Tracing-Problems in Dynamic Geometry

(Pfadverfolgungsprobleme aus der Dynamischen Geometrie)

Dissertation zur Erlangung des Doktorgrades

vorgelegt am

Fachbereich Mathematik und Informatik
der Freien Universität Berlin

2008

von

Britta Denner-Broser

Institut für Informatik

Freie Universität Berlin

Takustraße 9

14195 Berlin

broser@inf.fu-berlin.de

zum Teil gefördert durch die DFG im Rahmen des Europäischen

Graduiertenkollegs Combinatorics, Geometry, and Computation (GRK 588/3)

Betreuer: Prof. Dr. Helmut Alt
Institut für Informatik
Freie Universität Berlin
Takustraße 9
14195 Berlin
Germany
alt@inf.fu-berlin.de

Gutachter:	Prof. Dr. Helmut Alt	Prof. Dr. Ulrich Kortenkamp
	Institut für Informatik	Institut für Mathematik und Informatik
	Freie Universität Berlin	Pädagogischen Hochschule Schwäbisch Gmünd
	Takustraße 9	Oberbettringer Straße 200
	14195 Berlin	73525 Schwäbisch Gmünd
	Deutschland	Deutschland
	alt@inf.fu-berlin.de	kortenkamp@cinderella.de

Vorlage zur Begutachtung: 29. Juli 2008
Termin der Disputation: 1. Dezember 2008

Preface

Quo vadis

This famous Latin verse goes back to the Bible[1] and can be translated as "Where are you going?". It perfectly reflects the subject of this thesis: Given is a geometric construction possibly done with a geometry software on a computer. In general, a geometric construction consists of independent objects, which are placed at arbitrary positions at the beginning, and constructed elements like a line connecting two given points. If an independent object from the construction is moved, then an important question is what happens with the remaining geometric objects? Where are they going? This problem is essential to make geometry dynamic: After making a geometric construction with a geometry software, we can move the construction by dragging independent objects with the mouse, expecting that the whole construction adjusts automatically. As usual, the devil is in the details: The two words "adjusts automatically" make the field of Dynamic Geometry difficult and thus interesting. If angular bisectors, circles or even conics are involved, then ambiguities arise. "Adjust automatically" now means to choose the "right" possibility among a certain number of choices at each time of the motion. But which choice is the "right" one? This answer is easy: The right choice is the one the user of a Dynamic Geometry Software expects. But the user could be grabby and could expect

Continuity: In the motion of the construction, no jumps occur;

Conservatism: Reversing moves leads back to the initial configuration;

Determinism: For each placement of the free objects there is at most one configuration of the construction; and

Consistency: Degenerate situations are treated consistently and, thus, the behavior at degeneracies is somehow predictable.

[1] e.g. Genesis 16:8: dixit ad eam Agar ancilla Sarai unde venis et **quo vadis** quae respondit a facie Sarai dominae meae ego fugio; He said, "Hagar, servant of Sarai, where have you come from, and **where are you going**?" She replied, "I'm running away from my mistress, Sarai.", Judges 19:17, or John 13:36.

These are four wishes, and unfortunately it is not possible to satisfy all of them simultaneously. In my opinion, the first wish is the most important one since unmotivated and unpredictable jumps are difficult to handle. The last wish "Consistency" is related to "Continuity". In this thesis, I consider the beautiful model for Dynamic Geometry that has been developed by Kortenkamp and Richter-Gebert as the foundation of the interactive Geometry Software *Cinderella*. The continuity of the induced motions is the natural center of the mathematical model. The consistent treatment of critical points is achieved by the creative idea to allow complex coordinates for the geometric objects. In my thesis, I investigate an algebraic variant of this model for Dynamic Geometry. I greatly enjoyed to see that this nice model perfectly fits into the areas of pure and applied mathematics! Based on this knowledge, we (Ulli and myself) found a reliable algorithm for a continuous realization of the drag mode. Ulli told me about Dynamic Geometry at a party at Stefan Felsner's, and I am really glad that Stefan invited both of us! This party was the beginning of my research in the field of Dynamic Geometry.

I would like to thank Ulli and Helmut for their friendly, persistent and very constructive support! I would like to say a special thank you to Ulli for all the interesting, motivating and helpful discussions, he *always* took time for me when I needed some help! He directed my attention to the area of Dynamic Geometry and proposed the subject of my thesis together with Helmut. I would like to give sincere thanks to Helmut for the fruitful discussions, he suggested to consider Tarski-formulas. This idea led to our proof of the decidability of the Tracing Problem and the Reachability Problem by considering semi-algebraic sets. This result was an important step of my thesis. Helmut and Ulli always encouraged and supported me to run my family and to write my thesis! Their attitude and support was and is very helpful for my whole family! I would like to thank all members of the research group "Theoretical Computer Science" of FU-Berlin for the very nice research atmosphere, special thanks to Géraldine, Tobias and Jens who shared and share the office with me! Particular thanks to Domique Michelucci (Université de Bourgogne) and Martin Weiser (Zuse Institut Berlin), who proposed independently to use interval arithmetic for solving the Tracing Problem! I greatly appreciate Hanfried Lenz (born 1916), he immediately answered my e-mail and recommended to look at Bieberbach's book about geometric constructions. I would like to thank the members of the Department of Mathematics and Computer Science of FU-Berlin. I had the opportunity to borrow a large number of famous old books from the library of our department.

At this point, I would particularly like to thank Elmar Vogt and Karin Gatermann, my mathematical knowledge is based on their lectures, seminars, and my master's thesis (Diplomarbeit)! I am very sad that Karin died on the first of January 2005. I would like to thank Dirk and Elke for very helpful proofreading ! I also would like to say a special thank you to Marco, Stephan and Beni for their friendship and for all the lectures and seminars we attended together! Grit gives me whenever necessary a chance to escape the stressful everyday life, thanks a lot! I would like to thank Dirk for very careful proofreading!

Special thanks to my whole family, who supported me all the time. My parents abandoned one week of their holidays to enable me to finish this thesis. My grandmother (born 1921) took care of Lisa when she was ill, my parents in law always walk the dog with Lisa and Svenja when the girls feel too sick for visiting preschool. My sister Tanja supports me with her amazing educational background, she found the Bible verses for this preface. Similar to our situation, my brother Matthias and his friend Felissa manage their (PhD-) studies and a family, as well. This common background is very helpful! At this point, I would like to commemorate my three grandparents, who have passed away. Opa Waldemar would have been very happy to witness the success of my PhD-studies at FU-Berlin.

In particular, I would like to thank my husband Philip and our children Lisa and Svenja, they had to put a lot of effort in my thesis, as well. During the last months, Philip had to combine his full-time position in a pharmaceutical company with two children and a wife, who was always busy with her thesis. Lisa and Svenja had to abandon a lot of time[2] with me and had a very impatient mother in the last months! To compensate this lack of attention, Lisa visited us almost every night.

[2]Lisa: "Mama, ich fahre jetzt ganz langsam, weil ich noch ein bisschen bei Dir sein möchte. Heute ist doch der letzte Tag, an dem Du nicht nach Hause kommst!"; Svenja: "Mama Uni!"

Zusammenfassung

Unter dynamischer Geometrie versteht man das interaktive Erstellen von geometrischen Konstruktionen am Computer. Ein Dynamisches Geometrie System ist ein Geometriesystem, in dem es möglich ist, geometrische Konstruktionen durchzuführen, und das einen Zugmodus hat. Im Zugmodus können geometrische Objekte, die mindesten einen Freiheitsgrad haben, mit der Maus bewegt werden. Dabei paßt sich die gesamte geometrische Konstruktion der Bewegung an, indem der Computer das entstehende Pfadverfolgungsproblem löst. In dem von uns verwendeten Modell für dynamische Geometrie steht die Stetigkeit der resultierenden Bewegungen im Vordergrund, es wurde von Kortenkamp und Richter-Gebert entwickelt und ist die Grundlage für die Geometriesoftware Cinderella. Wir arbeiten den Zusammenhang dieses Modells zu Riemannschen Flächen algebraischer Funktionen heraus.

Im Rahmen dieser Doktorarbeit zeigen wir, wie sich eine algebraische Variante des Modells für Dynamische Geometrie von Kortenkamp und Richter-Gebert sowohl in die angewandte als auch in die reine Mathematik einfügt. Daraus resultiert ein numerisches Verfahren für das Tracing Problem, das auf einer allgemeinen Prediktor-Korrektor-Methode aufbaut. Wie bei den meisten numerischen Verfahren gibt es hierbei keine Garantie dafür, dass die Schrittweite klein genug gewählt ist, um auf dem richtigen Lösungsweg zu bleiben. Das bedeutet, dass ein korrekter Umgang mit Mehrdeutigkeiten nicht garantiert werden kann. Wir haben einen weiteren Algorithmus entwickelt, bei dem die Schrittweite mit Hilfe von Intervallrechnung so gewählt wird, dass die Korrektheit der Lösung garantiert ist. Kritische Punkte werden durch Umwege umgangen, bei denen die geometrischen Objekte bzw. die entsprechenden Variablen in einem algebraischen Modell *komplexe* Koordinaten haben können. Dabei hängt die erreichte Konfiguration wesentlich von dem gewählten Umweg ab. Diese Idee von Kortenkamp und Richter-Gebert führt zu einer konsistenten Behandlung von kritischen Punkten und kommt in der interaktiven Geometriesoftware Cinderella zum Einsatz.

Abstract

Dynamic Geometry is the field of interactively doing geometric constructions using a computer. Usually, the classical ruler-and-compass constructions are considered. The available tools are simulated by the computer. A Dynamic Geometry System is a system to do geometric constructions that has a drag mode. In the drag mode, geometric elements with at least one degree of freedom can be moved, and the remaining part of the geometric construction adjusts automatically. Thus, the computer has to trace the paths of the involved geometric objects during the motion.

In this thesis, we focus on the beautiful model by Kortenkamp and Richter-Gebert that is the foundation of the geometry software *Cinderella*. We embed an algebraic variant of this model into different fields of pure and applied mathematics, which leads to different approaches for realizing the drag mode practically. We develop a numerical method to solve the Tracing Problem that is based on a generic Predictor-Corrector method. Like most numerical methods, this method cannot guarantee the correctness of the computed solution curve, hence ambiguities are not treated satisfactorily. To overcome this problem, we develope a second algorithm that uses interval analysis. This algorithm is robust, and the computed step length is small enough to break up all ambiguities. Critical points are bypassed by detours, where the geometric objects or the corresponding variables in the algebraic model can have *complex* coordinates. Here, the final configuration depends essentially on the chosen detour, but this procedure due to Kortenkamp and Richter-Gebert leads to a consistent treatment of degeneracies.

We investigate the connection of the used model for Dynamic Geometry to Riemann surfaces of algebraic functions.

Contents

Chapter 1

Introduction

1.1 What is Dynamic Geometry?

Dynamic Geometry is the field of interactively doing geometric constructions using a computer. Usually, the classical ruler-and-compass constructions are considered. The available tools are simulated by the computer. A *Dynamic Geometry System* (DGS) is a system to do geometric constructions that has a "drag mode" (German: Zugmodus). In the drag mode, geometric elements with at least one degree of freedom can be moved, and the remaining part of the geometric construction adjusts automatically; see Figure 1.1. Thus, the computer has to *trace* the paths of the geometric objects involved during the motion.

The main application of Dynamic Geometry is in math education. Doing geometric constructions with a computer avoids inaccuracies due to an imprecise usage of the construction tools like ruler, compass and the used pen. The drag mode allows to develop a better intuition for geometric coherences and helps to understand whether a conjectured theorem is true or false. If you start a geometric construction with an arbitrary quadrilateral, then most people draw an "almost" square. Without noticing it, the drawn shape has some kind of symmetry. This phenomenon makes it difficult to decide whether an observed property of a geometric construction that is based on this particular shape is a universal property of the shape or not. If we consider the two diagonals of the drawn quadrilateral, we could observe that they bisect each other. This is a property of parallelograms that does not hold for general quadrilaterals; see Figure 1.2.

Dynamic Geometry is an important tool for scientists, as well. One application is the investigation of rigid motions of mechanisms [24, 42]. The enhancements of the Dynamic Geometry Systems enlarge the range of use. The additional functionality includes the treatment of conics, plotting loci of constructed points,

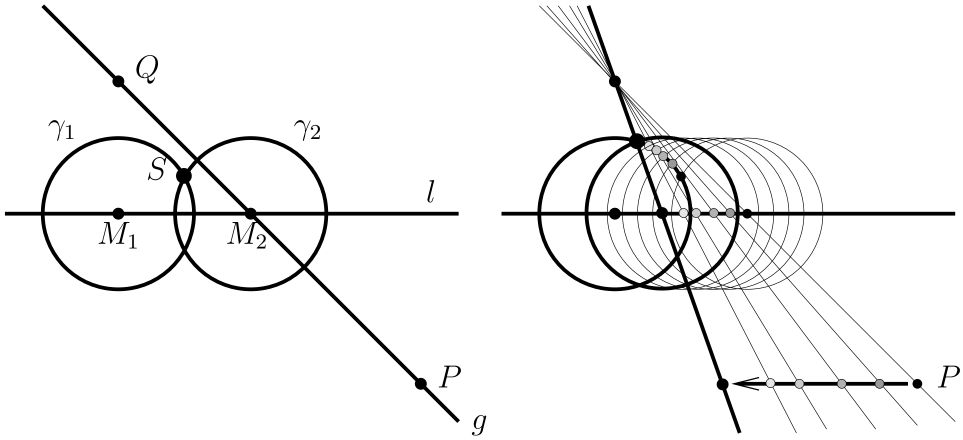


Figure 1.1: The left figure shows a geometric construction. The line g connects the two points P and Q. The intersection point M_2 of g and the line l is the center of the circle γ_2. The point S is the upper intersection point of the two circles. The right figure visualizes the drag mode: If the point P is moved, then the entire construction follows the motion of P.

dealing with transformations, doing (simple) simulations of physical laws, the connection to computer-algebra, or the possibility to switch to other geometries [42, 41].

To implement the drag mode in a Dynamic Geometry System is a challenging task. The main problem is to deal with the ambiguities due to the involvement of circles or angular bisectors. For example, a circle and a line intersect in two, one or no points, two intersecting lines have two angular bisectors, which are orthogonal. These ambiguities lead to the necessity to choose one of the two possible intersection points or angular bisectors. In the initial drawing, the user of the Dynamic Geometry Software makes these choices. In the drag mode, the computer has to make these decisions autonomously. At each time of the motion, the computer should come to the same decision as the user of the software would come to in the current configuration. Thus, we expect that there are no "jumps" in the motions of the geometric objects, and we prefer a continuous behavior of the Dynamic Geometry System.

In fact, to realize the drag mode, we have to solve a path tracking problem. This problem is called the *Tracing Problem from Dynamic Geometry*. We are given a starting configuration that was fixed by the user. Additionally, we are given an abstract description of the geometric construction and paths of the free objects; they implicitly define the resulting motion of the dependent elements. The starting configuration acts as an initial value.

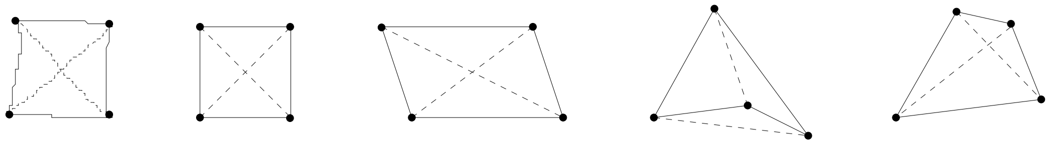


Figure 1.2: Different quadrilaterals with their diagonals are shown.

The field of Dynamic Geometry has been founded in the late eighties and the beginning of the nineties of the twentieth century. At this time, the first geometry softwares Cabri Géomètre and Geometer's Sketchpad were developed that realize the drag mode. Up to now, several software packages for Dynamic Geometry have been designed. They all have the functionality to do geometric constructions and provide a drag mode. Additionally, they have different features as described above. However, their drag modes are implemented in different ways and have different properties that arise from the underlying mathematical models.

In this thesis, we focus on the beautiful model by Kortenkamp and Richter-Gebert [40, 43] that is the foundation of the geometry software *Cinderella* [42, 41] and that leads to a continuous behavior of the drag mode. We embed this model into different fields of pure and applied mathematics, which leads to different approaches for realizing the drag mode practically.

1.2 A Model for Dynamic Geometry

The results of this thesis are based on the model for Dynamic Geometry developed by Kortenkamp and Richter-Gebert [40, 43], which is the theoretical foundation of the geometry software *Cinderella* [42, 41]. We explain this model and some resulting problems in this section. A detailed description is given in Chapter 3.

In Dynamic Geometry, the step-by-step-procedure of doing geometric constructions can be represented by *Geometric Straight-Line Programs* (GSP). A GSP is a sequence of instructions that describe the single construction steps or introduce new, independent points; see Figure 1.3. These independent points are called free points. The remaining geometric objects are called dependent elements, they are defined by instructions that belong to a construction step like the computation of a line connecting two points, the intersection point of two lines, or one of the at most two intersection points of a line and a circle. Throughout this thesis, let k be the number of free points in the GSP and n the number of dependent elements.

An instance of a GSP is an assignment of fixed values to all free parameters and choices; see [43, 44]. Thus, instances of a GSP correspond to concrete drawings of the underlying geometric construction. To deal with singularities like the intersection of two identical circles or of a circle and a tangent is important, as well. The corresponding point in the configuration space is called a *critical point*.

Since we work on Dynamic Geometry, we have to formalize movements of constructions. This is done via *continuous evaluations* [43]: Continuous paths $p_l(t)$, $t \in [0,1]$ and $l = -k+1,\ldots,0$, of the free points are given. A continuous evaluation under the movement $\{p_l\}$ is an assignment of continuous paths v_i, $i = 1,\ldots,n$, to all the dependent elements, such that the objects

$$(p_{-k+1}(t),\ldots,p_0(t),v_1(t),\ldots,v_n(t))$$

form an instance of the GSP for all times $t \in [0,1]$. Thus a Dynamic Geometry System that is based on the notion of continuous evaluations supports continuous motions of constructions.

There are two problems arising naturally from this setup:

Problem 1. *(Reachability Problem)*
Two instances A and B of a GSP are given, where A is called a starting instance and B a final instance.
Decide whether there are paths $\{p_l\}$ of the free points for which a continuous evaluation from A to B exists.

Problem 2. *(Tracing Problem)*
As in the Reachability Problem, we are given a starting instance A and a final instance B. Let p_A be the position of the free points at instance A, and p_B their position at B. Furthermore, a movement $\{p_l\}$ of the free points from p_A to p_B is given. We assume that the resulting continuous evaluation starting at A exists.
Decide: Does this continuous evaluation end at the given instance B?

In [43], Kortenkamp and Richter-Gebert show by a reduction of 3-SAT that these two problems are NP-hard (in $\mathbb{R}$). In [13], decidability aspects are discussed. In this work, we focus on the Tracing Problem and develop different approaches. We give an algorithmic solution with interval analysis [14]; see [1, 50] for standard references for interval arithmetic.

We have to solve the Tracing Problem to realize the drag mode. Assume that a user of a Dynamic Geometry software has done a geometric construction. Then what is seen on the screen is an instance A of the GSP describing the underlying construction. Now the user drags a free point with his mouse. This describes a continuous path of the free points: The path of the dragged point is piecewise linear whereas the paths of the other free points are constant. The Dynamic Geometry software has to decide which instance B should be drawn after the motion (and at all intermediate positions).

$$
\begin{aligned}
P &\leftarrow \text{FREE}\\
l_x &\leftarrow \text{JOIN}((0,0),(-1,0))\\
l_y &\leftarrow \text{JOIN}((0,0),(0,1))\\
l &\leftarrow \text{JOIN}((0,1),P)\\
Q &\leftarrow \text{MEET}(l,l_x)\\
C_1 &\leftarrow \text{CIRCLE}((0,0),\tfrac{1}{2})\\
C_2 &\leftarrow \text{CIRCLE}(Q,\tfrac{1}{2})\\
M &\leftarrow \text{MEET}(C_1,C_2)
\end{aligned}
$$

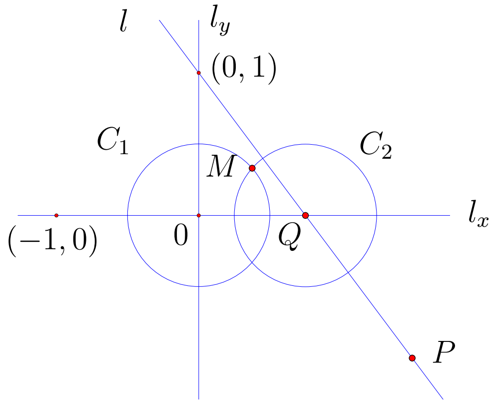


Figure 1.3: A GSP Γ and an instance of Γ are shown. By construction, Q has to stay on the line l_x.

A solution to the Reachability Problem could help in automated theorem proving [62, 45]. Here a geometric problem is given as an instance A of a certain GSP. If a probabilistic approach is used as in [40], we have to create instances B of the geometric problem at random. For generating an instance, we have to assign fixed values to all free parameters *and* to the choices; e.g. we have to specify which intersection point of a line and a circle is chosen. If we can reach the instance B with a continuous evaluation starting at A, then we either have found a counterexample and know that the conjecture is wrong or a "positive example", which increases the probability that the conjecture holds; see [40].

Allowing Complex Coordinates. In some situations, considering complex coordinates of the objects is useful as well. The complex Tracing Problem occurs when we are dealing with singularities like the intersection of two identical circles or of a circle and a tangent: Depending on the path p of the free variables, tracing along p might force us to consider such degenerate situations; see Figures 1.3 and 1.4: Let the free point P move on the linear path $p(t) = \binom{1.5}{-1} + t \cdot \binom{-3}{0}$. Then at time $t_0 = \frac{1}{2}$ the point P lies on the y-axis l_y. Hence, Q reaches the origin, and the two circles are identical. Thus we have a degenerate situation and the intersection point M is not defined.

To avoid a singularity $S = p(t_0)$, we might bypass it with a detour by modifying p in a neighborhood of t_0. Unfortunately, this might be impossible as our example shows; see Figures 1.3 and 1.4: By construction, the dependent point Q is always incident to the line l_x, and the singularity occurs when Q is moved to the origin. Since Q has to stay on l_x by construction, the singularity S cannot be avoided by modifying the path p of the free point P. A way out of this problem is to consider detours for that the coordinates of the free points may have non-real coordinates.

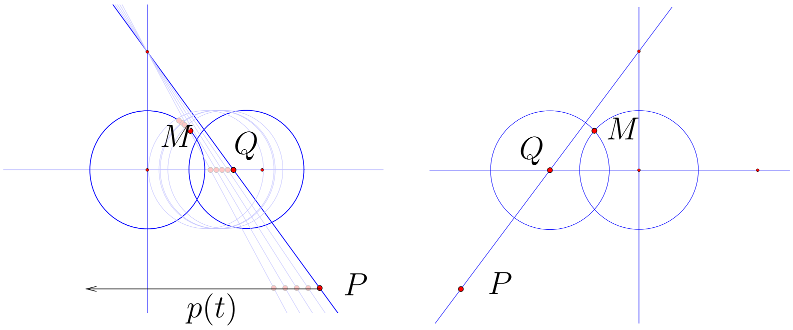


Figure 1.4: A singularity occurs if the free point P moves on the path $p(t)$ across the y-axis.

Now the line l_x of our example becomes a two-dimensional object, and the point Q might bypass S without leaving l_x. Additionally, if the singularity is *removable*, the instance being reached after finishing the detour does not depend on the detour itself (as long as the detour does not "catch" other singularities). This is an application of the complex Tracing Problem since we have to trace the complex detour instead of the original real path p of the free point P.

The complex Reachability Problem seems to be useful for automated theorem proving. This is due to the fact that if a theorem holds over $\mathbb{C}$ then it also holds over $\mathbb{R}$.

Concerning the Tracing Problem, switching to complex coordinates does not cause many changes since we are tracing given paths. In contrast to this, the Reachability Problem over $\mathbb{R}$ and the complex Reachability Problem are somehow different problems; see Figure 1.4: In the real situation, the right instance in Figure 1.4 cannot be reached from the left one, whereas it can be reached via a complex path as described previously. The consequences of using complex coordinates are discussed in Section 2.6.

Switching to an Algebraic Model. The geometric situation translates easily into an algebraic model where the objects are numbers (real or complex) and the operations are addition, subtraction, multiplication, division, and taking square roots; see Chapter 2. In this thesis, we discuss algorithms for the Tracing Problem and the Reachability Problem in this algebraic model. We focus on the Tracing Problem.

Model of Computation. In our algorithms, we assume that each arithmetic operation and each interval operation is executed exactly and in constant time.

1.3 Properties and Advantages of the Chosen Model for Dynamic Geometry

In the chosen model for Dynamic Geometry, the focus lies on a *continuous* behavior of the Dynamic Geometry System (DGS) and on a *consistent* treatment of degeneracies. A DGS is a system to do geometric constructions that has a "drag mode" (German: Zugmodus).

The drag mode allows the user of a Dynamic Geometry Software to move geometric objects with at least one degree of freedom. The resulting path of the picked object induces a motion of the *entire* construction. In fact, the motion of the entire construction is composed by several paths: For every geometric object, we have exactly one path that describes the motion of the corresponding object. Non-moving elements belong to constant paths. At each time of the motion, the constraints of the geometric construction must be fulfilled. Additionally, we prefer a continuous behavior of the DGS: If the picked object is moved on a continuous path, then the induced motion of the construction should be continuous as well. The concept of *continuity* avoids "jumps" of the construction.

In our model for Dynamic Geometry that has been developed by Kortenkamp and Richter-Gebert [40, 43], the movements of geometric constructions are modeled via *continuous evaluations*. This notion formalizes and combines the concepts explained above and guarantees a continuous behavior of the DGS. If a "free" object is moved on a continuous path, then the remaining objects have to move on continuous paths as well, and at each time the given constraints are fulfilled; see page 4 and Section 3.3. The only constraints that are fixed in our model are the *relations* between the geometric objects. We neither bound nor cut the ranges of the geometric objects. In particular, we consider the configuration space as a whole and do not split it up into regions that are for example separated by degeneracies. These cuts conflict with a global continuous behavior of the DGS, since we might be in trouble if the given motion crosses some of these cuts.

The idea of detouring around critical points as introduced by Kortenkamp [40] leads to a *consistent* treatment of degeneracies; see Section 7.2. Consistency is achieved since all degeneracies that occur at a certain time in the motion are treated simultaneously in the same manner. The idea behind this concept is that the path $p(t)$ of the driving free element is changed by replacing the time parameter t with

$$\begin{array}{rcl} t(s)\colon [0,1] & \to & \mathbb{C} \\ s & \mapsto & \frac{1-e^{i\pi s}}{2} \end{array} \qquad \text{resp. } \begin{array}{rcl} t(s)\colon [t_1,t_2] & \to & \mathbb{C} \\ s & \mapsto & t_1 + \frac{1-e^{i\pi s}}{2}(t_2-t_1) \end{array} \qquad (*)$$

We describe the concept of detouring around critical points in Example 1.3.1. Note that here only a single degeneracy occurs.

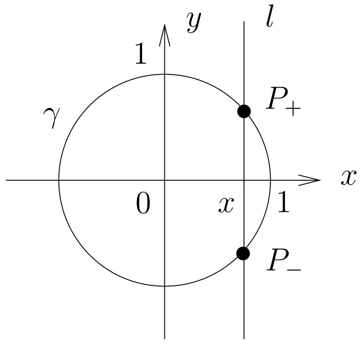


Figure 1.5: The line l intersects the circle γ in two points P_+ and P_- if $-1 < x < 1$.

Example 1.3.1. We consider the following geometric construction shown in Figure 1.5. Let γ be the circle with radius 1 having the origin $(0,0)$ as midpoint. Let l be the line that is parallel to the y-axis and that passes through the point $(x,0)$. For $-1 < x < 1$, the line l and the circle γ have two intersection points P_+ and P_-. Let P_+ be the intersection point that lies above the x-axis. Thus, $P_+ = (x, +\sqrt{1-x^2})$ and $P_- = (x, -\sqrt{1-x^2})$. At the beginning, we set $x = -1/2$ and consider the path $p_1\colon [0,1] \to \mathbb{R}$, $t \mapsto -1/2 + t$. If x moves on p_1, then the intersection point P_+ moves on the path $t \mapsto (p_1(t), +\sqrt{1+(p_1(t))^2}\,) = (-1/2 + t, +\sqrt{3/4+t-t^2}\,)$, and P_- moves on the path $t \mapsto (-1/2+t, -\sqrt{3/4+t-t^2}\,)$. Note that the paths of P_+ and P_- only differ in the sign of the y-coordinate.

Now let $p_2[0,1] \to \mathbb{R}$, $t \mapsto 1/2 + t$. If x moves on p_2, then at time $t = 1/2$ the line l is tangent to γ and we have $P_+ = P_-$. Instead of $p_2(t)$ we consider the path $\hat{p}_2(s) := p_2(t(s))\colon [0,1] \to \mathbb{C}$, $s \mapsto p_2(t(s)) = 1/2 + (1 - e^{i\pi s})/2 = 1 - e^{i\pi s}/2$ describing a semi-circle in the complex plane $\mathbb{C}$. Note that in most applications, we choose much smaller detours around critical points; this example is used to illustrate the ideas. Following the path $p_2(t(s))$, the point P_+ moves on the path $t \mapsto (p_2(t(s)), +\sqrt{1-(p_2(t(s)))^2}\,) = (1 - e^{i\pi s}/2, +\sqrt{e^{i\pi s} - e^{2i\pi s}/4}\,)$, and P_- moves on the path $t \mapsto (1 - e^{i\pi s}/2, -\sqrt{e^{i\pi s} - e^{2i\pi s}/4}\,)$. These new paths do not pass through the degeneracy $x_0 = 1$; they end up in the points $(2/3, i\sqrt{5}/2) \in \mathbb{R} \times \mathbb{C}$ and $(2/3, -i\sqrt{5}/2)$, respectively. Since the degeneracy $x_0 = 1$ is avoided, we can always distinguish the two points P_+ and P_-; see Figure 1.6.

What happens if we reverse the two moves induced by p_1 and p_2? Reversing the moves means to first traverse p_2 and then p_1 in converse direction. Thus we consider the paths $-p_2\colon [0,1] \to \mathbb{R}$, $t \mapsto p_2(1-t) = 3/2 - t$, and $-p_1\colon [0,1] \to \mathbb{R}$, $t \mapsto p_1(1-t) = 1/2 - t$. Since the path $-p_2$ passes through the critical point, we switch to the path $\tilde{p}_2(s) := (-p_2)(t(s))\colon \to \mathbb{C}$, $s \mapsto (-p_2)(t(s)) = 1 + e^{i\pi s}/2$. The starting position of P_+ in the backward move is the final position $(2/3, i\sqrt{5}/2)$ of the forward move of P_+. Altogether, we have the composed path $p_1 + p_2 - p_2 - p_1$.

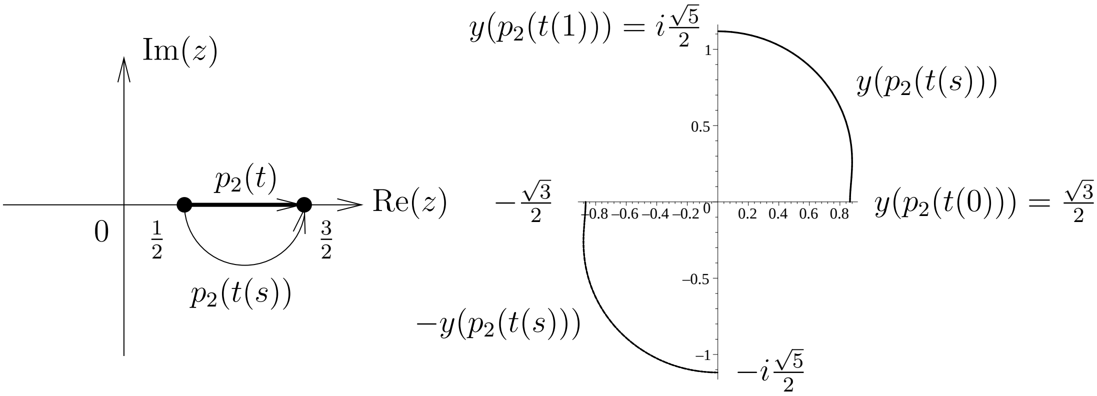


Figure 1.6: The left figure shows the path $\hat{p}_2(s) := p_2(t(s))$, the right figure displays the paths $\pm\sqrt{1-\hat{p}_2(s)^2} = y(\hat{p}_2(s))$ with $y(x) = \sqrt{1-x^2}$ that describe the paths of the y-coordinates of the intersections points P_+ and P_-. At each time s, the two paths and consequently the two points P_+ and P_- are separated.

Since we replaced $p_2(t)$ by $p_2(t(s))$ and $(-p_2)(t)$ by $(-p_2)(t(s))$, we have the path $p_1 + \hat{p}_2 + \tilde{p}_2 - p_1$. We observe that the path $\hat{p}_2 + \tilde{p}_2$ winds around the singularity at $x = p_2(1/2) = 1$ once; see Figure 1.7. This singularity is a branch point of the algebraic function $y = y(x) = \pm\sqrt{1-x^2}$. Thus, the branch of the root function of $1 - x^2$ is changed during the motion, and the points P_+ and P_- are swapped; see Figure 1.7. Consequently, following the path $p_1 + p_2 - p_2 - p_1$ in the DGS interchanges the two intersection points P_+ and P_- as well.

We have seen in Example 1.3.1 that reversing motions does not always lead back to the initial position, hence our model does not show a *conservative* behavior [40, Sect. 5.2]. However, this is a overhasty conclusion. A closer look shows that applying our rule for surrounding critical points to the paths p_2 and $-p_2$ leads to the paths $\hat{p}_2$ and $\tilde{p}_2$ with $\hat{p}_2 \neq -\tilde{p}_2$; see Figure 1.7. Thus, we do not go back and forth but follow paths that have different traces. This leads to a paradox: If a user of a Dynamic Geometry Software that is based on our model for Dynamic Geometry moves a construction and reverses the move afterward, then the final position can differ from the initial position. Macroscopically, the DGS does not show a continuous behavior. Since continuous evaluations are unique as long as no critical points are hit, the system must be conservative, microscopically: If we reverse the paths that have been traced in reality, then the system returns to the initial configuration [40].

The advantage of this fixed rule for detouring is the *consistent* treatment of degeneracies. We illustrate this important property with Example 1.3.2 shown in Figure 1.8 that is taken from [40].

Example 1.3.2. Let M_1 and M_2 be two distinct points on the x-axis l_x. Let γ_1

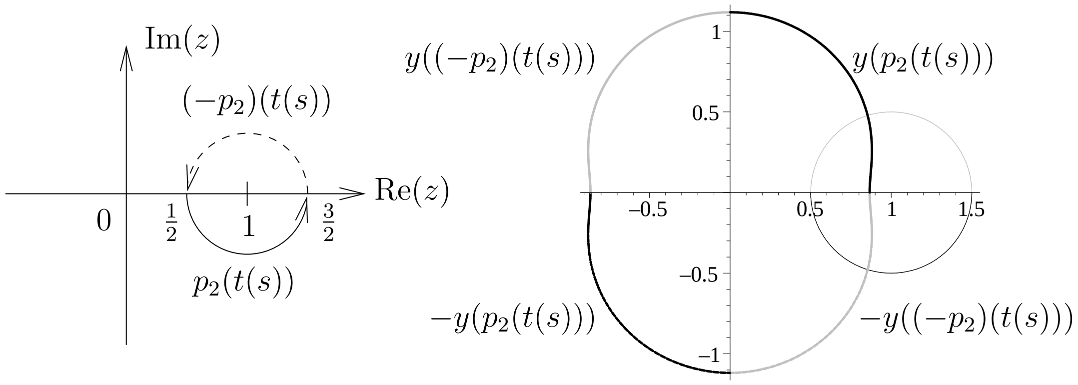


Figure 1.7: The left figure shows the paths $\hat{p}_2(s) = p_2(t(s))$ and $\tilde{p}_2(s) = (-p_2)(t(s))$. The right figure shows the path of the y-coordinate of the point P_+ in the motion induced by $\hat{p}_2 + \tilde{p}_2$ (upper dome) and the corresponding path of P_- (lower dome). Observe that the intersection points P_+ and P_- are swapped by this motion.

and γ_2 be two circles with radius 1 having M_1 and M_2 as centers, respectively. Let l be the line that is parallel to the x-axis and that passes through the point $Y = (0, y)$. For $-1 < y < 1$, each of the two circles γ_1 and γ_2 has two intersection points with l. Let P_1 be the rightmost intersection point of γ_1 and l and P_2 the rightmost intersection point of γ_2 and l. For $y = \pm 1$, the line l is tangent to the two circles and we have a degenerate situation. Starting at a position of Y with $-1 < y < 1$, we move the line l up beyond the degeneracy at $y = 1$ and down again. After this concatenated motion, the two points P_1 and P_2 are the leftmost intersection points of the circles γ_1 and γ_2 with the line l. This happens, since both points P_1 and P_2 have a degenerate position at the same time and since both degeneracies are treated in the same way. The behavior of the intersection points of a line and a single circle is described in Example 1.3.1.

We have seen that formula $(*)$ from page 7 leads to local and global consistency: First, similar situations are *always* treated similarly. Second, at each time we treat *all* degeneracies of the entire construction in the same way as demonstrated with Example 1.3.2.

Consistency is strongly related with continuity. To see this, we consider the distance between P_1 and P_2, which is the difference $x_2 - x_1$ of the x-coordinates x_1 and x_2 of P_1 and P_2 if γ_2 is the rightmost circle. This distance equals the distance $m_2 - m_1$ of the two midpoints $M_1 = (m_1, 0)$ and $M_2 = (m_2, 0)$. Note that we have $P_1 = (m_1 + \sqrt{1 - y^2}, y)$ and $P_2 = (m_2 + \sqrt{1 - y^2}, y)$. We observe that even if we follow the complex detour around the degeneracy as described above, this distance remains constant since the "choice" for the square root operation

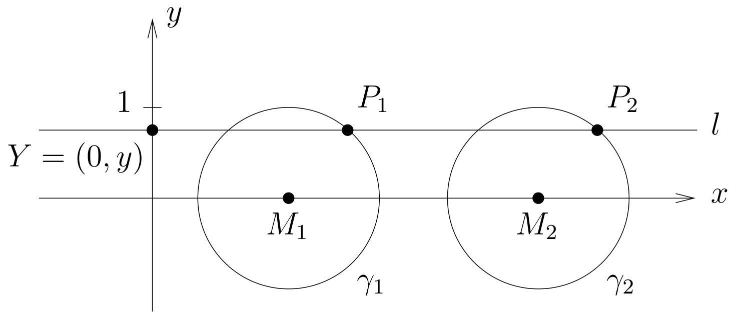


Figure 1.8: The line l is parallel to the x-axis and passes through the point $Y = (0, y)$; l intersects the two circles γ_1 and γ_2 with radius 1 in the points P_1 and P_2. After moving l above the point $(0, 1)$ and back again, the points P_1 and P_2 have changed due to detouring around the degeneracy at $y = 1$.

must be the same for both points P_1 and P_2 at each time of the motion. The used branch of the square root function changes simultaneously. In contrast to this, a non-consistent treatment of the degeneracy at $y = 1$ would lead to a change of the distance function: After the motion, the distance between P_1 and P_2 would be changed, and it would no longer describe a constant function. This dramatic change of the structure due to a non-consistent treatment of singularities could lead to a non-continuous behavior of the Dynamic Geometry System.

Another property of DGSs is *determinism*, where we require that for every given position of the geometric objects with at least one degree of freedom, there is at most one configuration of the corresponding construction. This property contradicts the concept of continuity if angular bisectors, circles or conics are involved [40, Sect. 6.2]. Figure 1.9 shows a concise example using iterated angular bisectors taken from [40, Sect. 6.2]. Since our model induces a continuous behavior, the resulting DGS cannot be determined.

To sum up, our model for Dynamic Geometry leads to Dynamic Geometry Systems with a continuous behavior, where degeneracies are treated consistently. The price for the continuity is that we cannot achieve determinism. The treatment of degeneracies prohibits a conservative behavior close to critical points. Note that point-line constructions are continuous, determined and conservative because no ambiguities occur [40, Sect. 5.2].

