_2 x$. Let τ denote the height of Q. Then $\mathrm{bit}(\tau) = \beta$ is an upper bound on the bitsizes of the coefficients of Q and

$$\begin{aligned} \beta &\le 1 + \log_2 \tau, \\ \mathrm{bit}(N) &\le 1 + \log_2 N, \\ \mathrm{bit}(2d+3) &\le 1 + \log_2(2d+3). \end{aligned}$$

Using these inequalities, we simplify the bound R in (4.15) to

$$\begin{aligned} R &\le n^{1/2}(N+1)2^{ND\left(6+\log_2\left(\tau N(2d+3)\right)\right)} \\ &= n^{1/2}(N+1)2^{6ND}\big(\tau N(2d+3)\big)^{ND}. \end{aligned} \tag{4.16}$$

Using the inequalities

$$\begin{aligned} N &\le 3^n(d+1)^n, \\ D &\le 2n(d+1), \\ ND &\le n(3d+3)^{n+1}, \end{aligned}$$

we update our bound from (4.16) to be

$$\begin{aligned} R &\le n^{1/2}(N+1)\big(2\tau N(2d+3)\big)^{ND} \\ &\le n3^n(d+1)^n\big(2\tau 3^n(d+1)^n(2d+3)\big)^{n(3d+3)^{n+1}}. \end{aligned} \tag{4.17}$$

Using very pessimistic upper bounds, we simplify (4.17) further to

$$R \le n3^n(d+1)^n\big(2\tau 3^n(d+1)^n(2d+3)\big)^{n(3d+3)^{n+1}} \le n(60d\tau)^{4n^2(6d)^{n+1}}. \tag{4.18}$$

We wish to find an upper bound on τ. From Lemma 4.13 we find

$$\begin{aligned}\tau &= \operatorname{hgt}(Q)\\ &\leq 2\max\left\{\operatorname{hgt}\left(A_2 f^2\right), \operatorname{hgt}\left(A_1 U^{d+1}\right)\right\}\\ &= 2\max\left\{A_2 \operatorname{hgt}\left(f^2\right), A_1 \operatorname{hgt}\left(U^{d+1}\right)\right\}.\end{aligned} \tag{4.19}$$

We now will calculate bounds on the height of f^2 and U^{d+1}. Using Lemma 4.13 we find

$$\operatorname{hgt}(f^2) \leq 2^{\deg\left(f^2\right)+(2-1)n} H^2 = H^2 2^{2d+n} \tag{4.20}$$

and

$$\operatorname{hgt}(U) = \max\left\{1, |-2c_1|, \ldots, |-2c_n|, c_1^2 + \cdots + c_n^2 + 1\right\} = c_1^2 + \cdots + c_n^2 + 1 \tag{4.21}$$

so

$$\begin{aligned}\operatorname{hgt}\left(U^{\gamma}\right) &\leq 2^{\deg\left(U^{d+1}\right)+nd} \operatorname{hgt}(U)^{d+1}\\ &\leq 2^{d(n+2)+2}\left(c_1^2 + \cdots + c_n^2 + 1\right)^{d+1}.\end{aligned} \tag{4.22}$$

We calculate an upper bound on τ using (4.19), (4.20), and (4.22) to be

$$\begin{aligned}\tau &\leq 2\max\left\{A_2 H^2 2^{2d+n}, A_1 2^{d(n+2)+2}\left(c_1^2 + \cdots + c_n^2 + 1\right)^{d+1}\right\}\\ &\leq A_1 A_2 H^2 2^{d(n+2)+3}\left(c_1^2 + \cdots + c_n^2 + 1\right)^{d+1}.\end{aligned} \tag{4.23}$$

Combining (4.18) with (4.23), we find

$$\begin{aligned}R &\leq n\left(60 d A_1 A_2 H^2 2^{d(n+2)+3}\left(c_1^2 + \cdots + c_n^2 + 1\right)^{d+1}\right)^{4n^2(6d)^{n+1}}\\ &\leq n\left(120 A_1 A_2 H d(c_1^2 + \cdots + c_n^2 + 1)\right)^{4n^3(6d)^{3n}}.\end{aligned}$$

$\square$

Our goal now is to put a ball around the level set of g corresponding to the routing point that is lowest in height. More precisely, we want to put a ball around $\{g = M\}$ where $M = \min_{r\in R} g(r)$ and R is the set of routing points of g. For each routing point r, $g(r) > 0$, so we expect $M > 0$. We can then apply the previous lemma to find such a ball.

Lemma 4.24. *Let R be the set of routing points of g and $M = \min_{r\in R} g(r)$. Then*

$$M \geq \left(\left(2dH\left(c_1^2 + \cdots + c_n^2 + 2\right)\right)^{104n^3(5d)^{5n}}\right)^{-1}$$

Proof. Let R be the set of routing points of g and $M = \min_{r\in R} g(r)$. The set of routing points is defined to be

$$R = \{x \in \mathbb{R}^n \mid \nabla g(x) = 0 \wedge g(x) \neq 0\}.$$

The gradient of g is

$$\nabla g = \frac{f}{U^{d+2}}\big(2\nabla f U - (d+1) f \nabla U\big).$$

Hence

$$R = \{x \in \mathbb{R}^n \mid 2\nabla f(x)U(x) - (d+1)f(x)\nabla U(x) = 0 \wedge f(x) \neq 0\}$$

because for all x, $U(x) \neq 0$ and $g(x) = 0$ if and only if $f(x) = 0$. Let

$$\mathcal{F} = \left\{2(\partial_{x_i} f)U - (d+1)f(\partial_{x_i} U)\right\}_{i=1}^n.$$

The set $V(\mathcal{F})$ is the zero-locus in $\mathbb{R}^n$ of the polynomials in $\mathcal{F}$. It is the union of the set of routing points of g and the singular points of f. The function f has no singular points by assumption, so $V(\mathcal{F})$ is exactly the set of routing points of g. We are interested in finding a bound on the minimum value of $g(x)$ where $x \in V(\mathcal{F})$. If z is this minimum value, then $z = g(x)$ for some $x \in V(\mathcal{F})$. Furthermore, $g(x) = z$ if and only if $f(x)^2 - zU(x)^{d+1} = 0$. Hence, it suffices to find a lower bound on $|z|$ where $(x, z) \in \mathbb{R}^n \times \mathbb{R}$ is a solution to the polynomial system with $n + 1$ equations

$$\begin{aligned} 2\nabla f(x)U(x) - (d+1)f(x)\nabla U(x) &= 0 \\ f(x)^2 - zU(x)^{d+1} &= 0. \end{aligned} \tag{4.25}$$

Let $\mathcal{P} = \mathcal{F} \cup \{f^2 - zU\}$ be a family of $n + 1$ polynomials. The set $V(\mathcal{P})$ is zero-dimensional because g has finitely many routing points. Let $(x, z) \in V(\mathcal{P})$. According to [Emi10, Theorem 3, Section 2, pp. 4],

$$|z| \geq (2^D \rho C)^{-1} := A^{-1} \tag{4.26}$$

where

$$
\begin{aligned}
D &\leq (n+1)(2d+3)^{2(n+1)},\\
\rho &\leq 2^{(n+1)(2d+3)^{(n+1)-1}}(2d+3)^{(n+1)^2(2d+3)^{(n+1)-1}},\\
C &\leq 2^{(n+1)((2d+3)\tau)^{(n+1)-1}},
\end{aligned}
\tag{4.27}
$$

and β is a bound on the maximum bitsize of the coefficients of polynomials in $\mathcal{P}$. We wish to simplify our bound in (4.26). We do so by simplifying A. Observe that for all $x > 0$, $\operatorname{bit}(x) = \lceil \log_2 x \rceil \leq 1 + \log_2 x$. Let τ denote the height of $\mathcal{P}$. Then $\operatorname{bit}(\tau) = \beta$ is an upper bound on the maximum bitsize of the coefficients of polynomials in $\mathcal{P}$ and $\beta \leq 1 + \log_2 \tau$. Using this inequality and the inequalities from (4.27) we find

$$
\begin{aligned}
A &\leq 2^{(n+1)(2d+3)^{2(n+1)}} 2^{(n+1)(2d+3)^n}(2d+3)^{(n+1)^2(2d+3)^n}(2\tau)^{(n+1)(2d+3)^n}\\
&\leq \big(8\tau(2d+3)\big)^{(n+1)^2(2d+3)^{2n+2}}\\
&\leq (40\tau d)^{(2n)^2(5d)^{4n}}
\end{aligned}
\tag{4.28}
$$

We will now calculate an upper bound on τ. To do so, we must first calculate the height of each of the $n+1$ polynomials in $\mathcal{P}$. Using (4.21) and Lemma 4.13, we find for all $1 \leq i \leq n$,

$$
\begin{aligned}
\operatorname{hgt}\big((\partial_{x_i} f)U\big) &\leq 2^{\deg\left((\partial_{x_i} f)U\right)+n(2-1)} \operatorname{hgt}(\partial_{x_i} f)\operatorname{hgt}(U)\\
&\leq 2^{(d+1)+n} dH(c_1^2 + \cdots + c_n^2 + 1)
\end{aligned}
$$

and

$$
\begin{aligned}
\operatorname{hgt}\big(f(\partial_{x_i} U)\big) &\leq 2^{\deg\left(f(\partial_{x_i} U)\right)+n(2-1)} \operatorname{hgt}(f)\operatorname{hgt}(\partial_{x_i} U)\\
&\leq 2^{(d+1)+n} H \max\big\{2, -|2c_1|, \ldots, -|2c_n|\big\},
\end{aligned}
$$

hence

$$
\begin{aligned}
\operatorname{hgt}\big(2(\partial_{x_i} f)U - (d+1)f(\partial_{x_i} U)\big) &\leq 2\max\left\{\operatorname{hgt}\big(2(\partial_{x_i} f)U\big), \operatorname{hgt}\big((d+1)f(\partial_{x_i} U)\big)\right\}\\
&= 2\max\left\{2\operatorname{hgt}\big((\partial_{x_i} f)U\big), (d+1)\operatorname{hgt}\big(f(\partial_{x_i} U)\big)\right\}\\
&\leq (d+1)H2^{(d+1)+n+2}(c_1^2 + \cdots + c_n^2 + 2).
\end{aligned}
\tag{4.29}
$$

Now, we use (4.20) and apply Lemma 4.13 again to find

$$\begin{aligned}\operatorname{hgt}\left(zU^{d+1}\right) &\leq 2^{\deg\left(zU^{d+1}\right)+(n+1)(d+2-1)}\operatorname{hgt}(z)\operatorname{hgt}(U)^{d+1}\\ &\leq 2^{(2d+3)+(n+1)(d+1)}\left(c_1^2+\cdots+c_n^2+1\right)^{d+1},\end{aligned}$$

hence

$$\begin{aligned}\operatorname{hgt}\left(f^2-zU^{d+1}\right) &\leq 2\max\left\{\operatorname{hgt}\left(f^2\right),\operatorname{hgt}\left(zU^{d+1}\right)\right\}\\ &\leq 2\max\left\{2^{2d+n}H^2, 2^{(2d+3)+(n+1)(d+1)}\left(c_1^2+\cdots+c_n^2+1\right)^{d+1}\right\} \qquad (4.30)\\ &\leq 2^{5+n+d(3+n)}H^2\left(c_1^2+\cdots+c_n^2+1\right)^{d+1}.\end{aligned}$$

From (4.29) and (4.30) we deduce

$$\tau \leq 2^{5+n+d(3+n)}(d+1)H^2\left(c_1^2+\cdots+c_n^2+2\right)^{d+1}. \qquad (4.31)$$

Using (4.31), we will simplify (4.28) to find

$$\begin{aligned}A &\leq \left(40\cdot 2^{5+n+d(3+n)}(d+1)H^2\left(c_1^2+\cdots+c_n^2+2\right)^{d+1}d\right)^{(2n)^2(5d)^{4n}}\\ &\leq \left(2dH\left(c_1^2+\cdots+c_n^2+2\right)\right)^{\left(12+n+d(3+n)\right)(2n)^2(5d)^{4n}}\\ &\leq \left(2dH\left(c_1^2+\cdots+c_n^2+2\right)\right)^{104n^3(5d)^{5n}}.\end{aligned}$$

Hence,

$$|z| \geq A^{-1} \geq \left(\left(2dH\left(c_1^2+\cdots+c_n^2+2\right)\right)^{104n^3(5d)^{5n}}\right)^{-1}$$

as desired. ∎

Lemma 4.32. *Suppose $p, q \in \mathbb{Q}^n \cap D$ where D is a connected component of $\{g \neq 0\}$. Let*

$$\frac{A_1}{A_2} = \min\left\{g(p), g(q), \frac{1}{\left(2dH\left(c_1^2+\cdots+c_n^2+2\right)\right)^{104n^3(5d)^{5n}}}\right\}$$

be an irreducible fraction with $A_1, A_2 > 0$. There exists a ball, centered at the origin, of radius

$$n\left(120A_1A_2Hd(c_1^2+\cdots+c_n^2+1)\right)^{4n^3(6d)^{3n}}$$

containing any connectivity path for p and q.