1.4 Scientific Contribution of this Thesis

Our results are based on the model for Dynamic Geometry that has been developed by Kortenkamp and Richter-Gebert [40, 43]. We show how the basic

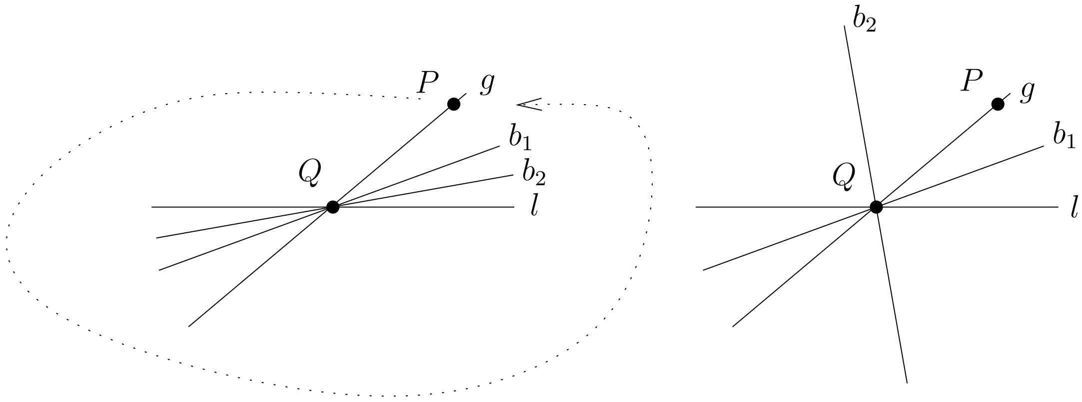


Figure 1.9: The lines l and g intersect in the point Q; the line b_1 is an angular bisector of l and g, and b_2 is an angular bisector of l and b_1. After moving P once around the point Q, all geometric objects but the line b_2 return to their initial positions; the bisector b_2 is flipped.

notions like Geometric Straight-Line Programs, continuous evaluations and critical points integrate into different fields of pure and applied mathematics. These reformulations illuminate the field of Dynamic Geometry from different points of view and lead to several instruments for solving the Tracing Problem and the Reachability Problem. We define the new notions of the derivative of a GSP with respect to time (Section 3.5) and of the interval-GSP of a given Geometric Straight-Line Program (Section 3.6). Interval-GSPs serve as a link between Geometric Straight-Line Programs and interval arithmetic.

We give an algorithm for the Tracing Problem based on the notion of interval-GSPs that uses interval arithmetic (Chapter 6). We refine this algorithm by an improved step length control that uses the derivative GSP of the underlying Geometric Straight-Line Program. Due to the application of interval arithmetic we achieve that the correctness of the computed solution is guaranteed. Since the given algorithm is reliable and robust (Section 6.5), it might be a useful tool for automated theorem proving and could be used for path-tracking in homotopy methods for solving algebraic systems of equations [18]. In Chapter 7, we extend the algorithm so that it can detect *and* treat (possible) critical points as well. For the treatment of critical points, we use Kortenkamp's approach of detouring critical points by following a detour in the complex plane [40]. Using a buffer zone might improve the practical results.

The interpretation of continuous evaluations as implicitly defined curves enables us to use numerical continuation methods; see Chapter 5. We adapt these methods to get a numerical algorithm for the Tracing Problem. Since the resulting method is efficient, it might be a good choice although the correctness of the solution is not guaranteed. This method was a significant step for the development of our algorithm for the Tracing Problem from Chapter 6.

Moreover, we show that the Tracing Problem for polynomial paths and the Reachability Problem are decidable. We obtain this result by defining suitable semi-algebraic sets. To solve the Reachability Problem and the Tracing Problem, we have to check whether the given starting and final instances lie in the same connected component of the corresponding semi-algebraic set. This result is important for determining the complexity class of the two problems since it was only known that these problems are NP-hard.

We point out the connection to algebraic functions and their Riemann surfaces as they are studied in complex analysis. We show that continuous evaluations of a Geometric Straight-Line Program Γ with one free variable correspond to liftings to the Riemann Surfaces of the algebraic functions that come from the dependent variables of Γ. We prove that a GSP Γ with one free variable z either has finitely many critical points or every position of z can be extended to a critical point.

An implementation of the algorithms from Chapters 6 and 7 is planned to verify the practical performance and to solve further problems with our methods.

1.5 Organization of the Thesis

In Chapter 2, we show how to translate the geometric situation into an algebraic setup and discuss the consequences of allowing complex coordinates. We underline the step-by-step procedure of doing geometric constructions that suggests to model geometric constructions via Geometric Straight-Line Programs. The basic notions from Dynamic Geometry that are used in our model are explained in Chapter 3. We introduce the new concepts of the derivative of a GSP and of using intervals in GSPs. In Chapter 4, we describe decision algorithms for the Tracing Problem and the Reachability Problem in the algebraic context. Here, for the Tracing Problem we only allow (piecewise) polynomial paths of the free variables. In Chapter 5, numerical solutions are applied to the Tracing Problem. The most important result are the algorithms for the Tracing Problem from Chapter 6. Since they are based on interval arithmetic, we introduce interval arithmetic in Appendix A.

The aim of Chapter 6 is to give an algorithm for the Tracing Problem from Dynamic Geometry in the algebraic context. We have to deal with two main problems occurring in this setup:

1. Critical points: Division by 0 and square roots of 0;
2. Ambiguity of the root function $\sqrt{\_}\colon \mathbb{C} \to \mathbb{C}$, $z \mapsto \pm\sqrt{z}$, e.g. $\sqrt{4} = \pm 2$).

The idea of our algorithm is to avoid these problems in advance using interval arithmetic. We achieve this by regarding the radicands of the square root operations and the divisors of the division operations. Our algorithm for the Tracing Problem proceeds stepwise. In each step, the chosen step length guarantees that the problems 1 and 2 can be handled. It turns out that our algorithm is robust. The treatment of critical points is discussed in Chapter 7. In Chapter 8, we point out the relationship between Geometric Straight-Line Programs and algebraic functions. We discuss the uniqueness and existence of continuous evaluations using covering maps.

Chapter 2

Geometry versus Algebra

In this thesis, we are dealing with Dynamic Geometry, and we need a model for geometric constructions *and* its movements. In this chapter, we focus on the geometric part. We show, how we can translate geometric constructions into an algebraic model. The dynamic aspect is treated in Chapter 3.

We compare the geometric and the algebraic situation that are explained in Sections 2.1 and 2.2. We investigate linear, quadratic and cubic constructions in Sections 2.3, 2.4 and 2.5. In Section 2.6, we discuss the change to complex coordinates and the consequences for the Tracing Problem, the Reachability Problem and automated theorem proving. The usage of complex coordinates has been motivated in the introduction, they are important for dealing with degeneracies; see Chapter 7.

2.1 The Geometric Situation

We describe different tools like ruler and compass to obtain geometric constructions. First, we give a definition for geometric constructions taken from the Encyclopaedia [34].

Definition 2.1.1. *[34, p. 588, Vol. 1]* Geometric Construction Problem
A geometric construction problem is a problem of drawing a figure satisfying given conditions using certain prescribed tools only a finite number of times.

Bieberbach [6] gives a beautiful scientific treatise on the theory of geometric constructions. We cite the first sentences of his introduction:

> `Die Theorie der geometrischen Konstruktionen lehrt,`
> `welche Konstruktionsaufgaben mit gegebenen Konstruktions-`

> `mitteln lösbar sind oder auch welche Konstruktionsmittel zur Lösung gegebener Konstruktionsaufgaben herangezogen werden muüssen. Unter einer Konstruktionsaufgabe versteht man die Aufgabe, auf dem Zeichenblatt mit gegebenen Hilfsmitteln aus gegebenen Punkten und Linien gesuchte Punkte und Linien zu finden. ...`

In other words, to make a geometric construction, we have a finite number of tools and start with a given finite configuration of geometric objects. In each step, we can make a *single* construction step using a tool from the given finite set. In the step, a new geometric object is constructed based on the objects from the initial configuration and on the objects that have already been constructed. After a finite number of these steps, the construction is finished. This stepwise procedure makes it possible to describe geometric constructions with Geometric Straight-Line Programs (GSPs) as described by Kortenkamp [40]; see Definition 3.1.1 for GSPs in an algebraic context.

We classify geometric constructions according to the used classes of geometric objects with the common operations; see Figure 2.1.

	Geometric Objects	Geometric Operations
1.	points, lines	a) line connecting two points, b) intersection point of two lines
2.	points, lines, circles	c) circle passing through three points, d) intersection point of a circle and a line, e) intersection of two circles, ...
3.	points, lines, conics	f) conic passing through five points, g) intersection of a line and a conic, h) intersection of two conics, ...

Bieberbach [6] considers a wider range of construction tools. An interesting tool is to extend the point-line constructions by an operation to compute an angular bisector of two intersecting lines. This operation isolates the problem of ambiguities and is used by Kortenkamp and Richter-Gebert [43] to show the NP-hardness of the Tracing Problem and the Reachability Problem. Note that two intersecting lines lead to two possible angular bisectors; see Figure 2.2.

In all three mentioned classes of geometric constructions, we have to deal with degenerate situations like the intersection of two identical lines or circles or the intersection of a circle and a tangent line. Point-Line constructions are *deterministic*, since for every given initial configuration, there is at most one possibility to

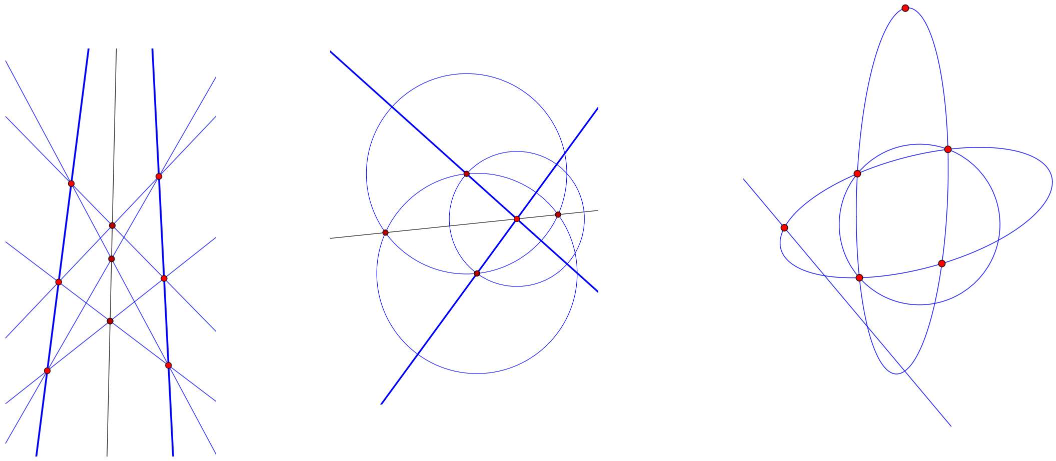


Figure 2.1: The three figures show geometric construction using points and lines; points, lines and circles; and points, lines, and conics. The left figure shows Pappus' theorem; the figure in the middle shows a construction of an angular bisector that depends on the choices for the intersection points.

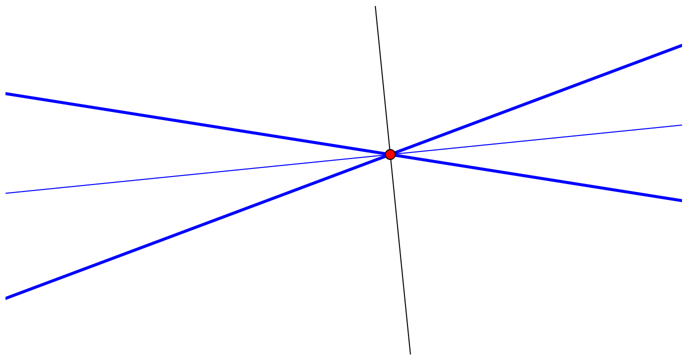


Figure 2.2: The two angular bisectors of two intersecting lines are shown.

make the wanted construction. In contrast to this, constructions involving circles or conics are *not deterministic* since here we can choose between two possible intersection points of a line and a circle, for example.

2.2 The Algebraic Situation

We translate the geometric situation to an algebraic context. Here, instead of geometric objects we consider real or complex numbers. A construction step is replaced by one of the algebraic operations addition, subtraction, multiplication, division, or square root. Instead of doing geometric constructions, we now build up algebraic expressions, and instead of given geometric objects we now have variables and constants. An *algebraic expression* in the given variables and constants is obtained by performing a finite number of the mentioned algebraic operations.

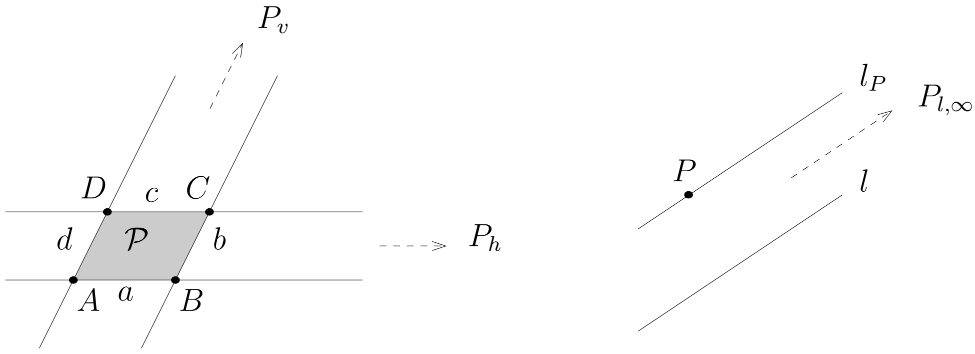
We classify algebraic expressions according to the allowed algebraic operations.

	Object Set	Algebraic Operations
1.	$\mathbb{R}$ or $\mathbb{C}$	$+, -, \cdot, /$
2.	$\mathbb{R}$ or $\mathbb{C}$	$+, -, \cdot, /, \sqrt{\_}$
3.	$\mathbb{R}$ or $\mathbb{C}$	$+, -, \cdot, /, \sqrt{\_}, \sqrt[3]{\_}$
4.	$\mathbb{R}$ or $\mathbb{C}$	$+, -, \cdot, /, \sqrt[n]{\_}$ for $n = 2, 3, 4, \ldots$

An algebraic expression without variables represents an algebraic number, an algebraic expression with variables defines an algebraic function. Note that even if we allow taking roots of arbitrary order, we cannot describe all algebraic numbers or functions as algebraic expressions. This is a consequence of the Abel-Ruffini theorem. All of the algebraic numbers or functions that cannot be written as an algebraic expression are solutions to polynomials of degree ≥ 5.

In all four mentioned classes of algebraic expressions, we have to deal with degeneracies. A division by zero is not defined and leads to a "definition gap" (German: Definitionslücke). If the object set is $\mathbb{R}$, then the square root and more generally the roots of even order are not defined for negative radicands. Although the nth root of zero is defined to be zero, taking the root of zero leads to a degeneracy since here the n distinct solutions meet in a single point.

Algebraic expressions involving the four basic arithmetic operations addition, subtraction, multiplication, and division, only, are *deterministic* since each of these operations allows at most one output on a given input. In contrast to this,



Construction of the line l_∞ Construction of parallel lines

Figure 2.3: Construction of the line at infinity l_∞ and of parallel lines in projective geometry.

algebraic expressions that contain radicals are *not deterministic*. Here we have to choose one of the n solutions for the nth root.

2.3 Linear Constructions

We can model geometric constructions of points and lines with algebraic expressions using the four arithmetic operations addition, subtraction, multiplication, and division and vice versa. For this correspondence, the vertices of a parallelogram must be constructible from the initial geometric configuration. This parallelogram $\mathcal{P}$ induces a coordinate system that builds the base of the promised analogy. Instead of Euclidean geometry, we use projective geometry and describe all points and lines by their homogeneous coordinates.

To describe a translation from the geometric setting to the algebraic one, we observe that for computing the line connecting two points and the intersection point of two lines, we have to solve linear equations. Thus, the coordinates of all constructed points and lines can be derived from the coordinates of the initially given points and lines using the four basic arithmetic operations.

For dealing with the converse direction, we need the von-Staudt constructions. These constructions allow to perform the four basic arithmetic operations geometrically. They are shown in Figures 2.4 and 2.5. The addition and subtraction operations are based on a parallel translation of two congruent triangles, and the multiplication, and division operations are based on the theorem on intersecting lines. Hence, to make these constructions we must be able to construct parallel lines that pass through a given point. In projective geometry, parallel lines can be constructed using the two pairs of parallel lines given by the parallelogram $\mathcal{P}$

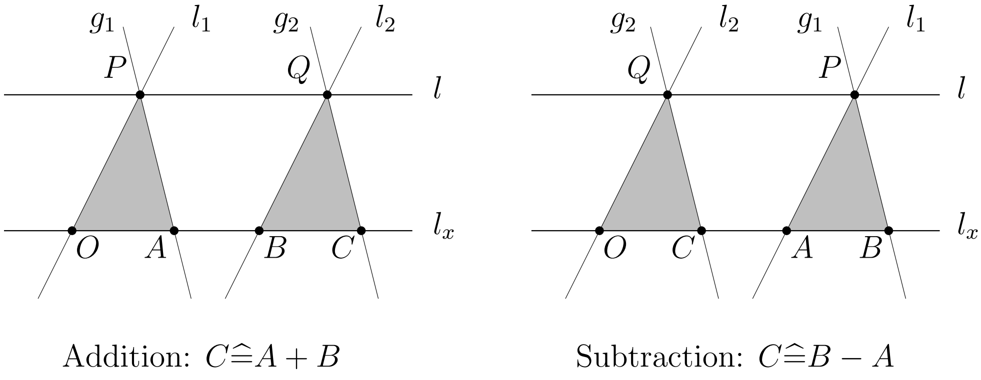


Figure 2.4: Von-Staudt constructions for geometric addition and subtraction

that defines the coordinate system. Here, two parallel lines intersect in a point on the line at infinity l_∞.

Let A, B, C and D be the four vertices of $\mathcal{P}$ in counterclockwise order. Let

a be the line connecting A and B;
b be the line connecting B and C;
c be the line connecting C and D;
d be the line connecting D and A;
P_h be the intersection point of a and c;
P_v be the intersection point of b and d; and
l_∞ be the line connecting P_h and P_v.

Since a and c are distinct parallel lines, the point P_h is a point at infinity; and since b and d are distinct parallel lines, the point P_v is a point at infinity with $P_v \neq P_h$. Thus the line l_∞ is indeed the line at infinity; see Figure 2.3. Let l be a given line and P be a point not incident to l. Let $P_{l,\infty}$ be the intersection point of l and l_∞, and let l_P be the line connecting P and $P_{l,\infty}$. Since l and l_P meet in the point $P_{l,\infty}$ at infinity, they must be parallel lines; see Figure 2.3.

We remark that if the given parallelogram $\mathcal{P}$ is a square, then it is even possible to drop a perpendicular to a given line l through a given point P, to bisect a right angle, and to rotate a line segment by 90°; see [6, §4].

Addition and Subtraction (von-Staudt Construction). We focus on the geometrical addition of two real numbers a and b. The difference $b - a$ of a and b can be computed similarly as shown in Figure 2.4. Let $O = (0, 0, 1)$ be the origin of the given coordinate system given in homogeneous coordinates. Let $A = (a, 0, 1)$ and $B = (b, 0, 1)$. The aim is to construct the point $C = (a + b, 0, 1)$.

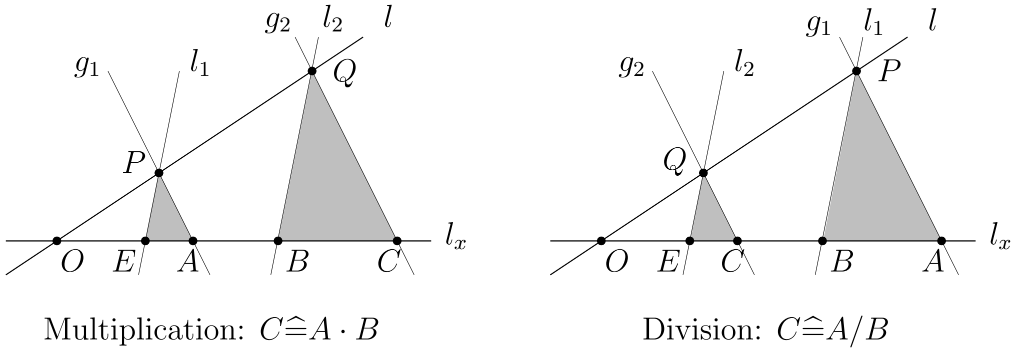


Figure 2.5: Von-Staudt constructions for geometric multiplication and division

Let P be a point that does not lie on the x-axis l_x, and let l be the parallel line to the x-axis that passes through the point P. Let

l_1 be the line connecting O and P;
g_1 be the line connecting P and $A = (a, 0, 1)$;
l_2 be the parallel line to l_1 through the point $B = (b, 0, 1)$;
Q be the intersection point of l_2 and l;
g_2 be the parallel line to g_1 through the point Q;
C be the intersection point of g_2 and the x-axis l_x.

Hence, the two triangles OAP and BCQ are congruent, and the line segments $\overline{OA}$ and $\overline{BC}$ have the same (directed) length, which equals a. Thus we have $C = (a + b, 0, 1)$. The construction even works for $a = 0$ or $b = 0$.

Multiplication and Division (von-Staudt Construction). We explain the geometrical multiplication of two real numbers a and b; the division a/b can be computed by reversing the construction for the multiplication; see Figure 2.5. As for the addition, let $O = (0, 0, 1)$ be the origin of the given coordinate system in homogeneous coordinates, and let $A = (a, 0, 1)$ and $B = (b, 0, 1)$. Furthermore, let $E = (1, 0, 1)$ be the point that represents the neutral element 1 of the multiplication. The aim is to construct the point $C = (a \cdot b, 0, 1)$. Let P be a point that does not lie on the x-axis l_x, and let l be the line connecting P and O. Let

l_1 be the line connecting E and P;
g_1 be the line connecting P and $A = (a, 0, 1)$;
l_2 be the parallel line to l_1 through the point $B = (b, 0, 1)$;
Q be the intersection point of l_2 and l;
g_2 be the parallel line to g_1 through the point Q;
C be the intersection point of g_2 and the x-axis l_x.

First, we assume $a \neq 0$ and $b \neq 0$. By construction, the triangles EAP and BCQ are similar. The theorem of intersecting lines implies

$$\frac{|\overline{OC}|}{|\overline{OA}|} = \frac{|\overline{OQ}|}{|\overline{OP}|} = \frac{|\overline{OB}|}{|\overline{OE}|}.$$

Hence we have

$$\frac{|c|}{|b|} = \frac{|\overline{OC}|}{|\overline{OB}|} = \frac{|\overline{OA}|}{|\overline{OE}|} = \frac{|a|}{1},$$

and $|c| = |a| \cdot |b|$. Let l_+ be the half-line of the x-axis l_x containing E and l_- the other one. We observe that the point C lies on l_+ if and only if both points A and B either lie on l_+ or on l_-, and C lies on l_- if and only if one of the points A and B lies on l_+ and the other one on l_-. Thus the sign of the product $a \cdot b$ is determined by the half-line that contains the point C. Note that the construction even works if $a = 0$ or $b = 0$. In this case we get $C = O$. If the divisor b in the geometric division is zero, then we have $l_1 = l$, and the lines l and l_2 are parallel. Thus the intersection point Q of l and l_2 is a point at infinity, g_2 is the line at infinity and the resulting point C is a point at infinity as expected.

2.4 Quadratic Constructions

To deal with circles as well, we have to allow square root operations in the algebraic model. We show that ruler-and-compass constructions can be transformed to algebraic expressions using addition, subtraction, multiplication, division, and square roots and vice versa.

To compute the coordinates of all points, lines and circles in a geometric construction, we have to solve quadratic equations. Thus, the coordinates of the constructed elements can be computed from the coordinates of the initially given points, lines and circles using the five mentioned algebraic operations.

To investigate the converse direction, we have to compute square roots of non-negative real numbers geometrically; see Figure 2.6. This can be done using the altitude theorem in right triangles. Let $a \geq 0$ be a given real number. As before, let $O = (0, 0, 1)$ be the origin given in homogeneous coordinates, let $E = (1, 0, 1)$ be the point representing 1 and $A = (a, 0, 1)$. The aim is construct the two points $C_+ = (+\sqrt{a}, 0, 1)$ and $C_- = (-\sqrt{a}, 0, 1)$. To simplify the notation, we assume that we are given a Cartesian coordinate system. Let l_y be the y-axis and $F = (0, 1, 1)$.

1. Construct the point $M = (\frac{a-1}{2}, 0, 1)$ with the methods from Section 2.3.

2. Let γ be the circle having M as midpoint that passes through the point A.

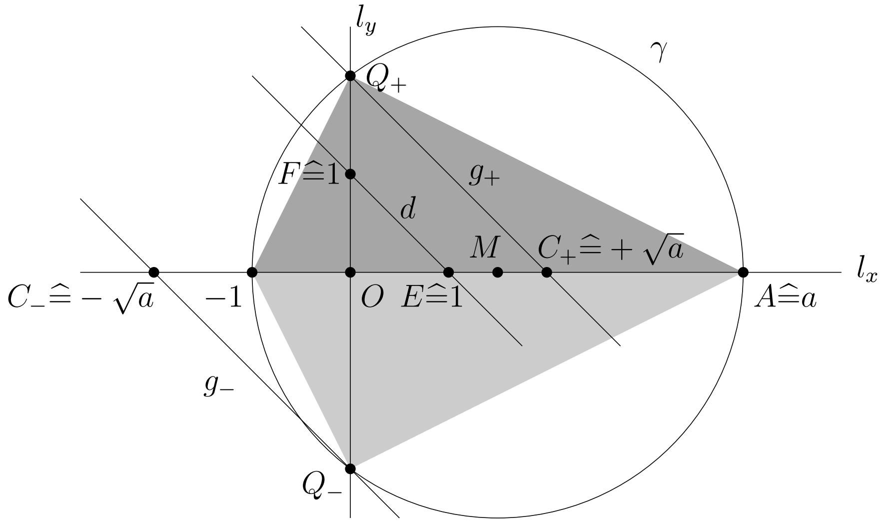


Figure 2.6: Geometric construction of square roots using the altitude theorem.

3. Let Q_+ and Q_- be the intersection points of the circle γ and the line l_y; we choose Q_+ to be the intersection point that lies on the same half-line of l_y as the point F. Note that if and only if $a = 0$, we have $A = O$ and the line l_y is tangent to the circle γ. In this situation, we have $Q_+ = Q_- = O$.

4. Let d be the line connecting E and F.

5. Let g_+ be the parallel line to d that passes through the point Q_+, and let g_- be the parallel line to d that passes through the point Q_-.

6. Let C_+ be the intersection point of g_+ and the x-axis l_x, and let C_- be the intersection point of g_- and l_x.

If $a = 0$, then we have $C_+ = C_- = O = (0,0,1) = (\sqrt{0},0,1)$. Now, we look at the case $a > 0$. Let $B = (-1,0,1)$ be the second intersection point of the circle γ and the x-axis l_x. The triangles BAQ_+ and BAQ_- are right triangles with height $h = |\overline{OQ_+}| = |\overline{OQ_-}|$. The altitude theorem implies $h^2 = |\overline{BO}| \cdot |\overline{OA}| = 1 \cdot a = a$. Thus we have $h = \pm\sqrt{a}$. The theorem of intersecting lines implies $|h| = |\overline{OC_+}| = |\overline{OC_-}|$.

If we interpret the Euclidean plane as complex plane $\mathbb{C}$, we can also compute square roots of complex numbers geometrically. Let $z = re^{i\phi}$ be a complex number given by its polar coordinates. To determine the two square roots of z, we have to compute $\sqrt{r}$ and to bisect the angle ϕ. We have just seen, how to determine $\sqrt{r}$ using the circle γ. We can bisect an angle using ruler and compass as shown in Figure 2.1.

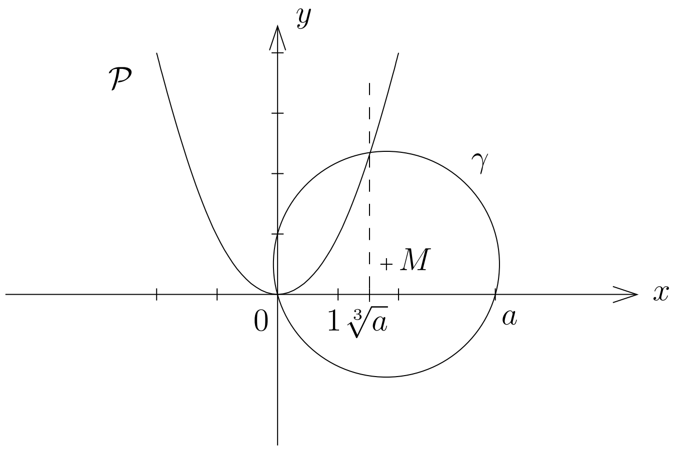


Figure 2.7: The geometric construction of cubic roots is shown. The cubic root $\sqrt[3]{a}$ is the x-coordinate of the intersection point of the unit parabola $\mathcal{P}$ and the circle γ with midpoint $M = (a/2, 1/2, 1)$.

Bieberbach [6, §5] shows that ruler-and-compass constructions are as powerful as constructions with a ruler if only one circle and its midpoint are given. These constructions are called Poncelet-Steiner constructions. In fact, only an arc of the given circle is needed.

2.5 Cubic Constructions

Now we add conics to the set of geometric objects and consider geometric constructions, where points, lines, and conics are involved. Note, that circles are a special case of conics. We show that geometric constructions with points, lines, and conics can be described by algebraic expressions using the operations addition, subtraction, multiplication, division, square root, and cubic root and vice versa.

The coordinates of the constructed points, lines, and conics can be obtained by solving polynomial equations of degree at most 3. Kortenkamp [40] gives a beautiful and practice-related transformation from geometric constructions involving points, lines, and conics to algebraic expressions using the mentioned six algebraic operations.

For the transformation from algebra to geometry, we have to compute cubic roots of real numbers geometrically; see Figure 2.7. To simplify the notation, we assume that we are given a Cartesian coordinate system. Let $a \in \mathbb{R}$.

1. Let $\mathcal{P}$ be the parabola defined by the equation $y = x^2$. Thus, $\mathcal{P}$ is the unique conic that passes through the five points $(-2, 4, 1)$, $(-1, 1, 1)$, $(0, 0, 1)$, $(1, 1, 1)$ and $(2, 4, 1)$ given in homogeneous coordinates. These five points can be constructed using ruler and compass, only.

2. Let $M = (a/2, 1/2, 1)$, and let γ be the circle that has M as midpoint and that passes through the origin $O = (0, 0, 1)$. The circle γ is described by the equation

$$0 = \left(x - \frac{a}{2}\right)^2 + \left(y - \frac{1}{2}\right)^2 - \left(\left(\frac{a}{2}\right)^2 + \left(\frac{1}{2}\right)^2\right) = x^2 - ax + y^2 - y.$$

3. Let $P_i = (x_i, y_i, z_i)$, $i = 1, \ldots, 4$, be the (at most) four intersection points of the conic $\mathcal{P}$ and the circle γ. The values for the x-coordinates x_1, x_2, x_3 and x_4 are obtained by plugging in the formula $y = x^2$ of the parabola $\mathcal{P}$ into the formula for the circle γ:

$$0 = x^2 - ax + y^2 - y = x^2 - ax + x^4 - x^2 = x^4 - ax = x(x^3 - a)$$

   Solving this equation leads to the wanted values for x_i, $i = 1, \ldots, 4$. Note that two solutions are complex.

If we interpret the Euclidean plane as complex plane $\mathbb{C}$, we can also compute cubic roots of complex numbers geometrically. Let $z = re^{i\phi}$ be a complex number given by its polar coordinates. To determine the three cubic roots of z, we have to compute $\sqrt[3]{r}$ and to trisect the angle ϕ. We have just seen, how to determine $\sqrt[3]{r}$ using a parabola. In [26, Chap. XLI (author: Otto Böklen), p. 436] is shown, how to trisect a given angle using an ellipse. Bieberbach [6] discusses how to solve arbitrary equations of degree 3 and 4 geometrically using related arguments. Solving equations of degree 4 can be reduced to solving equations of degree at most 3 using the cubic resolvent.

2.6 Using Complex Coordinates

We have seen in the previous sections that we can translate the geometric situation to an algebraic model, where the geometric operations are described by a finite number of additions, subtractions, multiplications, divisions, square and cubic roots. Thus, using complex coordinates for the geometric objects implies that the algebraic operations are performed in the field of complex numbers $\mathbb{C}$ instead in $\mathbb{R}$.

First, we observe that solutions of the used algebraic operations over $\mathbb{R}$ are solutions over $\mathbb{C}$ as well. At first view, we are interested in geometric objects

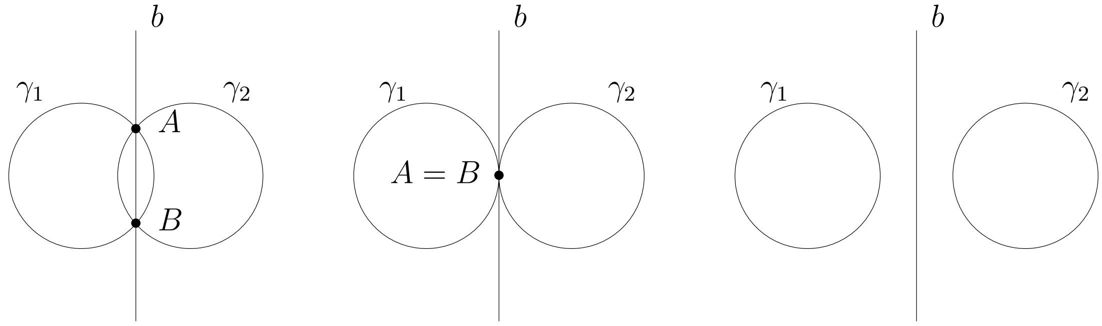


Figure 2.8: The line b connects the two intersection points A and B of the circles γ_1 and γ_2 with identical radii. In the degenerate situation, we have $A = B$ and the line b is a tangent. In the right figure, the circles γ_1 and γ_2 do not intersect. In fact, they have two intersection points, which have complex coordinates. The line connecting these two points has real coordinates again.

that correspond to solutions over $\mathbb{R}$ since in a Dynamic Geometry System, we only display these solutions. However, switching to complex coordinates has several advantages. Since we do not need to distinguish between positive and negative radicands in roots over $\mathbb{C}$, several special cases are omitted, which simplifies the computations. Even if intermediate results are complex numbers with a non-vanishing imaginary part, later results could be real numbers again. Figure 2.8 shows an example for this phenomenon taken from [40]. Kortenkamp [40, Chap. 6] uses the field of complex numbers $\mathbb{C}$ for the algebraic modeling of circles and conics. In projective geometry, circles are conics that pass through the points $I = (i, 1, 0)$ and $J = (i, -1, 0)$ given in homogeneous coordinates. Moreover, using complex coordinates enables us to use the theory of complex analysis, which turns out to be quite helpful.

Allowing complex coordinates influences the properties and the behavior of the underlying Dynamic Geometry system. The consequences of the change to complex coordinates depend on the investigated problem.

Consequences for the Tracing Problem. As long as no degeneracies are hit in the tracing process, the change to complex coordinates has no consequences. In this case, the coordinates of the geometric objects remain real numbers at each time t. If the given paths in the Tracing Problem are paths in $\mathbb{R}$, then we only might switch to complex coordinates at degenerate positions. This holds since we consider *continuous paths*. The only operations that might lead to complex results for a real input are the root operations. Due to the intermediate value theorem, before the radicand of a square root operation becomes negative, it must have been zero, and we must have passed through a degeneracy. The same holds for cubic roots: Before we switch from a real cubic root to one of the two conjugate complex roots, we must have passed through zero where we have hit a degeneracy.

By contrast, we exploit the change to complex coordinates to deal with degeneracies. Instead of running into a degenerate situation, we follow a detour around the critical point in the complex plane $\mathbb{C}$. This concept was developed by Kortenkamp [40] and is implemented in the Dynamic Geometry Software *Cinderella* [42, 41]. We gave a motivation in Chapter 1, and we can integrate this beautiful concept into our algorithms as shown in Chapter 7.

Consequences for the Reachability Problem. Allowing complex coordinates extensively extends the configuration space of a given geometric construction or algebraic expression. In some situations, a given configuration of a construction can be transformed into another one by following a continuous path in the complex plane $\mathbb{C}$ without hitting degeneracies whereas there are no continuous paths in $\mathbb{R}$ that could be used for the transformation. This fact is explained on page 6 of Chapter 1 and illustrated by Figure 1.4.

Consequences for Automated Theorem Proving. Most of the geometric theorems considered in automated theorem proving so far are closure theorems [16, p. 363]. For a certain configuration of points, lines or circles is claimed that some of these points lie on a line or a circle or that some the lines intersect in a common point. Since $\mathbb{R} \subset \mathbb{C}$, all of these theorems that hold over $\mathbb{C}$ also hold over $\mathbb{R}$. The converse direction of this claim is the difficult part as stated by Dolzmann, Sturm, and Weispfenning [16, p. 358]:

> `It is an amazing fact, which does not appear to have a sufficient theoretical explanation up to now, that for the overwhelming majority of theorems in the plane geometry of points, lines, circles, and cones the algebraic translation` ϕ `- if done ‘‘properly’’ - does hold in the field of complex numbers. Trivial exceptions may occur if the theorem asserts properties of points that do not exist in the real plane but exist in the complex plane; ...`

In [16], several examples of geometric theorems are considered including a theorem that holds over $\mathbb{R}$ but not over $\mathbb{C}$:

Theorem 2.6.1. *[16, Ex. 7, p. 372] Consider eight points A, B, C, D, E, F, G and H such that the eight triples ABD, BCE, CDF, DEG, EFH, FGA, GHB and HAC are collinear. Then all eight points lie on a line.*

	g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8
P_1			×	×	×			
P_2						×	×	×
P_3	×		×			×		
P_4	×			×			×	
P_5	×				×			×
P_6		×	×			no	⊗	×
P_7		×		×		×	no	⊗
P_8		×			×	⊗	×	no

		g_1	g_2	g_3	g_4	g_5	g_6	g_7	g_8
→	P_1	×	×		×				
	P_2		×	×		×			
→	P_3			×	×		×		
	P_4				×	×		×	
	P_5					×	×		×
	P_6	×					×	×	
→	P_7		×					×	×
→	P_8	×		×					×

Table 2.3: Incidence tables for point-line configurations with eight points and lines such that every points lies on three lines and every line is incident to three points.

Tracing back the references given in [16] shows that this theorem goes back to Levi [49, Chap. III]. In his book [49], Levi investigates point-line configurations. In Theorem 2.6.1, we are given eight points and eight lines and require that every line is incident to exactly three of the given eight points and that every point lies on exactly three of the given eight lines.

Proof. We follow Levi's approach [49]. For each configuration point P_1 there is exactly one configuration point P_2 which is not connected with P_1 since the three lines that are incident to P_1 contain exactly 6 configuration points in addition to P_1. Let $P_3, \dots, P_8$ be these six points. Every point $P_i \in \{P_3, \dots, P_8\}$ is connected with P_1 and with P_2. Additionally, P_i lies on a line that neither contains P_1 nor P_2. There are exactly two lines g_1 and g_2 that neither contain P_1 nor P_2. The left of Table 2.3 summarizes the previous observations. Levi [49, p. 3] calls this table "Inzidenztafel" (incidence table). Note that the position filled by "no" cannot be used for incidences. If for example the point P_6 lies on the line g_6, then the lines g_3 and g_6 would both contain the points P_3 and P_6 and either the two lines or the two points would be identical. If the points P_3 and P_6 are identical, then the line g_1 would contain the four points P_3, P_4, P_5 and P_6, which is not allowed. If the lines g_3 and g_6 are identical, then the line g_6 would contain the four points P_1, P_2, P_3 and P_6, which is not allowed. The two remaining possibilities for the lower right submatrix lead to equivalent tables since they can be transformed into each other by renaming points and lines.

Thus we have shown that there is just one incidence table up to renaming points and lines. For simplicity of notation, we rename the eight points and lines such that the incidence table has the form shown in the right of Table 2.3. This can be done since every point lies on exactly three lines and every line is incident to exactly three points, again.

To compute a realization of this particular point-line configuration, we observe that the four points P_1, P_3, P_7 and P_8 must be the vertices of a non-degenerate quadrilateral since each line contains at most two of these four points. We choose a projective coordinate system such that

$$P_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}; \quad P_3 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}; \quad P_7 = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}; \quad P_8 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

Using the incidences given by the eight lines, we can compute the coordinates of the remaining points and get

$$P_5 = \begin{pmatrix} 1 \\ 1 \\ -\varepsilon \end{pmatrix}; \quad P_2 = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}; \quad P_4 = \begin{pmatrix} 1 \\ 1+\varepsilon \\ 0 \end{pmatrix}; \quad P_6 = \begin{pmatrix} 1 \\ 0 \\ -\varepsilon \end{pmatrix}.$$

Finally, we have to choose ε such that the points P_4, P_6 and P_7 are colinear as required by column 7. Hence, we have to solve the equation

$$0 = \det \begin{pmatrix} 1 & 1+\varepsilon & 0 \\ 1 & 0 & -\varepsilon \\ 1 & 1 & 1 \end{pmatrix} = -\varepsilon^2 - \varepsilon - 1,$$

which has the two complex solutions $\varepsilon_{1/2} = -1/2 \pm i\sqrt{3}/2$. Plugging in the solutions for ε into the coordinates of the points $P_1, \ldots, P_8$ gives a realization of this point-line configuration over $\mathbb{C}$. Simultaneously, we have shown that there is no realization over $\mathbb{R}$.

The previous observation shows that Theorem 2.6.1 does not hold over $\mathbb{C}$: The computed realizations of the configuration are counter-examples. Now, we consider the field of real numbers $\mathbb{R}$. At the beginning of the proof, we considered the quadrilateral defined by the points P_1, P_3, P_7 and P_8. Since there is no realization of the configuration over $\mathbb{R}$, this quadrilateral must degenerate. Using the given incidences, we can show that all points $P_1, \ldots, P_8$ must be collinear. If for example the points P_1, P_3 and P_7 are collinear, then they lie on the line g_4 like the point P_4; thus these four points lie on the line g_7 like the point P_6; and so on. □

To sum up, the consequences of the change to complex coordinates are not known, yet. It seems that we do not need to count many problems although we have explained a theorem from projective geometry that holds over $\mathbb{R}$ but not over $\mathbb{C}$. The Reachability Problem over $\mathbb{C}$ could be a useful tool for automated theorem proving. We have seen above that the Reachability Problem depends on the underlying field $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$. We do not focus on automated theorem proving in this thesis.

Chapter 3

Some Basic Concepts from Dynamic Geometry

We explain basic notions from Dynamic Geometry introduced by Kortenkamp and Richter-Gebert [40, 43] like Geometric Straight-Line Programs (Section 3.1), critical points (Section 3.2), and continuous evaluations (Section 3.3). In Section 3.4, we define the Tracing Problem and the Reachability Problem which have been introduced by Kortenkamp and Richter-Gebert in [43], the main ideas can already be found in [40]. In Section 3.5, we define the derivative GSP $\dot{\Gamma}$ of a GSP Γ with respect to time [14]. This new concept is crucial for the Cone Algorithm given in Section 6.3. In Section 3.6, we show how we could use intervals in GSPs. We define the interval-GSP Γ_{int} that results from a GSP Γ by replacing all operations of Γ by the corresponding interval operations. The variables of Γ_{int} take intervals as values. This new notion is important for our algorithms for the Tracing Problem in Chapter 6.

3.1 Geometric Straight-Line Programs

As mentioned in the introduction, geometric constructions can be described by Geometric Straight-Line Programs (GSP); see [43, 44]. The objects are points, lines and circles (or more generally conics) with the standard geometric operations like computing the line connecting two points, the intersection point of two lines, or one of the at most two intersection points of a circle and a line. Here, we consider an algebraic situation: Our objects are real or complex numbers with the operations addition, subtraction, multiplication, division, and square root. In this context, Geometric Straight-Line Programs describe algebraic expressions like $\sqrt{z^2 - 1}$ instead of geometric constructions.

Definition 3.1.1. Geometric Straight-Line Program Γ over $\mathbb{R}$ or $\mathbb{C}$.
Let $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$ be either the field of real numbers or the field of complex numbers. A GSP Γ over $\mathbb{K}$ is a finite sequence of instructions γ_j of the form

$$\begin{array}{llll} \gamma_j: & v_j & \leftarrow & \text{FREE} & \\ \gamma_j: & v_j & \leftarrow & v_a + v_b & (+) \\ \gamma_j: & v_j & \leftarrow & v_a - v_b & (-) \\ \gamma_j: & v_j & \leftarrow & v_a \cdot v_b & (\cdot) \\ \gamma_j: & v_j & \leftarrow & v_a / v_b & (/) \\ \gamma_j: & v_j & \leftarrow & \sqrt{v_a}. & (\sqrt{\_}) \end{array}$$

We say that the instruction γ_j defines the variable v_j and require that the variables v_a and v_b are defined before the variable v_j, i.e., the variables v_a and v_b in the definition of γ_j are defined by instructions γ_a and γ_b with $a < j$ and $b < j$.

The instruction $v_j \leftarrow$ FREE is used to generate free variables*; it has no input and allows the output to be any element of $\mathbb{K}$. Variables that are created by one of the instructions $+, -, \cdot, /$, or $\sqrt{\_}$ are called* dependent variables. *For given values of the variables v_a and v_b, the instructions $+, -, \cdot$, or $/$ compute the corresponding sum, difference, product, or quotient, respectively. For a given value of v_a, the instruction $\sqrt{\_}$ describes one of the at most two solutions of the equation $v_j^2 = v_a$, the "sign" of v_a is not fixed. Zero is defined not to be a valid input for the $\sqrt{\_}$-operation. For $\mathbb{K} = \mathbb{R}$, only positive numbers are valid inputs for the $\sqrt{\_}$-operation. The divisor of a division operation must not be zero.*

Definition 3.1.1 is a practical description of the notion of Geometric Straight-Line Programs, a formal and detailed definition is given by Kortenkamp and Richter-Gebert and can be found in [40, 43]. Their definition allows the treatment of different object types. For simplicity of notation, we assume that the first k variables $v_{-k+1} = z_{-k+1}, \dots, v_0 = z_0$ of a GSP Γ are free variables and the following n variables $v_1, \dots, v_n$ are dependent variables. Example 3.1.3 below is a running example of this thesis.

The operations $+$, $-$, $\cdot$, $/$ and $\sqrt{\_}$ of a GSP Γ can be formulated by relations using addition and multiplication, only.

Operation of Γ	Corr. Relation
$v_c \leftarrow v_a + v_b$	$v_c = v_a + v_b$
$v_c \leftarrow v_a - v_b$	$v_a = v_c + v_b$
$v_c \leftarrow v_a \cdot v_b$	$v_c = v_a \cdot v_b$
$v_c \leftarrow v_a / v_b$	$v_a = v_c \cdot v_b$
$v_c \leftarrow \sqrt{v_a}$	$v_a = v_c \cdot v_c$

This formulation by relations is important for the $\sqrt{\_}$-operation. For a complex number $z \neq 0$ or a real number $z > 0$ the operation $w \leftarrow \sqrt{z}$ is defined by the relation $w^2 = z$, and the two values "$+\sqrt{z}$" and "$-\sqrt{z}$" are candidates for the output. The sign of the output is not specified by the GSP Γ. Thus, the $\sqrt{\_}$-operation is non-deterministic in contrast to the operations $+$, $-$, $\cdot$, and $/$.

Definition 3.1.2. *[40, 43]* Instance of a GSP Γ
An instance *of a GSP Γ is an assignment of valid values to the variables of Γ that fulfill all relations of the GSP Γ. The* configuration space *of a GSP Γ is the set of all instances of Γ. If Γ is a GSP with k free variables and n dependent ones, then the set of all instances $A = (a_{-k+1}, \ldots, a_0, a_1, \ldots, a_n)$ with $a_{-k+1} = \tilde{a}_{-k+1}, \ldots, a_0 = \tilde{a}_0$ is called the* fiber *of the point $(\tilde{a}_{-k+1}, \ldots, \tilde{a}_0)$. The instance A* lies above *the position $(a_{-k+1}, \ldots, a_0) = (\tilde{a}_{-k+1}, \ldots, \tilde{a}_0)$ of the free variables.*

In an instance, the "signs" for the square root operations are fixed in contrast to the underlying GSP Γ. Definitions 3.1.1 and 3.1.2 imply that, in an instance, the values of the divisor variables of the division operations and the values of the radicand variables of the root operations are non-zero, and for $\mathbb{K} = \mathbb{R}$, the values of the radicands are positive. Due to square root instructions, the fiber of a point might contain more than one instance. Thus, we might have more than one instance that lies above a fixed position of the free variables.

Example 3.1.3. The expression $\sqrt{z^2 - 1}$ can be described by the GSP Γ:

$$\begin{array}{rrcll} \Gamma: & z & \leftarrow & \text{FREE} & \\ & v_1 & \leftarrow & z \cdot z & \\ & v_2 & \leftarrow & v_1 - 1 & \\ & v_3 & \leftarrow & \sqrt{v_2}\,; & \qquad \text{described by } v_3^2 = v_2 \text{ and } v_2 \underset{\neq}{>} 0. \end{array}$$

If $\mathbb{K} = \mathbb{C}$ then $(0, 0, -1, i)$ and $(0, 0, -1, -i)$ are instances of this GSP, whereas $(1, 1, 0, 0)$ is not an instance since 0 is not a valid input for the square root operation. The two instances $(0, 0, -1, i)$ and $(0, 0, -1, -i)$ lie above the point $z = 0$. For $\mathbb{K} = \mathbb{R}$, none of the three tuples is an instance.

The notion of Geometric Straight-Line Programs is derived from the notion of Straight-Line Programs described by Bürgisser et al. in [9, Chapter 4]. The main difference is that the concept of Geometric Straight-Line Programs allows non-deterministic operations like the square root operation. Straight-Line Programs are not designed for dealing with non-determinism.

Every dependent variable of a GSP Γ describes an algebraic function in the free variables $z_{-k+1}, \ldots, z_0$ of Γ. Formally, this fact can be shown by induction using Theorem 8.2.6 from page 127; see Section 8.4.

3.2 Critical Points

We discuss the notion of critical points. In the geometric setting, a critical point occurs if the intersection of a circle and a tangent line is considered, for example. Here, the two intersection points of a line and a circle in the non-degenerate situation merge to a single point, and we cannot distinguish these two points anymore. This causes problems since we consider dynamic constructions, where we keep track of the motions of *all* geometric objects. At a critical point, one of the two intersection points of a line and a circle "disappears". In the algebraic model, a critical point occurs if we take the square root of zero or if we divide by zero in the computation of an instance of a GSP. For the square root operation, we have the same situation as with the intersection of a line and a circle in the geometric situation. If the radicand is not zero, we always have two possible values for the output. Only if the radicand is zero, the two solutions merge. A division by zero cannot be executed. It corresponds to the intersection of two parallel lines, for example. If we use homogeneous coordinates as in the Dynamic Geometry Software *Cinderella* [42, 41], divisions are omitted.

We use a less restrictive definition of a critical point than in [13]. Before we give the definition, we introduce the notion of an m-head of a GSP Γ. As before, let $\mathbb{K}$ be one of the fields $\mathbb{R}$ or $\mathbb{C}$.

Definition 3.2.1. m-Head $\Gamma^{(m)}$ of a GSP Γ
Let Γ be a GSP with k free variables $z_{-k+1}, \ldots, z_0$ and n dependent ones $v_1, \ldots, v_n$, let $m \in \{1, 2, \ldots, n\}$. We call the GSP that arises from Γ by cutting off the variables $v_{m+1}, v_{m+2}, \ldots, v_n$ the m-head $\Gamma^{(m)}$ of Γ.

If $A = (a_{-k+1}, \ldots, a_0, a_1, \ldots, a_m, \ldots, a_n)$ is an instance of Γ, then $A^{(m)} := (a_{-k+1}, \ldots, a_0, a_1, \ldots, a_m)$ is an instance of $\Gamma^{(m)}$, and we call $A^{(m)}$ the m-head of the instance A.

Definition 3.2.1 implies that $\Gamma^{(m)}$ has k free variables $z_{+k-1}, \ldots, z_0$ like Γ, and m dependent variables $v_1, v_2, \ldots, v_m$. Each dependent variable of $\Gamma^{(m)}$ is defined by the same operation as the corresponding dependent variable of Γ.

Definition 3.2.2. m-Critical Point of a GSP Γ
Let Γ be a GSP over $\{\mathbb{K}, +, -, \cdot, /, \sqrt{\_}\}$ with k free variables $z_{-k+1}, \ldots, z_0$, and n dependent ones $v_1, \ldots, v_n$. Let $m \in \{1, 2, \ldots, n\}$, and let $\tilde{C} = (c_{-k+1}, \ldots, c_0, c_1, c_2, \ldots, c_{m-1}) \in \mathbb{K}^{k+m-1}$ be an instance of $\Gamma^{(m-1)}$ that cannot be extended to an instance of $\Gamma^{(m)}$. That is, there is no $c_m \in \mathbb{K}$ such that $(c_{-k+1}, \ldots, c_{m-1}, c_m)$ is an instance of $\Gamma^{(m)}$. Then, $\tilde{C}$ is called an m-critical point of Γ, and the variable v_m causes the m-critical point $\tilde{C}$.

Definition 3.2.3. Critical Point of Γ
A point $C = (c_{-k+1}, \ldots, c_0, c_1, c_2, \ldots, c_n)$ is a critical point of a GSP Γ with k

free variables and n dependent ones if there is an $m \in \{1, 2, \dots, n\}$ such that the $m-1$-head $C^{(m-1)}$ of C is an m-critical point of Γ. All critical points with the same $(m-1)$-head are identified, because the values $c_m, \dots, c_n$ are arbitrary.

Definition 3.2.4. Critical Value of Γ
Let Γ be a GSP with k free variables $z_{-k+1}, \dots, z_0$ and n dependent ones. Let $c_{-k+1}, \dots, c_0 \in \mathbb{K}$ be values of the free variables $z_{-k+1}, \dots, z_0$. If $(c_{-k+1}, \dots, c_0) \in \mathbb{K}^k$ can be extended to a critical point of Γ, then $(c_{-k+1}, \dots, c_0)$ is called a critical value *of Γ. Otherwise, $(c_{-k+1}, \dots, c_0)$ is called a* regular value *of Γ.*

We have chosen the term critical value of a GSP according to the notion of critical values of algebraic functions. In the theory of algebraic functions, the critical values are called exception points, as well; see page 131 in Section 8.3. The critical values of a GSP Γ are related to the critical values of the algebraic functions that correspond to the dependent variables of Γ; see Section 8.4.

Lemma 3.2.5. *The set of critical points of a GSP Γ is finite if and only if the set of critical values of Γ is finite.*

Proof. Let Γ be a GSP with k free variables and n dependent ones. The set of critical values of Γ is obtained by projecting the set of critical points to the k coordinates of the free variables. Thus, if the set of critical points is finite, then the set of critical values is finite as well. To prove the converse direction, we observe that for every square root operation, the number of instances lying above a fixed position of the free variables is at most doubled. A square root operation doubles the number of instances if and only if its radicand variable is not zero. For the four elementary arithmetic operations, the number of instances remains unchanged. Hence, the number of instances lying above a regular value of Γ is 2^s where s is the number of square root operations of Γ. Furthermore, the number of instances lying above a critical value of Γ is smaller or equal to 2^s. Consequently, if the set of critical values is finite, then the set of critical points is finite as well. □

Example 3.2.6. We consider the GSP Γ that describes the algebraic expression $f(z) = \frac{z}{\sqrt{z}-2}$:

$$\begin{array}{rcll} \Gamma: \quad z & \leftarrow & \text{FREE} & \\ v_1 & \leftarrow & \sqrt{z} & // \; v_1 = \sqrt{z} \\ v_2 & \leftarrow & v_1 - 2 & // \; v_2 = \sqrt{z} - 2 \\ v_3 & \leftarrow & \frac{z}{v_2} & // \; v_3 = \frac{z}{\sqrt{z}-2} \end{array}$$

The value $z = 0$ is a 1-critical point. Since 0 is a forbidden input for the $\sqrt{\_}$-operation, it cannot be extended to an instance of $\Gamma^{(1)}$. The point $(4, 2, 0)$ is a 3-critical point, since it causes a division by 0 for the variable v_3. Hence, for $\mathbb{K} = \mathbb{C}$, the critical points of Γ are $(0, 0, 0, 0)$ and $(4, 2, 0, 0)$, and the critical values are 0 and 4. For $\mathbb{K} = \mathbb{R}$, all points $(r, 0, 0, 0)$ with $r < 0$ are 1-critical points, additionally. Hence, the notion of critical points depends on the underlying field.

Example 3.2.7. We describe the expression $\sqrt{z - \sqrt{z^2}}$ by the GSP Γ_1.

$$\begin{array}{rrcll} \Gamma_1: & z & \leftarrow & \text{FREE} & \\ & v_1 & \leftarrow & z \cdot z & // \; v_1 = z^2 \\ & v_2 & \leftarrow & \sqrt{v_1} & // \; v_2 = \sqrt{z^2} = \pm z \\ & v_3 & \leftarrow & z - v_2 & // \; v_3 = z - \sqrt{z^2} \\ & v_4 & \leftarrow & \sqrt{v_3} & // \; v_4 = \sqrt{z - \sqrt{z^2}} \end{array}$$

Here, every value for z can be extended to a critical point: $(z, z^2, z, 0) \in \mathbb{C}^4$ is an instance of the 3-head $\Gamma_1^{(3)}$ of Γ_1 for every $z \in \mathbb{C}$. It cannot be extended to an instance of $\Gamma_1^{(4)} = \Gamma_1$. Thus, the set of critical values is the complex plane $\mathbb{C}$. However, if we choose $v_2 = -z$, then $(z, z^2, z, \pm\sqrt{2z})$ are instances of Γ_1 for all $z \in \mathbb{C}$ with $z \neq 0$.

Example 3.2.8. We consider $\sqrt{z_{-1} + z_0}$ and the GSP Γ_2.

$$\begin{array}{rrcll} \Gamma_2: & z_{-1} & \leftarrow & \text{FREE} & \\ & z_0 & \leftarrow & \text{FREE} & \\ & v_1 & \leftarrow & z_{-1} + z_0 & // \; v_1 = z_{-1} + z_0 \\ & v_2 & \leftarrow & \sqrt{v_1} & // \; v_2 = \sqrt{z_{-1} + z_0} \end{array}$$

Let z_{-1} and z_0 move on the paths $p_{-1}, p_0 \colon [0,1] \to \mathbb{C}$, i.e., we consider the values $(p_{-1}(t), p_0(t))$ for (z_{-1}, z_0) with $t \in [0,1]$. We assume that the paths p_{-1} and p_0 can be extended to analytic functions that are defined on a connected neighborhood of $[0,1]$. A critical point occurs at time t if and only if $p_{-1}(t) = -p_0(t)$. By a well known theorem from complex analysis (German: "Identitätssatz"), $p_{-1}(t) = -p_0(t)$ holds either for all $t \in [0,1]$ or for at most finitely many $t \in [0,1]$.

We formalize the situations from the previous examples. If the GSP Γ has just one free variable z and if $\mathbb{K} = \mathbb{C}$, then the set of critical points is either finite or each point can be extended to a critical point.

Lemma 3.2.9. *If Γ is a GSP with just one free variable z and if $\mathbb{K} = \mathbb{C}$, then the set of critical points is either finite or every value for z can be extended to a critical point.*

Proof. Induction on the length of the GSP Γ.
If Γ only consists of the free variable z, then Γ has no critical points. We assume that Lemma 3.2.9 holds for GSPs with n dependent and one free variable. Now, let Γ be a GSP with $n+1$ dependent variables and one free variable z. Then the n-head $\Gamma^{(n)}$ of Γ is a GSP with one free variable z and n dependent ones. By the

induction hypothesis, $\Gamma^{(n)}$ has either finitely many critical points, or every value of z can be extended to a critical point of $\Gamma^{(n)}$. We observe that every critical point of $\Gamma^{(n)}$ induces a critical point of the original GSP Γ. Thus, if every value of z can be extended to a critical point of $\Gamma^{(n)}$, then every value of z can be extended to a critical point of Γ as well.

It remains to investigate the situation where $\Gamma^{(n)}$ has finitely many critical points. If the $n+1$st variable v_{n+1} of Γ is defined by an addition, subtraction, or multiplication, then v_{n+1} does not cause critical points. Hence, Γ has finitely many critical points. If v_{n+1} is defined by a division or square root operation, we regard the divisor or radicand variable v_c of v_{n+1}. The variable v_c is either the free variable z or a dependent variable v_j of Γ with $j \in \{1, 2, \ldots, n\}$. The $n+1$st variable v_{n+1} of Γ causes a critical point if $v_c = 0$. For $v_c = z$, this only occurs if $z = 0$, and the number of critical points remains finite. For $v_c = v_j$ with $j \in \{1, 2, \ldots, n\}$, the variable v_c in combination with the j-head GSP $\Gamma^{(j)}$ of Γ defines an algebraic function $v_c(z)$ in z on $\mathbb{C}$. We consider the corresponding Riemann surface (X_c, η_c, f_c) of $v_c(z)$ (German: "Riemannsches Gebilde" of $v_c(z)$), where X_c is a Riemann Surface, $\eta_c \colon X_c \to \mathbb{C}$ a covering, and $f_c \colon X_c \to \mathbb{C}$ a meromorphic function with $v_c(z) = f_c(\eta_c^{-1}(z))$. By construction of (X_c, η_c, f_c) we have

$$v_c(z) = \{w \in \mathbb{C} \mid w = v_c \text{ for a sign choice in } \Gamma^{(j)}\} = f_c\left(\eta_c^{-1}(z)\right);$$

see Section 8.3. The Riemann surface (X_c, η_c, f_c) might decompose into $m \geq 1$ components $(X_{c,i}, \eta_{c,i}, f_{c,i})$. Each component $(X_{c,i}, \eta_{c,i}, f_{c,i})$ is the Riemann surface of an algebraic function $v_{c,i}(z)$, and we have

$$v_c(z) = f_c\left(\eta_c^{-1}(z)\right) = \bigcup_{i=1}^{m} f_{c,i}\left(\eta_{c,i}^{-1}(z)\right) = \bigcup_{i=1}^{m} v_{c,i}(z).$$

Each algebraic function $v_{c,i}(z)$ has either finitely many zeros or is constantly zero; see Theorem 8.1.6. If all functions $v_{c,i}(z)$ have a finite number of zeros, then $v_c(t)$ has a finite number of zeros. Since the zeros of $v_c(z)$ are responsible for the additional critical points which are caused by the variable v_{n+1}, the set of critical points remains finite.

If at least one of the functions $v_{c,i}(z)$ is constantly zero, then $0 \equiv v_{c,i}(z) \in v_c(z)$ for all $z \in \mathbb{C}$. This implies that for every $z \in \mathbb{C}$ there is a sign choice for the square root operations of $\Gamma^{(j)}$ with $v_c = 0$, and every point $z \in \mathbb{C}$ can either be extended to a critical point of $\Gamma^{(n)}$ or to a zero of v_c in $\Gamma^{(j)}$. Thus, all values $z \in \mathbb{C}$ are critical values of Γ. □

Lemma 3.2.10. *Let Γ be a GSP over $\mathbb{C}$ with k free variables $z_{-k+1}, \ldots, z_0$. Let $U_{[0,1]} \subset \mathbb{C}$ be a connected neighborhood of the time interval $[0, 1]$, and let*

$p_{-k+1}, \dots, p_0 \colon U_{[0,1]} \to \mathbb{C}$ be polynomial functions in t. We assign the values $p_{-k+1}(t), \dots, p_0(t)$ to the free variables $z_{-k+1}, \dots, z_0$ for $t \in [0,1]$. Then either each time $t \in [0,1]$ induces a critical point or there are only finitely many values for t inducing a critical point.

Proof. Since the functions $p_l(t)$ are polynomials in t, they can be described by GSPs Γ_l with only one free variable t. In Γ, we replace the operations $z_l \leftarrow$ FREE by the operations of Γ_l. Since all functions $p_l(t)$ are polynomials in the variable t, we only need one operation $t \leftarrow$ FREE. The resulting GSP $\tilde{\Gamma}$ is a complex GSP with one free variable t. By construction of $\tilde{\Gamma}$, a point $t \in U_{[0,1]}$ induces a critical point of $\tilde{\Gamma}$ if and only if $(p_{-k+1}(t), \dots, p_0(t))$ induces a critical point of Γ. By Lemma 3.2.9, $\tilde{\Gamma}$ has either finitely many critical points or every value for t can be extended to a critical point. □

Note that Lemma 3.2.10 even holds for functions $p_l(t)$ in one variable t that can be described by a GSP.

Corollary 3.2.11. *Let Γ be a GSP over $\mathbb{C}$ with k free variables $z_{-k+1}, \dots, z_0$ together with k linear paths $p_{-k+1}, \dots, p_0$ for the free variables. Then, either every point $(p_{-k+1}(t), \dots, p_0(t))$ with $t \in [0,1]$ can be extended to a critical point of Γ, or there are only finitely many times t_c for which $(p_{-k+1}(t_c), \dots, p_0(t_c))$ can be extended to a critical point.*

Proof. Corollary 3.2.11 is a consequence of Lemma 3.2.10. □

We can get rid of the critical points that are caused by a division operation if we switch from $\mathbb{C}$ to the Riemann Sphere $\hat{\mathbb{C}}$ [32]. The basic idea of the Riemann Sphere is to add a point ∞ to the set of complex numbers $\mathbb{C}$ such that a division by zero can be defined. Hence, a division operation on the Riemann Sphere does not cause a critical point.

3.3 Continuity

In Dynamic Geometry, we are dealing with *dynamic constructions*: If a free point is moved in a continuous way, the whole construction should follow continuously. Whenever the free points move on continuous paths, the dependent elements have to move on continuous paths as well (as long as no critical points lie on the paths). This concept is formalized in the notion of continuous evaluations defined by Kortenkamp and Richter-Gebert in [43]. As before, we consider the algebraic situation.

Definition 3.3.1. Continuous Evaluation
Let Γ *be a GSP having* k *free variables and* n *dependent elements; w.l.o.g. let the first* k *variables* $z_{-k+1}, \ldots, z_0$ *be the free variables and* $v_1, \ldots, v_n$ *be the dependent ones. Furthermore, for each free variable* z_l *we are given a continuous path* $p_l(t) : [0,1] \to \mathbb{K}$. A continuous evaluation *of the GSP* Γ *under the movement* $\{p_l(t)\}$ *is an assignment of continuous functions*

$$v_i(t) : [0,1] \to \mathbb{K}$$

to the dependent variables, i.e. for each dependent variable v_i *there is a function* $v_i(t)$ *such that for all* $t \in [0,1]$

$$\big(p_{-k+1}(t), \ldots, p_0(t), v_1(t), \ldots, v_n(t)\big)$$

is an instance of the GSP Γ.

If the GSP Γ does not contain $\sqrt{\_}$-operations, then the values of the dependent variables are determined by the values of the free variables. Since $+$, $-$, $\cdot$ and $/$ are continuous functions (unless there is a division by zero), there is exactly one continuous evaluation for a given continuous motion of the free variables (avoiding a division by zero).

If the GSP Γ contains $\sqrt{\_}$-operations, the values of the dependent variables are determined by the values of the free variables *up to a number of binary choices* that correspond to the two branches of the $\sqrt{\_}$-function. The definition of continuous evaluation ensures that we always choose the "right" branch and do not jump from one branch to the other one.

Continuous evaluations are unique as discussed in Section 8.6 and in [13]. If a starting instance is fixed (i.e. values for $v_1(0), \ldots, v_n(0)$) and if we do not hit a critical point, then there is exactly one continuous evaluation for the given motion starting at this starting instance. Note that the notion of critical points in [13] differs slightly from ours.

At a critical point, the radicand or divisor variable of a root or division operation, respectively, becomes zero. Thus we are interested in the zeros of the corresponding paths in a continuous evaluation. Lemma 3.3.2 is a consequence of a famous theorem from complex analysis (German: "Identitätssatz").

Lemma 3.3.2. *Let* Γ *be a GSP over* $\mathbb{C}$ *with* k *free variables and* n *dependent variables* $v_1, \ldots, v_n$. *Let* A *be an instance of* Γ, *and let* $p_{-k+1}, \ldots, p_0\colon [0,1] \to \mathbb{C}$ *be paths of the free variables of* Γ *that can be extended to analytic functions on a connected neighborhood* $U_{[0,1]} \subset \mathbb{C}$ *of the time interval* $[0,1]$ *and for which the corresponding continuous evaluation along* $p_{-k+1}, \ldots, p_0$ *starting at* A *exists. Let* $v_i(t)$ *be the path of the dependent variable* v_i *in the continuous evaluation. Then* $v_i(t)\colon [0,1] \to \mathbb{C}$ *either has at most a finite number of zeros, or this path is constantly zero.*

Proof. Since the paths $p_{-k+1}(t), \ldots, p_0(t)$ can be extended to analytic functions on $U_{[0,1]}$ and since the corresponding continuous evaluation exists, the path $v_i(t)$ of the dependent variable v_i can be extended to an analytic function $\hat{v}_i$ on a (possibly smaller) neighborhood $\hat{U}_{[0,1]}$ of $[0,1]$ as well; see Lemma 8.6.13 of Appendix 8.6.1. A famous result from complex analysis implies that either the zeros of $\hat{v}_i$ form a discrete set in $\hat{U}_{[0,1]}$ or $\hat{v}_i$ is the zero function. If $\hat{v}_i$ is the zero function, then $v_i = \hat{v}_{i|_{[0,1]}}$ is the zero function as well. If the zeros of $\hat{v}_i$ form a discrete set in $\hat{U}_{[0,1]}$, then only a finite number of them is contained in the compact interval $[0,1]$: Since $\hat{v}_i$ is continuous, the zero set $\hat{v}_i^{-1}(0)$ and hence $\hat{v}_i^{-1}(0) \cap [0,1]$ are closed sets, and $[0,1] \setminus \hat{v}_i^{-1}(0)$ is an open subset of $[0,1]$. Since the zeros of $\hat{v}_i$ form a discrete set in $\hat{U}_{[0,1]}$, every zero ξ has an open neighborhood U_ξ that does not contain another zero of $\hat{v}_i$. Thus, the set $\mathcal{U} := \{U_\xi \cap [0,1] \mid \xi \in [0,1] \text{ with } \hat{v}_i(\xi) = 0\} \cup \{[0,1] \setminus \hat{v}_i^{-1}(0)\}$ is an open cover of $[0,1]$. Every zero ξ of $\hat{v}_i$ in $[0,1]$ is covered by exactly one element of $\mathcal{U}$. Since $[0,1]$ is compact, $\mathcal{U}$ has a finite subcover. This finite subcover can cover at most a finite number of zeros $\xi \in [0,1]$ of $\hat{v}_i$. □

3.4 The Tracing Problem and the Reachability Problem

We define the Tracing Problem and the Reachability Problem [43, 13]. Recall that we are considering GSPs using the operations addition, subtraction, multiplication, division, and square root, and the object set is $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$. Let Γ be a GSP with k free and n dependent variables.

Problem 1. Tracing Problem

Given: *A GSP Γ with k free and n dependent variables, a starting instance $A = (z_A, v_A) \in \mathbb{K}^k \times \mathbb{K}^n$, a final instance $B = (z_B, v_B) \in \mathbb{K}^k \times \mathbb{K}^n$ of Γ, and continuous paths $p_{-k+1}, \ldots p_0 : [0,1] \to \mathbb{K}$ of the free variables of Γ with $(p_{-k+1}(0), \ldots, p_0(0)) = z_A$ and $(p_{-k+1}(1), \ldots, p_0(1)) = z_B$ are given. We assume that the resulting continuous evaluation starting at A exists.*

Decide: *Does this continuous evaluation end at the given instance B?*

Recall that critical points of a GSP Γ cannot be instances of Γ and that, in a continuous evaluation, we have an instance at each time $t \in [0,1]$. Thus, critical points on the motion are excluded in advance by the requirement that the corresponding continuous evaluation exists.

Here, we have formulated the Tracing Problem as a decision problem. To solve this problem, we "just" have to decide whether the unique continuous evaluation under the given movement $p_{-k+1}, \ldots, p_0$ and the starting instance A ends at B.

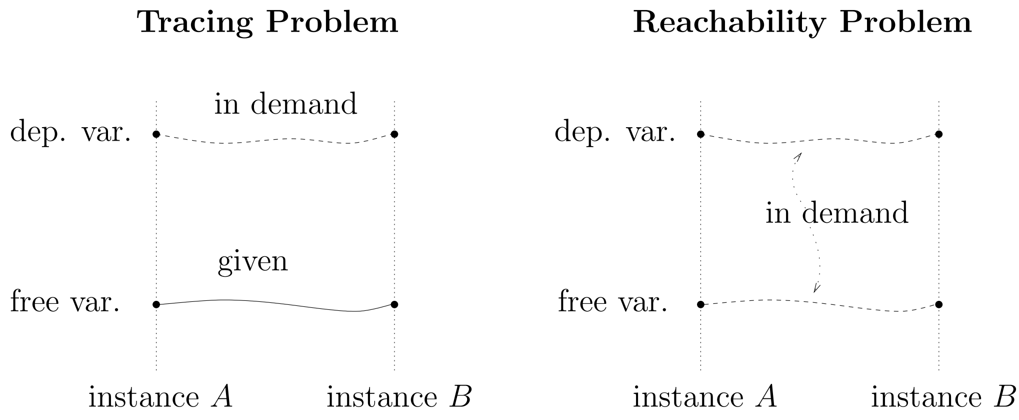


Figure 3.1: Exemplification of the difference between the Tracing and the Reachability Problem. As usual, A denotes the starting instance and B the final instance.

In many applications like the drag mode in a Dynamic Geometry Software, we are interested in the entire continuous evaluation, as well. In addition to the final instance, we also want to determine the values of the dependent variables at some intermediate times $t \in (0, 1)$.

Problem 2. The Reachability Problem

Given: *A GSP* Γ *with* k *free and* n *dependent variables, a starting instance* $A = (z_A, v_A) \in \mathbb{K}^k \times \mathbb{K}^n$ *and a final instance* $B = (z_B, v_B) \in \mathbb{K}^k \times \mathbb{K}^n$ *of* Γ *are given.*

Decide: *Are there continuous paths* $p_l : [0, 1] \to \mathbb{K}$ *of the free variables for which there is a corresponding continuous evaluation from* A *to* B*?*

The difference between both problems is that for the Tracing Problem, the paths of the free variables are given, whereas for the Reachability Problem they are not given (see Figure 3.1).

Both problems are NP-hard as shown in [43]. The Reachability Problem is decidable, and the Tracing Problem is decidable for piecewise polynomial paths; see [13] and Chapter 4. The algorithms for the Tracing Problem from Chapter 6 deal with linear paths $p_{-k+1}, \ldots, p_0$ of the free variables of Γ. There is indicated how the algorithms can be adapted to a more general situation. In Chapter 7, we investigate the usage of semi circular paths $p_{-k+1}, \ldots, p_0$ to bypass a critical point.

3.5 The Derivative of a GSP with Respect to Time

For the Tracing Problem, we are given a GSP Γ, paths $p_{-k+1}, \ldots, p_0$ of the free variables, and a starting instance A. They define the corresponding continuous evaluation $(v_1(t), \ldots, v_n(t))$ implicitly if it exists. The aim of this section is to define the derivative $\dot{\Gamma}$ of the GSP Γ, which implicitly describes the derivative $(\dot{v}_1(t), \ldots, \dot{v}_n(t))$ of the continuous evaluation $(v_1(t), \ldots, v_n(t))$ with respect to time. We assume that the paths $p_{-k+1}, \ldots, p_0$ of the free variables of Γ are continuously differentiable. Before explaining the general case, we look at an example.

Example 3.5.1. We consider the GSP Γ describing $\sqrt{z^2-1}$ from Example 3.1.3. The GSP Γ has one free variable z and three dependent ones v_1, v_2, v_3. We choose the path $p\colon [0,1] \to \mathbb{C},\ t \mapsto 2t+3$, for the free variable z and the starting instance $A = (3, 9, 8, +\sqrt{8})$.

$$
\begin{array}{lrcll rcl}
\text{GSP } \Gamma: & z & \leftarrow & \text{FREE} & // & p(t) & = & 2t+3 \\
& v_1 & \leftarrow & z \cdot z & // \ v_1 = z^2, & v_1(t) & = & 4t^2+12t+9 \\
& v_2 & \leftarrow & v_1 - 1 & // \ v_2 = z^2 - 1, & v_2(t) & = & 4t^2+12t+8 \\
& v_3 & \leftarrow & \sqrt{v_2} & // \ v_3 = \pm\sqrt{z^2-1}, & v_3(t) & = & +\sqrt{4t^2+12t+8}
\end{array}
$$

We are interested in the derivatives $\dot{v}_1(t)$, $\dot{v}_2(t)$ and $\dot{v}_3(t)$. By the chain rule,

$$
\begin{array}{rclcl}
\dot{p}(t) & = & 2 & & \\
\dot{v}_1(t) & = & 2p(t)\dot{p}(t) & = & 8t+12 \\
\dot{v}_2(t) & = & \dot{v}_1(t) & = & 8t+12 \\
\dot{v}_3(t) & = & \frac{\dot{v}_2(t)}{2v_3(t)} & = & \frac{8t+12}{+2\sqrt{4t^2+12t+8}}.
\end{array}
$$

In a GSP-like notation we get

$$
\begin{array}{rcl}
\dot{z} & \leftarrow & \text{FREE} \\
\dot{v}_1 & \leftarrow & 2 \cdot z \cdot \dot{z} \\
\dot{v}_2 & \leftarrow & \dot{v}_1 \\
\dot{v}_3 & \leftarrow & \frac{\dot{v}_2}{2v_3}.
\end{array}
$$

Hence, the derivative $(\dot{p}(t), \dot{v}_1(t), \dot{v}_2(t), \dot{v}_3(t))$ with respect to time of the continuous evaluation $(p(t), v_1(t), v_2(t), v_3(t))$ is implicitly defined by Γ, $p(t)$, the starting instance A, and the variables $\dot{z}, \dot{v}_1, \dot{v}_2, \dot{v}_3$. For the definition of the variables $\dot{v}_1$ and $\dot{v}_3$, the variables z and v_3 of the GSP Γ are needed. These observations immediately lead to Definition 3.5.2 and Lemma 3.5.3.

Definition 3.5.2. Derivative GSP $\dot{\Gamma}$ of a GSP Γ with respect to time
Let Γ be a GSP over $\{\mathbb{R}, +, -, \cdot, /, \sqrt{\_}\}$ or $\{\mathbb{C}, +, -, \cdot, /, \sqrt{\_}\}$ with k free variables

$z_{-k+1}, \ldots, z_0$ and n dependent variables $v_1, \ldots, v_n$. Then the derivative GSP $\dot{\Gamma}$ consists of $2k$ free variables $\dot{z}_{-k+1}, \ldots, \dot{z}_0, z_{-k+1}, \ldots, z_0$ and at most $6n$ dependent ones including $\dot{v}_1, \ldots, \dot{v}_n$ and $v_1, \ldots, v_n$ according to the operations of Γ. The operations of $\dot{\Gamma}$ are defined as follows:

Operation of Γ	*Derivative* $\dot{v}_a(t)$ *of* $v_a(t)$	*Operations of* $\dot{\Gamma}$
$v_a \leftarrow v_b + v_c$	$\dot{v}_a(t) = \dot{v}_b(t) + \dot{v}_c(t)$	$v_a \leftarrow v_b + v_c$ $\dot{v}_a \leftarrow \dot{v}_b + \dot{v}_c$
$v_a \leftarrow v_b - v_c$	$\dot{v}_a(t) = \dot{v}_b(t) - \dot{v}_c(t)$	$v_a \leftarrow v_b - v_c$ $\dot{v}_a \leftarrow \dot{v}_b - \dot{v}_c$
$v_a \leftarrow v_b \cdot v_c$	$\dot{v}_a(t) = \dot{v}_b(t)v_c(t) + v_b(t)\dot{v}_c(t)$	$v_a \leftarrow v_b \cdot v_c$ $v_a^{(1)} \leftarrow \dot{v}_b \cdot v_c$ $v_a^{(2)} \leftarrow v_b \cdot \dot{v}_c$ $\dot{v}_a \leftarrow v_a^{(1)} + v_a^{(2)}$
$v_a \leftarrow v_b / v_c$	$\dot{v}_a(t) = \frac{\dot{v}_b(t)v_c(t) - v_b(t)\dot{v}_c(t)}{(v_c(t))^2}$	$v_a \leftarrow v_b / v_c$ $v_a^{(1)} \leftarrow \dot{v}_b \cdot v_c$ $v_a^{(2)} \leftarrow v_b \cdot \dot{v}_c$ $v_a^{(3)} \leftarrow v_a^{(1)} - v_a^{(2)}$ $v_a^{(4)} \leftarrow v_c \cdot v_c$ $\dot{v}_a \leftarrow v_a^{(3)} / v_a^{(4)}$
$v_a \leftarrow \sqrt{v_c}$	$\dot{v}_a(t) = \frac{\dot{v}_c(t)}{2v_a(t)}$	$v_a \leftarrow \sqrt{v_c}$ $v_a^{(1)} \leftarrow 2 \cdot v_a$ $\dot{v}_a \leftarrow \dot{v}_c / v_a^{(1)}$

We observe that the derivative GSP $\dot{\Gamma}$ contains all variables of the GSP Γ. The variables $\dot{v}_a$ describe the "derivatives" of the variables v_a of Γ. The variables $v_a^{(1)}, \ldots, v_a^{(4)}$ are auxiliary variables, they are needed for translating the differentiation rules to a GSP. They are not important for our applications and will be omitted.

Lemma 3.5.3. *Let A be an instance of the GSP Γ and $\dot{a}_{-k+1}, \ldots, \dot{a}_0$ values for the free variables $\dot{z}_{k+1}, \ldots, \dot{z}_0$ of $\dot{\Gamma}$. Then A and $\dot{a}_{-k+1}, \ldots, \dot{a}_0$ define a unique instance $\dot{A}$ of $\dot{\Gamma}$.*

Let $\dot{\Gamma}$ be the derivative GSP of Γ. Let $p_{-k+1}, \ldots, p_0$ be continuously differentiable paths of the free variables $z_{-k+1}, \ldots, z_0$ of Γ, and let

$$A = \big(p_{-k+1}(0), \ldots, p_0(0), a_1, \ldots, a_n\big)$$

be an instance of Γ. There is a continuous evaluation of Γ along the paths $p_{-k+1}, \ldots, p_0$ starting at A if and only if there is a continuous evaluation of $\dot{\Gamma}$ along the paths $\dot{p}_{-k+1}, \ldots, \dot{p}_0, p_{-k+1}, \ldots, p_0$ starting at $\dot{A}$ with $\dot{a}_l = \dot{p}(0)$; $l = -k+1, \ldots, 0$.

Let $\dot{v}_i(t)$ be the path of the variable $\dot{v}_i$ in a continuous evaluation of $\dot{\Gamma}$. Then the path $\dot{v}_i(t)$ is the derivative of $v_i(t)$ with respect to time, $i = 1, \ldots, n$.