Proof. Suppose $p, q \in \mathbb{Q}^n \cap D$ and let

$$\frac{A_1}{A_2} = \min\left\{ g(p), g(q), \frac{1}{\Big(2dH\big(c_1^2 + \cdots + c_n^2 + 2\big)\Big)^{104n^3(5d)^{5n}}} \right\}$$

be an irreducible fraction with $A_1, A_2 > 0$. Since $\frac{A_1}{A_2} > 0$, according to Lemma 4.14, there exists a ball B, centered at the origin, of radius

$$n\big(120A_1A_2Hd(c_1^2 + \cdots + c_n^2 + 1)\big)^{4n^3(6d)^{3n}}$$

containing $\left\{g = \frac{A_1}{A_2}\right\}$. Suppose $M = \min_{r \in R} g(r)$, where R is the set of routing points of g. From Lemma 4.24, we know

$$M \geq \frac{1}{\Big(2dH\big(c_1^2 + \cdots + c_n^2 + 2\big)\Big)^{104n^3(5d)^{5n}}},$$

hence the ball B contains $\{g = M\}$, $\{g = g(p)\}$, and $\{g = g(q)\}$. Furthermore, B contains $\{g \geq M\}$, $\{g \geq g(p)\}$, and $\{g \geq g(q)\}$ since these sets are compact (Lemma 2.7). We can connect any two points in a connected component by steepest ascent paths using outgoing eigenvectors (Theorem 1.33) and since g increases along a trajectory of ∇g (Lemma 2.4), these steepest ascent paths must lie in $\left\{g \geq \frac{A_1}{A_2}\right\}$. In particular, any connectivity path of p and q must lie in B. $\square$

4.2 Proof of Main Result

The final stage of our proof is to compute a bound on the length of a connectivity path between two points in a same connected component. We will build the connectivity path by looking at trajectories of ∇g between level sets of g. We then use the bounds we have derived earlier to bound the length of the entire connectivity path. The proof of Theorem 1.47 is extremely similar to the proof given in [DK04, Section 10, Theorem 10.3, pp. 18].

Proof of Theorem 1.47. Let $f \in \mathbb{Z}[x_1, \ldots, x_n]$, $n \geq 2$, degree $d \geq 2$ with no singular points. Suppose $(c_1, \ldots, c_n) \in \mathbb{Z}^n$ such that

$$g = \frac{f^2}{\big((x_1 - c_1)^2 + \cdots + (x_n - c_n)^2 + 1\big)^{d+1}}$$

is a routing function. Let $H = \operatorname{hgt}(f)$. Let $\Theta(g)$ be the gradient extremal of g. Let D be a connected component of $\{f \neq 0\}$ and $p, q \in \mathbb{Q}^n \cap D$. Let B be a ball of radius

$$r = n\big(120A_1A_2Hd(c_1^2 + \cdots + c_n^2 + 1)\big)^{4n^3(6d)^{3n}}$$

where

$$\frac{A_1}{A_2} = \min\left\{g(p), g(q), \frac{1}{\Big(2dH\big(c_1^2 + \cdots + c_n^2 + 2\big)\Big)^{104n^3(5d)^{5n}}}\right\}$$

is an irreducible fraction with $A_1, A_2 > 0$. Suppose $\Theta(g) \cap \overline{B}$ is a compact rectifiable curve. According to Lemma 4.32, B contains any connectivity path for p and q.

Consider the connected components of sets $\{g \geq a\} \cap B$, where a is a variable. We write a decomposition into connected components like so:

$$\{g \geq a\} \cap B = \bigcup_{i=1}^{e_a} C_i^a$$

where C_i^a is a connected component. Note that $e_a < \infty$ for any given $a > 0$. Let $\widetilde{\Omega}(g)$ be the ridge and valley set of g restricted to B; that is,

$$\widetilde{\Omega}(g) = \left\{x \in B \;\middle|\; \|\nabla g\|^2 \text{ has a local minimum at } x \in g^{-1}\big(g(x)\big) \cap B\right\}.$$

Let $a > 0$ be arbitrary. Note that for $a \geq \max_{r \in R} g(r)$, the set $\{g > a\} \cap B$ is empty because g is bounded above by $\max_{r \in R} g(R)$ (Lemma 2.6), so we may assume $a < \max_{r \in R} g(r)$. We also assume $a \geq \frac{A_1}{A_2}$ so that every component C_i^a is contained in B.

We claim that any two points in C_i^a can be joined by a connectivity path of length not greater than $2\operatorname{Length}\left(\widetilde{\Omega}(g) \cap C_i^a\right)$. If the claim is true, we can complete the proof in the following way. We fix $a = A_1/A_2$, then we find from Lemma 4.12 that

$$2\operatorname{Length}\left(\widetilde{\Omega}(g) \cap C_i^a\right) \leq 2\operatorname{Length}\left(\widetilde{\Omega}(g)\right) \leq 2 \cdot 2nr(6d+4)^{n-1} = 4nr(6d+4)^{n-1}$$

as desired.

To prove our claim, we use induction on the number of routing points of g in C_i^a. As a base case, suppose C_i^a contains one and only one routing point m_i.

By fixing a particular value of a, in Figure 4.33 we illustrate an example where

$$\{g \geq a\} \cap B = C_1^a \cup C_2^a \cup C_3^a \cup C_4^a.$$

and each C_i^a is drawn as a gray region. The component C_3^a contains one and only one routing point of g (red point).

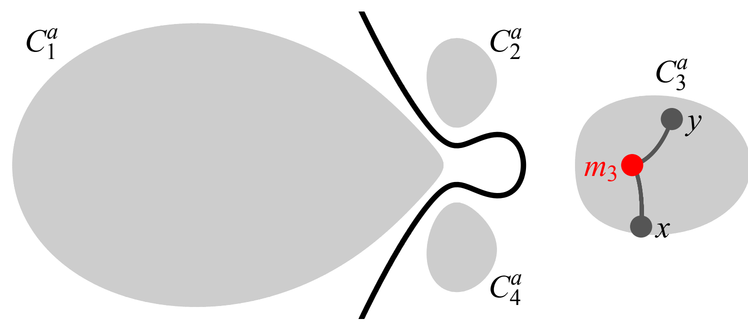


Figure 4.33 Illustration of the induction base case.

Take two points $x, y \in C_i^a$ with $x \neq y$. These points are represented as gray dots in Figure 4.33. We may assume without loss of generality that $\nabla g(x) \neq 0$ since C_i^a contains one and only one routing point of g. As $\nabla g(x) \neq 0$, we know there exists a trajectory of ∇g through x using $\widehat{\nabla g(x)}$ whose destination is m_i (Lemma 2.11). Similarly, if $\nabla g(y) \neq 0$, there exists a trajectory of ∇g through y using $\widehat{\nabla g(y)}$ whose destination is m_i, otherwise $y = m_i$. We see m_1, ϕ_x, ϕ_y is a connectivity path for x and y. In Figure 4.33, we illustrate the corresponding trajectories for x and y as gray curves. From Lemma 4.12,the length of each of the two trajectories is bounded by

$$\text{Length}\left(\widetilde{\Omega}(g) \cap C_i^a\right)$$

hence the sum of the trajectory lengths is bounded by

$$2\,\text{Length}\left(\widetilde{\Omega}(g) \cap C_i^a\right).$$

We now continue with our induction step. Suppose the claim holds for connected components C_i^a containing $m \geq 1$ or less routing points. Consider a connected component C_i^a containing $m+1$ routing points.

By fixing a particular value of a, in Figure 4.34 we illustrate an example where

$$\{g \geq a\} \cap B = C_1^a \cup C_2^a,$$

the gray regions being the respective connected components. We focus on the set C_2^a because it contains more than one routing point (the red points).

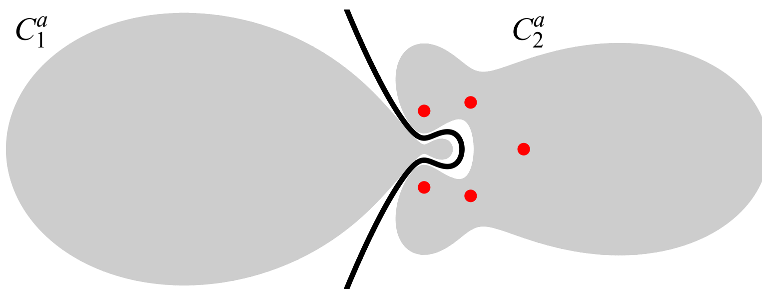


Figure 4.34 A connected component containing more than one routing point.

Let $b = \min_r g(r)$ where the minimum is taken over all routing points r of g lying in C_i^a. Let us denote by $z_1, \ldots, z_\ell$ the routing points that satisfy $g(z_j) = b$. Note that for all j, z_j cannot have index n, otherwise we contradict minimality. Consider the connected components of

$$\{g > b\} \cap C_i^a = \bigcup_{j=1}^{e_b} D_j^b.$$

Note that $e_b < \infty$.

Building off our last figure, in Figure 4.35 we illustrate C_2^a as the dark gray region and D_1^b, D_2^b, D_3^b as the three light gray regions. There are two routing points z_1, z_2 such that $g(z_1) = g(z_2) = b$.

Take two points $x, y \in C_i^a$ with $x \neq y$. We consider several cases.

Case 1. Suppose $g(x) < b$ and $g(y) < b$. It follows that $\nabla g(x) \neq 0$ and $\nabla g(y) \neq 0$, so there exist trajectories ϕ_x and ϕ_y of ∇g through x and y using $\widehat{\nabla g(x)}$ and $\widehat{\nabla g(y)}$, respectively. Consider the portion of the image of ϕ_x and ϕ_y lying in $g^{-1}\big((a,b)\big)$; that is, there exist $\alpha_x, \alpha_y > 0$ such that

$$\phi_x\big((0,\alpha_x)\big) \subseteq g^{-1}\big((a,b)\big) \qquad \text{and} \qquad \phi_y\big((0,\alpha_y)\big) \subseteq g^{-1}\big((a,b)\big).$$

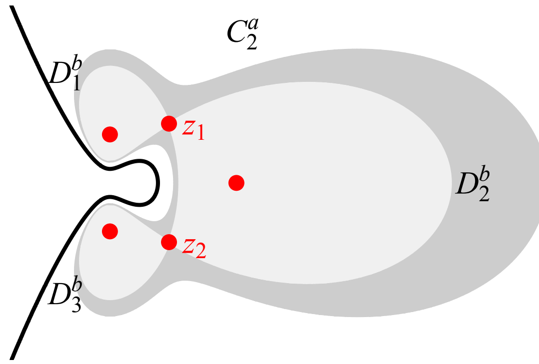


Figure 4.35 Superlevel set of routing point lowest in height.