Proof. Lemma 3.5.3 holds by construction and can be proved by induction on n. Note that the divisor or radicand variables of $\dot{\Gamma}$ are exactly the divisor or radicand variables of the original GSP Γ. □

3.6 GSPs and Interval Arithmetic

Formally, we can use interval arithmetic in GSPs. We define the interval-GSP Γ_{int} induced by a GSP Γ. Here, the variables take intervals as values and the operations are performed in interval arithmetic. As before, let $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$, and let $\mathrm{I}(\mathbb{K}) \in \{\mathrm{I}(\mathbb{R}), \mathrm{R}(\mathbb{C}), \mathrm{K}(\mathbb{C})\}$ be the set of real closed intervals, rectangular complex intervals, or circular complex intervals, respectively. We give an introduction to interval arithmetic in Appendix A.

Definition 3.6.1. Interval-GSP Γ_{int}
Let Γ be a GSP over $\{\mathbb{K}, +, -, \cdot, /, \sqrt{\_}\}$ having the free variables $z_{-k+1}, \ldots, z_0$ and the dependent variables $v_1, \ldots, v_n$. Then, Γ_{int} is a GSP-like structure over $\{I(\mathbb{K}), +, -, \cdot, /, \sqrt{\_}\}$ with k free variables $Z_{-k+1}, \ldots, Z_0$ and n dependent variables $V_1, \ldots, V_n$. Every dependent variable V_i of Γ_{int}, $i \in \{1, \ldots, n\}$, is defined by the interval operation that corresponds to the operation defining the variable v_i of Γ:

Operation of Γ	*Operation of* Γ_{int}
$v_a \leftarrow v_b + v_c$	$V_a \leftarrow V_b + V_c$
$v_a \leftarrow v_b - v_c$	$V_a \leftarrow V_b - V_c$
$v_a \leftarrow v_b \cdot v_c$	$V_a \leftarrow V_b \cdot V_c$
$v_a \leftarrow v_b / v_c$	$V_a \leftarrow V_b / V_c$
$v_a \leftarrow \sqrt{v_c}$	$V_a \leftarrow \sqrt{V_c}$

We also use the notion of an instance in the context of interval arithmetic: An *instance* of Γ_{int} is an assignment of (real or complex) intervals to all variables of Γ_{int} such that all relations given by Γ_{int} are fulfilled. We assume that the divisor intervals of the division operations and the radicand intervals of the root operations do not contain 0.

Remark 3.6.2. For a GSP Γ over $\{\mathbb{R}, +, -, \cdot, /, \sqrt{\_}\}$ or $\{\mathbb{C}, +, -, \cdot, /, \sqrt{\_}\}$ the operations are defined by relations. This definition is important for the square root operation since it is the only operation that is not determined: The GSP Γ

does not fix the sign of the output of the root operation. The subtraction, division, and square root operations are defined as follows; see page 32:

Operation of Γ	Corresp. Relation
$v_c \leftarrow v_a - v_b$	$v_a = v_c + v_b$
$v_c \leftarrow v_a / v_b$	$v_a = v_c \cdot v_b$
$v_c \leftarrow \sqrt{v_a}$	$v_a = v_c^2$

For the interval-GSP Γ_{int}, this description does not lead to the expected results. For real interval arithmetic, we have for example

$$\begin{aligned}[1,2]-[0,3] &= [-2,2], \text{ but} \\ [-2,2]+[0,3] &= [-2,5] \neq [1,2].\end{aligned}$$

Hence, it does not make sense to reduce the operations addition, subtraction, multiplication, division, and square root to the two operations addition and multiplication, only. Instead, we use the original operations themselves. Again, the square root operation is not determined. As with GSPs over $\{\mathbb{R}, +, -, \cdot, /, \sqrt{\_}\}$ or $\{\mathbb{C}, +, -, \cdot, /, \sqrt{\_}\}$, the root operation of Γ_{int} does not "fix" the sign of the result of a root operation as indicated below. Let $\mathrm{I}(\mathbb{C}) \in \{\mathrm{R}(\mathbb{C}), \mathrm{K}(\mathbb{C})\}$ be the set of either rectangular or circular complex intervals. As before, $\mathrm{I}(\mathbb{R})$ is the set of real intervals.

I($\mathbb{R}$): Let $I = [a,b]$ be an interval with $a > 0$. Then $\sqrt{I} = \left[\sqrt{a}, \sqrt{b}\right]$ or $\sqrt{I} = \left[-\sqrt{b}, -\sqrt{a}\right] = -\left[\sqrt{a}, \sqrt{b}\right]$.

I($\mathbb{C}$): Let $I \in \mathrm{I}(\mathbb{C})$ with $0 \notin I$. Let $\sqrt[+]{\_}$ and $\sqrt[-]{\_}$ be the two branches of the root function that are defined on I. Then $\sqrt{I} = \sqrt[+]{I}$ or $\sqrt{I} = \sqrt[-]{I} = -\sqrt[+]{I}$.

We can estimate the range of a continuous evaluation with an instance of Γ_{int} as the following lemmas for $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$ show.

Lemma 3.6.3. *Let v_a be a dependent variable of a GSP Γ that is defined by the variables v_b and v_c of Γ. Let v_b and v_c be composed by one of the operations $+$, $-$, $\cdot$, $/$ or $\sqrt{\_}$. Let $A = (a_{-k+1}, \ldots, a_0, a_1, \ldots, a_n)$ be an instance of Γ, and let $p_{-k+1}, \ldots, p_0$ be paths of the free variables of Γ such that the corresponding continuous evaluation $v_1(t), \ldots, v_n(t)$ exists. Let I_b and I_c be intervals in $\mathbb{K}$ with $v_b([t_1,t_2]) \subset I_b$ and $v_c([t_1,t_2]) \subset I_c$. Then, an interval I_a with $v_a([t_1,t_2]) \subset I_a$ can be computed in the following way:*

Operation of Γ	*Interval I_a*
$v_a \leftarrow v_b + v_c$	$I_a = I_b + I_c$
$v_a \leftarrow v_b - v_c$	$I_a = I_b - I_c$
$v_a \leftarrow v_b \cdot v_c$	$I_a = I_b \cdot I_c$
$v_a \leftarrow v_b / v_c$	$I_a = I_b / I_c$, *if* $0 \notin I_c$
$v_a \leftarrow \sqrt{v_c}$	$I_a = \sqrt{I_c} \ni v_a(t_1)$, *if* $0 \notin I_c$

Proof. Let $t \in [t_1, t_2]$. By the assumptions of Lemma 3.6.3, we have $v_b(t) \in I_b$ and $v_c(t) \in I_c$. Hence $v_a(t) = v_b(t) \circ v_c(t) \in I_b \circ I_c = I_a$ holds for $\circ \in \{+, -, \cdot, /\}$. If v_a is defined by a root operation, then $v_a(t) = \sqrt{v_c(t)} \in \sqrt{I_c} = I_a$. Since $0 \notin I_c$, we have $\sqrt{I_c} \cap -\sqrt{I_c} = \emptyset$, and the condition $v_a(t_1) \in \sqrt{I_c}$ uniquely defines the "sign" of $I_a = \sqrt{I_c}$. □

Lemma 3.6.4. *Let Γ be a GSP, $p_{-k+1}, \ldots, p_0 \colon [0,1] \to \mathbb{K}$ be paths of the free variables of Γ, and let $A = (a_{-k+1}, \ldots, a_0, a_1, \ldots, a_n)$ be a starting instance such that the corresponding continuous evaluation $(v_1(t), \ldots, v_n(t))$ exists. Let Γ_{int} be the interval-GSP induced by Γ, and let $(I_{-k+1}, \ldots, I_0, I_1, \ldots, I_n)$ be an instance of Γ_{int} with $a_i \in I_i$ for $i = -k+1, \ldots, n$ and $p_l([0,1]) \subset I_l$ for $l = -k+1, \ldots, 0$. Then $v_i([0,1]) \subset I_i$ holds for $i = 1, \ldots, n$.*

Proof. Lemma 3.6.4 can be proved by induction on n using Lemma 3.6.3. □

By reparametrization, Lemma 3.6.4 holds for arbitrary time intervals $[t_1, t_2]$ as well. Let $J_{-k+1}, \ldots, J_0, J_1, \ldots, J_n$ be intervals with $I_m \subset J_m$ for all $m \in \{-k+1, \ldots, 0, 1, \ldots, n\}$. Then Lemma 3.6.4 still holds if the interval-instance $(I_{-k+1}, \ldots, I_0, I_1, \ldots, I_n)$ is replaced by $(J_{-k+1}, \ldots, J_0, J_1, \ldots, J_n)$.

For interval-GSPs, we take interval dependency [35, p. 4] into account as shown in Example 3.6.5.

Example 3.6.5. We look at the polynomial $f(x) = x^2 - x = x(x-1)$ from Example A.1.4 on page 153 and describe f by two GSPs Γ and $\tilde{\Gamma}$:

$$\begin{array}{rrcl} \Gamma: & z & \leftarrow & \text{FREE} \\ & v_1 & \leftarrow & z - 1 \\ & v_2 & \leftarrow & z \cdot v_1 \;//\; v_2 = z(z-1) \end{array} \qquad \begin{array}{rrcl} \tilde{\Gamma}: & \tilde{z} & \leftarrow & \text{FREE} \\ & \tilde{v}_1 & \leftarrow & \tilde{z} \cdot \tilde{z} \\ & \tilde{v}_2 & \leftarrow & \tilde{v}_1 - \tilde{z} \;//\; \tilde{v}_2 = \tilde{z}^2 - \tilde{z} \end{array}$$

Let $A = (a_0, a_1, a_2)$ be an instance of Γ, and let $\tilde{A} = (\tilde{a}_0, \tilde{a}_1, \tilde{a}_2)$ be an instance of $\tilde{\Gamma}$. Then $a_0 = \tilde{a}_0$ implies $a_2 = \tilde{a}_2$. Now, let $A_{\text{int}} = (I_0, I_1, I_2)$ be an instance of the interval-GSP Γ_{int} of Γ, and let $\tilde{A}_{\text{int}} = (\tilde{I}_0, \tilde{I}_1, \tilde{I}_2)$ be an instance of the interval-GSP $\tilde{\Gamma}_{\text{int}}$ of $\tilde{\Gamma}$. We show that $I_0 = \tilde{I}_0$ does not imply $I_2 = \tilde{I}_2$. We consider the real interval $[0,1]$ and the corresponding instances of Γ_{int} and of $\tilde{\Gamma}_{\text{int}}$:

$$\begin{array}{rrclclcl} \Gamma_{\text{int}}: & I_0 & = & [0,1] & & & & \\ & I_1 & = & I_0 - 1 & = & [0,1] - 1 & = & [-1,0] \\ & I_2 & = & I_0 \cdot I_1 & = & [0,1] \cdot [-1,0] & = & [-1,0] \end{array}$$

$$\begin{array}{rrclclcl} \tilde{\Gamma}_{\text{int}}: & \tilde{I}_0 & = & [0,1] & & & & \\ & \tilde{I}_1 & = & \tilde{I}_0 \cdot \tilde{I}_0 & = & [0,1] \cdot [0,1] & = & [0,1] \\ & \tilde{I}_2 & = & \tilde{I}_1 - \tilde{I}_0 & = & [0,1] - [0,1] & = & [-1,1] \end{array}$$

Although $I_0 = \tilde{I}_0 = [0,1]$, we have $I_2 \neq \tilde{I}_2$.

Chapter 4

Decidability Aspects in Dynamic Geometry

Investigating the decidability of the Reachability Problem and the Tracing Problem is an important step to determine the complexity class of the two problems. Kortenkamp and Richter-Gebert have shown that both problems are NP-hard [43]. The Reachability Problem and the Tracing Problem for polynomial paths of the free variables can be decided in the following way [13]: In a first step, the connected components of a suitable semi-algebraic set are computed. In a second step, we check whether the starting instance A and the final instance B lie in the same connected component. A semi-algebraic set is a set of points in an $\mathbb{R}^n$ satisfying a boolean combination of polynomial equalities and inequalities. In [4], Basu, Pollack and Roy give an algorithm for computing the connected components of semi-algebraic sets.

4.1 The Reachability Problem

First, we focus on the real situation $\mathbb{K} = \mathbb{R}$. Afterwards, we describe how the algorithm can be extended to the case $\mathbb{K} = \mathbb{C}$. A GSP Γ defines in a natural way a semi-algebraic set $\mathcal{R}_{\mathbb{R}}(\Gamma)$, as we first explain with Example 3.1.3 from page 33.

Example 4.1.1. We recall the GSP Γ from Example 3.1.3, which describes the algebraic expression $\sqrt{z^2 - 1}$.

$$\begin{array}{llll} \Gamma: & z & \leftarrow & \text{FREE} \\ & v_1 & \leftarrow & z \cdot z \\ & v_2 & \leftarrow & v_1 - 1 \\ & v_3 & \leftarrow & \sqrt{v_2}\,; \qquad \text{described by } v_3^2 = v_2 \text{ and } v_2 > 0. \end{array}$$

The corresponding semi-algebraic set is

$$\mathcal{R}_{\mathbb{R}}(\Gamma) := \{(z, v_1, v_2, v_3) \in \mathbb{C}^4 \mid v_1 = z^2 \wedge v_2 = v_1 - 1 \wedge v_3^2 = v_2 \wedge v_2 > 0\}.$$

Now we consider the general case, where we are given an arbitrary GSP Γ. For each operation γ of Γ describing a dependent variable we define a term $\tau(\gamma)$ as follows:

Operation γ	Term $\tau(\gamma)$
$a \leftarrow b + c$	$a = b + c$
$a \leftarrow b - c$	$a = b - c$
$a \leftarrow b \cdot c$	$a = b \cdot c$
$a \leftarrow b/c$	$b = a \cdot c \ \wedge \ c \neq 0$
$a \leftarrow \sqrt{b}$	$a^2 = b \ \wedge \ b > 0$

Furthermore, let Γ have k free variables and n dependent ones defined by the operations $\gamma_1, \ldots, \gamma_n$. Then we set

$$\mathcal{R}_{\mathbb{R}}(\Gamma) := \{(z_{-k+1}, \ldots, z_0, v_1, \ldots, v_n) \in \mathbb{R}^{k+n} \mid \tau(\gamma_1) \wedge \tau(\gamma_2) \wedge \cdots \wedge \tau(\gamma_n)\}.$$

To decide the Reachability Problem, we have to check whether the starting instance A and the final instance B lie in the same connected component of $\mathcal{R}_{\mathbb{R}}(\Gamma)$. This fact is stated in the following lemma.

Lemma 4.1.2. *Let $\mathbb{K} = \mathbb{R}$, and let A and B two instances of a GSP Γ. Then there is a continuous evaluation connecting A and B if and only if A and B lie in the same connected component of $\mathcal{R}_{\mathbb{R}}(\Gamma)$.*

Proof. Let A and B be instances of Γ. Then by definition, the coordinates of A and of B fulfill the relations of Γ, hence $A, B \in \mathcal{R}_{\mathbb{R}}(\Gamma)$. The same argument shows that a continuous evaluation is a continuous path in $\mathcal{R}_{\mathbb{R}}(\Gamma)$, and each path in $\mathcal{R}_{\mathbb{R}}(\Gamma)$ is a continuous evaluation. This implies that there is a continuous evaluation connecting A and B if and only if A and B lie in the same path-connected component of $\mathcal{R}_{\mathbb{R}}(\Gamma)$. Since $\mathcal{R}_{\mathbb{R}}(\Gamma)$ is a semi-algebraic set, the path-connected components of $\mathcal{R}_{\mathbb{R}}(\Gamma)$ coincide with the connected components of $\mathcal{R}_{\mathbb{R}}(\Gamma)$; see [4, Sect. 3.2 and Sect. 5.2]. □

The connected components of $\mathcal{R}_{\mathbb{R}}(\Gamma)$ can be computed with the algorithm of Basu et al. [4, Sect. 16.4]. For each connected component, this algorithm outputs a boolean combination of polynomial equalities and inequalities that describes the connected component. A connected component contains the instances A and B if and only if the coordinates of A and of B fulfill this boolean combination of polynomial equalities and inequalities. Thus, we have proven the following theorem.

Theorem 4.1.3. *The Reachability Problem is decidable for $\mathbb{K} = \mathbb{R}$.*

If we are dealing with the complex Reachability Problem ($\mathbb{K} = \mathbb{C}$), we can use the same approach. Then the term $\tau(\gamma)$ of a $\sqrt{\_}$-operation $\gamma = (a \leftarrow \sqrt{b})$ is defined by $\tau(\gamma) := (a^2 = b \wedge b \neq 0)$. Additionally, each complex variable v of the GSP Γ is split into two variables v_r and v_i representing the real and the imaginary part of v. We call the corresponding semi-algebraic set $\mathcal{R}_{\mathbb{C}}(\Gamma)$.

Corollary 4.1.4. *The Reachability Problem is decidable for* $\mathbb{K} = \mathbb{C}$.

4.2 The Tracing Problem

We restrict on *polynomial paths* for the free variables. As for the Reachability Problem, we start with the real situation, and we use the GSP Γ from Example 3.1.3 describing $\sqrt{z^2 - 1}$ to explain the algorithm; see Example 4.1.1. In addition to the GSP Γ, we are given a polynomial path $p(t)\colon [0,1] \to \mathbb{R}$ of the free variable z. This defines the following semi-algebraic set:

$$\begin{aligned}\mathcal{T}_{\mathbb{R}}(\Gamma_0) := \{\, (t, z, v_1, v_2, v_2) \in \mathbb{R} \times \mathbb{R}^4 \;|0 \leq t \leq 1 \;\wedge\; z = p(t) \;\wedge\; v_1 = z^2 \;\wedge \\ v_2 = v_1 - 1 \;\wedge\; v_3^2 = v_2 \;\wedge\; v_2 > 0\}\end{aligned}$$

More generally, let Γ be an arbitrary GSP having the k free variables $z_{-k+1}, \ldots, z_0$ and the n dependent ones $v_1, \ldots, v_n$. Let $p(t) = (p_{-k+1}(t), \ldots, p_0(t))\colon [0,1] \to \mathbb{R}^k$ be a polynomial path of the free variables. Using the same notation as in the definition of $\mathcal{R}_{\mathbb{R}}(\Gamma)$, we define

$$\begin{aligned}\mathcal{T}_{\mathbb{R}}(\Gamma) := \{\, (t, z_{-k+1}, \ldots, z_0, v_1, \ldots, v_n) \in \mathbb{R} \times \mathbb{R}^{k+n} \mid 0 \leq t \leq 1 \\ \wedge\; z_{-k+1} = p_{-k+1}(t) \wedge \cdots \wedge z_0 = p_0(t) \wedge \tau(\gamma_1) \wedge \tau(\gamma_2) \wedge \cdots \wedge \tau(\gamma_n)\}.\end{aligned}$$

To decide the Tracing Problem, we have to determine whether A and B lie in the same connected component of $\mathcal{T}_{\mathbb{R}}(\Gamma)$. Additionally, at A we must have $t = 0$, and at B we must have $t = 1$.

Lemma 4.2.1. *Let* $A = (z_A, v_A) \in \mathbb{R}^k \times \mathbb{R}^n$ *and* $B = (z_B, v_B) \in \mathbb{R}^k \times \mathbb{R}^n$ *be instances of a GSP* Γ *and* $p(t)$ *be a polynomial path of the free variables of* Γ *with* $p(0) = z_A$ *and* $p(1) = z_B$. *There exists a continuous evaluation along* $p(t)$ *starting at the instance* A, *and this continuous evaluation ends at* B, *if and only if* A *and* B *lie in the same connected component of* $\mathcal{T}_{\mathbb{R}}(\Gamma)$, *at* A *we have* $t = 0$, *and at* B *we have* $t = 1$.

Proof. The proof is similar to the proof of Lemma 4.1.2. Two continuous evaluations only could meet at an instance where the radicand of a $\sqrt{\_}$-operation is zero. Since these are excluded in $\mathcal{T}_{\mathbb{R}}(\Gamma)$, a continuous evaluation along the path $p(t)$ is a connected component of $\mathcal{T}_{\mathbb{R}}(\Gamma)$. □

We can adapt this algorithm for the Tracing Problem to piecewise polynomial paths $p(t) = (p_{-k+1}(t), \ldots, p_0(t))\colon [0,1] \to \mathbb{R}^k$ in a straight forward way. A path $p(t) = (p_{-k+1}(t), \ldots, p_0(t))\colon [0,1] \to \mathbb{R}^k$ is a piecewise polynomial path, if there is a subdivision of the time interval $[0,1]$ into finitely many intervals $[t_i, t_{i+1}]$ such that the restriction $p_{|[t_i,t_{i+1}]}$ is a polynomial path. Thus, in the definition of the semi-algebraic set $\mathcal{T}_{\mathbb{R}}(\Gamma)$, we have to replace the term

$$0 \le t \le 1 \wedge z_{-k+1} = p_{-k+1}(t) \wedge \cdots \wedge z_0 = p_0(t)$$

by

$$\Big(\Big(0 \le t \le t_1 \wedge z_{-k+1} = p_{-k+1|_{[0,t_1]}}(t) \wedge \cdots \wedge z_0 = p_{0|_{[0,t_1]}}(t)\Big) \vee \cdots \vee \\ \Big(t_m \le t \le 1 \wedge z_{-k+1} = p_{-k+1|_{[0,t_1]}}(t) \wedge \cdots \wedge z_0 = p_{0|_{[0,t_1]}}(t)\Big)\Big).$$

For the complex Tracing Problem, we choose the same approach as for the complex Reachability Problem and define the semi-algebraic set $\mathcal{T}_{\mathbb{C}}(\Gamma)$.

Combined with the algorithm of Basu et al. [4, Sect. 16.4] for the computation of the connected components of semi-algebraic sets, the previous observations lead to the following theorem:

Theorem 4.2.2. *The Tracing Problem is decidable for piecewise polynomial paths of the free variables.*

4.3 An Algorithm for the Reachability Problem for Complex GSPs with one Free Variable and a Finite Number of Critical Points

The Reachability Problem for complex GSPs with one free variable and at most finitely many critical points can be reduced to the Tracing Problem. Since this approach gives information about the structure of the Reachability Problem, we discuss this quite restrictive situation separately. We begin with an overview of the algorithm.

Algorithm 1. Reachability Problem with one free variable over $\mathbb{C}$ and a finite number of critical points.

1. We check whether the number of critical values is finite. If this is true, we compute the critical values and a bounding box $\mathcal{D} = [-C, C] \times [-iC, iC]$ containing all of them in its interior.

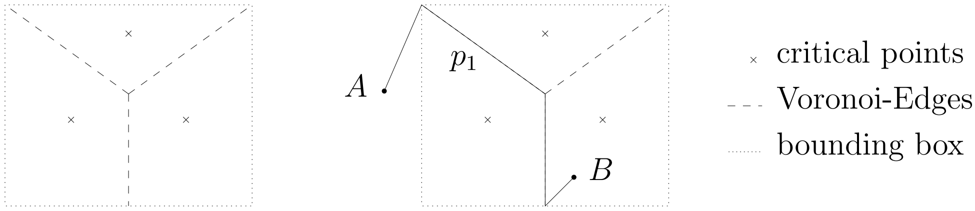


Figure 4.1: The left figure shows the Voronoi diagram of three points in a bounding box; in the right figure, a path p_1 from A to B is drawn that is essentially composed of edges of the Voronoi diagram.

2. We compute a finite set of paths that suffices to decide the Reachability Problem. For this purpose, we compute the Voronoi diagram [5, 58] of the critical values in the bounding box $\mathcal{D}$. The considered paths basically consist of edges of this Voronoi diagram; see Figure 4.1.

3. To decide the Reachability problem, we trace every path computed in step 2.

Details of Algorithm 1. We discuss the details of each step of Algorithm 1. We begin with step 2, since this step is the quintessence of the algorithm.

Ad 2: We construct a finite set of paths with the following property: After tracing along these paths, we can easily decide the Reachability Problem. This construction is based on three concepts:

i) The Voronoi Diagram:

First, we recall the common notion of the Voronoi diagram of a finite set $S = \{c_1, \ldots, c_f\}$ of points in the plane $\mathbb{R}^2 \widehat{=} \mathbb{C}$; see Figure 4.1. In our situation, these points are the critical values of the GSP Γ. The Voronoi region of a point c_i contains all points of $\mathbb{R}^2 \widehat{=} \mathbb{C}$ that are closer to c_i than to all other points of S. The line segments (or rays) separating two Voronoi regions are called Voronoi edges, their endpoints are the Voronoi vertices. A formal definition can be found in [58]. The Voronoi Diagram is a planar graph having the following properties:

(a) The diagram has finitely many edges, and the edges are line segments or rays,
(b) no critical point lies on an edge of the diagram, and
(c) each facet (Voronoi region) contains exactly one critical point.

We place the computed Voronoi diagram into the bounding box $\mathcal{D}$ computed in step 1 of Algorithm 1. At the intersection points of the Voronoi edges with the boundary of $\mathcal{D}$, we introduce new vertices and cut off the (old) Voronoi edges.

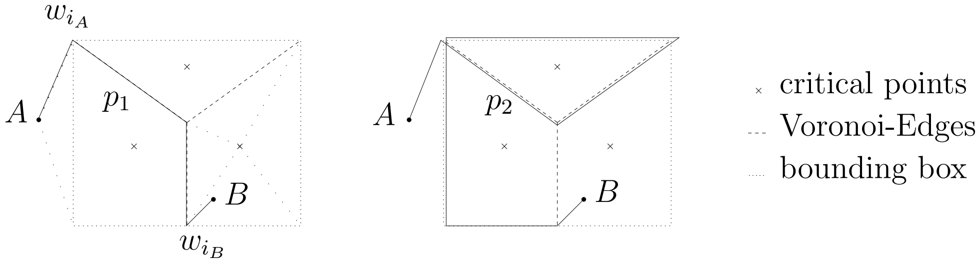


Figure 4.2: Two paths p_1 and p_2 from A to B are drawn that we have to consider in our algorithm. In the left figure the construction of the Voronoi vertices w_{i_A} and w_{i_B} is indicated.

ii) Homotopy Classes of Paths:

The next aim is to define a discrete set of paths of the free variable z of Γ such that we can decide the Reachability Problem by tracing along these paths. We will see that these paths basically consist of Voronoi edges from (i).

Here, the main observation is that "similar" paths (i.e. homotopic paths) of the free variable z lead to "similar" continuous evaluations; i.e., if the continuous evaluations have the same starting instance, then they also have the same final instance. Roughly speaking, two paths with the same endpoints are called homotopic, if one of them can be continuously transformed into the other one, and the transformation leaves the endpoints of the curves fixed; see Definition 8.6.10. This defines an equivalence relation on the set of continuous paths, the equivalence classes are called homotopy classes.

Let $c_1, \ldots, c_f$ be the critical values of the GSP Γ. Lemma 4.3.1 is a consequence of Corollary 8.6.12.

Lemma 4.3.1. *Let Γ be a GSP over $\mathbb{C}$ with one free variable z and n dependent variables $v_1, \ldots, v_n$. Let $S = \{c_1, \ldots, c_f\} \subset \mathbb{C}$ be the set of critical values of Γ. Let $p\colon [0,1] \to \mathbb{C} \setminus S$ and $p'\colon [0,1] \to \mathbb{C} \setminus S$ be homotopic continuous paths of the free variable z of Γ, and let $A = (a_0, \ldots, a_n) \in \mathbb{C}^{n+1}$ be an instance of Γ with $a_0 = p(0) = p'(0)$. Then the continuous evaluations of p and of p' starting at A end at the same instance of Γ.*

Hence, to decide the Reachability Problem, it suffices to take one path $p\colon [0,1] \to \mathbb{C} \setminus \{c_1, \ldots, c_f\}$ per homotopy class of paths $[0,1] \to \mathbb{C} \setminus \{c_1, \ldots, c_f\}$ starting and ending at the corresponding instances. The two paths p_1 and p_2 in Figure 4.2 are not homotopic and represent different homotopy classes.

Since each Voronoi cell contains exactly one critical value, every homotopy class of paths connecting two Voronoi vertices has a representative that is composed by a finite number of Voronoi edges of the Voronoi diagram of the critical points $c_1, \ldots, c_f$; see Figure 4.2 for an example.

If the starting instance $A = (z_A, v_A)$ or the final instance $B = (z_B, v_B)$ does not lie above a Voronoi vertex, we choose a vertex w_{i_A} or w_{i_B} of the diagram and consider the linear path γ_A or γ_B connecting z_A and w_{i_A}, or z_B and w_{i_B}. The vertices w_{i_A} and w_{i_B} can be chosen such that the linear paths γ_A and γ_B do not hit critical values; see Remark 4.3.2 and Figure 4.2.

Remark 4.3.2. The Voronoi vertices w_{i_A} (or w_{i_B}) can be chosen as follows: If z_A lies outside the bounding box, then w_{i_A} is the nearest vertex on the bounding box such that the line segment $|z_A w_{i_A}|$ does not intersect the interior of the bounding box. Otherwise, z_A is contained in a Voronoi cell. We triangulate this cell by adding all edges connecting a vertex of this cell with the critical value defining the cell. Then, w_{i_A} is the nearest Voronoi vertex of the triangle containing z_A. If in the first step of Algorithm 1 the critical values have been approximated, we have to choose w_{i_A} and w_{i_B} more carefully.

As long as there is no critical value on the linear paths γ_A and γ_B, it does not matter which Voronoi vertices are chosen for w_{i_A} and w_{i_B}: For another choice of $w_{i'_A}$ or $w_{i'_B}$, the linear path connecting z_A and $w_{i'_A}$, or z_B and $w_{i'_B}$ is homotopic to a concatenation of γ_A or γ_B and some paths that follow edges of the Voronoi diagram.

iii) A Graph $G = (V, E)$ on a Riemann Surface:

Finally, we find a *finite* subset of paths from (ii) of the free variable z of Γ such that we can decide the Reachability Problem by tracing along these paths. Our construction combines the idea of Voronoi diagrams and covering spaces or Riemann Surfaces of algebraic functions, respectively.

Let $l = 2^s$ be the number of instances lying above a regular point, where s is the number of square root operations of the GSP Γ; see page 35. Let VD be the Voronoi diagram in the bounding box $\mathcal{D}$ of the critical values of the GSP Γ having the Voronoi vertices $w_1, \ldots, w_g \in \mathbb{C} = \mathbb{R}^2$. We define the graph $G = (V, E)$ as follows: The vertex set V consists of l copies $w_{i,1}, \ldots, w_{i,l}$ of each Voronoi vertex w_i. Two vertices $w_{i,j}$ and $w_{i',j'}$ are connected by an edge if the following holds: There is a Voronoi edge in the Voronoi diagram between the Voronoi vertices w_i and $w_{i'}$, and the continuous evaluation of this Voronoi edge starting at the instance $w_{i,j}$ ends at $w_{i',j'}$.

Using the graph G from above, the Reachability Problem can be decided as follows:

(a) We consider a linear path $\gamma_A \colon [0,1] \to \mathbb{C}$ from the starting instance A (or more precisely z_A) to a Voronoi vertex w_{i_A} and another path $\gamma_B \colon [0,1] \to \mathbb{C}$ from the final instance B to a Voronoi vertex w_{i_B}. The corresponding continuous evaluations starting at A or B end at the instances of the GSP that correspond to the vertices $w_{i_A,j}$ or $w_{i_B,j'}$ of our graph G; see Remark 4.3.2.

(b) Check whether the vertices $w_{i_A,j}$ and $w_{i_B,j'}$ are in the same connected component of the graph G.

Ad 3 The tracing process is included into the construction of the graph $G = (V, E)$ defined in (iii) of step 2.

Ad 1 Lemma 3.2.9 implies that the set of critical values of Γ is either finite or every point $z \in \mathbb{C}$ is a critical value of Γ. Thus, Γ has finitely many critical values if and only if the set of critical values is bounded. Recall that the set of critical values is finite if and only if the set of critical points is finite; see Lemma 3.2.5.

The critical values are solutions of polynomial systems of equations in one variable. Unfortunately, there seem to be no algorithms to compute the solutions of such a system, exactly. However, in step 1 of Algorithm 1 , it suffices to approximate the critical values up to the precision $\frac{1}{6}\epsilon$, where ϵ is the minimum distance between two critical values. Then, the Voronoi diagram of the approximated points still has the properties required in step 2 of Algorithm 1.

Let $S = \{c_1, \dots, c_f\} \subset \mathbb{C}$ be the set of critical values of Γ. The distance of two points $c_i \neq c_j \in S$ is at least ϵ. For every point $c_i \in S$ we choose a point c'_i that is contained in the disc with center c_i and radius $\frac{1}{6}\epsilon$. Let $S' := \{c'_1, \dots, c'_f\}$. By construction of the points $c'_i \in S'$, the distance of two points $c'_i \neq c'_j \in S'$ is at least $\frac{2}{3}\epsilon$. We consider the Voronoi diagram $\mathrm{VD}(S')$ of S'. Since the distance of two different points of S' is at least $\frac{2}{3}\epsilon$, the Voronoi edges that are incident to the Voronoi region of the approximated critical value c'_j have at least distance $\frac{1}{3}\epsilon$ to c'_j. Thus, the Voronoi region of a point c'_i contains the disc with center c'_i and radius $\frac{1}{6}\epsilon$. Hence, $c_i \in S$ is the only point of S that is contained in the Voronoi Region of c'_i in $\mathrm{VD}(S')$. We choose the bounding box $\mathcal{D}$ large enough.

We can determine ϵ or at least a lower bound for the minimum distance of two critical values using the decidability of Tarski formulas [4, Sect. 14.1] and binary search; see Example 4.3.3. For the approximation of the critical values, we could use an Interval Newton Method; see Section 7.4 and Appendix A.5.

Example 4.3.3. Again, we use the GSP Γ describing the expression $\sqrt{z^2 - 1}$ from Example 3.1.3. The minimum distance of two critical values is at least ϵ if and only if the following formula is true:

$$\begin{aligned}
&\forall z_0, v_{01}, v_{02}, v_{03},\ z, v_1, v_2, v_3 :\\
&\quad \big(v_{01} = z_0^2 \wedge v_{02} = v_{01} - 1 \wedge v_{03}^2 = v_{02} \wedge v_{02} = 0 \wedge z_0 \neq z \wedge\\
&\quad\ v_1 = z^2 \wedge v_2 = v_1 - 1 \wedge v_3^2 = v_2 \wedge v_2 = 0\big)\\
&\quad \Longrightarrow \ \|z_0 - z\| \geq \epsilon
\end{aligned}$$

Using this Tarski formula, we can determine a lower bound for the minimum distance: We start with an initial guess for ϵ and halve this value until the Tarski formula is fulfilled.

The previous observations prove the following corollary:

Corollary 4.3.4. *Let Γ be a GSP with just one free variable and at most finitely many critical points. Then Algorithm 1 solves the Reachability Problem for every given pair of instances A and B.*

We finish this section with two interesting examples.

Example 4.3.5. We give a GSP Γ, whose configuration space is not path-connected. This GSP describes the expression $\frac{\sqrt{z^2}}{z}$. Recall that the configuration space of Γ is the set of all instances.

$$\begin{array}{rrcl} \Gamma: & z & \leftarrow & \text{FREE} \\ & v_1 & \leftarrow & z \cdot z \\ & v_2 & \leftarrow & \sqrt{v_1} \\ & v_3 & \leftarrow & v_2/z \end{array}$$

If $z \neq 0$ we have $v_3 = \pm 1$, and the projection to the coordinate v_3 of the configuration space consists of the two points $+1$ and -1. The critical value of Γ is $z = 0$. It induces the 2-critical point (0,0). Additionally, $z = 0$ causes a division by zero.

There might be exponentially many critical values of a GSP Γ on a path p of the free variables of Γ as the following example shows.

Example 4.3.6. The expression $\sqrt{z^{2^m} - 1}$ can be described by a GSP Γ of length $m + 2$ in the following way:

$$\begin{array}{rrcll} \Gamma: & z & \leftarrow & \text{FREE} & \\ & v_1 & \leftarrow & z \cdot z & // \quad v_1 = z^2 \\ & v_2 & \leftarrow & v_1 \cdot v_1 & // \quad v_2 = z^4 \\ & & \vdots & & \\ & v_m & \leftarrow & v_{m-1} \cdot v_{m-1} & // \quad v_m = z^{2^m} \\ & v_{m+1} & \leftarrow & v_m - 1 & // \quad v_{m+1} = z^{2^m} - 1 \\ & v_{m+2} & \leftarrow & \sqrt{v_{m+1}} & // \quad v_{m+2} = \sqrt{z^{2^m} - 1} \end{array}$$

The critical values are the 2^m roots of unity given by the radicand $z^{2^m} - 1$. All critical values lie on the path p: $[0, \sqrt{\frac{1}{2}}] \to \mathbb{C}$, $t \mapsto (+\sqrt{1 - t^2} + it)^8$ whose graph is the unit circle.

We can extend Γ by $m - 1$ iterated square root operations, such that the last variable v_{2m+1} describes the expression $\sqrt[2^m]{z^{2^m} - 1}$. Then, we have 2^m instances that lie above a regular value of Γ.

Chapter 5

The Tracing Problem and Continuation Methods

We can solve the Tracing Problem numerically using continuation methods. Continuation Methods are a well-established and useful field of modern mathematics; in [2], Allgower and Georg give an overview. These methods are used to solve systems of nonlinear equations. We describe a typical application. Let $F\colon \mathbb{R}^N \to \mathbb{R}^N$ be a smooth function, and we want to solve the equation $F(x) = 0$. Let us assume that we know the solutions of another, possibly simpler system $G(x) = 0$ where $G\colon \mathbb{R}^N \to \mathbb{R}^N$ is a smooth function as well. We consider the function $H\colon \mathbb{R}^N \times \mathbb{R} \to \mathbb{R}^N$, $(x, \lambda) \mapsto \lambda G(x) + (1 - \lambda)F(x)$, which is called convex homotopy [2, p. 2]. To solve the equation $F(x) = 0$, we have to solve $G(x) = 0$ and to trace the solutions from $\lambda = 1$ to $\lambda = 0$. Tracing these *implicitly defined* solution curves numerically is the aim of numerical continuation methods. A geometric application for homotopy methods is given in [48].

Homotopy methods have successfully been applied to solve *polynomial systems of equations* and have lead to the new area *Numerical Algebraic Geometry* [59]. Here, the homotopies are chosen such that no singularities occur along the solution curves and every isolated solution of multiplicity m is reached by exactly m paths. As already mentioned, these paths are traced using numerical methods. In [19], a robust algorithm to trace curves that are implicitly defined by algebraic equations is given. The algorithm is based on interval analysis.

To solve the Tracing Problem, we consider polynomial systems, as well. The underlying Geometric Straight-Line Program Γ together with the given paths of the free variables of Γ lead to a polynomial function $H\colon \mathbb{R}^{N+1} \to \mathbb{R}^N$. We will see in Section 5.1 that the continuous evaluation is an implicit curve defined by the function H. In Section 5.2, we develop a numerical solution that captures the particular structure of the Tracing Problem. In the Tracing Problem, the wanted continuous evaluation is a parametrized curve in contrast to the situation considered in [19].

5.1 Continuous Evaluations as Implicit Curves

We discuss how a continuous evaluation along paths $p_{-k+1}(t), \ldots, p_0(t)$ of the k free variables $z_{-k+1}, \ldots, z_0$ of a GSP Γ over a field $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$ is given as an implicit curve of a function $H\colon \mathbb{K}^{k+n} \times [0,1] \to \mathbb{K}^{k+n}$ where n is the number of dependent variables. Hence we assume that the paths $p_l(t)$ of the free variables z_l are continuously differentiable. To define the function H, we assign to every dependent variable $a = v_j$ of Γ a multivariate polynomial $\mathcal{P}_a = \mathcal{P}_{v_j}$:

Dependent Variable a	Polynomial $\mathcal{P}_a$
$a \leftarrow b + c$	$a - (b + c)$
$a \leftarrow b - c$	$a - (b - c)$
$a \leftarrow b \cdot c$	$a - b \cdot c$
$a \leftarrow b/c$	$a \cdot c - b$
$a \leftarrow \sqrt{b}$	$a^2 - b$

The polynomials $\mathcal{P}_a$ are polynomials in at most three variables; we have $\mathcal{P}_a = 0$ if and only if the relation that defines the dependent variable a is fulfilled. Using the polynomials $\mathcal{P}_a = \mathcal{P}_{v_j}$, we define the functions

$$\begin{aligned} F = F(\Gamma)\colon \mathbb{K}^{k+n} &\to \mathbb{K}^{k+n} \\ (z_{-k+1}, \ldots, z_0, v_1, \ldots, v_n) &\mapsto (z_{-k+1}, \ldots, z_0, \mathcal{P}_{v_1}, \ldots, \mathcal{P}_{v_n}) \end{aligned}$$

and $H = H(\Gamma, p_{-k+1}(t), \ldots, p_0(t))\colon \mathbb{K}^{k+n} \times [0,1] \to \mathbb{K}^{k+n}$,

$$\begin{aligned} (z_{-k+1}, \ldots, z_0, v_1, \ldots, v_n, t) \mapsto & (z_{-k+1} - p_{-k+1}(t), \ldots, z_0 - p_0(t), \mathcal{P}_{v_1}, \ldots, \mathcal{P}_{v_n}) \\ = & F(z_{-k+1}, \ldots, z_0, v_1, \ldots, v_n) \\ & -(p_{-k+1}(t), \ldots, p_0(t), 0 \ldots, 0), \end{aligned}$$

where $p_l(t)$ is the path of the free variable z_l; $l = -k+1, \ldots, 0$. We observe that a point $(a_{-k+1}, \ldots, a_0, a_1, \ldots, a_n) \in \mathbb{K}^{k+n}$ fulfills the relations of Γ if and only if

$$F(a_{-k+1}, \ldots, a_0, a_1, \ldots, a_n) - (a_{-k+1}, \ldots, a_0, 0, \ldots, 0) = (0, \ldots, 0).$$

Similarly, if a tuple of paths $(v_1(t), \ldots, v_n(t))$ is a continuous evaluation along the given paths $p_{-k+1}(t), \ldots, p_0(t)$ of the free variables $z_{-k+1}, \ldots, z_0$, then

$$H(p_{-k+1}(t), \ldots, p_0(t), v_1(t), \ldots, v_n(t), t) = (0, \ldots, 0).$$

For formal reasons, we identify $\mathbb{C} \widehat{=} \mathbb{R}^2$. If Γ is a GSP over $\mathbb{K} = \mathbb{C}$, then we split all variables in their real and imaginary parts and consider the space $\mathbb{R}^{2(k+n)} \widehat{=} \mathbb{C}^{k+n}$.