The length of each of these steepest ascent paths are bounded by

$$\text{Length}\left(\widetilde{\Omega}(g) \cap C_i^a \cap g^{-1}\big((a,b)\big)\right)$$

according to Lemma 4.12, hence the sum of their lengths is bounded by

$$2\,\text{Length}\left(\widetilde{\Omega}(g) \cap C_i^a \cap g^{-1}\big((a,b)\big)\right). \tag{4.36}$$

Let $r_1, \ldots, r_s$, $s \geq 1$, be routing points and $\phi_x, \ldots, \phi_y$ be $s+1$ functions defining a connectivity path P for x and y. We wish to pick a point x' on the connectivity path P for x and y that is arbitrarily close to $\lim_{t\to\alpha_x} \phi_x(t)$ and $g(x') > s$.

If for all j,

$$\lim_{t\to\alpha_x} \phi_x(t) \neq z_j$$

we can simply choose $x' = \phi_x(\alpha_x + \varepsilon)$ for small $\varepsilon > 0$. Similarly for y, if for all j,

$$\lim_{t\to\alpha_y} \phi_y(t) \neq z_j.$$

we choose $y' = \phi_y(\alpha_y + \varepsilon)$ for small $\varepsilon > 0$.

In Figure 4.37a, we illustrate a connectivity path P in blue for a specific choice of x and y. In Figure 4.37b we illustrate the possibility discussed above where a critical point does not lie on the steepest ascent paths corresponding to ϕ_x and ϕ_y. In this figure, x, x', y, y' are the gray points and the gray curves are the images of ϕ_x, ϕ_y, respectively.

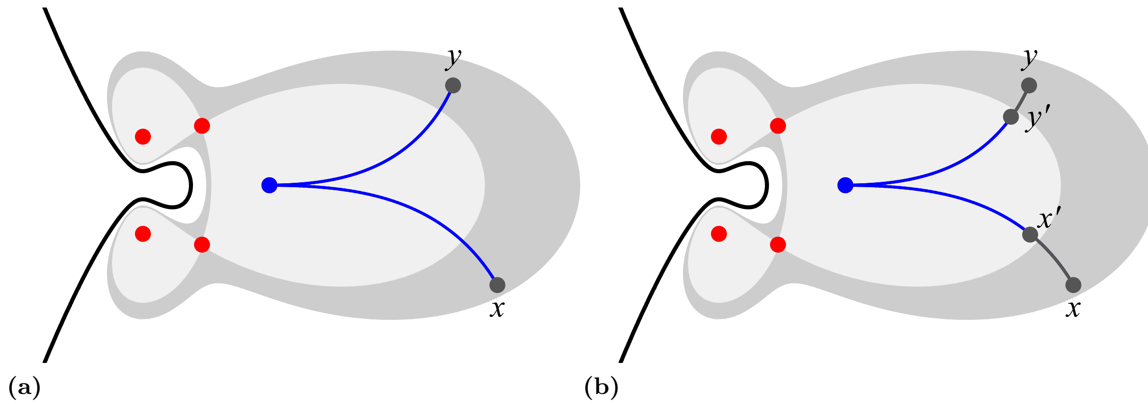


Figure 4.37 Illustration of points x' and y'.

On the other hand, if for some j

$$\lim_{t \to \alpha_x} \phi_x(t) = z_j$$

then $r_1 = z_j$, $s > 1$, and there exists an outgoing eigenvector v of $(\operatorname{Hess} g)(z_j)$ such that r_2 is reachable from r_1 using g and v. Let φ_{r_1} be a trajectory through r_1 using g and v. Pick $x' = \varphi_{r_1}(\varepsilon)$ for small $\varepsilon > 0$. Similarly for y, if for some j

$$\lim_{t \to \alpha_y} \phi_y(t) = z_j$$

then $r_s = z_j$, $s > 1$, and there exists an outgoing eigenvector v of $(\operatorname{Hess} g)(z_j)$ such that r_{s-1} is reachable from r_s using g and v. Let φ_{r_s} be a trajectory through r_s using g and v. Pick $y' = \varphi_{r_s}(\varepsilon)$ for small $\varepsilon > 0$.

In Figure 4.38a, we illustrate a connectivity path P in blue for a specific choice of x and y. The white arrows are the outgoing eigenvectors needed to connect the three blue routing points. In Figure 4.37b we illustrate the possibility discussed above where a critical point z_1, z_2, lies on the steepest ascent paths corresponding to ϕ_x, ϕ_y, respectively. In this figure, x, x', y, y' are the gray points and the gray curves are the images of ϕ_x, ϕ_y, respectively.

We now consider two subcases.

Case 1.1. Suppose x' and y' are in different components of $\{g > b\} \cap C_i^a$; that is, suppose without loss of generality $x' \in D_1^b$ and $y' \in D_{e_b}^b$. Note that $e_b \geq 2$ since the routing points $z_1, \ldots, z_\ell$ are nondegenerate. Let us denote by $\{z'_1, \ldots, z'_{\ell'}\} \subseteq \{z_1, \ldots, z_\ell\}$ the subset of routing

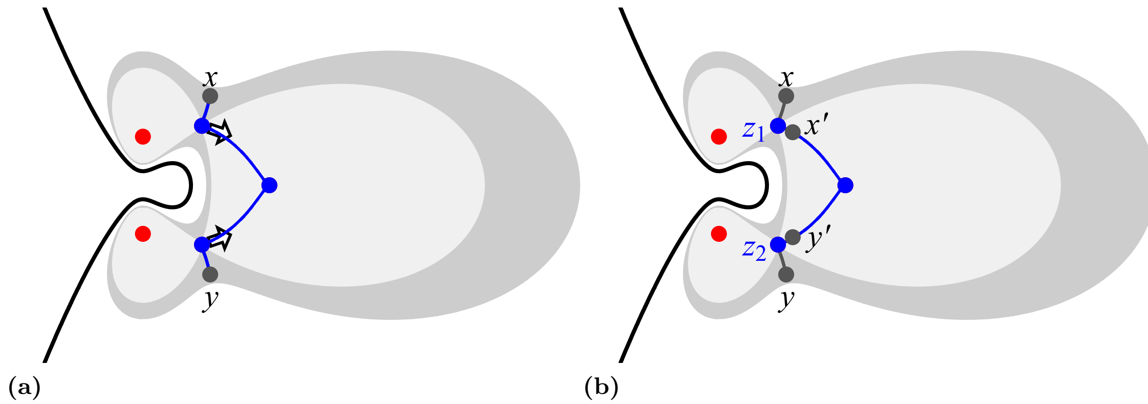


Figure 4.38 Illustration of points x' and y'.

points that lie on the connectivity path P. We consider two subcases.

Case 1.1.1 Suppose $\ell' = 1$. In Figure 4.39a, we illustrate a connectivity path P in blue for a specific choice of x and y. The white arrows are the outgoing eigenvectors needed to connect the three blue routing points. In Figure 4.39b, we show a choice of $x' \in D_1^b$ and $y' \in D_2^b$ as gray points and the routing point z_1' lying on the connectivity path P.

By the induction hypothesis, we can join in D_1^b the point x' with a point x'' lying on the connectivity path P that is very close to z_1' by a connectivity path of length not greater than

$$2\,\mathrm{Length}\left(\widetilde{\Omega}(g) \cap D_1^b\right).$$

Similarly, we can join in $D_{e_b}^b$ the point y' with a point y'' lying on the connectivity path P that is very close to z_1' by a path of length not greater than

$$2\,\mathrm{Length}\left(\widetilde{\Omega}(g) \cap D_{e_b}^b\right).$$

The total length of these curves is not greater than

$$2\left(\mathrm{Length}\left(\widetilde{\Omega}(g) \cap D_1^b\right) + \mathrm{Length}\left(\widetilde{\Omega}(g) \cap D_{e_b}^b\right)\right)$$

In Figure 4.39c, we show the choice of $x'' \in D_1^b$ and $y'' \in D_2^b$ as gray points near the point z_1'.

Finally we can join in C_i^a the point x'' with y'' by a short trajectory of ∇g that is part of the connectivity path P. From (4.36), we deduce the the total length of the connectivity path

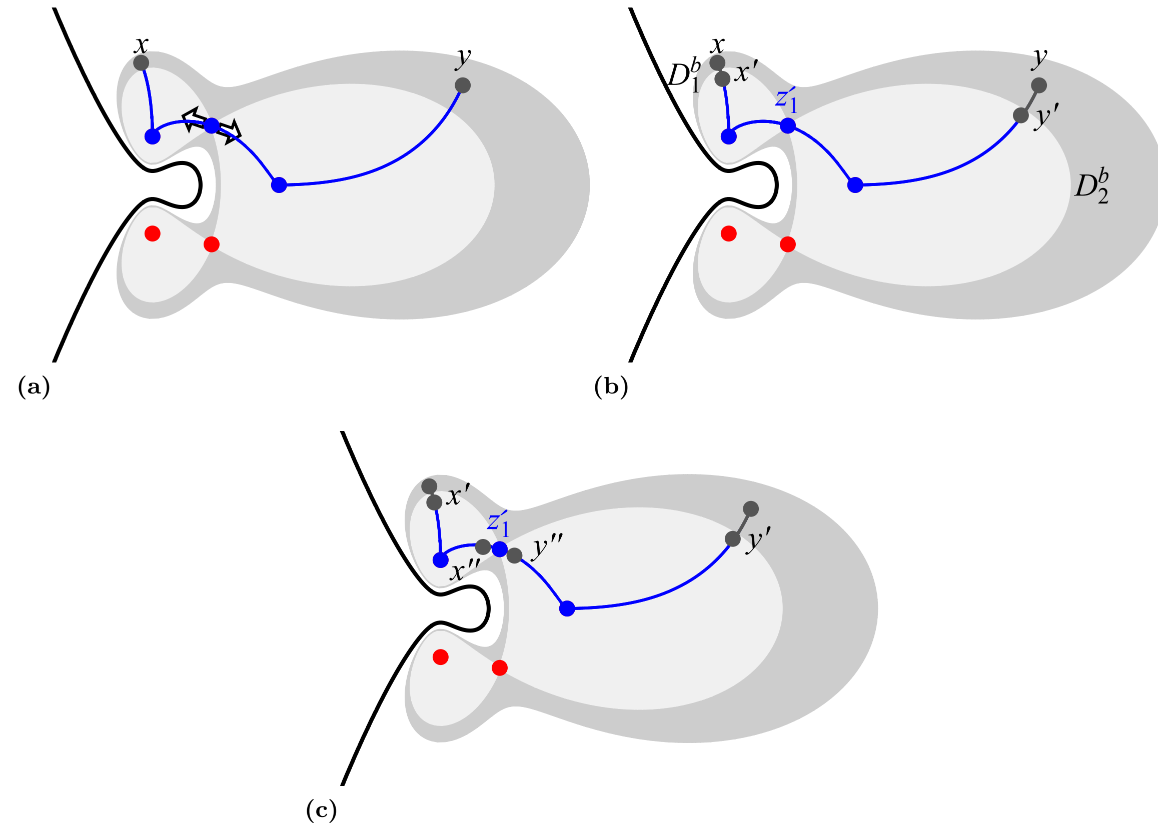


Figure 4.39 Illustration of Case 1.1.1.

joining x with y in C_i^a is not greater than $2\,\mathrm{Length}\left(\widetilde{\Omega}(g) \cap C_i^a\right)$ as desired.

Case 1.1.2 Suppose $\ell' > 1$. The routing points $z_1', \ldots, z_\ell'$ lie on the boundaries of the components $D_1^b, \ldots, D_{e_b}^b$. In Figure 4.40a, we illustrate a connectivity path P in blue for a specific choice of x and y. The white arrows are the outgoing eigenvectors needed to connect the five blue routing points. In Figure 4.40b, we show a choice of $x' \in D_1^b$ and $y' \in D_3^b$ as gray points and the routing points z_1', z_2' lying on the boundaries of D_1^b, D_2^b, D_3^b.

Suppose $z_1' \in \partial D_1^b$. By the induction hypothesis, we can join in D_1^b the point x' with a point x_1' lying on the connectivity path P that is very close to z_1' by a connectivity path of length not greater than

$$2\,\mathrm{Length}\left(\widetilde{\Omega}(g) \cap D_1^b\right).$$

We illustrate a choice of the point x_1' in Figure 4.41a.

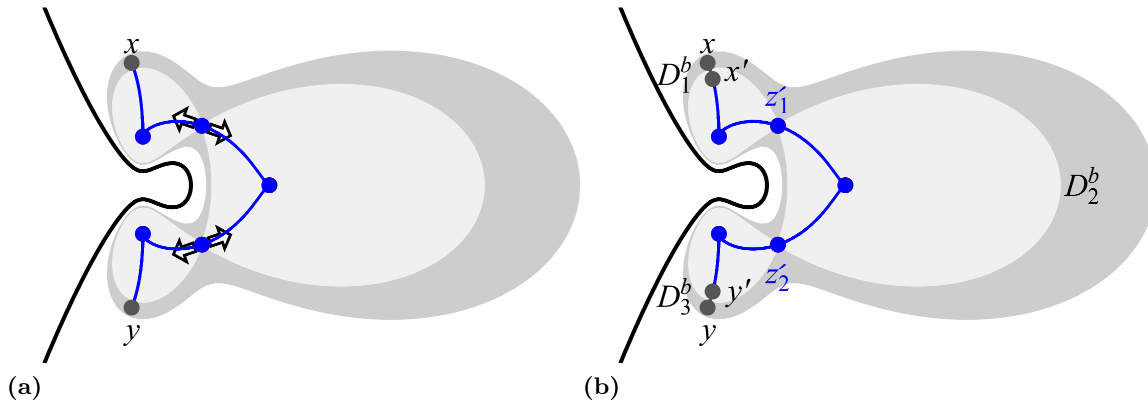


Figure 4.40 Illustration of Case 1.1.2.

Suppose $z_1', z_2' \in \partial D_2^b$. As mentioned earlier, we can find points $x_2', x_3' \in D_2^b$ that are very close to z_1', z_2', respectively, that lie on the connectivity path P. Again, by the induction hypothesis, we can join in D_2^b the point x_2' and x_3' by a connectivity path of length not greater than

$$2\,\mathrm{Length}\left(\widetilde{\Omega}(g) \cap D_2^b\right).$$

We illustrate a choice of the points x_2', x_3' in Figure 4.41b.

We continue this process to generate a sequence of points $x_1', \dots, x_h'$ where $x_h' \in D_{e_b}^b$ and x_h' is very close to $z_{\ell'}$. We illustrate a choice for the sequence of points x_1', x_2', x_3', x_4' in Figure 4.41c.

The total length of these curves is not greater than

$$2\left(\mathrm{Length}\left(\widetilde{\Omega}(g) \cap D_1^b\right) + \cdots + \mathrm{Length}\left(\widetilde{\Omega}(g) \cap D_{e_b}^b\right)\right)$$

Finally we can join the points x_{j-1}' with x_j' by a short trajectory of ∇g that is part of the connectivity path P. From (4.36), we deduce the the total length of the connectivity path joining x with y in C_i^a is not greater than $2\,\mathrm{Length}\left(\widetilde{\Omega}(g) \cap C_i^a\right)$ as desired.

Case 1.2. Suppose x' and y' are in the same component of $\{g > b\} \cap C_i^a$; that is, suppose $x', y' \in D_j^b$. Such a scenario is illustrated in Figures 4.37b and 4.38b because x', y' are both in D_2^b. The component D_j^b has m or less routing points, so by the induction hypothesis, we can connect x' and y' by a connectivity path of length not greater than

$$2\,\mathrm{Length}\left(\widetilde{\Omega}(g) \cap D_j^b\right).$$

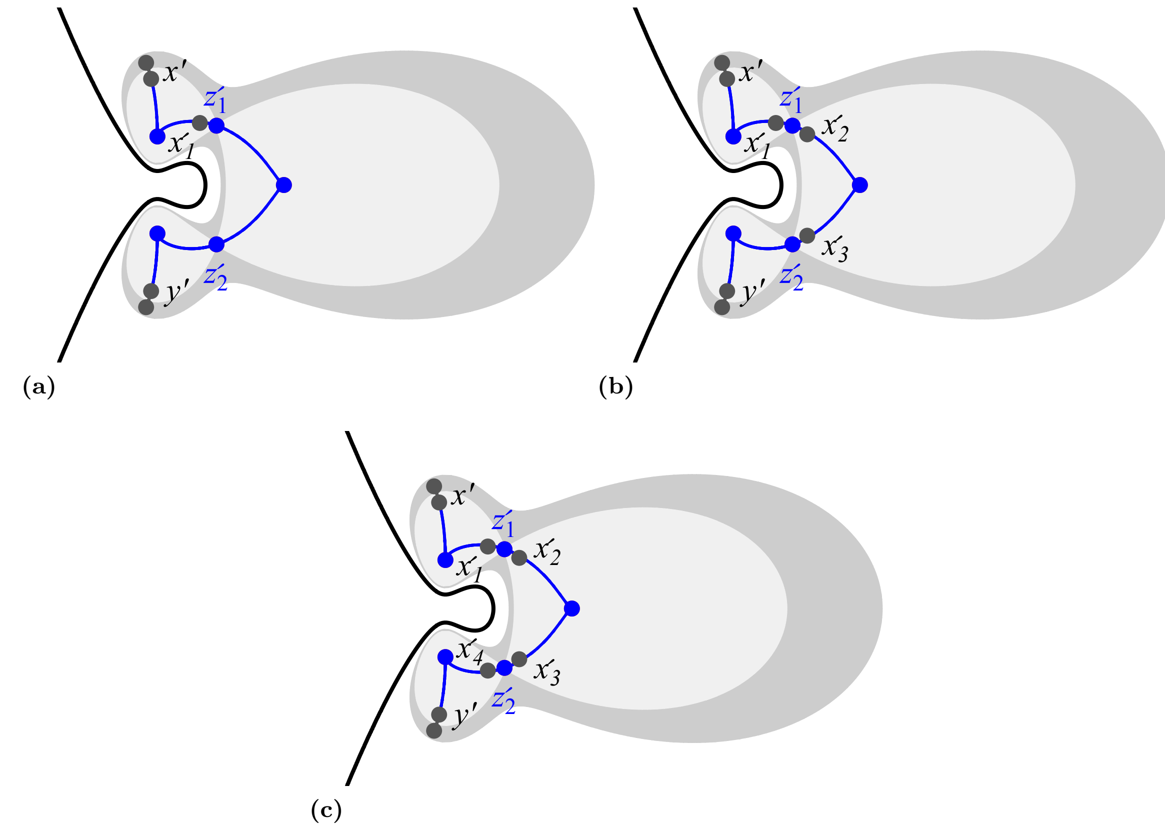


Figure 4.41 Illustration of Case 1.1.2.

Hence by (4.36), the total length of a connectivity path joining x with y in A_i^t is not greater than

$$2\,\mathrm{Length}\left(\widetilde{\Omega}(g) \cap C_i^a \cap g^{-1}\big((a,b)\big)\right) + 2\,\mathrm{Length}\left(\widetilde{\Omega}(g) \cap D_j^b\right) \le 2\,\mathrm{Length}\left(\widetilde{\Omega}(g) \cap C_i^a\right)$$

as desired.

The remaining cases where $g(x) < b$ and $g(y) > b$, or, $g(x) > b$ and $g(y) > b$ can be handled analogously.

$\square$

Chapter 5

Experimental Results

In this chapter we give experimental results for different size inputs to estimate the running time of **Connectivity**. The algorithm **Connectivity** was implemented in Maple 17 on top of a 64-bit Windows 7 system running an Intel Core i7-920 processor at 2.67 GHz with 6 GB of RAM. In order to measure the performance, we first need to fix the implementation details of several steps. We have made the following choices.

- To find routing points, we use the Maple command `RootFinding[Isolate]`, or when it fails, the `RegularChains[SemiAlgebraicSetTools][RealRootIsolate]` command. Both of these commands return a list of boxes with each box isolating exactly one routing point. We then took the center of each box computed to be the routing point.

- To implement **Destination**(g, R, p, v), we construct an approximation of the steepest ascent path through p using v and then use the endpoint of this path to determine the index of the point in R to return. In our implementation, we first let $q = p + 0.01v$, then approximate the steepest ascent path through q by taking steps of length 0.01 in the direction of the normalized gradient of g. After each gradient ascent step we check to see which of the points in R is closest, and terminate ascent when one is found within a tolerance of 0.01. The index of this closest point is the output of **Destination**(g, R, p, v).

In the first section we visualize the connectivity path for six non-trivial input polynomials having $n = 2$ or $n = 3$ variables and give timing results. In the second section we give some raw data on the computation time for running **Connectivity** on randomly generated input polynomials having $n = 2$ variables.

5.1 Non-Trivial Examples

In this section, we present several non-trivial examples using input polynomials in two and three variables. Each example will illustrate the routing points and all possible connectivity paths for any two routing points.

Example 5.1. Let

$$\begin{aligned} f = {} & 1280000x_1^{10} + 2560000x_1^8x_2^2 - 2016000x_1^8 + 1280000x_1^7x_2 + 1280000x_1^6x_2^4 \\ & - 2336000x_1^6x_2^2 + 793800x_1^6 - 1280000x_1^5x_2 - 1280000x_1^4x_2^4 + 1056000x_1^4x_2^2 \\ & - 59080x_1^4 + 2560000x_1^2x_2^4 - 738560x_1^2x_2^2 + 736x_1^2 + 1280000x_1x_2^3 - 1280x_1x_2 \\ & + 1280000x_2^6 + 222720x_2^4 + 57576x_2^2 - 45. \end{aligned}$$

In Figure 5.3a, the curve $\{f \neq 0\}$ is shown in black while the routing points and connectivity path are shown in red. The connectivity matrix formed had size 21×21 and took 2.36 seconds to find. Of those 2.36 seconds, 0.55 seconds were dedicated to finding the routing points. We randomly generated 100 pairs of points uniformly over $[-3.68, 3.68] \times [-1.29, 1.29]$ and used the connectivity matrix to determine the connectivity of these 100 pairs of points. The computing time was 0.14 seconds per pair on average.