Lemma 5.1.1. *Let Γ be a GSP over $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$ with k free variables and n dependent ones.*

1. *A point $(a_{-k+1}, \ldots, a_n) \in \mathbb{K}^{k+n}$ is an instance of the GSP Γ if and only if $F(a_{-k+1}, \ldots, a_n) - (a_{-k+1}, \ldots, a_0, 0, \ldots, 0) = 0$ and $\det F'(a_{-k+1}, \ldots, a_n) \neq 0$ hold.*

2. *If a point $(a_{-k+1}, \ldots, a_n) \in \mathbb{K}^{k+n}$ is a critical point of the GSP Γ, then $\det F'(a_{-k+1}, \ldots, a_n) = 0$ holds. A point $(a_{-k+1}, \ldots, a_n) \in \mathbb{K}^{k+n}$ is an m-critical point of the GSP Γ if and only if $\det F(\Gamma^{(m)})'(a_{-k+1}, \ldots, a_m) = 0$, $\det F(\Gamma^{(m-1)})'(a_{-k+1}, \ldots, a_{m-1}) \neq 0$ and $F(\Gamma^{(m-1)})(a_{-k+1}, \ldots, a_{m-1}) - (a_{-k+1}, \ldots, a_0, 0, \ldots, 0) = 0$ hold where $\Gamma^{(m)}$ and $\Gamma^{(m-1)}$ are the m-head and the $(m-1)$-head of Γ.*

3. *Let $p_{-k+1}(t), \ldots, p_0(t)$ be continuously differentiable paths of the free variables $z_{-k+1}, \ldots, z_0$ of the GSP Γ, and let $A = (a_{-k+1} = p_{-k+1}(0), \ldots, a_0 = p_0(0), a_1, \ldots, a_n)$ be an instance of Γ. If $\det F'(A) = \det \frac{\partial H}{\partial A} \neq 0$, then the corresponding continuous evaluation $(v_1(t), \ldots, v_n(t))$ exists locally, i.e., there is an $\varepsilon > 0$ such that the paths $v_j(t)$ are defined for $t \in (-\epsilon, \epsilon)$. Moreover, the paths $v_j(t)$ are continuously differentiable over $(-\epsilon, \epsilon)$; $j = 1, \ldots, n$.*

Proof. To prove Lemma 5.1.1, we consider the Jacobian $F'(a_{-k+1}, \ldots, a_n)$ of F in the point $(a_{-k+1}, \ldots, a_n)$. Since a GSP only uses variables defined before, this $(k+n) \times (k+n)$-matrix is a lower triangular matrix. The rows that belong to a dependent variable have at most three nonzero entries. The entries of the diagonal are 1 for the free variables z_l, the entries of the diagonal for the dependent variables $a = v_j$ are the partial derivatives $\frac{\partial \mathcal{P}_{v_j}}{\partial v_j} = \frac{\mathcal{P}_a}{\partial a}$, they are listed in the following table:

Dependent Variable a	Polynomial $\mathcal{P}_a$	$\frac{\partial \mathcal{P}_a}{\partial a}$
$a \leftarrow b + c$	$a - (b + c)$	1
$a \leftarrow b - c$	$a - (b - c)$	1
$a \leftarrow b \cdot c$	$a - b \cdot c$	1
$a \leftarrow b/c$	$a \cdot c - b$	c
$a \leftarrow \sqrt{b}$	$a^2 - b$	$2a$

We have

$$\det F'(a_{-k+1}, \ldots, a_0, a_1, \ldots, a_n) = \det \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ * & \cdots & * & \frac{\mathcal{P}_{v_1}}{\partial v_1} & & \\ \vdots & & & \ddots & \ddots & \\ * & \cdots & \cdots & \cdots & * & \frac{\mathcal{P}_{v_n}}{\partial v_n} \end{pmatrix}$$

$$= \frac{\mathcal{P}_{v_1}}{\partial v_1} \cdot \ldots \cdot \frac{\mathcal{P}_{v_n}}{\partial v_n}.$$

Hence, $\det F'(a_{-k+1}, \ldots, a_n) = 0$ holds if and only if a root variable or the divisor variable of a division variable is zero. By construction of F, a tuple $(a_{-k+1}, \ldots, a_n)$ fulfills all relations of Γ if and only if $F(a_{-k+1}, \ldots, a_n) - (a_{-k+1}, \ldots, a_0, 0, \ldots, 0) = 0$ holds. Since a tuple $(a_{-k+1}, \ldots, a_n)$ is an instance of Γ if and only if all relations given by Γ are fulfilled and no division by zero and no root of zero occur, part 1 is proven.

Part 2 can be shown using the same arguments. Part 3 is a consequence of the implicit function theorem. Since A is an instance at $t = 0$, we have $H(A, 0) = 0$. □

To apply the implicit function theorem properly, we have to assume that the paths $p_l(t)$ of the free variables z_l are continuously differentiable in a neighborhood $U_{[0,1]}$ of the time interval $[0, 1]$.

We summarize the consequences of the implicit function theorem: There is an open interval $J \subset U_{[0,1]}$ with $0 \in J$ and a continuously differentiable curve $\alpha \colon J \to \mathbb{K}^{k+n}$ with the following properties:

1. $\alpha(0) = A$, i.e., $\alpha(0)$ is the given starting instance A.
2. $H(\alpha(t), t) = 0$ holds for all $t \in J$, hence $\alpha(t)$ fulfills all relations of the GSP Γ.
3. $\det F'(\alpha(t)) = \det\left(\frac{\partial H}{\partial A}(\alpha(t), t)\right) \neq 0$ holds for all $t \in J$; this implies that the $(k+n) \times (k+n+1)$-matrix $H'(\alpha(t), t)$ has full rank $k+n$ for all $t \in J$.

Combining Properties 1-3 shows that the curve $\alpha(t)$ is the wanted continuous evaluation $(p_{-k+1}(t), \ldots, p_0(t), v_1(t), \ldots, v_n(t))$. Deriving the equation $H(\alpha(t), t) = 0$ from 2 leads to

$$H'(\alpha(t), t) \cdot (\dot{\alpha}(t), 1)^t = 0,$$

and $(\dot{\alpha}(t), 1)$ is a nonzero vector of the kernel of $H'(\alpha(t), t)$. Since the rank of $H'(\alpha(t), t)$ is $k + n$, the vector $(\dot{\alpha}(t), 1)$ spans this kernel. Using these observations, we can describe the curve $\alpha(t)$ and hence the continuous evaluation $(v_1(t), \ldots, v_n(t))$ by the following initial value problem: Let $(u(t), 1) = (u, 1) \in \ker H'(\alpha(t), t)$, then α is described by

- $\dot{\alpha} = u$, $\dot{t} = 1$;
- $\alpha(0) = A$, $t(0) = 0$.

It remains to show that $u(t)$ is a continuous function. To see this, we consider the tangent vector $\tau(M) \in \mathbb{R}^{k+n+1}$ of an $(k+n) \times (k+n+1)$-matrix M with $\operatorname{rank}(M) = k+n$; see [2, Def. 2.1.7, p. 9]. The vector $\tau(M)$ is the unique vector satisfying the conditions $M\tau = 0$, $\|\tau\| = 1$, and $\det \begin{pmatrix} M \\ \tau^t \end{pmatrix} > 0$. By [2, Lem. 2.1.8], the function $M \mapsto \tau(M)$ is smooth. Thus, the function $u(t)$ can be described by $(u(t), 1) = \tau(H'(\alpha(t), t))/\tau_{k+n+1}$ where τ_{k+n+1} is the last entry of the vector $\tau(H'(\alpha(t), t))$. Since $\ker(H'(\alpha(t), t)$ is spanned by $(\dot{\alpha}, 1)$, we have $\tau_{k+n+1} \neq 0$, and $u(t)$ is a continuous function.

The previous observation shows that the Tracing Problem can be interpreted as an initial value problem that has the corresponding continuous evaluation as solution curve. Additionally, this fact implies the existence and uniqueness of continuous evaluations. The approach via initial value problems illuminates the Tracing Problem from the viewpoint of Dynamical Systems. Two interesting structural properties are given in [2, Lem. 2.1.13 and Thm. 2.1.14].

5.2 About Numerical Solutions for the Tracing Problem

Various numerical methods have been developed to trace implicitly defined curves [2, 59, 19, 15] and to solve initial value problems [15] that could be used to solve the Tracing Problem from Dynamic Geometry. We adapt a generic Predictor-Corrector method to the Tracing Problem. The resulting method is an increment-and-fix path following method since the continuation parameter t remains fixed in the corrector step [59, p. 10]. Allgower and Georg give a well-founded introduction to Predictor-Corrector methods in [2]. They investigate a general situation where $H\colon \mathbb{R}^{N+1} \to \mathbb{R}^N$ is a smooth function and the starting point u_0 is a regular value [2, Sect. 2.2]. The aim is to trace the curve $\beta\colon \mathbb{R} \supset J \to \mathbb{R}^{N+1}$ with $\beta(0) = u_0$ and $H(\beta(s)) = 0$ for all $s \in J$ as long as we have $\operatorname{rank}(H'(\beta(s)) = N$ and $\beta'(s) \neq 0$. In their general context, the function H does not give rise to a parametrization of the solution curve β; Allgower and Georg choose the parametrization with respect to the arclength. In our situation explained in Section 5.1, the paths $p_l(t)$ of the free variables z_l are part of the definition of the function $H\colon \mathbb{R}^{k+n} \times [0,1] \to \mathbb{R}^{k+n}$. The paths $p_l(t)$ are parametrized curves with respect to the time t, and their parametrization induces a parametrization of the solution curve $\beta(t) = (\alpha(t), t)$ as seen in Section 5.1. Recall that the curve $\alpha(t)$ is the desired continuous evaluation $(p_{-k+1}(t), \ldots, p_0(t), v_1(t), \ldots, v_n(t))$. To harmonize the notations, we set $N := k+n$.

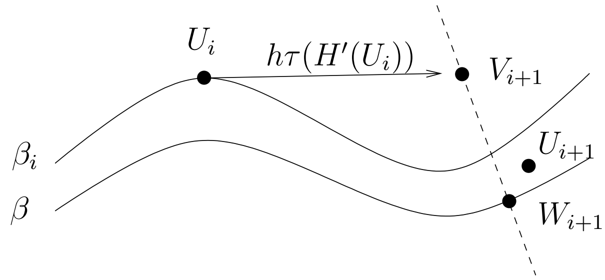


Figure 5.1: The predictor point V_{i+1} and the corrector point U_{i+1} are shown.

Let $A_0 := A = (a_{-k+1} = p_{-k+1}(0), \dots, a_0 = p_0(0), a_1 \dots, a_n)$ be the starting instance that is specified in the Tracing Problem. We set $U_0 := (A_0, 0) \in \mathbb{R}^{N+1}$. The aim of Predictor-Corrector methods is to determine iteratively a sequence of points $U_1, U_2, U_3, \dots \in \mathbb{R}^{N+1}$ alongside the curve β that fulfill a certain tolerance criterion like $H(U_i) \le \epsilon$ for some $\epsilon > 0$. If ϵ is chosen small enough, we can expect that the points U_i are close to the solution curve β. We assume that the points U_i are regular points of the function H. For every U_i, we have a unique maximal solution curve $\beta_i \colon \mathbb{R} \supset J \to \mathbb{R}^{k+n+1} = \mathbb{R}^{N+1}$ of the initial value problem

$$\begin{aligned} \dot{\beta} &= \tau(H'(\beta)); \\ \beta(0) &= U_i. \end{aligned}$$

To obtain a new point U_{i+1}, we first make a *predictor step*; see Figure 5.1. Like Allgower and Georg in [2], we choose an Euler predictor

$$V_{i+1} := U_i + h\tau(H'(U_i)),$$

where $h > 0$ represents a step length and $\tau(H'(U_i))$ is the tangent vector [2, Def. 2.1.7] defined on page 61. The vector $\tau(H'(U_i))$ has unit length and is tangent to the curve β_i in the point U_i.

The second step is called *corrector step*. Starting with the predictor point V_{i+1}, the closest point W_{i+1} on the solution curve β is approximated. This process leads to the point U_{i+1}. Since we have $H(\beta) = 0$, we could use Newton-like methods for this purpose, and we expect a rapid convergence [2, Sect. 2.2]; see Figure 5.1. The Predictor-Corrector continuation method for approximating β consists of repeatedly performing predictor and corrector steps. As pointed out by Allgower and Georg in [2], to construct an efficient and robust Predictor-Corrector method, we have to consider the following problems:

1. Develop an effective step length adaptation;

2. Select an efficient implementation of the corrector step;

3. Include higher order predictors efficiently;

4. Handle or approximate special points on the solution curve like turning points or bifurcation points.

We adapt the general predictor and corrector steps to the Tracing Problem from Dynamic Geometry and to the function H from page 58. As indicated on page 61 and on page 60, we interpret the last coordinate of a computed point $U_i \in \mathbb{R}^{N+1}$ as time and get $U_i = (A_i, t_i)$ with $A_i \in \mathbb{R}^N = \mathbb{R}^{k+n}$. If a computed point U_i lies *on* the solution curve β, then we have $H(U_i) = 0$. By construction of the function H, we have $H(A_i, t_i) = H(U_i) = 0$ if and only if $A_i = (a_{i,-k+1}, \ldots, a_{i,0}, a_{i,1}, \ldots, a_{i,n})$ is an instance of the underlying GSP Γ with $a_{i,-k+1} = p_{-k+1}(t_i), \ldots, a_{i,0} = p_0(t_i)$. Since this observation holds for all points in $\mathbb{R}^{N+1} = \mathbb{R}^{k+n} \times \mathbb{R}$, the last coordinate of the solution curve β represents the time and we have shown $\beta(t) = (\alpha(t), t)$. Additionally, we have $\alpha(t) = (p_{-k+1}(t), \ldots, p_0(t), v_1(t), \ldots, v_n(t))$, where $(v_1(t), \ldots, v_n(t))$ is the wanted continuous evaluation.

If $H(U_i) = 0$, then the vector $\tau(H'(U_i))$ used in the predictor step is the tangent to the solution curve $\beta(t) = (\alpha(t), t)$ in the point U_i with length 1 pointing in forward direction. To keep track of the parametrization induced by the paths $p_l(t)$ of the free variables of the underlying GSP Γ, we choose the derivative $\dot\beta(t) = (\dot\alpha(t), 1)$ of $\beta(t) = (\alpha(t), t)$ at time t_i as tangent vector in the predictor step. To determine $\dot\alpha(t_i)$, we recall $\alpha(t) = (p_{-k+1}(t), \ldots, p_0(t), v_1(t), \ldots, v_n(t))$, where $(v_1(t), \ldots, v_n(t))$ is the wanted continuous evaluation. Thus, to determine $\dot\alpha(t_i)$ we have to compute the derivatives $\dot p_l(t_i)$ of the paths p_l and the derivative $(\dot v_1(t_i), \ldots, \dot v_n(t_i))$ of the continuous evaluation $(v_1(t), \ldots, v_n(t))$ at time t_i. We assume that the derivatives $\dot p_l(t)$ of the paths p_l of the free variables z_l are known. Since A_i is an instance at time t_i, we can efficiently determine $(\dot v_1(t_i), \ldots, \dot v_n(t_i))$ using the derivative GSP $\dot\Gamma$ defined in Section 3.5.

We consider the resulting predictor point $V_{i+1} = (A_i, t_i) + h\dot\beta(t_i) =: (\tilde A_{i+1}, t_{i+1})$ with $t_{i+1} := t_i + h$. If the step length h is chosen sufficiently small, we can expect that the predictor point $V_{i+1} = (\tilde A_{i+1}, t_{i+1})$ is sufficiently close to the point $\beta(t_{i+1})$ on the solution curve β. This implies that the point $\tilde A_{i+1}$ is sufficiently close to the instance $A_{i+1} = (p_{-k+1}(t_{i+1}), \ldots, p_0(t_{i+1}), v_1(t_{i+1}), \ldots, v_n(t_{i+1})) = \alpha(t_{i+1})$ lying on the wanted continuous evaluation $(p_{-k+1}(t), \ldots, p_0(t), v_1(t), \ldots, v_n(t))$.

Now, we treat the corrector step. A first possibility is to approximate the instance A_{i+1} using the Newton method for the function F defined on page 58 with the point $\tilde A_{i+1}$ as initial guess. Using this approach, we can only expect that the computed points lie close to the solution curve $\alpha(t)$. The Newton method uses the inverse matrix of the derivative F' of F. Similar to the derivative GSP $\dot\Gamma$, we could compute F' symbolically. If Γ has only one root operation, and this root operation defines the last dependent variable v_n of Γ, then the Newton method is in fact the Babylonian method for computing square roots. Since the corrector point U_{i+1} might not lie on the solution curve $\beta(t) = (\alpha(t), t)$, we cannot use the efficient method via the derivative GSP $\dot{\Gamma}$ to determine the tangent vector $\tau(H'(U_{i+1}))$ in the next predictor step.

In the second approach, we present a *discrete* corrector step that uses the special structure of GSPs. After the predictor step, we consider the time $t_{i+1} = t_i + h$, where h is the step length. The positions of the free variables z_l of the GSP Γ at time t_{i+1} are given by their paths p_l; we have $z_l = p_l(t_{i+1})$ for $l = -k+1, \dots, 0$. Following the operations of Γ, we can determine all instances at the position $(p_{-k+1}(t_{i+1}), \dots, p_0(t_{i+1}))$. For the corrector point $U_{i+1} = (A_{i+1}, t_{i+1})$, we choose the instance A_{i+1}, which is "closest" to the point $\tilde{A}_{i+1}$ of the predictor point $V_{i+1} = (\tilde{A}_{i+1}, t_{i+1})$. We determine the instance $A_{i+1} = (a_{i+1,-k+1} = p_{-k+1}(t_{i+1}), \dots, a_{i+1,0} = p_0(t_{i+1}), a_{i+1,1}, \dots, a_{i+1,n})$ closest to $\tilde{A}_{i+1} = (\tilde{a}_{i+1,-k+1}, \dots, \tilde{a}_{i+1,0}, \tilde{a}_{i+1,1}, \dots, \tilde{a}_{i+1,n})$ coordinatewise starting with the dependent variable v_1: The coordinate $a_{i+1,1}$ is the output of v_1 that is closest to $\tilde{a}_{i+1,1}$. Using the position $(a_{i+1,-k+1} = p_{-k+1}(t_{i+1}), \dots, a_{i+1,0} = p_0(t_{i+1}))$ *and* the coordinate $a_{i+1,1}$, we can compute the output $a_{i+1,2}$ of the dependent variable v_2 that is closest to $\tilde{a}_{i+1,2}$. We give the general formula of this iterative procedure: Assume the coordinates $\tilde{a}_{i+1,1}, \dots, \tilde{a}_{i+1,j-1}$ of the dependent variables $v_1, \dots, v_{j-1}$ are already determined. Using the position $(a_{i+1,-k+1} = p_{-k+1}(t_{i+1}), \dots, a_{i+1,0} = p_0(t_{i+1}))$ *and* the coordinates $a_{i+1,1}, \dots, a_{i+1,j-1}$, we can compute the output $a_{i+1,j}$ of the dependent variable v_j that is closest to $\tilde{a}_{i+1,j}$. This algorithm runs in $O(n)$ time in the real RAM-model, where n is the number of dependent variables of Γ, if the position $(a_{i+1,-k+1} = p_{-k+1}(t_{i+1}), \dots, a_{i+1,0} = p_0(t_{i+1}))$ is known. If we do not assume exact computation, we have to take rounding errors into account even in the predictor step.

The main disadvantage of the presented Predictor-Corrector methods is that there is no guarantee for the correctness of the computed solutions. If the chosen step length h in the predictor step is too large, the corrector point might jump to a wrong path of the dependent variables. To overcome this problem, we developed the algorithms presented in Chapter 6. These algorithms are based on a step length adaptation that guarantees the correctness of the solution curve. Critical points are surrounded by a detour; see Chapter 7.

Experimental Results We give some experimental results for the Predictor-Corrector method. We used the discrete corrector step presented in the second approach. Unfortunately, we did not use the derivative GSP $\dot{\Gamma}$ to determine the tangent vector. Instead, we computed the kernel of the matrix $H'(U_i)$. For the implementation, we used the computer algebra software *Maple* [30].

Example 5.2.1. We consider the GSP

$$\begin{array}{rrcl} \Gamma_1\colon & z & \leftarrow & \text{FREE} \\ & v_1 & \leftarrow & \sqrt{z}, \end{array}$$

the starting instance $A = (-1 + 0.1i, +\sqrt{-1 + 0.1i} \approx 0.05 + i)$, and the path $p_1\colon [0,1] \to \mathbb{C}$, $t \mapsto 2t - 1 + 0.1i$. The wanted final instance is $B = (1 + 0.1i, +\sqrt{1 + 0.1i} \approx 1 + 0.05i)$. The step length h and the resulting predictor and corrector points are shown in Figures 5.3(a)-5.3(c).

Example 5.2.2. The GSP Γ_2 describes the algebraic expression $\sqrt{(1 - z^2)^3}$:

$$\begin{array}{rrcl} \Gamma_2\colon & z & \leftarrow & \text{FREE} \\ & v_1 & \leftarrow & z \cdot z \\ & v_2 & \leftarrow & 1 - v_1 \\ & v_3 & \leftarrow & v_2 \cdot v_2 \\ & v_4 & \leftarrow & v_3 \cdot v_2 \\ & v_5 & \leftarrow & \sqrt{v_4} \end{array}$$

The starting instance $A = (0, 0, 1, 1, 1, 1)$ and the path $p_2\colon [0,1] \to \mathbb{C}$, $t \mapsto 1 - \cos(t\pi) - \sin(t\pi)$ are given. The resulting final instance in the Tracing Problem is $B = (2, 4, -3, 9, -27, -\sqrt{-27} \approx -5.2i)$. The step length h and the resulting predictor and corrector points are shown in Figures 5.3(d)-5.3(f).

Example 5.2.3. We consider the GSP Γ_3:

$$\begin{array}{rrcll} \Gamma_3\colon & z & \leftarrow & \text{FREE} & \\ & v_1 & \leftarrow & z \cdot z & \\ & v_2 & \leftarrow & v_1 - 1 & //v_2 = z^2 - 1 \\ & v_3 & \leftarrow & v_1 - 0.01 & //v_3 = z^2 - 0.01 \\ & v_4 & \leftarrow & v_2 \cdot v_3 & //v_4 = (z^2 - 1)(z^2 - 0.01) \\ & v_5 & \leftarrow & 10iv_4 & //v_5 = 10i(z^2 - 1)(z^2 - 0.01) \\ & v_6 & \leftarrow & z + v_5 & //v_6 = z + 10i(z^2 - 1)(z^2 - 0.01) \\ & v_7 & \leftarrow & \sqrt{v_6} & \end{array}$$

If we assume that z takes values in $\mathbb{R}$, then the dependent variables v_j describe functions $v_j(z)\colon \mathbb{R} \to \mathbb{C}$. The image of the function $v_6(z)$ is the set $\{(z, i\,10(z^2 - 1)(z^2 - 0.01)) \in \mathbb{C} \mid z \in \mathbb{R}\}$, which can be interpreted as the graph of the function $f\colon \mathbb{R} \to \mathbb{R}$, $z \mapsto 10(z^2 - 1)(z^2 - 0.01)$. The graph of f is shown in Figure 5.2.

For the Tracing Problem, we are given the path $p_3\colon [0,1] \to \mathbb{C}$, $t \mapsto 2.2t - 1.1$, of the free variable z and the starting instance

$$A = (-1.1, 1.21, 0.21, 1.2, 0.252, 2.52i, -1.1 + 2.52i, +\sqrt{-1.1 + 2.52i} \approx 0.9 + 1.4i).$$

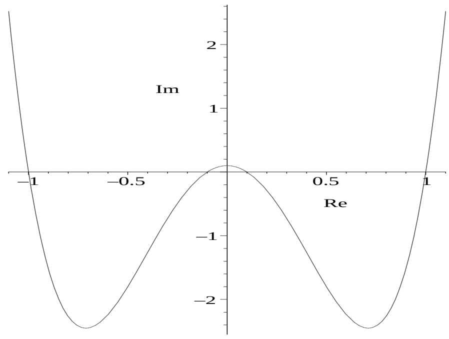


Figure 5.2: The graph of the function f is shown.

The resulting final instance is

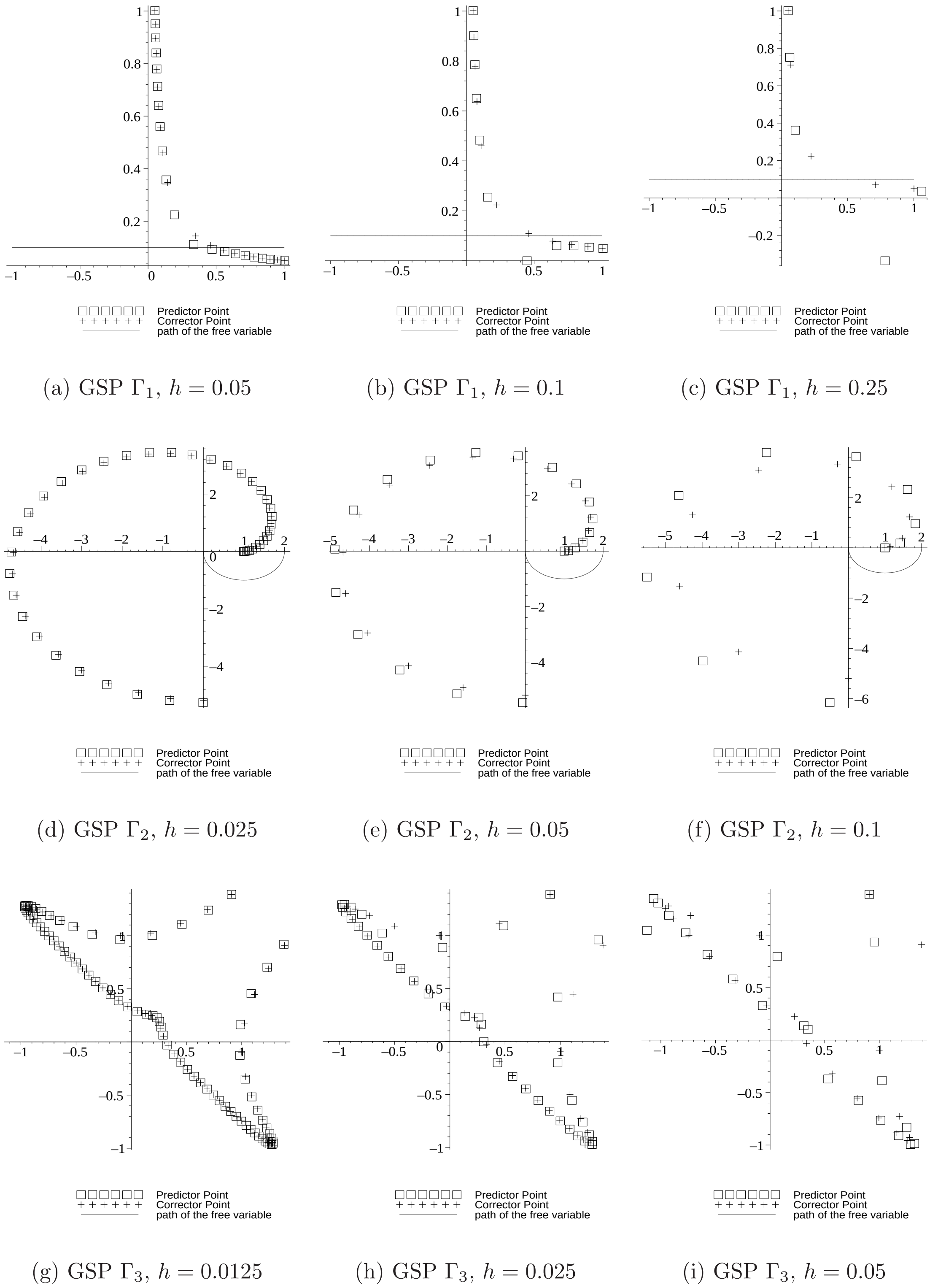
$$B = (1.1, 1.21, 0.21, 1.2, 0.252, 2.52i, 1.1 + 2.52i, +\sqrt{1.1 + 2.52i} \approx 1.4 + 0.9i).$$

The step length h and the resulting predictor and corrector points are shown in Figures 5.3(g)-5.3(i).

The experiments show that, for a further development of Predictor Corrector methods for the Tracing Problem from Dynamic Geometry, we could consider the curvature or the "speed" of the solution curve to work out an improved step length adaptation.

Related Methods Kearfott and Xing [36] give a continuation method that is based on interval arithmetic and guarantees the correctness of the solution. They use a so-called Gauss-Seidel-Sweep and consider boxes containing the solution curve. Their aim is to shrink the boxes in order to achieve that a box only contains one solution. This approach seems not to be efficient for solving the Tracing Problem from Dynamic Geometry: Here, we first determine a box that contains at least one solution curve. Afterwards, we decrease the step length and separate the solution curve from the other solution candidates; see Chapter 6.

Blum, Cucker, Shub and Smale [7, Sect. 14.3] investigate the complexity of homotopy continuation methods. They consider functions over $\mathbb{C}$ and combine the predictor and the corrector step in one single step. They estimate the number of Newton steps needed to follow the correct solution curve. As in [2], the solution curve is parametrized by arc length. The discussed estimates are based on a condition number, which might be difficult to determine. However, we do not know how to use this method to solve the Tracing Problem from Dynamic Geometry.



(a) GSP Γ_1, $h = 0.05$ (b) GSP Γ_1, $h = 0.1$ (c) GSP Γ_1, $h = 0.25$

(d) GSP Γ_2, $h = 0.025$ (e) GSP Γ_2, $h = 0.05$ (f) GSP Γ_2, $h = 0.1$

(g) GSP Γ_3, $h = 0.0125$ (h) GSP Γ_3, $h = 0.025$ (i) GSP Γ_3, $h = 0.05$

Figure 5.3: The predictor points (boxes) and corrector points (crosses) for Examples 5.2.1, 5.2.2 and 5.2.3 are shown for different step lengths. The values for the last dependent variable of the GSPs Γ_1, Γ_2 and Γ_3 are drawn in the complex plane $\mathbb{C}$.

Chapter 6

Reliable Algorithms for the Tracing Problem

We describe a *reliable* algorithm for the Tracing Problem for GSPs over $\mathbb{R}$ or $\mathbb{C}$ that is based on interval arithmetic [14]. We assume that there are no critical points on the paths $p_{-k+1}, \ldots, p_0$ of the free variables $z_{-k+1}, \ldots, z_0$, the treatment of critical points is discussed in Chapter 7. Furthermore, we restrict to linear paths $p_{-k+1}, \ldots, p_0$ which simplifies step 2 of Algorithm 2. The used interval arithmetic has to fulfill the inclusion monotonicity property.

We give the main ideas in Section 6.1. Algorithm 3 from Section 6.2 traces the continuous evaluation $v_1(t), \ldots, v_n(t)$ stepwise. For each step, a proper step length h_0 is determined *in advance*. To achieve this aim, we assign to each variable of Γ an interval that contains the range of the corresponding coordinate-path of the continuous evaluation. Algorithm 2 performs a single tracing step, Algorithm 3 uses Algorithm 2 and traces the whole continuous evaluation as long as there occurs no critical point. Algorithm 4 from Section 6.3 extends Algorithm 2 by an improved step length adaptation. It uses the derivative GSP $\dot{\Gamma}$.

In Section 6.4, we discuss the problem of overestimation due to the usage of interval arithmetic. In Section 6.5, we show that small modifications to the given algorithm combined with the treatment of critical points from Chapter 7 lead to a robust method for solving the Tracing Problem. Due to the robustness, this reliable method could be a useful tool in automated theorem proving and for path-tracking problems in homotopy methods for solving algebraic systems of equations. The extension of the algorithms to GSPs with cubic and higher roots is addressed in Section 6.6. At the end of this chapter, we deliberate about using affine arithmetic instead of interval arithmetic.

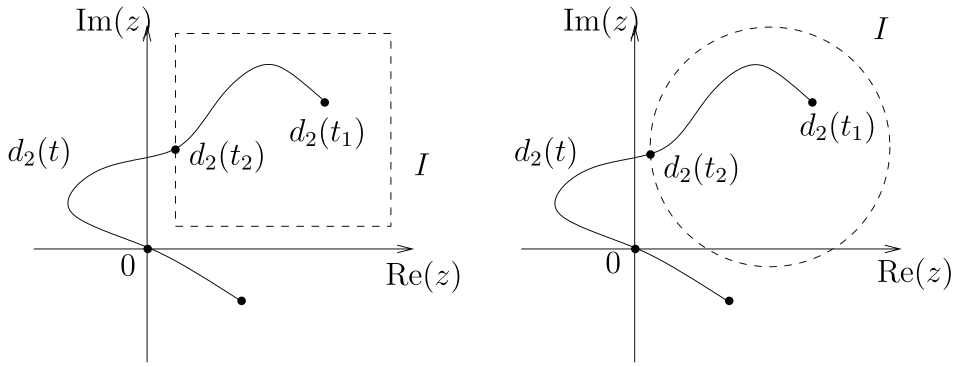


Figure 6.1: The path $d_2(t)$ of the divisor variable d_2, the point $d_2(t_2)$, and the interval I are shown for rectangular and for circular arithmetic.

6.1 Main Idea

We describe the main idea of our reliable algorithms for the Tracing Problem. Recall that we have to deal with the ambiguity due to the root function and with critical points, which are caused by a division by zero or a root of zero. Thus, the operations *division* and *root* need a special treatment. We explain how to determine the step length of a single tracing step.

Division: Let $v_m = d_1/d_2$, $d_1, d_2 \in \{z_{-k+1}, \ldots, z_0, v_1, \ldots, v_{m-1}\}$, be a division operation, and let $d_1(t), d_2(t) \colon [0,1] \to \mathbb{K}$ be the corresponding paths of the variables d_1 and d_2 in the continuous evaluation. In order to simplify the description, we assume that the paths $d_1(t)$ and $d_2(t)$ do not contain critical points, i.e., there is no $t \in [0,1]$ for that $\sqrt{0}$ or $\_/0$ occurs during the computation of $d_1(t)$ and $d_2(t)$. For further details concerning the treatment of critical points see Chapter 7.

For executing the division $v_m = d_1/d_2$ properly over a time interval $[t_1, t_2] \subset [0,1]$, we have to ensure that the path $d_2(t)$ does not pass through $0 \in \mathbb{C}$. If the image $d_2([t_1, t_2])$ of $d_2(t)$ is contained in an interval I with $0 \notin I$, then the path $d_2(t)$ has no zeros in $[t_1, t_2]$. We choose t_2 small enough; see Figure 6.1.

Root: Let $v_m = \sqrt{r}$ with $r \in \{z_{-k+1}, \ldots, z_0, v_1, \ldots, v_{m-1}\}$ be a root instruction. Let $r(t) \colon [0,1] \to \mathbb{C}$, and let $v_m(t) \colon [0,1] \to \mathbb{C}$ be the corresponding paths of the radicand variable r and of v_m in the continuous evaluation. For pointing out the main ideas, we assume that $v := v_m$ is defined by the first root-instruction of the GSP Γ and that the path $r(t)$ does not contain critical points itself, i.e., there is no $t \in [0,1]$ such that $\sqrt{0}$ or $\_/0$ occurs in the computation of $r(t)$. We describe the first step and determine a step length h for this step.

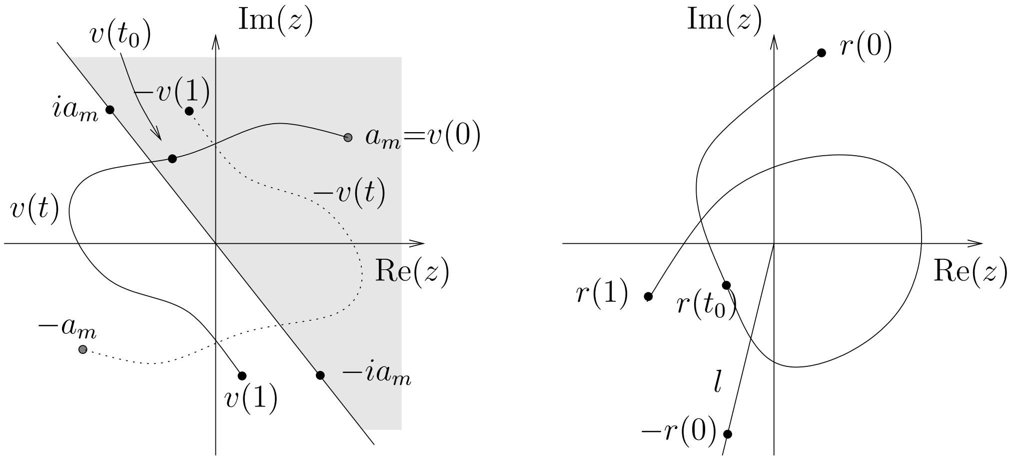


Figure 6.2: The left figure shows the path $v(t) := v_m(t)$, the bisector of $a_m = v(0)$ and $-a_m$ as well as the point $v(t_0)$. On the right hand side the path $r(t)$ of the radicand and the ray l is shown.

The main challenge is to break up the ambiguity of the root-function.

Observation: We can "walk" on the paths $r(t)$ and $v(t) = v_m(t)$ as long as we can "determine the right final instance". The difficulty is that we do not know these paths since they are *implicitly* given by the GSP Γ and the starting instance A. Instead, we can compute all possible instances at a fixed time $\tilde{t}$, i.e., all instances $C = (p_{-k+1}(\tilde{t}), \dots, p_0(\tilde{t}), c_1, \dots, c_m)$ having $p_{-k+1}(\tilde{t}), \dots, p_0(\tilde{t})$ as values for the free variables $z_{-k+1}, \dots, z_0$. From these instances, we have to detect the one that lies on the continuous evaluation at time $\tilde{t}$, which is

$$(p_{-k+1}(\tilde{t}), \dots, p_0(\tilde{t}), v_1(\tilde{t}), \dots, v_m(\tilde{t})).$$

We achieve this by choosing $t_0 \in [0, 1]$ with the property

$$\forall t \in [0, t_0] : \quad |v_m(t) - a_m| < |v_m(t) - (-a_m)|. \tag{6.1}$$

Here, a_m is the value of the variable v_m in the starting instance A. The unique value $b_m \in \{\pm\sqrt{r(t_0)}\}$ with $|b_m - a_m| < |b_m + a_m|$ is the m-th coordinate of the continuous evaluation at time t_0, and $h = t_0$ is a proper step length for the first root-instruction of the first step.

Condition (6.1) means that $v_m(t)$ has to stay in the half plane of $\mathbb{C}$ that is defined by the bisector of a_m and $-a_m$ and that contains a_m; see left of Figure 6.2. This is equivalent to the requirement that the path $r_{|[0,t_0]}(t)$ does not intersect the ray l starting in $0 \in \mathbb{C}$ and passing through the point $-r(0) = -a_m^2$; see right of Figure 6.2.

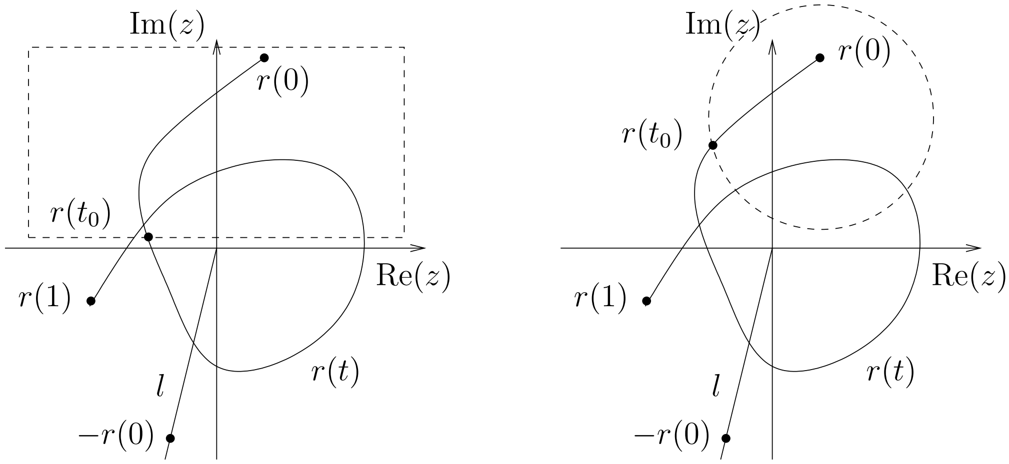


Figure 6.3: The path $r(t)$, the ray l, and the rectangular or circular interval, respectively, are shown.