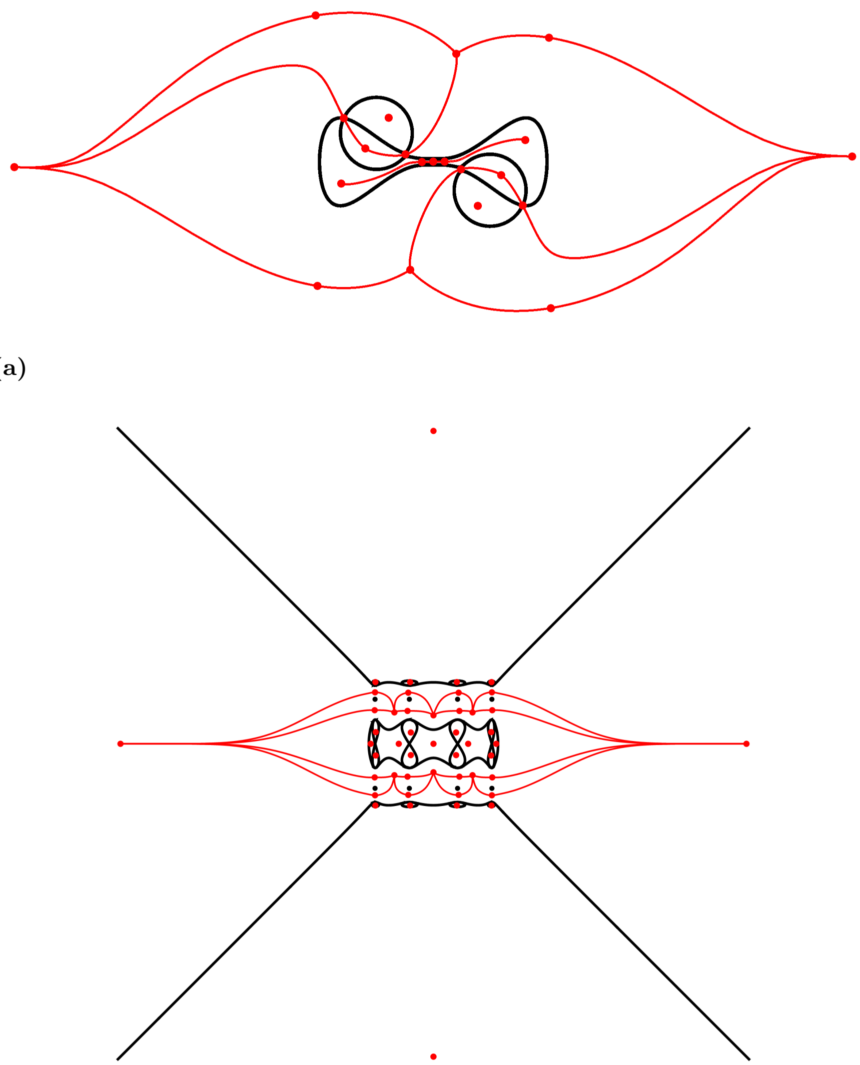
In Example 5.1 we see that the curve has many "narrow" gaps. The polynomial f was constructed so these gaps existed. The numeric methods for solving this problem would likely miss the narrow gaps, often producing wrong outputs. However, our algorithm presented in this thesis correctly catches all the narrow gaps.

Example 5.2. Let

$$\begin{aligned} f = {} & 4096x_1^{16} - 16384x_1^{14} + 26624x_1^{12} - 22528x_1^{10} - 1024x_1^8x_2^4 + 1024x_1^8x_2^2 \\ & + 10496x_1^8 + 2048x_1^6x_2^4 - 2048x_1^6x_2^2 - 2560x_1^6 - 1280x_1^4x_2^4 + 1280x_1^4x_2^2 \\ & + 256x_1^4 + 256x_1^2x_2^4 - 256x_1^2x_2^2 - 4096x_2^{16} + 16384x_2^{14} - 26624x_2^{12} \\ & + 22528x_2^{10} - 10560x_2^8 + 2688x_2^6 - 352x_2^4 + 32x_2^2 - 1. \end{aligned}$$

In Figure 5.3b, the curve $\{f \neq 0\}$ is shown in black while the routing points and connectivity path are shown in red. The connectivity matrix formed had size 47×47 and took 16.58 seconds to find. Of those 16.58 seconds, 5.55 seconds were dedicated to finding the routing points. We randomly generated 100 pairs of points uniformly over $[-4.96, 4.96]^2$ and used the connectivity matrix to determine the connectivity of these 100 pairs of points. The computing time was 0.51 seconds per pair on average.

In Example 5.2, the polynomial f was taken from [Lab10]. We chose this polynomial because plotting the implicit curve where $f = 0$ is very difficult. Our connectivity method can answer connectivity queries despite this difficulty.



(a)

(b)

Figure 5.3 Illustration of the connectivity path for examples with $n = 2$.

Example 5.4. Let

$$
\begin{aligned}
f = {} & 16000000000000000x_1^{20} - 6400000000000000x_1^{19} + 280000000000000000x_2^2x_1^{18} \\
& - 38400000000000000x_2x_1^{18} - 229120000000000000x_1^{18} - 110400000000000000x_2^2x_1^{17} \\
& + 15360000000000000x_2x_1^{17} + 96768000000000000x_1^{17} + 2025000000000000000x_2^4x_1^{16} \\
& - 518400000000000000x_2^3x_1^{16} - 2200960000000000000x_2^2x_1^{16} + 205824000000000000x_2x_1^{16} \\
& + 1755696000000000000x_1^{16} - 782400000000000000x_2^4x_1^{15} + 203520000000000000x_2^3x_1^{15} \\
& + 938048000000000000x_2^2x_1^{15} - 94617600000000000x_2x_1^{15} - 147835200000000000x_1^{15} \\
& + 7905000000000000000x_2^6x_1^{14} - 2786400000000000000x_2^5x_1^{14} - 8640200000000000000x_2^4x_1^{14} \\
& + 1518080000000000000x_2^3x_1^{14} - 1366374000000000000x_2^2x_1^{14} + 1439238400000000000x_2x_1^{14} \\
& + 579940358400000000x_1^{14} - 2966400000000000000x_2^6x_1^{13} + 1063680000000000000x_2^5x_1^{13} \\
& + 3737600000000000000x_2^4x_1^{13} - 730880000000000000x_2^3x_1^{13} - 675632000000000000x_2^2x_1^{13} \\
& - 499538176000000000x_2x_1^{13} - 219843271680000000x_1^{13} + 18390000000000000000x_2^8x_1^{12} \\
& - 7826400000000000000x_2^7x_1^{12} - 20083840000000000000x_2^6x_1^{12} + 5885408000000000000x_2^5x_1^{12} \\
& - 2694237000000000000x_2^4x_1^{12} + 10598894400000000000x_2^3x_1^{12} - 1895008505600000000x_2^2x_1^{12} \\
& - 1028177561600000000x_2x_1^{12} - 1636016433880000000x_1^{12} - 6614400000000000000x_2^8x_1^{11} \\
& + 2864640000000000000x_2^7x_1^{11} + 8709504000000000000x_2^6x_1^{11} - 2820710400000000000x_2^5x_1^{11} \\
& - 913001600000000000x_2^4x_1^{11} - 3631300096000000000x_2^3x_1^{11} + 798637992960000000x_2^2x_1^{11} \\
& + 807705640960000000x_2x_1^{11} + 240643564294400000x_1^{11} + 26730000000000000000x_2^{10}x_1^{10} \\
& - 12830400000000000000x_2^9x_1^{10} - 31461680000000000000x_2^8x_1^{10} + 13427328000000000000x_2^7x_1^{10} \\
& + 6441138240000000000x_2^6x_1^{10} - 3688223680000000000x_2^5x_1^{10} - 18838045801600000000x_2^4x_1^{10} \\
& + 4949577953280000000x_2^3x_1^{10} + 5762873629812000000x_2^2x_1^{10} - 6182966429024000000x_2x_1^{10} \\
& - 269292141035696000x_1^{10} - 9038400000000000000x_2^{10}x_1^9 + 4416000000000000000x_2^9x_1^9 \\
& + 12893440000000000000x_2^8x_1^9 - 5933568000000000000x_2^7x_1^9 - 2848442560000000000x_2^6x_1^9 \\
& + 3805657600000000000x_2^5x_1^9 + 7343349120000000000x_2^4x_1^9 - 1754397245440000000x_2^3x_1^9 \\
& - 3211337005312000000x_2^2x_1^9 + 1814309571072000000x_2x_1^9 + 878507589008640000x_1^9 \\
& + 24645000000000000000x_2^{12}x_1^8 - 12830400000000000000x_2^{11}x_1^8 - 3447472000000000000x_2^{10}x_1^8 \\
& + 18552640000000000000x_2^9x_1^8 + 22580358600000000000x_2^8x_1^8 - 10518047040000000000x_2^7x_1^8 \\
& - 11039219680000000000x_2^6x_1^8 - 658880768000000000x_2^5x_1^8 + 6771455274800000000x_2^4x_1^8 \\
& - 2109646442400000000x_2^3x_1^8 - 2549149923465600000x_2^2x_1^8 + 611154283384960000x_2x_1^8 \\
& + 201009929499102000x_1^8 - 7598400000000000000x_2^{12}x_1^7 + 4028160000000000000x_2^{11}x_1^7 \\
& + 11818048000000000000x_2^{10}x_1^7 - 6833152000000000000x_2^9x_1^7 - 8379111840000000000x_2^8x_1^7 \\
& + 4253558784000000000x_2^7x_1^7 + 5296576768000000000x_2^6x_1^7 - 1402340147200000000x_2^5x_1^7
\end{aligned}
$$

$$
\begin{aligned}
&- 6875755107392000000x_2^4x_1^7 + 1547662988620800000x_2^3x_1^7 + 379568919391616000x_2^2x_1^7 \\
&- 38620939604992000000x_2x_1^7 - 1494959424472416000x_1^7 + 14205000000000000000x_2^{14}x_1^6 \\
&- 78264000000000000000x_2^{13}x_1^6 - 265595600000000000000x_2^{12}x_1^6 + 160387840000000000000x_2^{11}x_1^6 \\
&+ 184516767600000000000x_2^{10}x_1^6 - 109785500800000000000x_2^9x_1^6 + 196432390880000000000x_2^8x_1^6 \\
&- 82163109632000000000x_2^7x_1^6 + 150851287279200000000x_2^6x_1^6 - 40942293980160000000x_2^5x_1^6 \\
&+ 176998962384192000000x_2^4x_1^6 - 94154955175897600000x_2^3x_1^6 - 41304058956936800000x_2^2x_1^6 \\
&+ 5456562753725760000x_2x_1^6 + 2616446089005160000x_1^6 - 378240000000000000000x_2^{14}x_1^5 \\
&+ 21235200000000000000x_2^{13}x_1^5 + 61140480000000000000x_2^{12}x_1^5 - 41289216000000000000x_2^{11}x_1^5 \\
&- 474511088000000000000x_2^{10}x_1^5 + 317160896000000000000x_2^9x_1^5 - 47738984448000000000x_2^8x_1^5 \\
&+ 61327417344000000000x_2^7x_1^5 + 26335874886400000000x_2^6x_1^5 - 1946273099008000000x_2^5x_1^5 \\
&- 63421541216473600000x_2^4x_1^5 + 26737441898854400000x_2^3x_1^5 - 5486983664575680000x_2^2x_1^5 \\
&+ 936526202576640000x_2x_1^5 + 167101456627238400x_1^5 + 48600000000000000000x_2^{16}x_1^4 \\
&- 27864000000000000000x_2^{15}x_1^4 - 142552000000000000000x_2^{14}x_1^4 + 89092800000000000000x_2^{13}x_1^4 \\
&+ 155343630000000000000x_2^{12}x_1^4 - 132145264000000000000x_2^{11}x_1^4 - 82210282336000000000x_2^{10}x_1^4 \\
&+ 33234046464000000000x_2^9x_1^4 + 169229546759000000000x_2^8x_1^4 - 30535950392320000000x_2^7x_1^4 \\
&- 300606415381984000000x_2^6x_1^4 + 50512733619929600000x_2^5x_1^4 + 223481250322622400000x_2^4x_1^4 \\
&- 42146648802669120000x_2^3x_1^4 + 15319082635830960000x_2^2x_1^4 - 710738466423974400x_2x_1^4 \\
&- 244202433980291810000x_1^4 - 99840000000000000000x_2^{16}x_1^3 + 58368000000000000000x_2^{15}x_1^3 \\
&+ 14336000000000000000x_2^{14}x_1^3 - 111206400000000000000x_2^{13}x_1^3 + 145503104000000000000x_2^{12}x_1^3 \\
&- 72506024960000000000x_2^{11}x_1^3 - 40564543488000000000x_2^{10}x_1^3 + 61556858880000000000x_2^9x_1^3 \\
&- 25215661832640000000x_2^8x_1^3 + 6292042747648000000x_2^7x_1^3 - 752738914086400000x_2^6x_1^3 \\
&+ 1971981791846400000x_2^5x_1^3 + 2679268139618912000x_2^4x_1^3 - 857792152334592000x_2^3x_1^3 \\
&- 19654575438739712000x_2^2x_1^3 + 1954136907896320000x_2x_1^3 + 78431494720079984000x_1^3 \\
&+ 8800000000000000000x_2^{18}x_1^2 - 51840000000000000000x_2^{17}x_1^2 - 49894400000000000000x_2^{16}x_1^2 \\
&+ 31086080000000000000x_2^{15}x_1^2 - 87283920000000000000x_2^{14}x_1^2 + 39317840000000000000x_2^{13}x_1^2 \\
&- 165097461376000000000x_2^{12}x_1^2 + 100394639718400000000x_2^{11}x_1^2 + 497910017785200000000x_2^{10}x_1^2 \\
&- 91107727988640000000x_2^9x_1^2 + 278774168750680000000x_2^8x_1^2 - 91487664754649600000x_2^7x_1^2 \\
&+ 55040267542816000000x_2^6x_1^2 + 190472562173884800000x_2^5x_1^2 - 245267864092028080000x_2^4x_1^2 \\
&+ 57447326101230592000x_2^3x_1^2 + 45459012287500361000x_2^2x_1^2 - 8928638845736344000x_2x_1^2 \\
&- 5692877331995819000x_1^2 - 10240000000000000000x_2^{18}x_1 + 61440000000000000000x_2^{17}x_1 \\
&+ 57344000000000000000x_2^{16}x_1 - 67174400000000000000x_2^{15}x_1 + 57014528000000000000x_2^{14}x_1 \\
&- 33685237760000000000x_2^{13}x_1 + 189080283136000000000x_2^{12}x_1 - 95697978982400000000x_2^{11}x_1 \\
&+ 32630827031040000000x_2^{10}x_1 + 22457344017920000000x_2^9x_1 - 45305339536537600000x_2^8x_1
\end{aligned}
$$