If we choose $t_0 \in [0,1]$ such that the path $r(t)\colon [0,t_0] \to \mathbb{K}$ stays in a rectangular or circular complex interval that does not contain $0 \in \mathbb{C}$, then the path $r_{|[0,t_0]}(t)$ cannot intersect the ray l; see Figure 6.3. Consequently, it does not pass through $0 \in \mathbb{C}$, and hence, it does not cause a critical point. By reparametrization, we can deal with arbitrary time intervals $[t_1, t_2]$ as well. The same construction can also be done over the field of real numbers $\mathbb{R}$. We summarize the previous observations:

Lemma 6.1.1. Radicand Lemma
Let $r\colon [0,1] \to \mathbb{C}^* = \mathbb{C} \setminus \{0\}$ be a continuous path and $r_0 := r(0)$ its starting point. Let $v\colon [0,1] \to \mathbb{C}^*$ be the unique continuous path with $(v(t))^2 = r(t)$ for all $t \in [0,1]$ and $v(0) = v_0$ for $v_0 \in \{\pm\sqrt{r_0}\}$. Let l be the ray that starts in $0 \in \mathbb{C}$ and passes through $-r_0$. If r does not intersect the ray l, then we have

$$|v(t) - v_0| < |v(t) + v_0|.$$

at any time $t \in [0,1]$. Hence, the path v stays in the Voronoi region of v_0 in the Voronoi diagram of the point set $\{v_0, -v_0\}$. The endpoint $v(1)$ is the unique point $v_1 \in \{\pm\sqrt{r(1)}\}$ with $|v_1 - v_0| < |v_1 + v_0|$.

Proof. (Indirect) Assume that there is a $\tilde{t} \in [0,1]$ with $|v(\tilde{t}) - v_0| \geq |v(\tilde{t}) + v_0|$. Then $v(\tilde{t})$ lies in the half plane that is defined by the bisector b of v_0 and $-v_0$ and that contains $-v_0$. The bisector b is the line passing through the points $0 \in \mathbb{C}$ and iv_0. Since v is a continuous path starting at v_0 and containing a point on the other side of the line b, there must be an intersection point $v(\tilde{t}_0)$ of v and b. We have $v(\tilde{t}_0) = isv_0$ for an $s \in \mathbb{R}$, hence

$$r(\tilde{t}_0) = (v(\tilde{t}_0))^2 = (isv_0)^2 = -s^2 v_0^2 = -s^2 r_0$$

lies on the ray l. This is a contradiction to the assumption of Lemma 6.1.1. □

Corollary 6.1.2. Radicand Corollary
If the path r stays in a rectangle or circle that does not contain $0 \in \mathbb{C}$, then $|v(t) - v_0| < |v(t) + v_0|$ holds for all $t \in [0, 1]$.

Proof. Since both the rectangle and the circle are convex, contain $r(0)$, and exclude $0 \in \mathbb{C}$, they cannot intersect the ray l that starts at $0 \in \mathbb{C}$ and passes through $-r(0)$, see Figure 6.3. Hence, the path r cannot intersect the ray l either, and we can apply Lemma 6.1.1. □

By reparametrization, Lemma 6.1.1 and Corollary 6.1.2 hold for arbitrary time intervals $[t_1, t_2]$, as well.

6.2 An Algorithm for the Tracing Problem

If there is no critical point on the paths $p_{-k+1}, \ldots, p_0$, then we have exactly one continuous evaluation $v_1(t), \ldots, v_n(t)$ starting at a given instance A. We trace this continuous evaluation stepwise. The considerations of Section 6.1 lead to an algorithm for a single tracing step; see Figure 6.4 for an example.

Algorithm 2.

Input: GSP Γ, linear paths $p_{-k+1}, \ldots, p_0$, starting time $t_0 \in [0, 1)$, starting instance $A = (p_{-k+1}(t_0), \ldots, p_0(t_0), a_1, \ldots, a_n)$.

Output: Step Length h, final instance $B = (p_{-k+1}(t_0 + h), \ldots, p_0(t_0 + h), b_1, \ldots, b_n)$.

1. Choose a step length $h > 0$ with $t_0 + h \leq 1$, and consider the time interval $[t_0, t_0 + h]$.

2. Compute intervals I_l with $p_l([t_0, t_0 + h]) = \{p_l(t) | t \in [t_0, t_0 + h]\} \subset I_l$ for $l = -k + 1, \ldots, 0$. Since $p_{-k+1}, \ldots, p_0$ are linear paths, the interval I_l can be chosen as the smallest interval containing the line segment $p_l([t_0, t_0 + h])$ in the used interval arithmetic.

3. Compute the interval-GSP Γ_{int}.
   Set the free variables Z_l of Γ_{int} to the intervals computed in step 2, i.e., set $Z_l := I_l$ for $l \in \{-k + 1, \ldots, 0\}$.
   Set the free variables z_l of Γ to their positions after the step of length h, i.e., set $z_l := b_l := p_l(t_0 + h)$ for $l \in \{-k + 1, \ldots, 0\}$.

$h := h/2$

GSP	A	$t \in [0,h]$	B	
$z_{-1} \leftarrow$ FREE	9	$p_{-1}(t) \in I_{-1} := \{2h+9; 2h\}$	$b_{-1} := p_{-1}(h)$	
$z_0 \leftarrow$ FREE	0	$p_0(t) \in I_0 := \{\frac{(-2+i)h}{2}; \frac{h}{2}\sqrt{5}\}$	$b_0 := p_0(h)$	
$v_1 \leftarrow z_{-1} + z_0$	9	$v_1(t) \in I_1 := I_{-1} + I_0$	$b_1 := b_{-1} + b_0$	
$v_2 \leftarrow v_1 \cdot z_{-1}$	81	$v_2(t) \in I_2 := I_1 \cdot I_{-1}$	$b_2 := b_1 \cdot b_{-1}$	$0 \in I_2$
$v_3 \leftarrow \sqrt{v_2}$	-9	$v_3(t) \in I_3 := \sqrt{I_2} \ni -9$	$b_3 := \sqrt{b_2} \in I_3$	
$v_4 \leftarrow v_3 - z_{-1}$	-18	$v_4(t) \in I_4 := I_3 - I_{-1}$	$b_4 := b_3 - b_{-1}$	$0 \in I_4$
$v_5 \leftarrow v_3/v_4$	1/2	$v_5(t) \in I_5 := I_3/I_4$	$b_5 := b_3/b_4$	

Figure 6.4: Algorithm 2 is visualized with an example. The paths of the free variables z_{-1} and z_0 are $p_{-1}(t) = 4t + 9$ and $p_0(t) = (-2 + i)t$, $t_0 = 0$. If the interval of a radicand or divisor variable contains zero, then the algorithm is restarted with a smaller step length. Here, we use circular interval arithmetic.

4. Follow the operations of Γ_{int} and Γ until the next division- or root-variable v_d is reached. Here, for each variable v_i the interval I_i and the i-th coordinate b_i of the final instance B is computed:

Operation of Γ	b_i	I_i
$v_i \leftarrow v_a + v_b$	$b_i := b_a + b_b$	$I_i := I_a + I_b$
$v_i \leftarrow v_a - v_b$	$b_i := b_a - b_b$	$I_i := I_a - I_b$
$v_i \leftarrow v_a \cdot v_b$	$b_i := b_a \cdot b_b$	$I_i := I_a \cdot I_b$

We consider the interval I_c of the divisor or radicand v_c of v_d.

5. **If** $0 \in I_c$

then // *I_c might contain a critical point.*
Restart from step 2 with step length $h/2$.

else // *I_c does not contain a critical point for v_d.*
if $v_d = v_j/v_c$ is a division variable **then**
$b_d := b_j/b_c$, $I_d := I_j/I_c$.
if $v_d = \sqrt{v_c}$ is a root variable **then**
choose $b_d \in \{\pm\sqrt{b_c}\}$ with $|b_d - a_d| < |b_d + a_d|$.
$I_d := \sqrt{I_c} \ni b_d$
GoTo step 4

6. Return h, $B = (b_{-k+1}, \ldots, b_0, b_1, \ldots, b_n)$.

This algorithm is based on the Radicand Corollary 6.1.2. If $0 \notin I_c$ and $v_d = \sqrt{v_c}$ in step 5, then we have $|v_d(t) - a_d| < |v_d(t) + a_d|$ for all $t \in [t_0, t_0 + h]$. Since $b_d = v_d(t_0 + h)$, the coordinate b_d is uniquely defined by the corresponding condition in step 5. The inequality $|b_d - a_d| < |b_d + a_d|$ could be checked using separation bounds [10].

Unfortunately, Algorithm 2 does not give an indication for a "good" choice for the step length h. Algorithm 4 from Section 6.3 extends Algorithm 2 by an improved step length adaptation.

We use the inclusion monotonicity property of the chosen interval arithmetic to show that Algorithm 2 terminates. For this reason, the intervals I_l in step 2 must be constructed in such a way that the inclusion monotonicity property holds: Let I_l be the interval of a free variable z_l with step length h in Algorithm 2, and let $I_{l\text{new}}$ be the interval of z_l with step length h_{new}. Thus, we consider the time intervals $[t_0, t_0+h]$ and $[t_0, t_0+h_{\text{new}}]$. We require that $h_{\text{new}} \leq h$ implies $I_{l\text{new}} \subset I_l$. If p_l is a linear path and if I_l is chosen as the smallest interval containing the line segment $p_l([t_0, t_0+h])$ as in step 2, then this property is fulfilled. Algorithm 2 also works for nonlinear paths p_l. Then the intervals I_l in step 2 have to be constructed such that the inclusion monotonicity property holds; see e.g. [1, Abschnitt 3] for the real case and [57, Chap. 2] for circular arithmetic.

In step 5, the square root $I_d = \sqrt{I_c}$ is uniquely determined by the condition $b_d \in \sqrt{I_c}$. This fact holds since $\sqrt{I_c} \cap -\sqrt{I_c} = \emptyset$ if $0 \notin I_c$; see Lemma A.2.10 resp. A.2.23 for complex interval arithmetic. Equivalently, we could require $a_d \in \sqrt{I_c}$.

Close to a critical point, the step length computed by Algorithm 2 becomes arbitrary small. At a critical point, $0 = v_c \in I_c$ holds. Only if $0 \notin I_c$ the algorithm continues with the next variable in step 5. In Chapter 7, we discuss how critical points can be treated.

Lemma 6.2.1. *Algorithm 2 terminates, and the correct final instance B after the step of length h is computed.*

Proof. Since A is an instance of the GSP Γ, the continuous evaluation $v_1(t), \ldots, v_n(t)$ along the paths $p_{-k+1}, \ldots, p_0$ starting at A exists at least locally over a time interval $[t_0, t_0 + h]$ for an $h \in (0, 1 - t_0]$. First, we prove that Algorithm 2 terminates. We show by induction that for every division or square root operation there are finitely many restarts in step 5. The proof uses the estimates from Lemma A.1.10, Lemma A.2.12, or Lemma A.2.25, depending on the chosen interval arithmetic. These three lemmas are based on the continuity of the five operations addition, subtraction, mulitplication, division, and square root.

We consider the first critical variable v_d and its radicand or divisor variable v_c. Since A is an instance of Γ, we have $a_c \neq 0$. As before, a_c is the coordinate of the variable v_c in the starting instance A. Let $\epsilon := \frac{|a_c|}{2}$. The estimates of Lemma A.1.10, Lemma A.2.12, and Lemma A.2.25 combined with an inductive argument imply that there is a step length $h_1 > 0$ with the following property: If Algorithm 2 is executed with h_1 as step length until the critical variable v_d is reached, then $I_c \subset a_c + E_\epsilon$, where E_ϵ is the interval with 0 as midpoint and diameter 2ϵ in the corresponding interval arithmetic. Hence, $0 \notin I_c$, and we have found a suitable step length for the first critical operation. We can determine such a step length h_1 by a binary search in finitely many steps. This is done in step 5.

Now, we assume that we have constructed a suitable step length $h_j > 0$ such that for the jth critical variable $v_{d,j}$ and its divisor or radicand variable $v_{c,j}$ we have $0 \notin I_{c,j}$. We consider the $j+1$st division or root variable $v_{d,j+1}$ and its radicand or divisor variable $v_{c,j+1}$. If $0 \notin I_{c,j+1}$, we are done. Otherwise there is –similar to the construction of h_1– a step length $h_{j+1} > 0$ for which $0 \notin I_{c,j+1}$ holds. Again, this can be shown by induction using the estimates of Lemma A.1.10, Lemma A.2.12, or Lemma A.2.25. Since we have chosen an interval arithmetic with the inclusion monotonicity property, we still have $0 \notin I_c$, $0 \notin I_{c,2}, \ldots,$ $0 \notin I_{c,j}$ since these intervals do not become larger if the step length is decreased.

At last, we show that the computed final instance B is the correct final instance of the step of length h, i.e., we show

$$\begin{aligned} B &= (p_{-k+1}(t_0+h), \ldots, p_0(t_0+h), b_1, \ldots, b_n) \\ &= \big(p_{-k+1}(t_0+h), \ldots, p_0(t_0+h), v_1(t_0+h), \ldots, v_n(t_0+h)\big). \end{aligned}$$

Thus, we have to prove that the choices for the square root operations in step 5 are correct. Lemma 3.6.4 on page 46 implies that the range of the continuous evaluation $v_1(t), \ldots, v_n(t)$ over the time interval $[t_0, t_0+h]$ is contained in the instance $(I_{-k+1}, \ldots, I_0, I_1, \ldots, I_n)$ of Γ_{int}. Step 5 ensures $0 \notin I_c$ for every radicand variable v_c of Γ. The Radicand Corollary 6.1.2 from page 73 implies that the choice in step 5 for b_d for every root variable v_d is correct. By Corollary 6.1.2, we have $|v_d(t) - a_d| < |v_d(t) + a_d|$ for all $t \in [t_0, t_0+h]$ since $0 \notin I_c$. In step 5, b_d is defined by $b_d \in \{\pm\sqrt{b_c}\}$ with $|b_d - a_d| < |b_d + a_d|$. Hence $b_d = v_d(t_0+h)$ holds. □

As long as there is no critical point on the paths $p_{-k+1}, \ldots, p_0$ there exists exactly one continuous evaluation $v_1(t), \ldots, v_n(t)$ starting at a given starting instance A. Algorithm 3 traces this continuous evaluation stepwise using Algorithm 2. In the process, the final instance of a previous step is the starting instance of the next step.

Algorithm 3.

Input: GSP Γ, linear paths $p_{-k+1}, \ldots, p_0$,
starting instance $A = (p_{-k+1}(0), \ldots, p_0(0), a_1, \ldots, a_n)$.

Output: final instance $B = (p_{-k+1}(1), \ldots, p_0(1), b_1, \ldots, b_n)$.

1. $\tilde{A} := A$ // starting instance
   $t := 0$ // starting time

2. **while** $t < 1$ **do**
   Run Algorithm 2 with starting time $t_0 := t$ and starting instance
   $A := \tilde{A}$, the given GSP Γ and the given paths $p_{-k+1}, \ldots, p_0$
   // the step length h and the final instance B are returned.
   $\tilde{A} := B$ // B is the starting instance for the next step
   $t := t + h$

3. **return** $\tilde{A}$.

Remark 6.2.2. 1. If there occurs a critical point along at least one of the paths, then Algorithm 3 does not terminate. The step length computed by Algorithm 2 in step 2 of Algorithm 3 becomes arbitrary small before the critical point is reached. This happens since the interval I_c in step 4 and step 5 of Algorithm 2 has to be arbitrary small in a neighborhood of a critical point. At a critical point, we have $0 = v_c \in I_c$. However, in step 5 is supposed that $0 \notin I_c$ in order to continue with the next variable. We discuss in Chapter 7, how critical points can be treated.

2. If the GSP Γ has real variables and if we know in advance that no critical point occurs, then no curve of a radicand or divisor variable in the continuous evaluation passes through $0 \in \mathbb{R}$. Thus, each curve is strictly positive or strictly negative, and the Tracing Problem could be solved easier. The problem with this approach is that we would have to guarantee in advance that there is no critical point on the continuous evaluation. Unfortunately, we do not know how to do this yet. For this reason, we propose to use Algorithm 3 combined with the treatment of critical points from Chapter 7.

Theorem 6.2.3. *Algorithm 3 terminates and computes the correct final instance B as long as no critical point occurs.*

Proof. Induction on the number j of root and division operations of Γ:
Since critical points are excluded, there exists a unique continuous evaluation along the paths $p_{-k+1}, \ldots, p_0$ starting at the instance A. We prove by induction on j that each step has at least step length h for a fixed $h > 0$. Furthermore, h

can be chosen such that it is a minimum step length for every starting time t_0. This proof is based on the uniform continuity of the five operations addition, subtraction, multiplication, division, and square root on compact sets S; for the division and for the square root, we must require that the divisor and the radicand is non-zero to obtain uniform continuity. Note that this uniform continuity is the base of the supplements of Lemmas A.1.10, A.2.12 and A.2.25.

Basic Step: Let v_{d_1} be defined by the first root or division operation of Γ, i.e., $v_{d_1} = v_j/v_c$ or $v_{d_1} = \sqrt{v_c}$, and let $v_{d_1}(t)$ be its path in the continuous evaluation. Let $v_c(t)$ be the path of v_c in the continuous evaluation. We consider the entire time interval $[0,1]$ and follow Algorithm 2 through step 4. We get the compact intervals $I_{-k+1}, \ldots, I_0, I_1, \ldots, I_{d_1-1}$. Since critical points are excluded, the path $v_c(t)$ does not pass through 0. Additionally, $[0,1]$ is a compact interval, and the mapping $t \mapsto |v_c(t)|$ is continuous. Thus $\delta := \min_{t\in[0,1]} |v_c(t)| > 0$ holds. The supplements of Lemma A.1.10, Lemma A.2.12 or Lemma A.2.25 combined with an inductive argument lead to a step length $h_1 > 0$ for a tracing step for the first $d_1 - 1$ dependent variables of Γ that only depends on δ and on $I_{-k+1}, \ldots, I_0, I_1, \ldots, I_{d_1-1}$. It neither depends on the starting time t_0 nor on a concrete instance. In the last step the step length is at least $\min\{h_1, 1 - t_0\}$.

Inductive Step: Let Γ be a GSP with $j + 1$ root or division operations. Let us consider the $j + 1$st root or division operation of Γ and its corresponding variable $v_{d_{j+1}}$. Let $\tilde{\Gamma} = \Gamma^{(d_{j+1}-1)}$ be the $(d_{j+1}-1)$-head GSP of Γ. In other words, $\tilde{\Gamma}$ is the GSP after cutting Γ before the variable $v_{d_{j+1}}$; see Definition 3.2.1. Since $\tilde{\Gamma}$ has j root or division operations, we can apply the induction hypothesis. Let h_j be a minimum step length for $\tilde{\Gamma}$ that is independent of the starting time t_0.

We partition the interval $[0,1]$ into $\lceil \frac{1}{h_j} \rceil$ intervals of length h_j at most. Since h_j is independent of the starting time t_0, we know that, for each of these subintervals, Algorithm 3 does not reduce the size of these time intervals for the first j root or division operations. This observation holds since we have chosen an interval arithmetic with the inclusion monotonicity property. Hence, for each of the subintervals we have the same situation as for the first root or division operation and the entire time interval $[0,1]$. Consequently, the inductive conclusion can be shown in the same way as the basic step. For each subinterval $[lh_j, (l+1)h_j] \subset [0,1]$ we get a step length $h_{j,l}$. Finally, $h_{j+1} := \min\{h_{j,l} \mid l = 1, 2, 3, \ldots, \lceil \frac{1}{h_j} \rceil\}$ is the wanted step length. In the last step the step length is at least $\min\{h_{j+1}, 1 - t_0\}$.

Lemma 6.2.1 implies that the correct final instance is computed. □

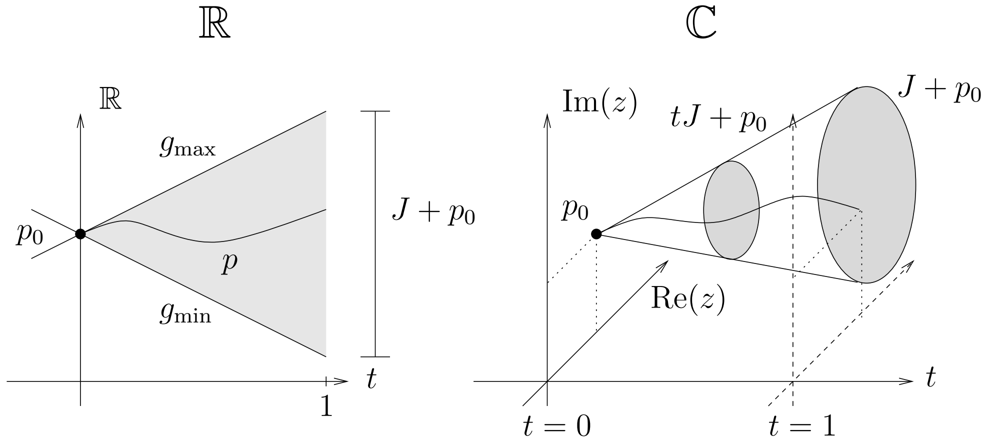


Figure 6.5: Illustration of the Cone Lemma: The left figure shows the situation over $\mathbb{R}$ with $J = [m_{\min}, m_{\max}]$, the right one shows the situation over $\mathbb{C}$ for circular intervals.

6.3 An Algorithm for the Tracing Problem Using $\dot{\Gamma}$

We extend Algorithm 2 from Section 6.2 by an improved step length adaptation. The resulting Cone Algorithm (Algorithm 4) uses the derivative GSP $\dot{\Gamma}$. Like Algorithm 3 from Section 6.2, Algorithm 4 works for GSPs with complex variables as well as with GSPs having real variables. Again, we restrict to linear paths $p_{-k+1}, \ldots, p_0$ of the free variables $z_{-k+1}, \ldots, z_0$ and we assume that there are no critical points on the paths $p_{-k+1}, \ldots, p_0$ of the free variables $z_{-k+1}, \ldots, z_0$. A generalization to arbitrary continuously differentiable paths $p_{-k+1}, \ldots, p_0$ is indicated. The treatment of critical points is discussed in Chapter 7.

6.3.1 The Cone Algorithm

The Cone Algorithm (Algorithm 4) is based on the Cone Lemma 6.3.1 and on the Radicand Corollary 6.1.2 from page 73 . The Cone Lemma is a reformulation of the mean value theorem, see Lemma 6.3.1 and Figure 6.5. Like Algorithm 2, Algorithm 4 uses an interval arithmetic that must fulfill the *inclusion monotonicity property.* The advantage over Algorithm 3 and Algorithm 2 is that for each step the step length is computed *directly.* The Cone Lemma is related to the Interval Newton Method in one dimension, since the same cone is considered; see Appendix A.5. A similar idea is used in [36, p. 900] for the real case.

Lemma 6.3.1. Cone Lemma; see Figure 6.5.

$\mathbb{R}$: *Let* $p\colon [0,1] \to \mathbb{R}$ *be a differentiable path and* $p_0 := p(0)$*. Furthermore, let* $m_{\min} \le \dot{p}(t) \le m_{\max}$ *for all* $t \in [0,1]$*. Let*

$$\begin{aligned} g_{\min}(t) &= t\,m_{\min} + p_0 \quad \textit{and} \\ g_{\max}(t) &= t\,m_{\max} + p_0 \end{aligned}$$

be the lines passing through the point $(0, p_0)$ *and having* $m_{\min}$ *and* $m_{\max}$ *as slopes. Then,* $g_{\min}(t) \le p(t) \le g_{\max}(t)$ *holds for all* $t \in [0,1]$*.*

If the time interval $[0,1]$ *is replaced by an arbitrary interval* $[h_1, h_2]$*, then we have* $p_0 := p(h_1)$*,* $m_{\min} \le \dot{p}(t) \le m_{\max}$ *for all* $t \in [h_1, h_2]$*,* $g_{\min}(t) = t\,m_{\min} + p_0 - h_1 m_{\min}$*, and* $g_{\max}(t) = t\,m_{\max} + p_0 - h_1 m_{\max}$*. Thus* $p(t) \in (t - h_1)[m_{\min}, m_{\max}] + p_0$ *holds.*

$\mathbb{C}$: *Let* $p\colon [0,1] \to \mathbb{C}$ *be a continuously differentiable path and* $p_0 := p(0)$*. Let* $J \subset \mathbb{C}$ *be a complex interval with* $\dot{p}(t) \in J$ *for all* $t \in [0,1]$*. Then,* $p(t) \in tJ + p_0 = \{tz + p_0 | z \in J\}$ *holds for all* $t \in [0,1]$ *.*

If the time interval $[0,1]$ *is replaced by an arbitrary interval* $[h_1, h_2]$*, then we have* $p_0 := p(h_1)$*, and* $p(t) \in (t - h_1)J + p_0$*.*

Proof. The real and the complex case are treated separately.

$\mathbb{R}$: Let $t \in (0,1]$. By the mean value theorem there is a $\tilde{t} \in (0,t)$ with $\dot{p}(\tilde{t}) = \frac{p(t)-p(0)}{t-0} = \frac{p(t)-p_0}{t}$, i.e.,

$$p(t) = t\dot{p}(\tilde{t}) + p_0 \begin{cases} \le & t\,m_{\max} + p_0 = g_{\max}(t) \\ \ge & t\,m_{\min} + p_0 = g_{\min}(t). \end{cases}$$

For $t = 0$ we have $g_{\min}(t) = p_0 = p(0) = g_{\max}(t)$.

$\mathbb{C}$: Let $t \in (0,1]$. By the mean value theorem,

$$p(t) - p_0 = t\left(\int_0^1 \mathrm{Re}(\dot{p}(ts))ds + i\int_0^1 \mathrm{Im}(\dot{p}(ts))ds\right) = t\int_0^1 \dot{p}(ts)ds.$$

It remains to show $\int_0^1 \dot{p}(ts)ds \in J$. This is done by considering the Riemann sum of a subdivision of the interval $[0,1]$. For any subdivision $0 = s_0 < s_1 < \cdots < s_m = 1$ of the interval $[0,1]$, the Riemann sum

$$\sum_{j=1}^{m} \mathrm{Re}(\dot{p}(ts_j)(s_j - s_{j-1}) + i\sum_{j=1}^{m} \mathrm{Im}(\dot{p}(ts_j)(s_j - s_{j-1}) \tag{6.2}$$

$$= \sum_{j=1}^{m} \underbrace{\dot{p}(ts_j)}_{\in J}(s_j - s_{j-1}) \tag{6.3}$$

is a convex combination of points of J. Since J is convex, the Riemann sum (6.3) is an element of J as well. The integral $\int_0^1 \dot{p}(ts)ds$ is the limit of these Riemann sums as $\max_{j=1,\ldots,m}(s_j - s_{j-1}) \to 0$. Since J is a closed subset of $\mathbb{C}$, this limit is contained in J, and $\int_0^1 \dot{p}(ts)ds \in J$. For $t = 0$ we have $p(t) = p_0 \in \{p_0\} = 0 \cdot J + p_0$.

If instead of $[0, 1]$ the time interval $[h_1, h_2]$ is considered, the mean value theorem implies

$$p(t) - p(h_1) = (t - h_1) \int_0^1 \dot{p}\big(h_1 + s(t - h_1)\big)ds,$$

and the claimed formula follows. □

We have exactly one continuous evaluation $v_1(t), \ldots, v_n(t)$ along the paths p_{-k+1}, $\ldots, p_0$ that starts at the instance A as long as we do not hit a critical point. As in Section 6.2, we trace this continuous evaluation stepwise. Algorithm 4 describes a single tracing step, Algorithm 5 uses Algorithm 4 and traces the whole continuous evaluation as long as no critical points occur.

Algorithm 4. *Cone Algorithm*

Input: GSP Γ, linear paths $p_{-k+1}, \ldots, p_0$, starting time $t_0 \in [0, 1]$
starting instance $A = (p_{-k+1}(t_0), \ldots, p_0(t_0), a_1, \ldots, a_n)$.

Output: Step Length h,
final instance $B = (p_{-k+1}(t_0 + h), \ldots, p_0(t_0 + h), b_1, \ldots, b_n)$.

1. Choose a step length $h > 0$ with $t_0 + h \leq 1$ and consider the time interval $[t_0, t_0 + h]$. Let T_h be the smallest interval containing $[0, h]$ in the used interval arithmetic.

2. Compute intervals I_l with $p_l([t_0, t_0 + h]) = \{p_l(t) | t \in [t_0, t_0 + h]\} \subset I_l$ for $l = -k + 1, \ldots, 0$. Since $p_{-k+1}, \ldots, p_0$ are linear paths, I_l can be chosen as the smallest interval containing the line segment $p_l([t_0, t_0 + h])$.

   Compute the derivative GSP $\dot{\Gamma}$ of the GSP Γ.

   Compute intervals $\dot{I}_l$ with $\dot{p}_l([t_0, t_0 + h]) = \{\dot{p}_l(t) | t \in [t_0, t_0 + h]\} \subset \dot{I}_l$ for $l = -k + 1, \ldots, 0$, where $\dot{p}_l$ is the derivative of p. Since p_l is a linear path, $\dot{I}_l$ can be chosen as $\dot{I}_l = \{\dot{p}_l(t_0)\}$.

3. Compute the interval-GSP $\dot{\Gamma}_{\text{int}}$ of the derivative GSP $\dot{\Gamma}$.
   Set the free variables Z_l and $\dot{Z}_l$ of $\dot{\Gamma}_{\text{int}}$ to the intervals computed in step 2, i.e., set $Z_l := I_l$ and $\dot{Z}_l := \dot{I}_l$ for $l \in \{-k + 1, \ldots, 0\}$.
   Set the free variables z_l of Γ to their positions after the step of length h, i.e., set $z_l := b_l := p_l(t_0 + h)$ for $l \in \{-k + 1, \ldots, 0\}$.

4. Follow the operations of $\dot{\Gamma}_{\text{int}}$ and Γ until the next division or root variable v_d is reached. Here, for each variable v_i the interval $\dot{I}_i$ and the i-th coordinate b_i of the final instance B is computed:

Operation of Γ	b_i	$\dot{I}_i$	I_i
$v_i \leftarrow v_a + v_b$	$b_i := b_a + b_b$	$\dot{I}_i := \dot{I}_a + \dot{I}_b$	$I_i := a_i + T_h \dot{I}_i$
$v_i \leftarrow v_a - v_b$	$b_i := b_a - b_b$	$\dot{I}_i := \dot{I}_a - \dot{I}_b$	$I_i := a_i + T_h \dot{I}_i$
$v_i \leftarrow v_a \cdot v_b$	$b_i := b_a \cdot b_b$	$\dot{I}_i := \dot{I}_a I_b + I_a \dot{I}_b$	$I_i := a_i + T_h \dot{I}_i$

Consider the variable v_c of the divisor or radicand of v_d, its coordinate a_c of A, the interval $I_c = a_c + T_h \dot{I}_c$, and the cone C in $[t_0, t_0+h] \times \mathbb{R}$ or $[t_0, t_0+h] \times \mathbb{C}$ with apex (t_0, a_c) and base $(t_0 + h, I_c) = \{(t_0 + h, z) | \, z \in I_c\}$; see Figure 6.6.

5. **If** the cone C intersects the t-axis

   then // *We might have* $0 \in a_c + (t - t_0)\dot{I}_c$ *for a* $t \in [t_0, t_0 + h]$.
   Compute the smallest time $t' \in [t_0, t_0 + h]$ for that $(t', 0)$ lies on the surface of the cone C.
   Restart from step 2 with the step length $h := \frac{2}{3}(t' - t_0)$.
   // *There is at most one restart per root or division*
   // *operation (see Theorem 6.3.3)!*

   else // $0 \notin I_c$ *holds and* $v_c(t)$ *does not pass through 0.*
   if $v_d = v_j / v_c$ is a division variable **then**
   $b_d := b_j / b_c$
   $\dot{I}_d := \frac{\dot{I}_j I_c - I_j \dot{I}_c}{I_c^2}$
   $I_d := a_d + T_h \dot{I}_d$
   if $v_d = \sqrt{v_c}$ is a root variable **then**
   choose $b_d \in \{\pm\sqrt{b_c}\}$ with $|b_d - a_d| < |b_d + a_d|$.
   $I_d := \sqrt{I_c} \ni b_d$
   $\dot{I}_d := \frac{\dot{I}_c}{2 I_d}$
   GoTo step 4

6. Return h, $B = (b_{-k+1}, \ldots, b_0, b_1, \ldots, b_n)$.

The choice of t' in step 5 of the algorithm above is shown in Figure 6.6 and treated in Section 6.3.2. Since we have $0 \in T_h$, we have $a_c \in I_c$, and the cone C intersects the t-axis if and only if $0 \in I_c = a_c + T_h \dot{I}_c$.

Remark 6.3.2. 1. As with Algorithm 2, we notice that the intervals I_i and $\dot{I}_i$ in step 2 must be constructed in such a way that the inclusion monotonicity property holds. This is crucial for the algorithm, otherwise it might not terminate. If they are defined as proposed in step 2, the inclusion monotonicity property holds.

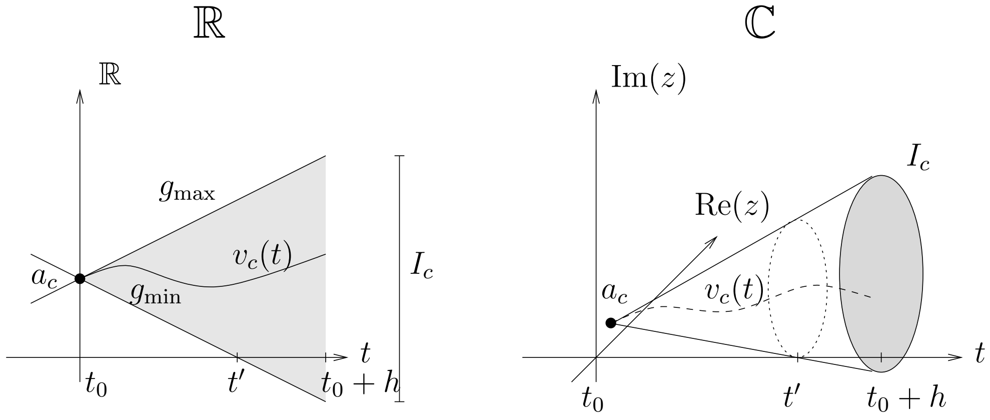


Figure 6.6: The cone C from step 4 and the definition of t' from step 5 of Algorithm 4 are shown. The left Figure shows the situation over $\mathbb{R}$, the right one the situation over $\mathbb{C}$ for circular intervals.

2. In step 5, the square root $I_d = \sqrt{I_c}$ is uniquely determined by the condition $b_d \in \sqrt{I_c}$. This fact holds since $\sqrt{I_c} \cap -\sqrt{I_c} = \emptyset$; see Lemma A.2.10, and A.2.23 for complex interval arithmetic.

3. In step 1, we define the interval T_h as the smallest interval containing the time interval $[t_0, t_0 + h]$ *in the used interval arithmetic*. For real interval arithmetic ($\mathrm{I}(\mathbb{R})$) and for rectangular complex arithmetic ($\mathrm{R}(\mathbb{C})$), we have $T_h = [t_0, t_0 + h]$, i.e., T_h is the time interval itself. In circular arithmetic ($\mathrm{K}(\mathbb{C})$), we have $T_h = \{t_0 + \frac{h}{2}\, ;\, \frac{h}{2}\}$, which is the circle having the time interval $[t_0, t_0 + h]$ as diameter. This overestimation is a disadvantage of circular interval arithmetic. In Improvement 6.3.8 from page 92 we discuss that T_h can be replaced by h in the definition of I_i if and only if $0 \in \dot{I}_i$.

Theorem 6.3.3. *For each division or root operation there is at most one restart in step 5 of Algorithm 4.*

Proof. Let v_d be a division or root variable and C be the corresponding cone defined in step 4. The variable v_d only causes a restart if its cone C intersects the t-axis. Hence, we have to show that the cone C of v_d intersects the t-axis in at most one run of Algorithm 4. If C intersects the t-axis, Algorithm 4 is restarted with a smaller step length in step 5, which we denote by h_{new}. Similarly, we denote the new values of all variables and intervals of the next run of Algorithm 4 by their old names and add "new" as index.

By construction of h_{new}, we ensure that in this next run of Algorithm 4 the new cone C_{new} does not intersect the t-axis since we use an interval arithmetic with the inclusion monotonicity property. For the new time interval $[t_0, t_0 + h_{\text{new}}]$, we have $[t_0, t_0 + h_{\text{new}}] \subset [t_0, t_0 + h]$ and $T_{h_{\text{new}}} \subset T_h$. This implies $I_{l,\text{new}} \subset I_l$ and $\dot{I}_{l,\text{new}} \subset \dot{I}_l$

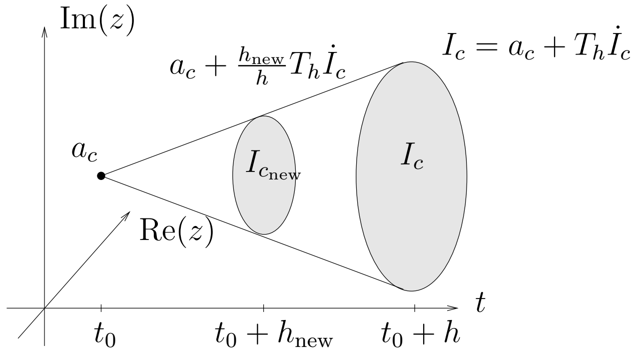


Figure 6.7: The cone C with apex (t_0, a_c) and base $(t_0 + h, I_c)$ and its section $(t_0 + h_{\text{new}}, a_c + \frac{h_{\text{new}}}{h} T_h \dot{I}_c)$ at time $t_0 + h_{\text{new}}$ are shown.

for the intervals defined in step 2 of Algorithm 4; see Remark 6.3.2. Hence, for every new interval I_{new} of an interval I that is computed by Algorithm 4 we have $I_{\text{new}} \subset I$. This implies that the base $(t_0 + h_{\text{new}}, a_c + T_{h_{\text{new}}} \dot{I}_{c,\text{new}})$ of C_{new} is a subset of the section

$$(t_0 + h_{\text{new}}, a_c + \frac{h_{\text{new}}}{h} T_h \dot{I}_c) = (t_0 + h_{\text{new}}, a_c + T_{h_{\text{new}}} \dot{I}_c)$$

of the old cone C at time $t_0 + h_{\text{new}}$; see Figure 6.7. Since by construction of h_{new}, the cone with base $(t_0 + h_{\text{new}}, a_c + \frac{h_{\text{new}}}{h} T_h \dot{I}_c)$ and apex $(t_0, 0)$ does not contain 0, we have $0 \notin C_{\text{new}}$. □

Theorem 6.3.4. *Algorithm 4 terminates in $O((n+k)n_0)$ time where n_0 is the total number of root and division operations of Γ. It computes a step length $h > 0$ and the correct final instance B of the step of length h.*

Proof. This proof has a similar structure as the proof of Lemma 6.2.1 from page 75. Theorem 6.3.3 implies that Algorithm 4 terminates. Let B be the final instance computed by Algorithm 4 and h the step length. We show that B is the correct final instance for the step of length h.

Since A is an instance, there is a time interval $[t_0, t_1]$ for that a continuous evaluation $v_1(t), \ldots, v_n(t)$ along the given paths $p_{-k+1}, \ldots, p_0$ starting at A exists. Hence it is also unique. We have to prove

$$B = \big(p_{-k+1}(t_0 + h), \ldots, p_0(t_0 + h), v_1(t_0 + h), \ldots, v_n(t_0 + h)\big). \tag{6.4}$$

Additionally, we prove $\dot{v}_i([t_0.t_0 + h]) \subset \dot{I}_i$ and $v_i([t_0, t_0 + h]) \subset I_i$ for $i = 1, \ldots, n$. This is done via induction on the number j of root or division operations of Γ. For the variables that are defined by addition, subtraction, or multiplication the correct value in the final instance B is computed in step 4 of Algorithm 4.

Let Γ be a GSP with one variable v_d that is defined by a root or division operation. Following the notation of Algorithm 4, we denote the radicand or divisor variable by v_c. Let $\dot{I}_c$ and I_c be the intervals computed in step 4. Firstly, we show $\dot{v}_c([t_0, t_0 + h]) \subset \dot{I}_c$ and $v_c([t_0, t_0 + h]) \subset I_c$ via Claim 6.3.5. Since Γ is a GSP, either $c \in \{1, \ldots, d-1\}$ holds, or v_c is a free variable.

Claim 6.3.5. The inclusions $\dot{v}_{i_0}([t_0, t_0 + h]) \subset \dot{I}_{i_0}$ and $v_{i_0}([t_0, t_0 + h]) \subset I_{i_0}$ hold for $i_0 \in \{1, \ldots, d-1\}$.

Proof. For the free variables z_l, $l \in \{-k+1, \ldots, 0\}$, the intervals $\dot{I}_l$ and I_l are defined in step 2 of Algorithm 4 such that $\dot{p}_l([t_0, t_0+h]) \subset \dot{I}_l$ and $p_l([t_0, t_0+h]) \subset I_l$ hold; see Remark 6.3.2.