$$
\begin{aligned}
&+ 23599098092492800000x_2^7x_1 - 19369965807319360000x_2^6x_1 + 8588970870144000 0x_2^5x_1 \\
&+ 29797343375877504000x_2^4x_1 - 11914877542740480000x_2^3x_1 + 10550642012657342400x_2^2x_1 \\
&- 1071787432531200000x_2x_1 - 3500193002386180800x_1 + 6400000000000000 0x_2^{20} \\
&- 3840000000000000 0x_2^{19} - 8652800000000000 0x_2^{18} + 5396480000000000 0x_2^{17} \\
&+ 1410240000000000 0x_2^{16} - 3065996800000000 00x_2^{15} + 23161462144000000000x_2^{14} \\
&- 13586078515200000000x_2^{13} + 11883006282720000000x_2^{12} + 2163636896000000x_2^{11} \\
&- 9668035555205760000 0x_2^{10} + 5438670924767360000 0x_2^9 + 6384683842946600000x_2^8 \\
&- 32088063513256640000x_2^7 + 10348948190067256000 0x_2^6 - 30732502775816064000x_2^5 \\
&- 24732429078459658000x_2^4 + 19786492596245821600x_2^3 - 29369716183334702000x_2^2 \\
&+ 2036619410857724000x_2 + 6647095911409240641.
\end{aligned}
$$

In Figure 5.5, the curve $\{f \neq 0\}$ is shown in black while the routing points and connectivity path are shown in red. The connectivity matrix formed had size 53×53 and took 169.12 seconds to find. Of those 169.12 seconds, 105.28 seconds were dedicated to finding the routing points. We randomly generated 100 pairs of points uniformly over $[-7.40, 7.74] \times [-7.23, 7.74]$ and used the connectivity matrix to determine the connectivity of these 100 pairs of points. The computing time was 1.73 seconds per pair on average.

Again, in Example 5.4 we see that the curve has many "narrow" gaps. The polynomial f was constructed so these gaps existed.

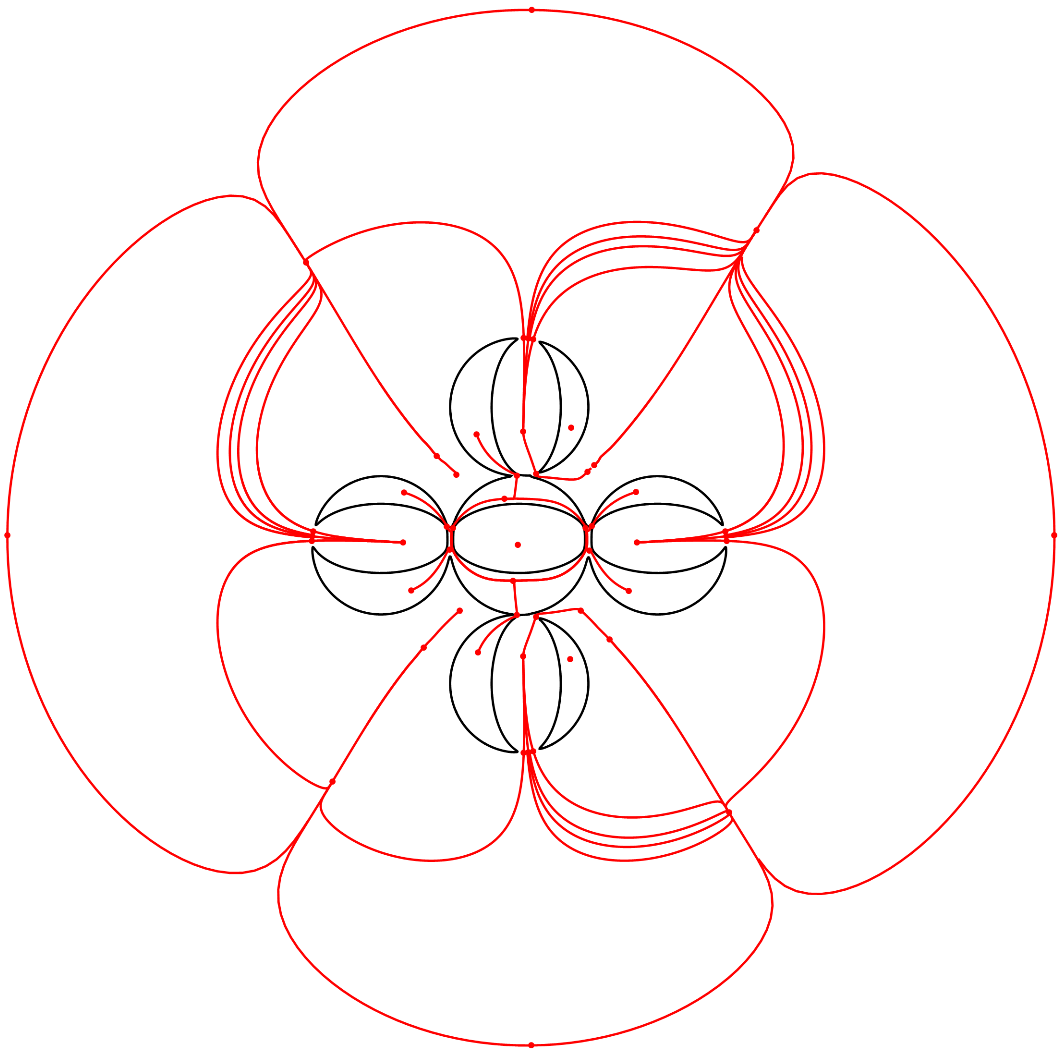


Figure 5.5 Illustration of the connectivity path for example with $n = 2$.

Example 5.6. Let

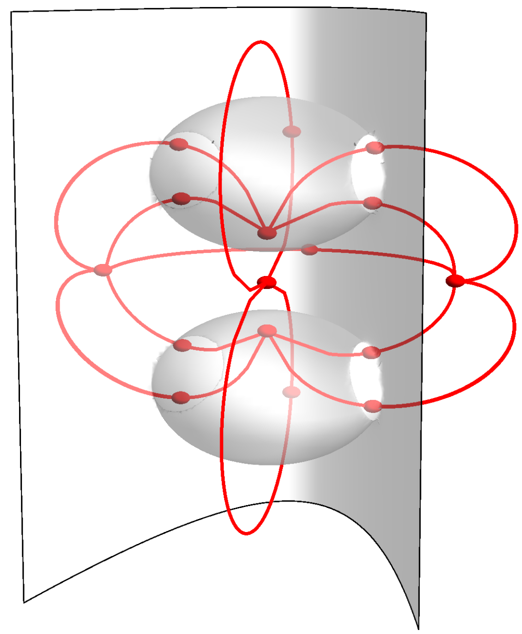
$$\begin{aligned} f = &-31 - 16x_1^2 + 8x_1^4 + 4x_1^6 + 16x_2 + 16x_1^2x_2 + 4x_1^4x_2 - 32x_2^2 + 8x_1^4x_2^2 + 16x_2^3 + 8x_1^2x_2^3 \\ &- 8x_2^4 + 4x_1^2x_2^4 + 4x_2^5 + 96x_3^2 - 64x_1^2x_3^2 + 8x_1^4x_3^2 - 48x_2x_3^2 + 8x_1^2x_2x_3^2 - 16x_2^2x_3^2 \\ &+ 8x_1^2x_2^2x_3^2 + 8x_2^3x_3^2 - 8x_3^4 + 4x_1^2x_3^4 + 4x_2x_3^4. \end{aligned}$$

In Figure 5.8a, the semi-algebraic set $\{f = 0\}$ consists of one connected component which we show in light gray while the routing points and connectivity path are shown in red. The connectivity matrix formed had size 16×16 and took 14.19 seconds to find. Of those 14.19 seconds, 9.2 seconds were dedicated to finding the routing points. We randomly generated 100 pairs of points uniformly over $[-2.24, 2.24] \times [-0.10, 2.98] \times [-2.6, 2.6]$ and used the connectivity matrix to determine the connectivity of these 100 pairs of points. The computing time was 0.23 seconds per pair on average.

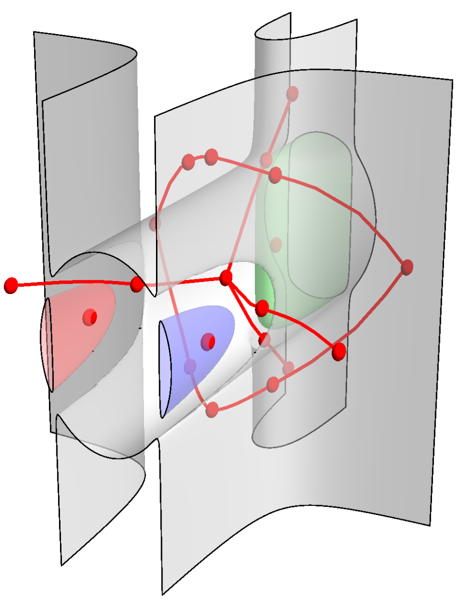
Example 5.7. Let

$$f = 20x_1^4x_2 + 20x_1^2x_2x_3^2 - 60x_1^2x_2 + 20x_1^2 - 20x_2x_3^2 + 40x_2 + 20x_3^2 - 41.$$

In Figure 5.8b, the semi-algebraic set $\{f = 0\}$ consists of four connected components which we show in light gray, light red, light blue, and light green, respectively, while the routing points and connectivity path are shown in red. The connectivity matrix formed had size 20×20 and took 3.62 seconds to find. Of those 3.62 seconds, 1.94 seconds were dedicated to finding the routing points. We randomly generated 100 pairs of points uniformly over $[-3.67, 3.67] \times [-2.22, 1.84] \times [-2.26, 2.26]$ and used the connectivity matrix to determine the connectivity of these 100 pairs of points. The computing time was 0.17 seconds per pair on average.



(a)



(b)

Figure 5.8 Illustration of the connectivity path for examples with $n = 3$.