Now, we assume $\dot{v}_i([t_0, t_0 + h]) \subset \dot{I}_i$ and $v_i([t_0, t_0 + h]) \subset I_i$ for $i < i_0$, and we consider the dependent variable v_{i_0}. Since $i_0 \in \{1, \ldots, d-1\}$ and Γ is a GSP, the variable v_{i_0} is defined as $v_{i_0} = v_a \pm v_b$ or $v_{i_0} = v_a \cdot v_b$ with $a,\ b \in \{-k+1, \ldots, 0, 1, \ldots, i_0 - 1\}$. By induction, we have

$$\begin{array}{rclrcl} v_a([t_0, t_0 + h]) & \subset & I_a, & \dot{v}_a([t_0, t_0 + h]) & \subset & \dot{I}_a, \\ v_b([t_0, t_0 + h]) & \subset & I_b, & \dot{v}_b([t_0, t_0 + h]) & \subset & \dot{I}_b. \end{array}$$

Thus, Lemma 3.6.3 applied to $\dot{\Gamma}$ implies $\dot{v}_{i_0}([t_0, t_0 + h]) \subset \dot{I}_{i_0}$. We consider the interval $I_{i_0} = a_{i_0} + T_h \dot{I}_{i_0}$. As in Algorithm 4, T_h is the smallest interval containing $[0, h]$ in the used interval arithmetic. We have $a_{i_0} + \tilde{h}\dot{I}_{i_0} \subset a_{i_0} + T_h \dot{I}_{i_0} = I_{i_0}$ for every $\tilde{h} \in [0, h]$. The Cone Lemma 6.3.1 implies $v_{i_0}(t_0 + \tilde{h}) \in a_{i_0} + \tilde{h}\dot{I}_{i_0}$. Hence $v_{i_0}([t_0, t_0 + h]) \subset I_{i_0}$ holds. □

To sum up, the variable v_c is the radicand or divisor of the variable v_d. By Claim 6.3.5, we know $v_c([t_0, t_0 + h]) \subset I_c$. In step 5 of Algorithm 4, the root or division instruction is only executed if $0 \notin I_c$. Recall that $0 \notin I_c$ if and only if the cone C does not intersect the t-axis. Thus, we can assume $0 \notin I_c$ when we consider the division or root operation that defines the variable v_d.

Let v_d be defined by the division operation ($v_d = v_j/v_c$). Since $0 \notin I_c$, the division in step 5 can be executed in interval arithmetic properly. Since $\dot{v}_j([t_0, t_0+h]) \subset \dot{I}_j$, $v_j([t_0, t_0 + h]) \subset I_j$, $\dot{v}_c([t_0, t_0 + h]) \subset \dot{I}_c$, and $v_c([t_0, t_0 + h]) \subset I_c \not\ni 0$ we have $\dot{v}_d([t_0, t_0 + h]) \subset \dot{I}_d$ by Lemma 3.6.3 and $v_d([t_0, t_0 + h]) \subset I_d = a_d + T_h \dot{I}_d$ by Lemma 6.3.1 (Cone Lemma).

Now, we investigate the case when v_d is defined by a root-operation, i.e., $v_d = \sqrt{v_c}$ in our notation. As in step 5 of Algorithm 4, let $b_d \in \{\pm\sqrt{b_c}\}$ with $|b_d - a_d| < |b_d + a_d|$ where a_d is the coordinate of v_d in the starting instance A. Since $0 \notin I_c$, we have $|v_d(t) - a_d| < |v_d(t) + a_d|$ for all $t \in [t_0, t_0+h]$ by Corollary 6.1.2 (Radicand Corollary). Thus we get $b_d = v_d(t_0 + h)$. The inclusions $v_d([t_0, t_0 + h]) \subset I_d$ and $\dot{v}_d([t_0, t_0 + h]) \subset \dot{I}_d$ hold by Lemma 3.6.3.

After the computation of b_d in step 5, the values $b_{d+1}, \dots, b_n$ of the coordinates of B of the remaining variables are computed correctly in step 4.

Inductive Step: We assume that (6.4) holds for GSPs Γ with j root- or division operations. Now, let Γ have $j+1$ root or division operations. Let $v_d = v_{d,j+1}$ be the variable of Γ that is defined by the $j+1$st of these operations. Let $\tilde{\Gamma} = \Gamma^{(d-1)}$ be the $(d-1)$-head GSP of Γ. In other words, $\tilde{\Gamma}$ is the GSP after cutting Γ before the variable v_d; see Definition 3.2.1. Then, by induction, Algorithm 4 computes a step length h, and the correct final instance $B = (p_{-k+1}(t_0 + h), \dots, p_0(t_0 + h), v_1(t_0 + h), \dots, v_{d-1}(t_0 + h))$.

Let $v_c = v_{c,j+1}$ be the radicand or divisor variable of $v_d = v_{d,j+1}$ and $I_c = I_{c,j+1}$ the corresponding interval from Algorithm 4. If $0 \notin I_c$ in step 5, then $b_d = b_{d,j+1}$ is computed correctly following the arguments from above. If $0 \in I_c$, then in step 5 the algorithm is restarted with a smaller step length h. Here, h is chosen, such that in the next run is guaranteed in step 5 that $0 \notin I_c$; see Theorem 6.3.3. This also remains true for the preceding root or division variables since their corresponding intervals of their radicands or divisors at most become smaller since we use an interval arithmetic with the inclusion monotonicity property. Thus, all intervals $\dot{I}_i$ and I_i are computed correctly by Lemma 3.6.3 and Lemma 6.3.1. Additionally, the coordinates $b_1, \dots, b_d$ of B are correct since we can apply Corollary 6.1.2 for all root variables. The remaining coordinates of B are correctly computed in step 4.

Hence, Algorithm 4 computes for Γ a step length h and the correct final instance B for the step of length h.

Finally, we investigate the runtime of Algorithm 4. The restart in step 5 of Algorithm 4 is executed at most once for each root or division operation of Γ; see Theorem 6.3.3. Step 1 and step 6 need constant time, whereas step 2 and step 3 need $O(n+k)$ time if we assume that an interval operation has constant cost; see page 6, Chapter 1. The costs of step 4 are linear for each restart and the costs of step 5 except the costs for the restart and goto are constant. Hence, the total costs are in $O((n+k)n_0)$ as claimed. □

Algorithm 4 performs a single tracing step. Algorithm 5 traces the whole continuous evaluation along the paths $p_{-k+1}, \dots, p_0$ starting at A if it exists. Algorithm 5 is similar to Algorithm 3 from page 76, but it uses Algorithm 4 (Cone Algorithm) instead of Algorithm 2 for the tracing steps. Again, the final instance of a preceding step is the starting instance of the next step.

Algorithm 5.

Input: GSP Γ, linear paths $p_{-k+1}, \ldots, p_0$,
starting instance $A = (p_{-k+1}(0), \ldots, p_0(0), a_1, \ldots, a_n)$.

Output: final instance $B = (p_{-k+1}(1), \ldots, p_0(1), b_1, \ldots, b_n)$.

1. $\tilde{A} := A$ // starting instance
   $t := 0$ // starting time
2. **while** $t < 1$ **do**
   Run Alg. 4 with starting time $t_0 := t$ and starting instance
   $A := \tilde{A}$, the given GSP Γ and the given paths $p_{-k+1}, \ldots, p_0$
   // the step length h and the final instance B are returned.
   $\tilde{A} := B$ // B is the starting instance for the next step
   $t := t + h$
3. **return** $\tilde{A}$.

Remark 6.3.6. *(Compare with Remark 6.2.2 from page 77.)*
If we have a critical point along the paths, then Algorithm 5 does not terminate. The step length computed by Algorithm 4 in step 2 of Algorithm 5 becomes arbitrary small before the critical point is reached. This happens since the interval I_c in step 4 and step 5 of Algorithm 4 has to be arbitrary small in a neighborhood of a critical point. At a critical point, we have $0 = v_c \in I_c$. However, in step 5 is supposed that $0 \notin I_c$ in order to continue with the next variable. We discuss in Chapter 7 how critical points can be treated.

Theorem 6.3.7. *Algorithm 5 terminates and computes the correct final instance B as long as no critical point occurs.*

Proof. Induction on the number j of root-and division-operations of Γ:
This proof has a similar structure as the proof of Theorem 6.2.3 on page 77. Since critical points are excluded, there exists a unique continuous evaluation along the paths $p_{-k+1}, \ldots, p_0$ starting at the instance A. We prove by induction on j that each step has at least step length h for a fixed $h > 0$. Furthermore, h can be chosen such that it is a minimum step length for every starting time $t_0 \in [0, 1)$.

Basic Step: Let v_{d_1} be defined by the first root or division operation of Γ, i.e., $v_{d_1} = v_j / v_c$ or $v_{d_1} = \sqrt{v_c}$, and let $v_{d_1}(t)$ be its path in the continuous evaluation. Let $v_c(t)$ be the path of v_c in the continuous evaluation. We consider the entire time interval $[0, 1]$ and follow Algorithm 4 through step 4. We get the intervals $\dot{I}_c$ and $I_c = a_c + T_h \dot{I}_c$ of the radicand or divisor variable v_c. Both intervals are bounded since the variable v_{d_1} is the first variable that is defined by a root or division operation, and hence, no division by zero occurs.

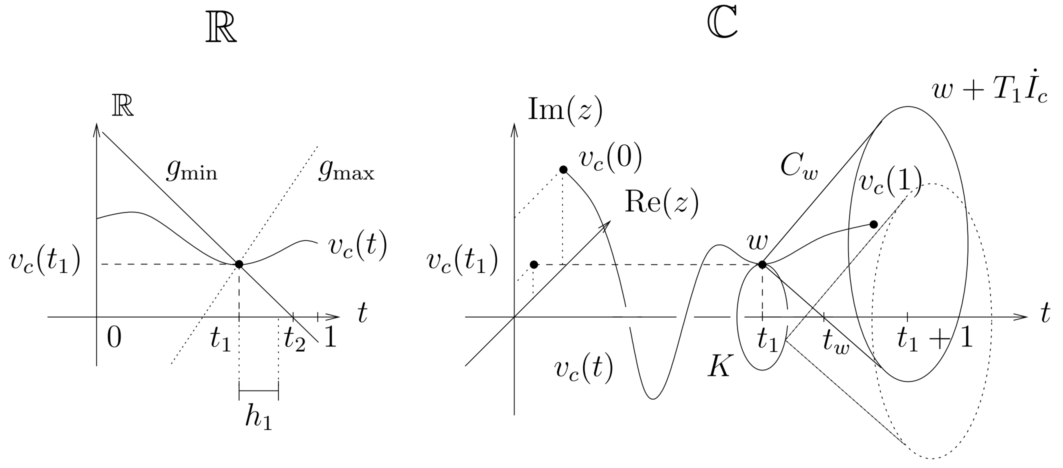


Figure 6.8: The construction of t_1 and $h_1 = \frac{2}{3}(t_2 - t_1)$ from the proof of Theorem 6.3.7 is illustrated: The left figure shows the real situation, and the right one shows the complex situation with the circle K and a cone C_w.

Since critical points are excluded, the path $v_c(t)$ does not pass through 0. Additionally, the time interval $[0, 1]$ is compact, and the mapping $t \mapsto |v_c(t)|$ is continuous. Hence $\delta := \min_{t\in[0,1]} |v_c(t)| > 0$. Let $t_1 \in [0, 1]$ with $|v_c(t_1)| = \delta$. We treat the real and the complex case separately:

$\mathbb{R}$: Since $v_c(t)$ does not pass through $0 \in \mathbb{R}$, either $v_c(t) \geq \delta$ for all $t \in [0, 1] = T_1$, or $v_c(t) \leq -\delta$ holds for all $t \in [0, 1]$. W.l.o.g. we assume $v_c(t) \geq \delta$ and look at the line $g_{\min}(t) = m_{\min}t + \delta - m_{\min}t_1$ with $m_{\min} = \min T_1\dot{I}_c$. This line passes through the point (t_1, δ) on the graph of $v_c(t)$ and has $m_{\min}$ as slope; compare with Lemma 6.3.1 and see Figure 6.8. According to Algorithm 4, let C be the cone with apex (t_1, δ) and base $(t_1 + 1, \delta + T_1\dot{I}_c)$. Hence, C is the triangle with the vertices (t_1, δ), $(t_1+1, \delta+m_{\min})$, and $(t_1+1, \delta+m_{\max})$ with $m_{\max} := \max T_1\dot{I}_c$. The lower edge of this triangle lies on the line $g_{\min}$.

If $0 \notin I_c$ we can choose $h := 1$ and we are done. In the other case, we observe that the interval $\dot{I}_c$ from our proof has to contain every interval $\dot{I}_c =: \dot{I}_{c_A}$ that is computed while Algorithm 5 runs. This inclusion holds due to the inclusion monotonicity property of the real interval arithmetic since $[t_0, t_0 + h] \subset [0, 1]$ holds for $0 \leq t_0, h \leq 1$ with $t_0 + h \leq 1$; see proof of Theorem 6.3.4 for a more detailed argumentation in a similar situation. Denote by C_A the cone of the divisor- or radicand-variable v_c from step 4 of a run of Algorithm 4 with the time interval $[t_0, t_0 + h]$.

Let $(t_2, 0)$ be the intersection point of $g_{\min}$ and the t-axis with $t_2 \in (t_1, t_1+1]$. Hence, t_2 is the earliest time such that $(t_2, 0)$ lies on the lateral surface of the cone C. We set $h_1 := \frac{2}{3}(t_2 - t_1)$. Let t' be the smallest time such that

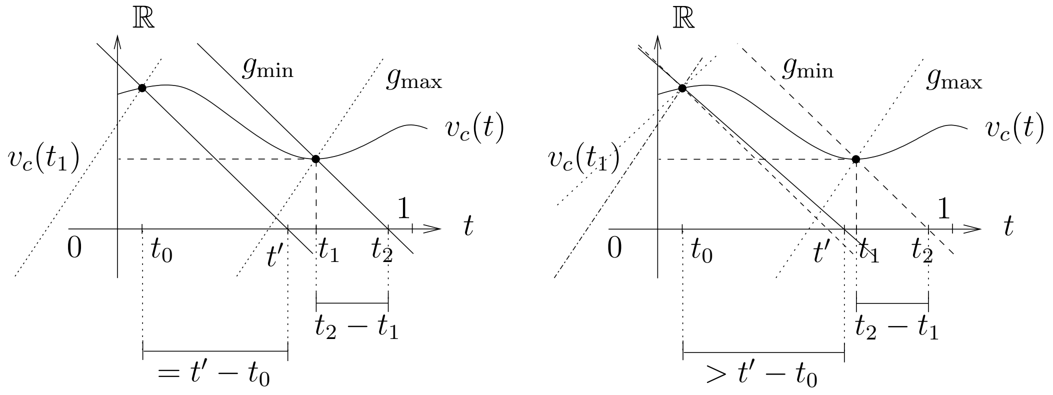


Figure 6.9: The proof of $t_2 - t_1 \leq t' - t_0$ is illustrated. In the left figure we have $\dot{I}_{c_A} = \dot{I}_c$, in the right one $\dot{I}_{c_A} \subsetneqq \dot{I}_c$ holds.

$(t', 0)$ lies on the lateral surface of the cone C_A as in step 5 of Algorithm 5. Since $v_c(t_0) \geq \delta$ and $\dot{I}_{c_A} \subset \dot{I}_c$, we have $t_2 - t_1 \leq t' - t_0$. Hence, $h \geq h_1$ holds by construction of δ, t_1 and h_1; see Figure 6.9. In the last step, the step length is at least $\min\{h_1, 1 - t_0\}$.

$\mathbb{C}$: Here, the situation and the construction of h_1 is slightly more complicated since the graph of $v_c(t)$ might wind around the t-axis.

We consider the circle K with center $(t_1, 0) \in [0, 1] \times \mathbb{C}$ and radius $\delta = |v_c(t_1)|$ lying in the plane that is orthogonal to the t-axis and that contains the point $(t_1, 0)$. Let w be a point of the boundary of K, and let C_w be the cone having w as apex and $(t_1 + 1, T_1\dot{I}_c + w) = \{(t_1 + 1, z) | z \in T_1\dot{I}_c + w\}$ as base; see right of Figure 6.8. Recall that T_1 is the smallest interval containing $[0, 1]$ in the used interval arithmetic. We define the set of cones $\mathcal{C}_K := \{\text{cone } C_w |\ w \text{ is on the boundary of } K\}$.

If none of the cones $C_w \in \mathcal{C}_K$ intersects the t-axis, then $0 \notin I_c$ always holds for the variable v_{d_1} in step 5 of Algorithm 4, and we can choose $h := 1$. Now, we investigate the situation where at least one of the cones C_w intersects the t-axis. The subset $\tilde{K} := \{w \in K |\ C_w \text{ intersects the } t\text{-axis}\}$ is compact, and for $w \in \tilde{K}$ let t_w be the time of the first intersection point of C_w and the t-axis. Since the mapping $\tilde{K} \ni w \mapsto t_w$ is continuous, it takes its minimum $t_2 \in [0, 1]$. The construction of t_2 ensures that t_2 is independent of the winding of $v_c(t)$ around the t-axis.

As in the real case ($\mathbb{R}$), we set $h_1 := \frac{2}{3}(t_2 - t_1)$ and claim that the step length h computed in step 5 of Algorithm 4 is at least h_1. We can show in the same way as for the real case that $h \geq h_1$ holds by construction of δ, t_1 and h_1. In the last step, the step length is at least $\min\{h_1, 1 - t_0\}$.

Inductive Step: The inductive step can be done with the same arguments as the inductive step of the proof of Theorem 6.2.3. Nevertheless, we state it again.

Let Γ be a GSP with $j+1$ root or division operations and consider the $j+1$st root or division operation of Γ and its corresponding variable $v_{d_{j+1}}$. Let $\tilde{\Gamma} = \Gamma^{(d_{j+1}-1)}$ be the $(d_{j+1}-1)$-head GSP of Γ. That is, $\tilde{\Gamma}$ is the GSP after cutting Γ before the variable $v_{d_{j+1}}$; see Definition 3.2.1. Since $\tilde{\Gamma}$ has j root or division operations, we can apply the induction hypothesis. Let h_j be a minimum step length for $\tilde{\Gamma}$ that is independent of the starting time t_0.

We partition the interval $[0,1]$ into $\lceil \frac{1}{h_j} \rceil$ intervals of length h_j at most. Since h_j is independent of the starting time t_0, we know that for each of these subintervals Algorithm 5 does not reduce the size of these time intervals for the first j root or division operations. This holds since we have chosen an interval arithmetic with the inclusion monotonicity property. Hence, for each of the subintervals, we have the same situation as for the first root or division operation and the entire time interval $[0,1]$. Consequently, the inductive conclusion can be shown in the same way as the basic step: For each subinterval $[lh_j, (l+1)h_j] \subset [0,1]$ we get a step length $h_{j,l}$. Finally, $h_{j+1} := \min\{h_{j,l} \mid l = 1,2,3,\ldots,\lceil \frac{1}{h_j} \rceil\}$ is the wanted step length. In the last step, the step length is at least $\min\{h_{j+1}, 1-t_0\}$. $\square$

6.3.2 Computation of the First Intersection of the Cone C with the t-Axis

In step 5 of Algorithm 4, the first intersection point $(t',0)$ of the cone C and the t-axis is needed. Recall that the cone C has the point (t_0, a_c) as apex and the set (t_0+h, I_c) as base. We give an outline of the computation steps. Recall that the variable v_c is the divisor or radicand of v_d.

If Γ is a real GSP, then $I_c = [a,b]$ for some $a \leq b \in \mathbb{R}$, and C is bounded by the lines $g_{\min}(t) = a_c + \frac{t-t_0}{h}(a-a_c)$ and $g_{\max}(t) = a_c + \frac{t-t_0}{h}(b-a_c)$. If $a_c > 0$, then the first intersection point of C and the t-axis is the intersection point of the line $g_{\min}$ and the t-axis -as far as the corresponding time lies in the time interval $[t_0, t_0+h]$. If $a_c < 0$ we consider the intersection point of $g_{\max}$ and the t-axis. The case $a_c = 0$ does not occur, since this would cause a d-critical point for the variable v_d, and A could not be an instance.

If Γ is a complex GSP, we distinguish between rectangular and circular interval arithmetic. For rectangular arithmetic, the situation is similar to the real one since we can treat the real and imaginary parts separately. Here, $I_c = a_c + ([a_1,b_1] + i[a_2,b_2])$. The cone C intersects the t-axis if and only if one of the four triangular surfaces of the boundary intersects the t-axis. Thus, the wanted intersection occurs at the earliest time $t' \in [t_0, t_0+h]$ with $0 \in \mathrm{Re}(a_c) + \frac{t'-t_0}{h}[a_1,b_1]$

and $0 \in \mathrm{Im}(a_c) + \frac{t'-t_0}{h}[a_2, b_2]$. Hence, the time t' can be computed as in the real situation.

For circular interval arithmetic, the situation is slightly more complicated. For simplicity of notation, we assume that the time interval is $[0,1]$. Let $I_c = \{m_c; r_c\}$, and let $q = m_c + r_c e^{i\phi}$ be a point on the boundary of $I = \{m_c; r_c\}$. Then $(t, a_c + t(q - a_c)) = (t, a_c + t(m_c - a_c) + tre^{i\phi})$ is a point on the lateral surface of C. Hence, for fixed $t \in [0,1]$, the corresponding points of the surface of C lie on the boundary of the circle $C_t := \{a_c + t(m_c - a_c)\,;\, tr_c\}$. We remark that the midpoint $a_c + t(m_c - a_c)$ of C_t is the corresponding point on the line $l(t) = a_c + t(m_c - a_c)$ connecting the points $(0, a_c)$ and $(1, m_c)$. The circle C_t is uniquely determined by the three points

$$\begin{array}{lllllll} q_1 & = & x_1 + iy_1 & := & a_c + t(m_c - a_c) + tr_c e^{i0} & = & a_c + t(m_c - a_c) + tr_c; \\ q_2 & = & x_2 + iy_2 & := & a_c + t(m_c - a_c) + tr_c e^{i\frac{\pi}{2}} & = & a_c + t(m_c - a_c) + tr_c i; \text{ and} \\ q_3 & = & x_3 + iy_3 & := & a_c + t(m_c - a_c) + tr_c e^{i\pi} & = & a_c + t(m_c - a_c) - tr_c. \end{array}$$

A point $q = x + iy$ lies on the boundary of the circle C_t if and only if

$$f(x,y,t) := \det \begin{pmatrix} x_1 & x_2 & x_3 & x \\ y_1 & y_2 & y_3 & y \\ x_1^2 + y_1^2 & x_2^2 + y_2^2 & x_3^2 + y_3^2 & x^2 + y^2 \\ 1 & 1 & 1 & 1 \end{pmatrix} = 0. \tag{6.5}$$

Since we are looking for an intersection point with the t-axis, we consider the point $q = 0 = 0 + 0i$. Equation (6.5), applied to the points q_1, q_2, q_3 and q, leads to the following equation with $m_c = c_1 + ic_2$.

$$\begin{aligned} 0 = f(0,0,t) = -2t^2 r_c^2 \Big(& t^2 \left((\mathrm{Re}(a_c) - c_1)^2 + (\mathrm{Im}(a_c) - c_2)^2 - r_c^2 \right) \\ & + t \;\; 2 \left(\mathrm{Im}(a_c)\, c_2 - (\mathrm{Re}(a_c))^2 - (\mathrm{Im}(a_c))^2 + \mathrm{Re}(a_c)\, c_1 \right) \\ & + \;\; (\mathrm{Re}(a_c))^2 + (\mathrm{Im}(a_c))^2 \Big) \end{aligned}$$

The cone C intersects the t-axis if and only if the quadratic equation $f(t) := f(0,0,t) = 0$ has a solution $t' \in [0,1]$. In this case, the smallest solution $t' \in [0,1]$ leads to the wanted intersection point $(t', 0)$ of the t-axis with the cone C.

6.3.3 Improvements of the Cone Algorithm

We describe some improvements of the Cone Algorithm (Algorithm 4) from page 81. The aim of these improvements is to reduce the size of the intervals I_i for $i \in \{1, \ldots, n\}$. This reduction might lead to less restarts in step 5 of Algorithm 4 and could improve the runtime.

Improvement 6.3.8. The usage of $\mathbf{T_h}$ is not always necessary.
The definition $I_i := a_i + T_h \dot{I}_i$ for $i \in \{1, \ldots, n\}$ in step 4 and step 5 of Algorithm 4 for the operations $+$, $-$, $\cdot$, and $:$ ensures that the range of the path $v_i(t)\colon [t_0, t_0 + h] \to \mathbb{K}$ is contained in I_i. By the Cone Lemma 6.3.1 from page 79, we have $v_i(t) \in a_i + (t - t_0)\dot{I}_i \subset a_i + T_h \dot{I}_i = I_i$ for $t \in [t_0, t_0 + h] \subset T_h$.

To choose $I_i := a_i + h\dot{I}_i$ is not sufficient. In this case, the range of the path $v_i(t)\colon [t_0, t_0+h] \to \mathbb{K}$ is contained in the cone with apex (t_0, a_i) and base (t_0+h, I_i) by the Cone Lemma 6.3.1. Unfortunately, the range of $v_i(t)$ is not contained in $a_i + h\dot{I}_i$, in general. But, we have $v_i(t) \in a_i + h\dot{I}_i$ for all $t \in [t_0, t_0 + h]$, if $a_i + \tilde{h}\dot{I}_i \subset a_i + h\dot{I}_i = I_i$ holds for all $\tilde{h} \in (0, h)$. The second assertion is equivalent to $0 \in \dot{I}_i$ as Lemma 6.3.9 states.

Lemma 6.3.9. *We have $a_i + \tilde{h}\dot{I}_i \subset a_i + h\dot{I}_i$ for all $\tilde{h} \in (0, h)$ if and only if $0 \in \dot{I}_i$.*

Proof. We consider the cases $\mathbb{K} = \mathbb{R}$ and $\mathbb{K} = \mathbb{C}$ separately:

$\mathbb{R}$**:** If $0 \in \dot{I}_i = [m_{\min}, m_{\max}]$, then $m_{\min} \le 0 \le m_{\max}$ holds, and $0 < \tilde{h} < h$ implies $hm_{\min} \le \tilde{h}m_{\min} \le \tilde{h}m_{\max} \le hm_{\max}$. Hence $\tilde{h}\dot{I}_i \subset h\dot{I}_i$ and consequently $a_i + \tilde{h}\dot{I}_i \subset a_i + h\dot{I}_i$.
If conversely $a_i + \tilde{h}\dot{I}_i \subset a_i + h\dot{I}_i$ with $0 < \tilde{h} < h$ holds, then $hm_{\min} \le \tilde{h}m_{\min} \le \tilde{h}m_{\max} \le hm_{\max}$. This chain of inequalities only holds if $m_{\min} \le 0$ and $m_{\max} \ge 0$.

$\mathbb{C}$**:** The proof for rectangular arithmetic can directly be reduced to the real case since $0 \in A = A_1 + iA_2$ if and only if $0 \in A_1$ and $0 \in A_2$.
For circular arithmetic, let $\dot{I}_i = \{c_i; r_i\}$. Then we have $\tilde{h}\dot{I}_i = \{\tilde{h}c_i; \tilde{h}r_i\}$ and $h\dot{I}_i = \{hc_i; hr_i\}$ for $0 < \tilde{h} < h$.
We have $0 \in \dot{I}_i$ if and only if $|c_i| - r_i \le 0$. Since $0 < \tilde{h} < h$, we have

$$\begin{array}{rrcl} & h(|c_i| - r_i) & \le & \tilde{h}(|c_i| - r_i) \\ \Leftrightarrow & (h - \tilde{h})\,|c_i| + \tilde{h}r_i & \le & hr_i \\ \Leftrightarrow & |hc_i - \tilde{h}c_i| + \tilde{h}r_i & \le & hr_i \\ \Leftrightarrow & \tilde{h}\dot{I}_i & \subset & h\dot{I}_i; \quad \text{see Figure 6.10.} \end{array}$$

□

Improvement 6.3.10. Use the Information of $\mathbf{\Gamma_{int}}$ as well.
We reduce the size of the intervals I_i and of the cone C by using the information of Γ_{int} as well. This improvement can only be used for real GSPs, or for complex GSPs in combination with rectangular interval arithmetic. Here, the intersection of two intervals is again an interval in contrast to circular interval arithmetic.

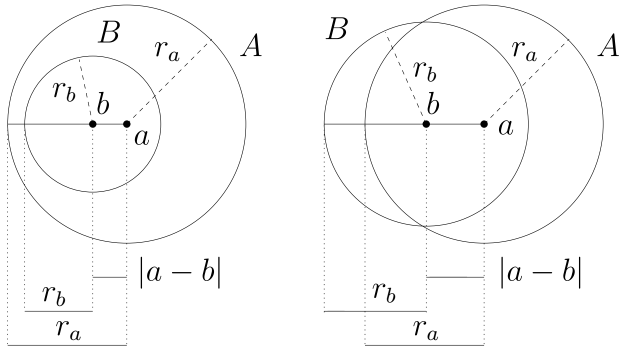


Figure 6.10: Two circles $A = \{a; r_a\}$ and $B = \{b; r_b\}$ are considered. We have $B \subset A$ if and only if $|a - b| + r_b \leq r_a$ holds.

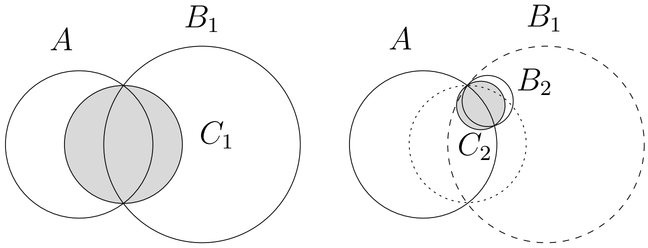


Figure 6.11: The circle C_1 is the smallest circle containing $A \cap B_1$, C_2 is the smallest circle containing $A \cap B_2$. We have $B_2 \subset B_1$ but not $C_2 \subset C_1$.

In step 4 and step 5 of Algorithm 4, we set

$$I_i := I_i \cap \left(a_i + T_h \dot{I}_i\right) \tag{6.6}$$

for the operations $+$, $-$, $\cdot$, $:$ and $\sqrt{\_}$. The interval I_i of the right hand side of (6.6) is the one that is computed by Γ_{int}. The interval I_i of the left hand side of (6.6) is used by Algorithm 4 for further computations. The resulting algorithm is somehow a combination of Algorithm 2 and Algorithm 4.

Remark 6.3.11. Improvement 6.3.10 does not work for circular interval arithmetic since here, the intersection of two intervals is not an interval, in general. Alternatively, we could choose the smallest circle containing the intersection of two intervals, but then, the inclusion monotonicity property gets lost (see Figure 6.11). This property is crucial for the correctness proofs of Algorithm 2 and Algorithm 4.

Improvement 6.3.12. Use the Backward Cone as well.
Like Improvement 6.3.10, Improvement 6.3.12 only works for real GSPs, and for

complex GSPs in combination with rectangular interval arithmetic. Here, the intersection of two intervals is again an interval.
The key idea of Improvement 6.3.12 is to apply the Cone Lemma 6.3.1 from page 79 to the paths $v_i(-t)$. This gives additional information that reduces the size of the interval I_i in the Cone Algorithm (Algorithm 4). We formulate the idea using the notation from the Cone Lemma.

Corollary 6.3.13. $\mathbb{R}$: *Let* $p\colon [0,1] \to \mathbb{R}$ *be a differentiable path and* $p_1 := p(1)$. *Furthermore, let* $m_{\min} \le \dot{p}(t) \le m_{\max}$ *for all* $t \in [0,1]$. *Let*

$$\begin{aligned} f_{\min}(t) &= t\,m_{\max} + p_1 - m_{\max}; \quad \textit{and} \\ f_{\max}(t) &= t\,m_{\min} + p_1 - m_{\min} \end{aligned}$$

be the lines passing through the point $(1, p_1)$ *and having* $m_{\max}$ *and* $m_{\min}$ *as slopes. Then,* $f_{\min}(t) \le p(1-t) \le f_{\max}(t)$ *holds for all* $t \in [0,1]$.

If the time interval $[0,1]$ *is replaced by an arbitrary interval* $[h_1, h_2]$, *then we get* $p_1 := p(h_2)$, $m_{\min} \le \dot{p}(t) \le m_{\max}$ *for all* $t \in [h_1, h_2]$, $f_{\min}(t) = t\,m_{\max} + p_1 - h_2 m_{\max}$, *and* $f_{\max}(t) = t\,m_{\min} + p_1 - h_2 m_{\min}$. *Thus* $p(t) \in (t-h_2)[m_{\min}, m_{\max}] + p_1$.

$\mathbb{C}$: *Let* $p\colon [0,1] \to \mathbb{C}$ *be a continuously differentiable path and* $p_1 := p(1)$. *Furthermore, let* $J \subset \mathbb{C}$ *be a complex interval with* $\dot{p}(t) \in J$ *for all* $t \in [0,1]$. *Then,* $p(t) \in (t-1)J + p_1 = \{(t-1)z + p_1 | z \in J\}$ *holds for all* $t \in [0,1]$.

If the time interval $[0,1]$ *is replaced by an arbitrary interval* $[h_1, h_2]$, *then we get* $p_1 := p(h_2)$, *and* $p(t) \in (t-h_2)J + p_1$.

Proof. Let J be a (real or complex) interval with $\dot{p}([0,1]) \subset J$, and let $p_1 := p(1)$. Furthermore, let $q(t) := p(1-t)$ for $t \in [0,1]$. Hence $\dot{q}(t) = -\dot{p}(1-t) \in -J$ for $t \in [0,1]$. By the Cone Lemma, we get $p(1-t) = q(t) \in -tJ + q(0) = -tJ + p_1$, and consequently, $p(t) \in (t-1)J + p_1$.

If we consider the time interval $[h_1, h_2]$, we set $q(t) := p(h_2 - t)$ for $t \in [0, h_2 - h_1]$. Then $\dot{q}(t) = -\dot{p}(h_2 - t)$ for $t \in [0, h_2 - h_1]$ and $p(h_2 - t) = q(t) \in t(-J) + q(0) = -tJ + p(h_2) = -tJ + p_1$ hold by the Cone Lemma. Hence $p(t) \in (h_2 - t)(-J) + p_1 = (t-h_2)J + p_1$.

If $J = [m_{\min}, m_{\max}]$ in the real situation, we get the corresponding formulas for the lines $f_{\min}$ and $f_{\max}$. □

If we combine the Cone Lemma 6.3.1 with Corollary 6.3.13, we get $p(t) \in ((t-h_1)J + p_0) \cap ((t-h_2)J + p_1)$. This directly leads to an improvement of the Cone Algorithm (Algorithm 4). In step 4 and step 5, we consider the intervals $I_i := (a_i + T_h \dot{I}_i) \cap (b_c - T_h \dot{I}_c)$ instead of $a_i + T_h \dot{I}_i$, which might reduce the number of restarts in step 5. Alternatively, we can consider the two cones $C = C_l$ and C_r

with apex (t_0, a_i) and $(t_0 + h, b_i)$ and base $(t_0 + h, a_i + T_h\dot{I}_i)$ and $(t_0, b_i + T_h\dot{I}_i)$, respectively. By Lemma 6.3.1 and Corollary 6.3.13, we have $v_i(t) \in C_r \cap C_l$ for all $t \in [t_0, t_0 + h]$. As before, $v_i(t)$ is the path of the considered variable v_i in the continuous evaluation. We can chose I_i as the smallest interval containing the projection of $C_r \cap C_l$ along the t-axis to the second coordinate, which is an element of $\mathbb{K}$.

Since this improvement uses the intersection of intervals, it does not work with circular complex interval arithmetic; see Remark 6.3.11.

6.4 The Problem of Overestimation

We discuss the problem of overestimation in Algorithm 2 and Algorithm 4. We have seen in Example 3.6.5 that interval dependency leads to an overestimation of the exact range of the algebraic function described by a GSP Γ. This overestimation depends heavily on the structure of the underlying GSP. An overestimation of the ranges of the algebraic functions that correspond to the dependent variables of Γ may lead to an underestimation of the step length computed in Algorithm 2 and Algorithm 4.

We compare the intervals I_i of the dependent variables v_i, $i = 1, \dots, n$, computed by Algorithm 2 or Algorithm 4 with the image $v_i([t_0, t_0 + h])$ of the corresponding paths in the continuous evaluation. We focus on the real situation ($\mathbb{K} = \mathbb{R}$). At the end of this section, we indicate how the complex case ($\mathbb{K} = \mathbb{C}$) could be treated, and we point out some difficulties for measuring the overestimation in complex interval arithmetic.

The Real Situation ($\mathbb{K} = \mathbb{R}$). As in the specifications of Algorithms 2 and 4, let A be a starting instance of a GSP Γ at time t_0, and let $p_{-k+1}, \dots, p_0 \colon [0, 1] \to \mathbb{R}$ be linear paths of the free variables of Γ. We assume that the corresponding continuous evaluation $(v_1(t), \dots, v_n(t))$ exists. By definition, every path $v_i(t)$ can be described as an expression involving the four elementary arithmetic operations and square roots by following the operations of Γ. Algorithm 2 does the same computations in real interval arithmetic in which for every square root operation the exact range is computed. Thus, the interval I_i in step 4 or step 5 of Algorithm 2 describes a natural interval extension $\mathbf{v_i}$ of the function $t \mapsto v_i(t)$; we have $I_i = \mathbf{v_i}([t_0, t_0 + h])$.

Theorem A.4.3 from Appendix A.4 implies that the interval extensions $\mathbf{v_i}$ of the functions $t \mapsto v_i(t)$ computed by Algorithm 2 are of first order, $i = 1, \dots, n$. Thus, for every dependent variable v_i there is a constant K_i with $\mathrm{w}(I_i) - \mathrm{w}(v_i([t_0, t_0 + h])) = \mathrm{w}(\mathbf{v_i}([t_0, t_0 + h])) - \mathrm{w}(v_i([t_0, t_0 + h])) \le K_i \mathrm{w}([t_0, t_0 + h]) = hK_i$. Note that the constant $K = K_i$ coming from the definition of the order of an interval extension (Definition A.4.2) depends on $i \leq n$ and the structure of the GSP Γ.

Now, we consider the Cone Algorithm (Algorithm 4). We observe that the intervals I_i computed by Algorithm 4 are mean value extensions $\mathbf{v_{i,2}}$ of the paths $t \mapsto v_i(t)$ from the continuous evaluation; see Definition A.4.4. These mean value extensions are centered at t_0. The intervals $\dot{I}_i$ are natural interval extensions $\dot{\mathbf{v}}_\mathbf{i}$ of the derivatives $\dot{v}_i(t)$, and we have $\mathbf{v_{i,2}}([t_0, t_0 + h]) = v_i(t_0) + \dot{\mathbf{v}}_\mathbf{i}([t_0, t_0 + h]) \cdot ([t_0, t_0 + h] - t_0) = a_i + \dot{I}_i \cdot [0, h] = a_i + T_h \dot{I}_i$. By Theorem A.4.5, the interval extensions $\mathbf{v_{i,2}}$ have order 2. Thus, for every dependent variable v_i there is a constant $K_{i,2}$ with $\mathrm{w}(I_i) - \mathrm{w}(v_i([t_0, t_0 + h])) \leq h^2 K_{i,2}$. As with Algorithm 2, the constant $K_2 = K_{i,2}$ from Definition A.4.2 of the interval extension computed by Algorithm 4 depends on $i \leq n$, the number k of free variables, and the structure of the GSP Γ.

Our observations show that the overestimation of the ranges of the paths $v_i(t)$ by the enclosing intervals I_i computed by Algorithm 2 or Algorithm 4 decreases if the step length h decreases and if h is small enough. The overestimation caused by Algorithm 4 should be smaller than the overestimation caused by Algorithm 2 if h is small. Thus, we expect that the step length computed by Algorithm 4 could be larger than the step length computed by Algorithm 2. This is an advantage of Algorithm 4 over Algorithm 2 in addition to the more efficient step length adaptation.

The Complex Situation ($\mathbb{K} = \mathbb{C}$): The complex situation is more complicated than the real situation. First, the range of the square root operation cannot be computed exactly since the square root of a complex interval is not an interval, in general. But, the exact computation of the interval square root operation is a requirement in the definition of natural interval extensions in Definition A.4.1. Thus in this case, we cannot reduce rectangular complex arithmetic to real interval arithmetic by treating the real and the imaginary part separately.

Interval extensions for circular interval arithmetic are treated in [57]. Here, Petković and Petković remark that it is not possible to show that the circular interval extension $\mathbf{F}(\mathbf{A})$ of a function F always includes the smallest disc containing the image $F(\mathbf{A})$; see [57, p. 41]. The results of [57, Sect. 2] imply that the interval extensions computed by Algorithm 2 or Algorithm 4 are of order two as long as no square root operations occur and if circular arithmetic is used. Additionally, Petković and Petković mention in [57, p. 47] that higher order forms improve the interval extensions. Thus we expect that Algorithm 4 gives better results for a small step length h than Algorithm 2 as in the real situation $\mathbb{K} = \mathbb{R}$.

6.5 Robustness

We have seen that Algorithms 3 and 5 compute the correct final instance if we assume exact computation and if no critical points lie on the traced continuous evaluation. Critical points are investigated in Chapter 7, but what happens if we use floating point arithmetic instead of exact computation? For the interval computations we can use rounded interval arithmetic, which is explained in Appendix A.3. If we use floating point arithmetic for all computations, then we have to deal with the following problems in Algorithms 2 and 4:

1. The coordinates $a_{-k+1}, \ldots, a_n$ of the starting instance A are represented as floating point numbers, as well. Thus the point $A = (a_{-k+1}, \ldots, a_n)$ is not an instance in general, but we can assume that A is close to an instance.

2. The coordinates $b_{-k+1}, \ldots, b_n$ are computed with floating point arithmetic, thus the resulting point $B = (b_{-k+1}, \ldots, b_n)$ will only be close to the desired final instance.