Example 5.9. Let

$$f = x_1^6 + 4x_1^4x_2^2 + 3x_1^4x_3^2 + 2x_1^4 + 5x_1^2x_2^4 + 8x_1^2x_2^2x_3^2 + 8x_1^2x_2^2 + 3x_1^2x_3^4 - 12x_1^2x_3^2$$
$$- 4x_1^2 + 2x_2^6 + 5x_2^4x_3^2 + 6x_2^4 + 4x_2^2x_3^4 - 24x_2^2x_3^2 + x_3^6 - 14x_3^4 + 28x_3^2 - 7.$$

In Figure 5.10, the semi-algebraic set $\{f = 0\}$ consists of three connected components which we show in light gray while the routing points and connectivity path are shown in red. The connectivity matrix formed had size 19×19 and took 9.31 seconds to find. Of those 9.31 seconds, 1.89 seconds were dedicated to finding the routing points. We randomly generated 100 pairs of points uniformly over $[-2.43, 2.43] \times [-1.78, 1.78] \times [-6.60, 6.60]$ and used the connectivity matrix to determine the connectivity of these 100 pairs of points. The computing time was 0.53 seconds per pair on average.

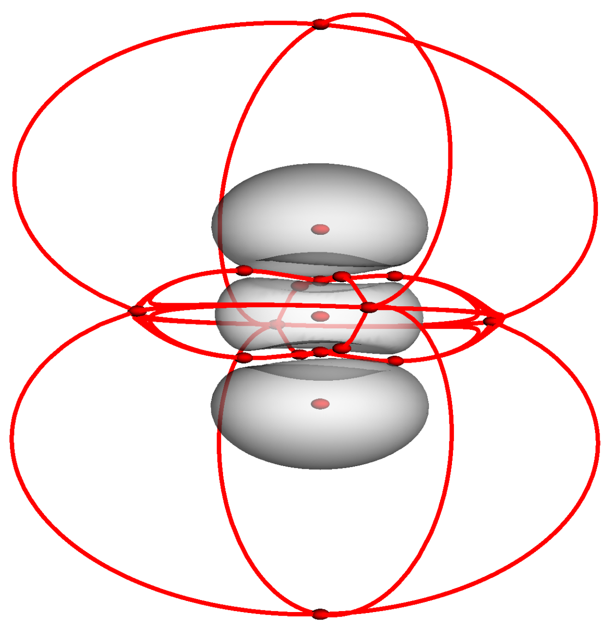


Figure 5.10 Illustration of the connectivity path for example with $n = 3$.

5.2 Computational Timings

In this section we will show that **Connectivity** typically runs faster for sparse polynomial inputs than for dense polynomial inputs. We use other computed results, such as the number of routing points calculated, to aid in our discussion of the running times. Throughout this section we use the abbreviations listed in Table 5.11.

Table 5.11 Abbreviations used throughout Section 5.2.

Abbreviation	Meaning
d	Degree of f
Time R	Average time to find routing points R
Time M	Average time to compute connectivity matrix M
Avg. No. CC	Average number of connected components of $\{f \neq 0\}$
Avg. No. R	Average number of routing points R
Avg. No. Non-Max	Average number of routing points having index less than n
Avg. No. Max	Average number of routing points having index n

We begin by explaining how we calculate each quantity in Table 5.11 given N polynomial instances $\{f_1, \ldots, f_N\}$. For each f_i, we compute the total CPU time R_i it takes to execute steps 1 through 3 of algorithm **Connectivity** and let

$$\text{Time } R = \frac{1}{N} \sum_{i=1}^{N} R_i.$$

Then for each f_i, we compute the total CPU time M_i it takes to execute steps 1 through 6 of algorithm **Connectivity** and let

$$\text{Time } M = \frac{1}{N} \sum_{i=1}^{N} M_i.$$

For each f_i, once step 5 of **Connectivity** has completed, we form a graph G_i using the adjacency matrix A_i and compute the number connected components C_i of the graph G_i. Then we let

$$\text{Avg. No. CC} = \frac{1}{N}\sum_{i=1}^{N} C_i.$$

For each f_i, once step 3 of **Connectivity** has completed, we count the number of routing points P_i computed and let

$$\text{Avg. No. } R = \frac{1}{N}\sum_{i=1}^{N} P_i.$$

Finally, for each f_i, once step 5 of **Connectivity** has completed, we compute the number of routing points having index less than $n = 2$, called O_i. Then we let

$$\text{Avg. No. Non-Max} = \frac{1}{N}\sum_{i=1}^{N} O_i$$

$$\text{Avg. No. Max} = \frac{1}{N}\sum_{i=1}^{N} (P_i - O_i).$$

To compute our instances, we used the Maple command `randpoly([x1, x2], degree=d)` to randomly generate 1000 sparse polynomial instances $\left\{f_1^S, \ldots, f_{1000}^S\right\}$ and the Maple command `randpoly([x1, x2], degree=d, dense)` to randomly generate 1000 dense polynomial instances $\left\{f_1^D, \ldots, f_{1000}^D\right\}$, where degree $2 \leq d \leq 13$. If in the course of running **Connectivity** a steepest ascent path using outgoing eigenvectors is computed to have length longer than 1500, then the instance was removed from calculation. The total number of instances we used are shown in Table 5.12.

We begin our discussion by presenting the average running times for sparse polynomial and dense polynomial instances in Table 5.13. We visualize this data in Figure 5.14 by plotting the degree versus the average time. It would appear from Figure 5.14 that the average time to find the routing points is roughly the same in both cases. However, by studying Table 5.13, we see the dense instances take slightly more time on average than the sparse instances. Interestingly, the average time required to compute the connectivity matrix in the sparse case is nearly constant as degree increases, which is not true for the dense case.

To understand why this might be true, we present four other computations: Avg. No. CC, Avg. No. R, Avg. No. Max, and Avg. No. Non-Max, in Table 5.15. By first studying Avg. No. R, we immediately see that in the dense case there are more routing points on average than in the sparse case. This may explain the slight difference in computation times for Time R. A consequence of having more routing points on average is that there may be more routing points having index less than n. Recall that for each routing point r having index less than n, we use the outgoing eigenvectors of $(\operatorname{Hess} g)(r)$ to compute a steepest ascent path. Hence, we expect in cases where there are more routing points of index less than n, the time required to compute the connectivity matrix should be higher as well. This is evidenced by the results in the Avg. No. Non-Max column. We see that dense polynomial instances have on average more routing points of index less than n than the sparse polynomial instances. This gives a reason for why computing the connectivity matrix takes longer in the dense case than in the sparse case.

If we study the Avg. No. CC column, we see that for a sparse polynomial instance f_i^S, the number of connected components of $\left\{f_i^S \neq 0\right\}$ is larger than the number of connected components of $\left\{f_i^D \neq 0\right\}$, where f_i^D is a dense polynomial instance. A consequence of having more connected components is that we have more routing points of index n. The reason being that in Lemma 2.16 we showed each connected component has at least one routing point of index n. This also explains why the sparse case has less routing points of index n on average than the dense case.

Table 5.12 Number of instances generated for each degree.

d	Number of Sparse Instances (N)	Number of Dense Instances (N)
2	1000	998
3	1000	998
4	1000	994
5	999	996
6	997	1000
7	997	997
8	995	998
9	994	999
10	990	996
11	996	998
12	995	998
13	991	995

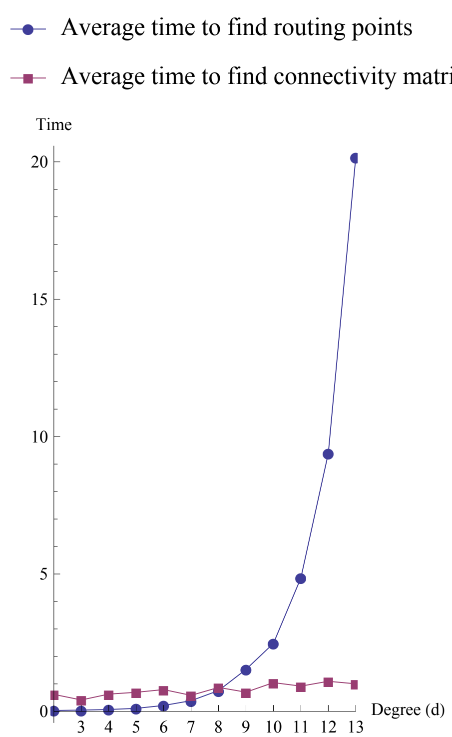
Table 5.13 Average running times for sparse and dense polynomial instances.

(a) Sparse polynomial instances.

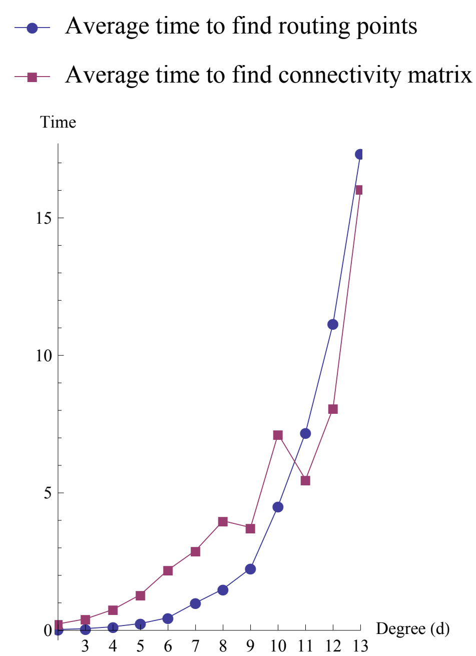
d	Time R	Time M
2	0.030853	0.617973
3	0.046903	0.415346
4	0.067771	0.628224
5	0.110179	0.694578
6	0.21477	0.793158
7	0.389042	0.582909
8	0.75642	0.871272
9	1.52629	0.699271
10	2.47424	1.04646
11	4.85625	0.919178
12	9.39308	1.09489
13	20.1636	1.0029

(b) Dense polynomial instances.

d	Time R	Time M
2	0.0311613	0.233683
3	0.0624419	0.415958
4	0.131546	0.757863
5	0.248677	1.30043
6	0.460523	2.1959
7	0.999128	2.90185
8	1.50305	3.98185
9	2.25379	3.74143
10	4.51773	7.13886
11	7.196	5.48099
12	11.158	8.08422
13	17.3462	16.0432



(a) Sparse polynomial instances.



(b) Dense polynomial instances.

Figure 5.14 Plot of the data from Table 5.13.

Table 5.15 Other computed averages for dense and sparse polynomial instances.

(a) Sparse Polynomial Instances

d	Avg. No. CC	Avg. No. R	Avg. No. Max	Avg. No. Non-Max
2	2.551	3.768	3.	0.768
3	3.018	5.783	4.311	1.472
4	3.536	6.88	5.098	1.782
5	3.95195	7.97998	5.85285	2.12713
6	4.334	9.01103	6.55366	2.45737
7	4.63089	9.90672	7.13139	2.77533
8	4.97588	10.6472	7.68141	2.96583
9	5.16298	11.1499	8.02414	3.12575
10	5.42626	11.7535	8.45556	3.29798
11	5.54719	12.3614	8.80522	3.55622
12	5.73467	12.6995	9.04322	3.65628
13	5.91423	13.4834	9.4995	3.98385

(b) Dense Polynomial Instances

d	Avg. No. CC	Avg. No. R	Avg. No. Max	Avg. No. Non-Max
2	2.52405	3.67936	2.94188	0.737475
3	2.5982	5.48497	3.89178	1.59319
4	2.91247	7.04628	4.69618	2.3501
5	2.89458	8.62851	5.4739	3.15462
6	3.136	9.93	6.123	3.807
7	3.16249	11.2919	6.80642	4.48546
8	3.38677	12.4649	7.37275	5.09218
9	3.40641	13.1131	7.71371	5.3994
10	3.64257	13.8976	8.10241	5.79518
11	3.57415	14.4269	8.3507	6.07615
12	3.76253	15.006	8.67936	6.32665
13	3.6593	15.4412	8.83216	6.60905

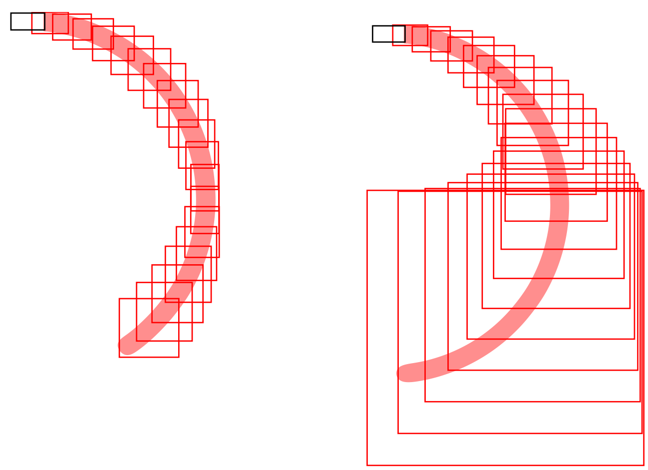
Chapter 6

Conclusion and Outlook

In this thesis we presented an algorithm **Connectivity** for determining whether two points lie in a same connected component of a semi-algebraic set defined by a single polynomial inequation. We proved the method to be partially correct using modified results from Morse theory assuming the correctness of a certified numeric subalgorithm **Destination**. Furthermore, we showed the algorithm terminates using results from semi-algebraic geometry. We gave an upper bound on the length of a connectivity path connecting two input points lying in a connected component of $\{g \neq 0\}$. To illustrate the efficacy of our method, we presented several non-trivial examples and used numerical experiments.

There are several future research topics which are related or motivated by the ideas in this thesis. As mentioned previously, we plan to describe the steps for **Destination** in a future paper. One possible implementation would require that we trace the steepest ascent paths using outgoing eigenvectors in a rigorous manner. A possible way to due this is using interval based methods [Moo09]. Researchers have used approaches like this in the past [Veg12], however their methods would need to be adapted carefully for our problem.

The steepest ascent paths we need to trace are solutions to an autonomous ordinary differential equation with initial value. Techniques have already been developed [MB03; Ned99] that allow us to put certified boxes enclosing the steepest ascent path, such as those seen in Figure 6.1a. Ideally we would like the boxes as small as possible, however, one major problem with verified integration is the *wrapping effect*. Illustrated in Figure 6.1b, the wrapping effect is a blow up in the size of the enclosures due to repeated arithmetic operations with intervals. Early computational tests show that when calculating steepest ascent paths using outgoing eigenvectors, the wrapping effect causes an unfavorable buildup of errors in the long term. More work needs to be done in this area.



(a) **(b)**

Figure 6.1 Interval ODE enclosures.

An alternative approach to implementing **Destination** could come from the field of dynamical systems. Much research has been done on how to compute invariant manifolds with several methods focusing on computing rigorous enclosures of (un)stable manifolds (see [Kra05] for a survey of such methods).

A second research direction is to improve the length bound given in Chapter 4. A reasonable first step would be to improve the radius of the bounding ball and the lower bound on the critical values of the routing points. We also want to remove the assumption that the gradient extremal of g is a compact rectifiable curve. Computational tests seem to suggest that that because g is a routing function, this assumption is already true.

A third research problem is to perform a full complexity analysis of **Connectivity**. Certainly this is not possible until an implementation of **Destination** has been fixed. To be competitive with existing methods that solve the connectivity problem, we hope **Connectivity** has complexity that is singly exponential. Once a full complexity analysis is done, we would like to generalize the method discussed in this thesis to help answer connectivity queries on smooth bounded semi-algebraic sets.

One last research direction is to focus on developing an algebraic path connecting any two points in a connected component of $\{f \neq 0\}$. In a recent paper [FK13], researchers have been answering connectivity queries using gradient extremal paths. It may be possible to adapt this idea.

BIBLIOGRAPHY

[BH04] Banyaga, A. & Hurtubise, D. *Lectures on Morse Homology*. Vol. 29. Kluwer Texts in the Mathematical Sciences. Kluwer Academic Publishers, 2004.

[BR13] Basu, S. & Roy, M.-F. "Divide and Conquer Roadmap for Algebraic Sets" (2013). arXiv: `1305.3211 [math.AG]`.

[Bas96] Basu, S. et al. "Computing roadmaps of semi-algebraic sets". *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. ACM. 1996, pp. 168–173.

[Bas00] Basu, S. et al. "Computing roadmaps of semi-algebraic sets on a variety". *Journal of the American Mathematical Society* **13**.1 (2000), pp. 55–82.

[Bas03] Basu, S. et al. *Algorithms in Real Algebraic Geometry*. 2nd. Vol. 10. Algorithms and Computation in Mathematics. Springer, 2003.

[Bas12] Basu, S. et al. "A baby step-giant step roadmap algorithm for general algebraic sets" (2012). arXiv: `1201.6439 [math.AG]`.

[BR10] Basu, S. & Roy, M.-F. "Bounding the radii of balls meeting every connected component of semi-algebraic sets". *Journal of Symbolic Computation* **45**.12 (2010), pp. 1270–1279.

[Can88] Canny, J. *The complexity of robot motion planning*. Cambridge, MA, USA: MIT Press, 1988.

[Can93] Canny, J. "Computing roadmaps of general semi-algebraic sets". *The Computer Journal* **36**.5 (1993), p. 504.

[Col75] Collins, G. "Quantifier elimination for real closed fields by cylindrical algebraic decompostion". *Automata Theory and Formal Languages 2nd GI Conference Kaiserslautern, May 20–23, 1975*. Springer. 1975, pp. 134–183.

[CM12] Coste, M. & Moussa, S. "Geodesic diameter of sets defined by few quadratic equations and inequalities". *Mathematische Zeitschrift* **272**.1-2 (2012), pp. 239–251.

[Cro68] Crofton, M. W. "On the theory of local probability, applied to straight lines drawn at random in a plane; the methods used being also extended to the proof of certain new theorems in the integral calculus". *Philosophical Transactions of the Royal Society of London* **158** (1868), pp. 181–199.

[DK04] D'Acunto, D. & Kurdyka, K. "Bounds for gradient trajectories of polynomial and definable functions with applications". 2004.

[DK05] D'Acunto, D. & Kurdyka, K. "Effective Lojasiewicz Gradient Inequality for Polynomials". *Preprint* (2005).

[DK06] D'Acunto, D. & Kurdyka, K. "Bounds for gradient trajectories and geodesic diameter of real algebraic sets". *Bulletin of the London Mathematical Society* **38**.6 (2006), pp. 951–965.

[Don96] Donaldson, S. "Symplectic submanifolds and almost-complex geometry". *J. Differential Geom* **44**.4 (1996), pp. 666–705.

[Emi10] Emiris, I. Z. et al. "The DMM bound: Multivariate (aggregate) separation bounds". *ISSAC*. Ed. by Watt, S. Munich, Germany: ACM, 2010, pp. 243–250.

[Fed96] Federer, H. *Geometric Measure Theory*. Springer, 1996.

[FK13] Filippidis, I. & Kyriakopoulos, K. J. "Roadmaps using gradient extremal paths". *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE. 2013, pp. 370–375.

[GR93] Gournay, L. & Risler, J. "Construction of roadmaps in semi-algebraic sets". *Applicable Algebra in Engineering, Communication and Computing* **4**.4 (1993), pp. 239–252.

[GV92] Grigor'ev, D. & Vorobjov, N. "Counting connected components of a semialgebraic set in subexponential time". *Computational Complexity* **2**.2 (1992), pp. 133–186.

[Hei90] Heintz, J. et al. "Single exponential path finding in semi-algebraic sets II: The general case". *Proc. 60th Birthday Conf. for S. Abhyankar*. 1990.

[HS00] Hindry, M. & Silverman, J. H. *Diophantine geometry: an introduction*. Vol. 201. Springer, 2000.

[Hof86] Hoffman, D. K. et al. "Gradient extremals". *Theoretica chimica acta* **69**.4 (1986), pp. 265–279.

[Hon10] Hong, H. "Connectivity in Semi-algebraic sets". *Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2010, 12th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*. IEEE. 2010, pp. 4–7.

[IC14] Iraji, R. & Chitsaz, H. "NuRA: Numerical Roadmap Algorithm". *CoRR* (2014).

[Ito09] Ito, D. *Robot Vision: Strategies, Algorithms and Motion Planning*. Commack, NY, USA: Nova Science Publishers, Inc., 2009.

[Kha86] Khatib, O. "Real-time obstacle avoidance for manipulators and mobile robots". *The international journal of robotics research* **5**.1 (1986), pp. 90–98.

[Kra05] Krauskopf, B. et al. "A survey of methods for computing (un) stable manifolds of vector fields". *International Journal of Bifurcation and Chaos* **15**.03 (2005), pp. 763–791.

[Lab10] Labs, O. "A List of Challenges for Real Algebraic Plane Curve Visualization Software". *Nonlinear Computational Geometry* (2010), pp. 137–164.

[Lat91] Latombe, J.-C. *Robot Motion Planning*. Norwell, MA, USA: Kluwer Academic Publishers, 1991.

[LaV06] LaValle, S. *Planning Algorithms*. New York, NY, USA: Cambridge University Press, 2006.

[LPW79] Lozano-Pérez, T. & Wesley, M. A. "An algorithm for planning collision-free paths among polyhedral obstacles". *Communications of the ACM* **22**.10 (1979), pp. 560–570.

[MB03] Makino, K. & Berz, M. "Taylor models and other validated functional inclusion methods". *International Journal of Pure and Applied Mathematics* **4**.4 (2003), pp. 379–456.

[Mat02] Matsumoto, Y. *An introduction to Morse theory*. Vol. 208. Amer Mathematical Society, 2002.

[Moo09] Moore, R. et al. *Introduction to interval analysis*. Society for Industrial Mathematics, 2009.

[Ned99] Nedialkov, N. et al. "Validated solutions of initial value problems for ordinary differential equations". *Applied Mathematics and Computation* **105**.1 (1999), pp. 21–68.

[Nic11] Nicolaescu, L. I. *An Invitation to Morse Theory*. Universitext (1979). Springer, 2011.

[Per01] Perko, L. *Differential Equations and Dynamical Systems*. 3rd. Vol. 7. Texts in Applied Mathematics. Springer-Verlag, 2001.

[Rei79] Reif, J. H. "Complexity of the movers problem and generalizations extended abstract". *Proceedings of the 20th Annual IEEE Conference on Foundations of Computer Science*. 1979, pp. 421–427.

[SEDS10] Safey El Din, M. & Schost, E. “A Baby Steps/Giant Steps Probabilistic Algorithm for Computing Roadmaps in Smooth Bounded Real Hypersurface”. *Discrete and Computational Geometry* (2010).

[SEDS13] Safey El Din, M. & Schost, É. “A nearly optimal algorithm for deciding connectivity queries in smooth and bounded real algebraic sets”. **arXiv:1307.7836** (2013).

[San04] Santaló, L. A. *Integral geometry and geometric probability*. Cambridge University Press, 2004.

[SS83] Schwartz, J. & Sharir, M. “On the piano movers problem: II. General techniques for computing topological properties of real algebraic manifolds”. *Advances in applied Mathematics* **4**.1 (1983), pp. 298–351.

[Sei54] Seidenberg, A. “A new decision method for elementary algebra”. *Annals of Mathematics* **60**.2 (1954), pp. 365–374.

[Tar51] Tarski, A. “A decision method for elementary algebra and geometry” (1951).

[Veg12] Vegter, G. et al. “Certified computation of planar morse-smale complexes”. *Proceedings of the 2012 symposuim on Computational Geometry*. ACM. 2012, pp. 259–268.