3. We have to show that, despite the rounding and truncation errors, we still have a_i, $b_i \in I_i$. In particular, we have to ensure that the choices from step 5 remain correct.

We denote a point $A = (a_{-k+1}, \ldots, a_n)$ a *floating point instance* of the GSP Γ, if the coordinates $a_1, \ldots, a_n$ are obtained from the values $a_{-k+1}, \ldots, a_0$ of the free variables of Γ by following the instructions of Γ using floating point arithmetic.

First, we have to define a correct starting instance $\tilde{A} = (\tilde{a}_{-k+1}, \ldots, \tilde{a}_0, \tilde{a}_1, \ldots, \tilde{a}_n)$. We observe that $\tilde{a}_l = p_l(t_0)$ is a natural choice for $l = -k+1, \ldots, 0$ since we want to trace along the paths $p_{-k+1}, \ldots, p_0$ and since t_0 is the start time. We choose $\tilde{a}_1, \ldots, \tilde{a}_n$ such that $\tilde{A}$ is the instance of the GSP Γ that is "close" to the point A. In the following paragraphs, we state the meaning of "close" precisely.

We evaluate the functions p_l at time t_0 with rounded interval arithmetic leading to the intervals P_l with a_l, $\tilde{a}_l \in P_l$; $l = -k+1, \ldots, 0$. Note that we can interpret a number as the interval that only contains this number. Following the instructions of Γ_{int} using rounded interval arithmetic, we compute intervals $P_1, \ldots, P_n$ with a_i, $\tilde{a}_i \in P_i$; $i = 1, \ldots, n$. Thus the resulting vector $\mathbf{P} = (P_{-k+1}, \ldots, P_n)$ of intervals is an interval-instance of Γ_{int} that contains an instance $\tilde{A}$ and the floating point instance A. We assume that the intervals that belong to radicand variables of square root operations do not contain zero. Otherwise we should increase the precision of the floating point numbers. Note that in this situation the point A might be close to a critical point and we could return to the previous step in the tracing process and continue with the methods from Chapter 7.

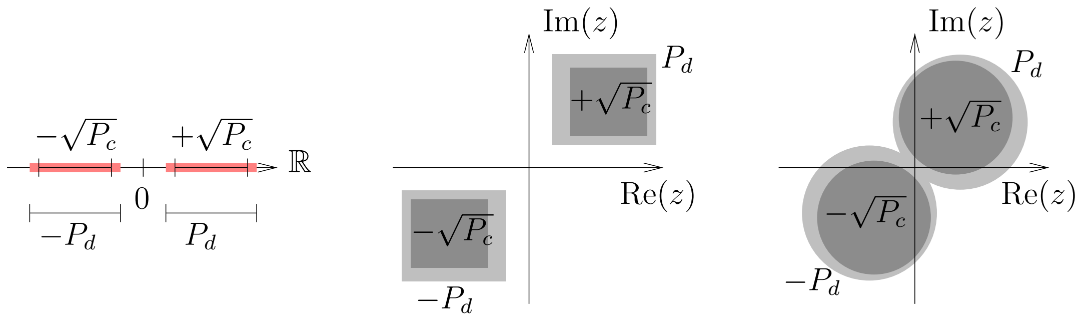


Figure 6.12: The intervals $P_d = \sqrt{I_c}$ are visualized. The smaller intervals are computed using exact arithmetic for determining the interval bounds, the bigger intervals are computed with rounded interval arithmetic. In rounded circular arithmetic, the intervals $+P_d$ and $-P_d$ can overlap. Here the radius of an interval computed by a square root operation must be chosen large enough to compensate the rounding errors of the midpoint calculation.

If another floating point instance $A' = (a_{-k+1}, \ldots, a_0, a'_1, \ldots, a'_n)$ lies above the point $(a_{-k+1}, \ldots, a_0)$, then we follow the instructions of Γ_{int} and compute intervals $P'_1, \ldots, P'_n$ with $a'_i \in P'_i$. If $A = A'$, then we have $a_i = a'_i$ for all $i = 1, \ldots, n$, and the intervals P_i and P'_i are computed by the same sequence of interval-operations, hence $P_i = P'_i$ holds for all $i = 1, \ldots, n$. The converse direction is also true: Let $A \neq A'$, then there is a smallest index $i_0 \in \{1, \ldots, n\}$ with $a_{i_0} \neq a'_{i_0}$. The only operation that might lead to different results on the same input is the square root operation. Thus the dependent variable v_{i_0} of Γ is defined by a square root operation $v_{i_0} = v_d \leftarrow \sqrt{v_c}$, and we have $a_{i_0} = a_d = -a'_d \neq 0$ and $P_d = -P'_d$. Since we assume $0 \notin P_c$, we have $0 \notin \sqrt{P_c}$ computed using exact computation; see Appendix A. Hence there must be a point $x_d \in P_d$ with $x_d \notin P'_d$, and $P_d \neq P'_d$; see Fig 6.12. Note that the intervals P_d and P'_d need not be disjoint due to the usage of rounded interval arithmetic.

If $A \neq A'$, then the vector $\mathbf{P}' = (P_{-k+1}, \ldots, P_0, P'_1, \ldots, P'_n)$ of intervals must contain an instance $\tilde{A}' = (\tilde{a}_{-k+1} = p_{-k+1}(t_0), \ldots, \tilde{a}_0 = p_0(t_0), \tilde{a}'_1, \ldots, \tilde{a}'_n)$ with $\tilde{a}'_d = -\tilde{a}_d \neq 0$. If $P_d \cap P'_d = \emptyset$ for the first index d with $P_d \neq P'_d$, then the instances $\tilde{A}$ and $\tilde{A}'$ are separated by the interval vectors $\mathbf{P}$ and $\mathbf{P}'$. If $\mathbf{P}$ separates $\tilde{A}$ from all other instances $\tilde{A}'$ from the fiber of the point $(\tilde{a}_{-k+1} = p_{-k+1}(t_0), \ldots, \tilde{a}_0 = p_0(t_0))$, then we say that the instance $\tilde{A}$ is close to A. In this situation, the instance $\tilde{A}$ is unique. If $P_d \neq P'_d$ and $P_d \cap P'_d \neq \emptyset$, then the interval vector $\mathbf{P} = (P_{-k+1}, \ldots, P_0, P_1, \ldots, P_n)$ might contain another instance $\tilde{A}' = (\tilde{a}_{-k+1}, \ldots, \tilde{a}_0, \tilde{a}'_1, \ldots, \tilde{a}'_n)$ from the fiber of the point $(\tilde{a}_{-k+1}, \ldots, \tilde{a}_0)$; see Figure 6.13. In this situation, the point A could be close to a critical point, and we might use the methods from Chapter 7. Another possibility to overcome this problem would be to increase the precision of the floating point numbers.

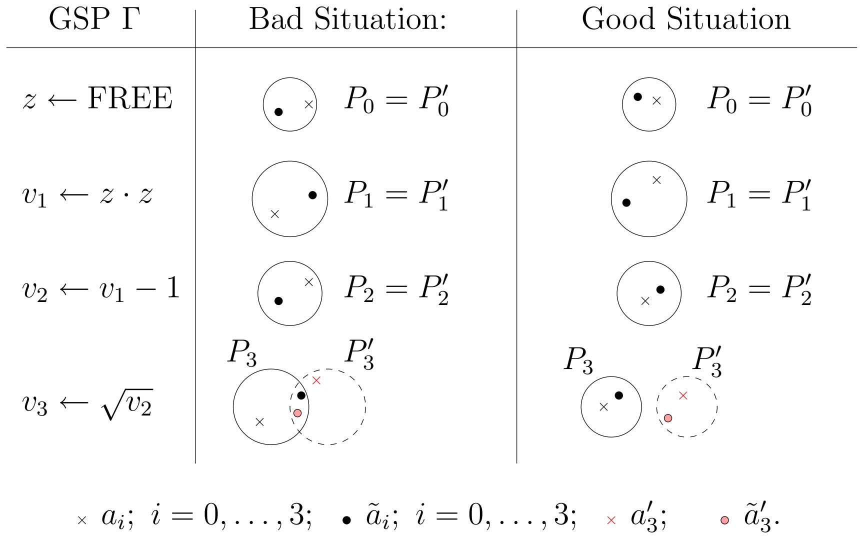


Figure 6.13: We consider the GSP Γ from Example 3.1.3. The intervals $P_d = P_3$ and $P_d' = P_3'$ are shown in a "bad" situation ($P_d \cap P_d' \neq \emptyset$) and in a "good" situation ($P_d \cap P_d' = \emptyset$). In the bad situation, the two solutions $\tilde{a}_3$ and $\tilde{a}_3'$ are not separated by the intervals P_3 and P_3'.

Similar considerations apply to the computed final instance B. Let $\tilde{B}$ be the correct final instance for a tracing step of length h if as starting instance $\tilde{A}$ was chosen. Then we have $b_l = p_l(t_0 + h)$ computed in floating point arithmetic and $\tilde{b}_l = p_l(t_0 + h)$ computed in exact arithmetic; $l = -k+1, \ldots, 0$. Then B should be the floating point instance that is closest to $\tilde{B}$, and the second problem is treated. Similar to the intervals P_i we define the intervals Q_i.

To address 1., we observe that, in Algorithm 2, the numbers a_i are only used to determine the right sign for the square root operation in step 5. We treat this problem below. Additionally, the points a_i are used to determine the intervals I_i in Algorithm 4. Here we have to use the intervals P_i instead of the values a_i. This might decrease the computed step length h.

Now, we discuss the third problem. The intervals I_l are the smallest intervals containing the line segments $p_l([t_0, t_0+h])$ with the endpoints $\tilde{a}_l = p_l(t_0)$ and $\tilde{b}_l = p_l(t_0+h)$; $l = -k+1, \ldots, 0$. To be on the safe side, we have to ensure a_l, $\tilde{a}_l$, b_l, $\tilde{b}_l \in I_l$. This is obtained if we set $I_l := p_l(t_0 + T_h)$ computed with rounded interval arithmetic. As in Algorithm 4, T_h is the smallest interval containing the time interval $[0, h]$ in the used interval arithmetic. We assume that t_0 and h are chosen such that $t_0 + h$ is computed correctly. If the step length h becomes too small, we might be close to a critical point and should apply the methods explained in Chapter 7.

Concerning Algorithms 2 and 4, the only problem might occur with the square root operations in step 5. We consider the inequality from step 5, which determines the sign choice for the square root operation:

$$\begin{array}{rrcl} & |b_d + a_d| & > & |b_d - a_d| \\ \Longleftrightarrow & (b_d + a_d)(\overline{b_d + a_d}) & > & (b_d - a_d)(\overline{b_d - a_d}) \\ \Longleftrightarrow & b_d\bar{a}_d + a_d\bar{b}_d & > & 0 \\ \Longleftrightarrow & \mathrm{Re}(b_d)\mathrm{Re}(a_d) + \mathrm{Im}(b_d)\mathrm{Im}(a_d) & > & 0 \end{array}$$

In addition to the correct sign choice for the floating point number b_d we have to ensure that this inequality leads to the correct choice for the exact value $\tilde{b}_d$, as well. We consider the intervals P_d and Q_d that are defined above and observe a_d, $\tilde{a}_d \in P_d$ and b_d, $\tilde{b}_d \in Q_d$. We replace the condition $\mathrm{Re}(b_d)\mathrm{Re}(a_d) + \mathrm{Im}(b_d)\mathrm{Im}(a_d) > 0$ by

$$[x_d, y_d] := \mathrm{Re}(Q_d)\mathrm{Re}(P_d) + \mathrm{Im}(Q_d)\mathrm{Im}(P_d) > 0. \tag{6.7}$$

If we use circular arithmetic, the real part $\mathrm{Re}(\{c; r\})$ of a circular interval $\{c; r\}$ is the closed interval $[\mathrm{Re}(c) - r, \mathrm{Re}(c) + r]$, where the lower bound is rounded down and the upper bound is rounded up. Similarly, we have $\mathrm{Im}(\{c; r\}) = [\mathrm{Im}(c) - r, \mathrm{Im}(c) + r]$. The imaginary part of a real interval is zero. We define that $[x_d, y_d] > 0$ if and only if $x_d > 0$. If Inequality 6.7 does not hold for both choices for Q_d, then we should either increase the precision of the floating point computations or we are close to a critical point and should apply the methods from Chapter 7.

Using an inductive argument, we know b_c, $\tilde{b}_c \in I_c$. Since the value b_d and the interval I_d are computed by the same branch of the square root function, we have $b_d \in I_d$. Since $\{\sqrt{z} | z \in I_c\} \subset \sqrt{I_c} \subset I_d$ for a fixed branch $\sqrt{\_}$ of the root function, we have $\tilde{b}_d \in I_d$.

To sum up, simple modifications to Algorithms 2 and 4 combined with the ideas from Chapter 7 concerning the treatment of (possible) critical points lead to a robust method for solving the Tracing Problem. We have seen that close to a critical point we could benefit from a higher precision of the floating point numbers. In [25], different packages for multiple precision interval arithmetic are compared.

6.6 Cubic and Higher Order Roots

We describe how the given algorithms for the Tracing Problem can be modified such that they can deal with GSPs with cubic and higher order roots, as well. The ideas given in Section 6.1 generalize directly to higher order roots, and we revisit the Radicand Lemma; see Figure 6.14.

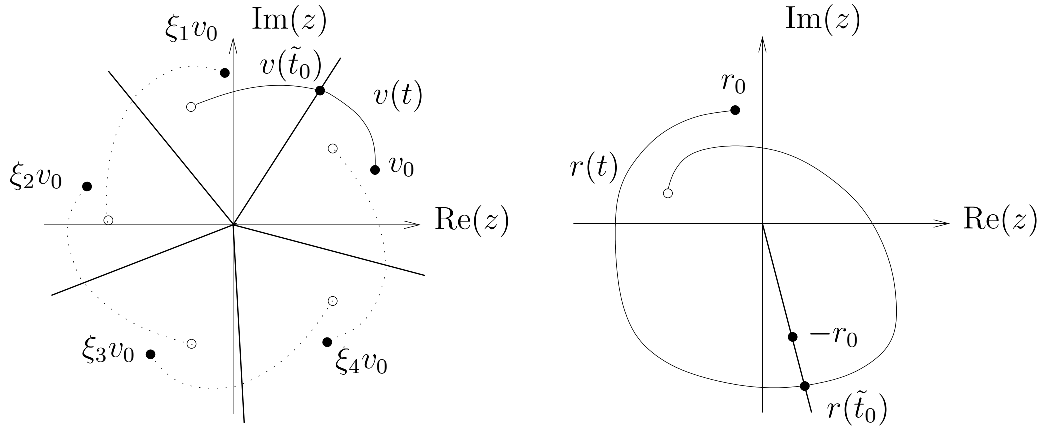


Figure 6.14: Illustration of the proof of the Radicand Lemma 6.6.1 for $n = 5$.

Lemma 6.6.1. Radicand Lemma for nth Roots
Let $n \in \mathbb{N}$ with $n \geq 2$. Let $\xi_0, \ldots, \xi_{n-1}$ be the n nth roots of unity. Let $r\colon [0,1] \to \mathbb{C}^* = \mathbb{C}\backslash\{0\}$ be a continuous path and $r_0 := r(0)$ its starting point. Let $v\colon [0,1] \to \mathbb{C}^*$ be the unique continuous path with $(v(t))^n = r(t)$ for all $t \in [0,1]$ and $v(0) = v_0$ for a fixed $v_0 = \xi_{j_0} \sqrt[n]{r_0}$ with $\xi_{j_0} \in \{\xi_0, \ldots, \xi_{n-1}\}$. Let l be the ray that starts in $0 \in \mathbb{C}$ and passes through $-r_0$. If r does not intersect the ray l, then at any time $t \in [0,1]$ we have

$$|v(t) - v_0| < |v(t) - \xi_j v_0| \text{ for } \xi_j \neq 1.$$

Lemma 6.6.1 states that the path v stays in the Voronoi region of v_0 in the Voronoi diagram of the point set $\{\xi_j \sqrt[n]{r_0} \mid j = 0, \ldots, n-1\}$. The endpoint $v(1)$ is the unique point $v_1 \in \{\xi_j \sqrt[n]{r(1)} \mid j = 0, \ldots, n-1\}$ with $|v_1 - v_0| < |v_1 - \xi_j v_0|$ for all nth roots of unity ξ_j with $\xi_j \neq 0$.

Proof. (Indirect) We consider the Voronoi diagram of the point set $\{\xi_j \sqrt[n]{r_0} \mid j = 0, \ldots, n-1\}$. To obtain a contradiction, we assume that there is a $\tilde{t} \in [0,1]$ and a root of unity $\xi_j \neq 1$ with $|v(\tilde{t}) - v_0| \geq |v(\tilde{t}) - \xi_j v_0|$. Thus, $v(\tilde{t})$ is contained in the Voronoi region of $\xi_j v_0 \neq v_0$. Since v is continuous, there must be an intersection point $v(\tilde{t}_0)$ of v and one of the two Voronoi edges bounding the Voronoi region of v_0. These two Voronoi edges are the rays starting in the origin $0 \in \mathbb{C}$ and passing through the points $e^{2\pi i/(2n)} v_0$ or $e^{-2\pi i/(2n)} v_0$, respectively. Thus we have $v(\tilde{t}_0) = s e^{2\pi i/(2n)} v_0$ or $v(\tilde{t}_0) = s e^{-2\pi i/(2n)} v_0$ for an $s > 0$, and we observe

$$r(\tilde{t}_0) = (v(\tilde{t}_0))^n = (s e^{\pm 2\pi i/(2n)} v_0)^n = s^n e^{\pm \pi i} v_0^n = -s^n r_0.$$

Hence, the point $r(\tilde{t}_0)$ lies on the ray l, and we have a contradiction to the assumption of Lemma 6.6.1. □

Corollary 6.6.2. Radicand Corollary
If the path r stays in a rectangle or circle that does not contain $0 \in \mathbb{C}$, then $|v(t) - v_0| < |v(t) - \xi_j v_0|$ holds for all $t \in [0,1]$ and the $n-1$ nth roots of unity $\xi_1, \ldots, \xi_{n-1}$.

Proof. Corollary 6.6.2 can be proved in the same way as Corollary 6.1.2. □

The Radicand Lemma is the fundament of step 5 in Algorithm 2 and 4. In order to adapt these algorithms to higher order roots, we have to replace the line

$$\text{choose } b_d \in \{\pm\sqrt{b_c}\} \text{ with } |b_d - a_d| < |b_d + a_d|$$

by

$$\text{choose } b_d \in \{\xi_j \sqrt{b_c} \mid j = 0, \ldots, n-1\} \text{ with } |b_d - a_d| < |b_d - \xi_j a_d| \text{ for all } n\text{th roots of unity } \xi_j \text{ with } \xi_j \neq 1$$

if we consider nth roots. Obviously, the assignment $I_d := \sqrt{I_c} \ni b_d$ in the next line must be replaced by $I_d := \sqrt[n]{I_c} \ni b_d$, so that we have to compute nth roots in interval arithmetic.

In real interval arithmetic, nth roots can be computed using Definition A.1.8. If n is even, then the nth root of an interval $[a,b]$ with $a > 0$ is either $[+\sqrt[n]{a}, +\sqrt[n]{b}]$ or $[-\sqrt[n]{b}, -\sqrt[n]{a}]$. If n is odd, then the nth root of an interval $[a,b]$ is $[\sqrt[n]{a}, \sqrt[n]{b}]$.

In rectangular complex interval arithmetic, the computation of nth roots seems to be more difficult; see page 162. Petković and Petković [56, 57] investigated the computation of nth roots in circular arithmetic; see page 168. If we have $n > 2$ and if the radius of the considered circular interval I_c is too large, then the n intervals $\sqrt[n]{I_c}$ might overlap; see [56, Thm. 1, p. 30]. Thus, we must choose $I_d = \sqrt[n]{I_c}$ in step 5 of Algorithms 2 and 4 more carefully. If the point b_d is contained in more than one disc $\sqrt[n]{I_c}$, then we could replace the requirement $b_d \in \sqrt[n]{I_c}$ by $a_d \in \sqrt[n]{I_c}$. If both points a_d and b_d are contained in more than one disc $\sqrt[n]{I_c}$, then we have to decrease the step length and to restart the algorithm.

6.7 Is Affine Arithmetic a Better Choice?

In our context of Dynamic Geometry, affine arithmetic captures the structure of GSPs more directly than interval arithmetic: In affine arithmetic, the difficult operations are the non-affine operations multiplication, division, and root. These operations also cause the main problems for the Tracing Problem: A square root operation doubles the number of instances that lie above a given point. In the language of Riemann surfaces of algebraic functions, a square root operation doubles the number of sheets of the covering space. A multiplication might enlarge the number or the multiplicity of critical points that are caused by a future division or square root operation. A division operation can be responsible for many critical points and seems to be difficult to handle.

An other advantage of affine arithmetic is the treatment of dependencies. In many GSPs, the number of free variables is much smaller than the number of dependent variables leading to many dependencies between the variables of the GSP. Thus, using affine arithmetic could reduce the overestimation of the ranges and, consequently, the underestimation of the step length.

However, affine arithmetic does not fulfill the inclusion monotonicity property, in general. Thus, replacing interval arithmetic by affine arithmetic should be done carefully. Maybe we achieve inclusion monotonicity in our particular situation since the dependencies remain similar if the step length is reduced.

The inclusion monotonicity property implies that if Algorithm 2 or Algorithm 4 treat a division or root variable v_i, then only variables v_j with $j \geq i$ can cause a restart of the algorithm. This property is fundamental for the termination proofs. Without the inclusion monotonicity property, there might be restarts for variables v_j with $j < i$. This fact complicates the correctness proofs of Algorithm 2 and Algorithm 4.

There are only few approaches for complex affine arithmetic. In [38], complex affine arithmetic is defined by treating the real and imaginary part of complex quantities separately similar to rectangular complex interval arithmetic. This approach leads to a straight forward definition of the four basic arithmetic operations addition, subtraction, multiplication, and division; a definition of a square root operation is missing. The square root operation for rectangular arithmetic from Definition A.2.8 does not consider dependencies and would not be a good choice for affine arithmetic.

An other possibility to define complex affine arithmetic would be to assume that the noise symbols ϵ_i are complex variables taking values in the unit circle $\{0; 1\}$. In this case, the challenge is to define the division, and the square root operation. We remark that assuming that the noise symbols take values in the square $[-1, 1] + i[-1, 1]$ is not a promising approach: A term $x_i \epsilon_i$ with $x_i \in \mathbb{C}$ does not describe an axisparallel square so that the geometric interpretation is difficult. Elaborating complex affine arithmetic for solving the Tracing Problem is an interesting future project.

Chapter 7

Detection and Treatment of Critical Points

The detection and treatment of critical points in the tracing process described in Chapter 6 is discussed; see Sections 7.1 and 7.2. A division or root variable causes a critical point if the path $v_c(t)$ of its divisor or radicand passes through zero[1]. After detecting a zero of $v_c(t)$ in the time interval $[t_0, t_0+h]$, we omit the critical point by a detour in the complex plane [40]. We remark that a similar idea is used for solving polynomial systems of equations via homotopy methods [59]. Here a randomly chosen complex parameter is used in the homotopy to avoid singularities along the solution paths. A crucial observation is that the instance reached after a detour around a critical point depends on the detour itself. We discuss consequences of detouring in Section 7.3. In Section 7.4, we indicate how all critical points of a Geometric Straight-Line Program with only one free variable that are contained in a box could be approximated using methods from interval analysis.

7.1 Detection of Critical Points

We describe how a critical point can be detected. For the Tracing Problem, we are given a starting instance A of a GSP Γ and paths $p_{-k+1}, \ldots, p_0$ of the free variables. Our Algorithms (Algorithm 3 and Algorithm 5) trace the corresponding continuous evaluation stepwise. The final instance of each step is determined coordinate-wise by Algorithm 2 or Algorithm 4. Here, for each (critical) vari-

[1]This fact also holds for root operations over $\mathbb{K} = \mathbb{R}$. Here, we can neglect the case $v_c(t) < 0$: At the starting instance A, the radicand variable is positive, otherwise A would not be an instance. Before the path $v_c(t)$ of the radicand variable v_c takes a negative real number as value, it must have passed through zero since $v_c(t)$ is a continuous function.

able v_m is ensured that no m-critical point is hit: Let v_c be the radicand or divisor variable of v_m, let $[t_0.t_0+h] \subset [0,1]$ be the time interval of the considered step. An m-critical point occurs at time $\tilde{t} \in [t_0, t_0+h]$ if $v_c(\tilde{t}) = 0$. In this situation we have $0 = v_c(\tilde{t}) \in v_c([t_0, t_0+h]) \subset I_c$. The inclusion $v_c([t_0, t_0+h]) \subset I_c$ holds by construction of the interval I_c. Algorithm 2 and Algorithm 4 only proceed if $0 \notin I_c$. In this way is guaranteed that no m-critical point occurs in the step of length h at starting time t_0. Otherwise, Algorithm 2 and Algorithm 4 are restarted with a smaller step length. Thus, a first indication of a critical point is that the step length computed by Algorithm 2 in Algorithm 3 or Algorithm 4 in Algorithm 5 becomes arbitrarily small; see Remark 6.2.2 from page 77 and Remark 6.3.6 from page 87. Hence, we should stop Algorithm 3 and Algorithm 5 if a suitable lower bound for the step length is reached.

Observation 1. *Let h_l be a lower bound for the step length.*
If the step length h, computed while Algorithm 3 or Algorithm 5 run, reaches the lower bound h_l at some time t_0, then there might be a critical point close to the instance $(p_{-k+1}(t_0), \dots, p_0(t_0), v_1(t_0), \dots, v_n(t_0))$.

Since the performance of the algorithms depends on the lower bound h_l of the step length h, the value of h_l should be chosen carefully. How to chose h_l properly is still an open problem. In practice, h_l could depend on the computational accuracy.

We investigate the zeros of the path $v_c(t)$ of a radicand or divisor variable v_c in a continuous evaluation of Γ. The variable v_c can only induce an m-critical point if its path $v_c(t)$ in the continuous evaluation $(v_1(t), \dots, v_c(t), \dots, v_{m-1}(t))$ of the $m-1$-head $\Gamma^{(m-1)}$ of Γ exists and has a zero in the time interval $[0,1]$. We assume that the paths $p_l\colon [0,1] \to \mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$ of the free variables z_l of Γ can be extended to analytic functions $\hat{p}_l\colon U_{[0,1]} \to \mathbb{C}$ on a neighborhood $U_{[0,1]} \subset \mathbb{C}$ of the time interval $[0,1] \subset \mathbb{R}$. In particular, this property holds for linear or circular paths p_l. Let $(v_1(t), \dots, v_n(t))$ be a continuous evaluation of Γ along the paths p_l. If the neighborhood $U_{[0,1]}$ is chosen small enough, then the paths $v_j(t)\colon [0,1] \to \mathbb{K}$ can be extended to analytic functions $\hat{v}_j\colon U_{[0,1]} \to \mathbb{C}$, as well; see Section 8.6.1. A famous theorem from complex analysis (German: "Identitätssatz") implies that the functions $\hat{v}_j$ are either constantly zero or their zeros form a discrete subset of $U_{[0,1]}$. Since the time interval $[0,1]$ is compact, a path $v_j(t)$ in a continuous evaluation is either constantly zero or has a finite number of zeros if the paths p_l of the free variables can be extended to analytic functions on $U_{[0,1]}$.

If Γ is a complex GSP and if the paths p_l of the free variables of Γ can be extended to analytic functions $\hat{p}_l$ on $U_{[0,1]}$, then the paths $v_j(t)\colon [0,1] \to \mathbb{C}$ are paths in the complex plane that can be extended to analytic functions $\hat{v}_j\colon U_{[0,1]} \to \mathbb{C}$. Lemma 7.1.1 implies that the real part $\mathrm{Re}(v_j(t))$ and the imaginary part $\mathrm{Im}(v_j(t))$ of $v_j(t)$ are either constantly zero on the interval $[0,1] \subset \mathbb{R}$ or have a finite number of zeros in $[0,1]$.

Lemma 7.1.1. *Let $U_{[0,1]} \subset \mathbb{C}$ be a neighborhood of the real interval $[0,1] \subset \mathbb{R}$. Let $f\colon U_{[0,1]} \to \mathbb{C}$, $z \mapsto f(z)$, be an analytic function. Then the real part $\mathrm{Re}(f)\colon U_{[0,1]} \to \mathbb{R}$ either has a constant number of zeros in the real interval $[0,1]$ or $\mathrm{Re}(f(z)) = 0$ holds for all points $z \in [0,1]$.*

Proof. Let $t_0 \in [0,1]$, and let

$$f(z) = \sum_{n=0}^{\infty} a_n (z - t_0)^n$$

be the power series of f in a neighborhood $U(t_0) \subset \mathbb{C}$ of t_0. For $t \in [0,1] \cap U(t_0)$ we have

$$\begin{aligned} f(t) &= \sum_{n=0}^{\infty} \underbrace{a_n}_{\in\mathbb{C}} \underbrace{(t - t_0)}_{\in\mathbb{R}} = \sum_{n=0}^{\infty} \big(\underbrace{\mathrm{Re}(a_n)(t - t_0)}_{\in\mathbb{R}} + \underbrace{i\mathrm{Im}(a_n)(t - t_0)}_{\in i\mathbb{R}} \big) \\ &= \underbrace{\sum_{n=0}^{\infty} \mathrm{Re}(a_n)(t - t_0)}_{\in\mathbb{R}} + i \underbrace{\sum_{n=0}^{\infty} \mathrm{Im}(a_n)(t - t_0)}_{\in\mathbb{R}} . \end{aligned}$$

This implies

$$\mathrm{Re}(f(t)) = \sum_{n=0}^{\infty} \mathrm{Re}(a_n)(t - t_0),$$

and $\mathrm{Re}(f)$ can locally be expressed as a power series. Hence the function $\mathrm{Re}(f)$ extends to an analytic function defined on a neighborhood $U \subset \mathbb{C}$ of $[0,1]$. A famous result from complex analysis implies that this analytic function is either constantly zero, or it has at most a finite number of zeros in the compact interval $[0,1] \subset \mathbb{R}$. □

The previous study leads to observation 2.

Observation 2. *We assume that the paths $p_l\colon [0,1] \to \mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$ of the free variables z_l of Γ can be extended to analytic functions $\hat{p}_l\colon U_{[0,1]} \to \mathbb{C}$ on a neighborhood $U_{[0,1]} \subset \mathbb{C}$ of the time interval $[0,1] \subset \mathbb{R}$. As mentioned above, this property holds for linear or circular paths p_l. Let $(v_1(t), \ldots, v_n(t))$ be a continuous evaluation of Γ along the paths p_l. If $\mathbb{K} = \mathbb{R}$, then the paths $v_j(t)$ either have a finite number of zeros in the time interval $[0,1]$ or they are constantly zero; $j = 1, \ldots, n$. If $\mathbb{K} = \mathbb{C}$, then the real part $\mathrm{Re}(v_j(t))$ and the imaginary part $\mathrm{Im}(v_j(t))$ of the paths $v_j(t)$ either have a finite number of zeros in the time interval $[0,1] \subset \mathbb{R}$, or they are constantly zero on the interval $[0,1]$; $j = 1, \ldots, n$.*

As mentioned above, it suffices to consider the case $\mathbb{K} = \mathbb{R}$ since in the complex situation $v_c(t) = 0$ holds if and only if $\mathrm{Re}(v_c(t)) = 0$ and $\mathrm{Im}(v_c(t)) = 0$. At

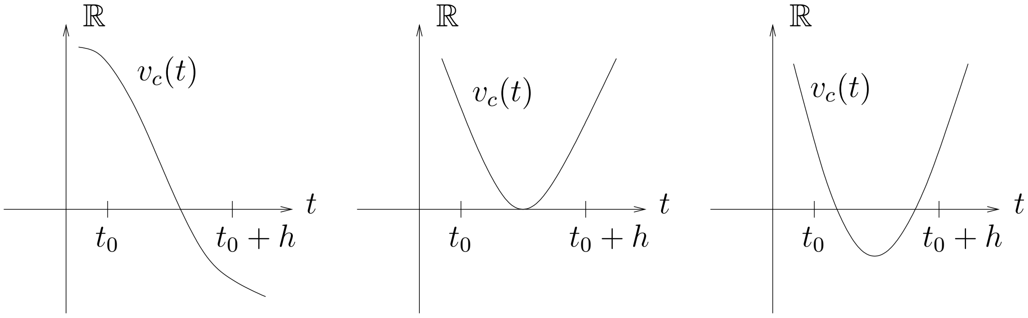


Figure 7.1: The local situation of $v_c(t)$ close to a zero is illustrated for $\mathbb{K} = \mathbb{R}$. The left figure shows the situation from observation 3, and the middle and the right one visualize observation 4.

the starting instance A at time $t_0 \in [0, 1]$ we have $v_c(t_0) \neq 0$ for every radicand or divisor variable. Otherwise, A would be a critical point and not an instance. Thus, for $\mathbb{K} = \mathbb{R}$ we can exclude the case $v_c(t) \equiv 0$. For $\mathbb{K} = \mathbb{C}$, at least one of the two functions $\mathrm{Re}(v_c(t))$ or $\mathrm{Im}(v_c(t))$ has at most a finite number of zeros in the time interval $[0, 1]$ and we can apply the same case distinction as in the real situation. Figure 7.1 shows the local situation of $v_c(t)$ close to a zero for $\mathbb{K} = \mathbb{R}$.

The intermediate value theorem leads to the following observation.

Observation 3. *Let $v_c(t)\colon [t_0, t_0 + h] \to \mathbb{R}$ with $v_c(t_0) \cdot v_c(t_0 + h) < 0$. Since $v_c(t)$ is a continuous real function, it has a zero in the time interval $[t_0, t_0 + h]$.*

In Algorithm 2 and Algorithm 4, the condition $a_c b_c = v_c(t_0) v_c(t_0 + h) < 0$ can be easily verified. Recall that b_c is the coordinate of v_c in the final instance B of the current step.

The path $v_c(t)$ might have a zero although $v_c(t_0)v_c(t_0 + h) > 0$ holds. Then, $v_c(t)$ has a local extremum at a time $\tilde{t} \in [t_0, t_0 + h]$, and $\dot{v}_c(\tilde{t}) = 0$ and $0 \in \{\dot{v}_c(t) \mid t \in [t_0, t_0 + h]\} \subset \dot{I}_c$ hold.

Observation 4. *If $0 \in \dot{I}_c$, then the function $v_c(t)$ might have a local extremum in the time interval $[t_0, t_0 + h]$. In this situation, the path $v_c(t)$ could have a zero in the time interval $[t_0, t_0 + h]$ although the condition $v_c(t_0)v_c(t_0 + h) < 0$ from observation 3 fails.*

Depending on the step length h, which is the length of the time interval $[t_0, t_0 + h]$, it might be important to approximate the zero of $v_c(t)$ causing the critical point. A first possibility is to use Algorithm 2 or Algorithm 4. Both algorithms could also be used backwards; see Improvement 6.3.12 from page 93. As usual, let a_c be the coordinate of v_c in the starting instance A, and b_c the coordinate of v_c in the final instance B of the current step. The coordinate b_c has already been computed since v_c is the radicand or divisor variable of the current (critical) variable v_m. We apply Algorithm 2 or Algorithm 4 to the path $v_c(1-t)$ at starting time t_0+h and starting point b_c. A second possibility for the approximation of the zero of $v_c(t)$ is to use a Newton-Iteration. This is possible since we can determine the first derivative $\dot{v}_c(t)$ of $v_c(t)$ by differentiating the GSP Γ as in Section 3.5 and by considering the corresponding coordinate of $\dot{v}_c(t)$ in the instance of $\dot{\Gamma}$ at a fixed time t.

In the situation of observation 4, we propose to bisect the time interval $[t_0, t_0+h]$ into the two subintervals $[t_0, t_0+\frac{h}{2}]$ and $[t_0+\frac{h}{2}, t_0+h]$ and to proceed with each subinterval separately.

If an m-critical point is found, we can determine the multiplicity of the zero of its radicand or divisor v_c. For this purpose, we have to compute higher order derivatives of the GSP Γ; see Section 3.5. This computation is possible since the derivative $\dot{\Gamma}$ of a GSP Γ is again a GSP. The corresponding instance of the GSP $\dot{\Gamma}$ can be determined using Lemma 3.5.3. Hence, we have to derive the GSP Γ and to determine the corresponding instances (so far as possible, i.e. the $m-1$-head) at this point until the corresponding coordinate of v_c that describes its μth derivative differs from zero. Then, $\mu-1$ is the multiplicity of the zero of v_c that causes the m-critical point.

7.2 Treatment of Critical Points

After the detection of a critical point, we have to deal with it. We choose Kortenkamp's approach [40] of detouring around degeneracies in the complex plane, which is used in the Dynamic Geometry Software *Cinderella* [42, 41]. This approach leads to a locally and globally consistent treatment of critical points; see [40, Sect. 6.3], Section 1.3 and Section 7.3.

As in our algorithms (Algorithm 2-5) we assume that the given paths of the free variables of the GSP Γ are linear paths $p = p_l$ with range $\mathbb{C}$, for example $p\colon [0,1] \to \mathbb{C}$, $t \mapsto a + t(b-a)$ for fixed $a, b \in \mathbb{C}$. Hence, p describes the line segment between a and b. Clearly, if and only if $a, b \in \mathbb{R}$, then the range of p is contained in $\mathbb{R}$. For simplicity, we use the time interval $[0,1]$; we can transfer all observations to arbitrary time intervals $[t_0, t_0+h]$ with $h > 0$ by reparametrization.

If a (possible) critical point is detected in the time interval $[0,1]$, we replace the line segment $\overline{ab}$ between a and b by the right semi circle having the line segment $\overline{ab}$

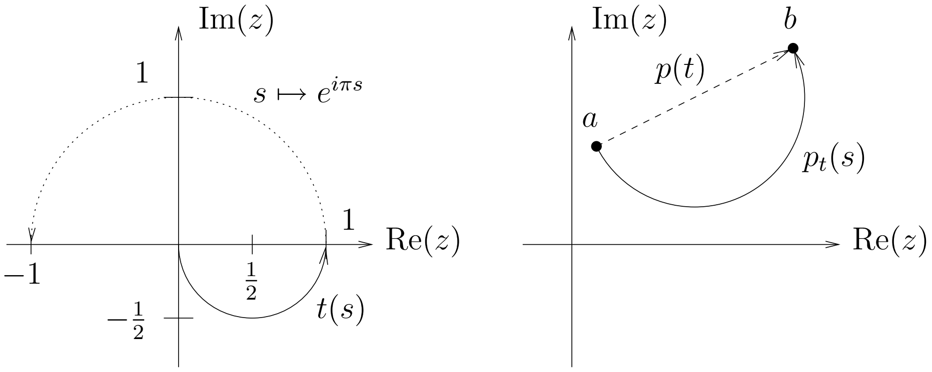


Figure 7.2: The left figure shows the function $t(s) = \frac{1-e^{i\pi s}}{2}$, and the right one shows the function $p_t(s) = a + t(s)(b-a) = \frac{a+b}{2} - \frac{b-a}{2}e^{i\pi s}$.

as diameter. For this purpose, we define the functions

$$\begin{array}{rccc} t(s): & [0,1] & \rightarrow & \mathbb{C} \\ & s & \mapsto & \frac{1-e^{i\pi s}}{2} \end{array}$$

and

$$\begin{array}{rccc} p_t: & [0,1] & \rightarrow & \mathbb{C} \\ & s & \mapsto & a + t(s)(b-a) \end{array}$$

as in [40] and on page 7. Hence, $p_t(s)$ moves once around the right semi circle with diameter $\overline{ab}$; see Figure 7.2.

Instead of tracing along the linear path $p(t) = p_l(t)$, we now have to trace along the circular path $p_t(s) = p_{l,t}(s)$. For this purpose, we can use Algorithm 3 or Algorithm 5. Note that the intervals I_l and $\dot{I}_l$ in step 2 of Algorithm 2, or step 2 of Algorithm 4 have to be determined, such that the inclusion monotonicity property is fulfilled. Recall that I_l contains the range $p_l([t_0, t_0+h])$ of the path p_l of the free variable z_l of the GSP Γ. The interval $\dot{I}_l$ contains the range $\dot{p}_l([t_0, t_0+h])$ of $\dot{p}_l$ over the time interval $[t_0, t_0+h]$. For the paths $p_{l,t}(s)$ we require $p_{l,t}([s_0, s_0+h]) \subset I_l$ and $\dot{p}_{l,t}([s_0, s_0+h]) \subset \dot{I}_l$. To fulfill the inclusion monotonicity property, we must define the intervals I_l and $\dot{I}_l$ such that $h' < h$ implies $I'_l \subset I_l$ and $\dot{I}'_l \subset \dot{I}_l$ for $l = -k+1, \ldots, 0$. Here, the intervals I'_l and $\dot{I}'_l$ are the corresponding intervals to I_l and $\dot{I}_l$ for the step length h'.

First, we describe the construction of the intervals I_l for $l = -k+1, \ldots, 0$. If rectangular interval arithmetic is used, we can choose the interval $I = I_l$ as the smallest axis-parallel rectangle containing the set $\{p_t(s) \,|\, s \in [s_0, s_0+h]\}$. For circular interval arithmetic, the situation is slightly more complicated. Here, we chose the interval I as the smallest circle $\{c; r_c\}$ with the following properties; see Figure 7.3:

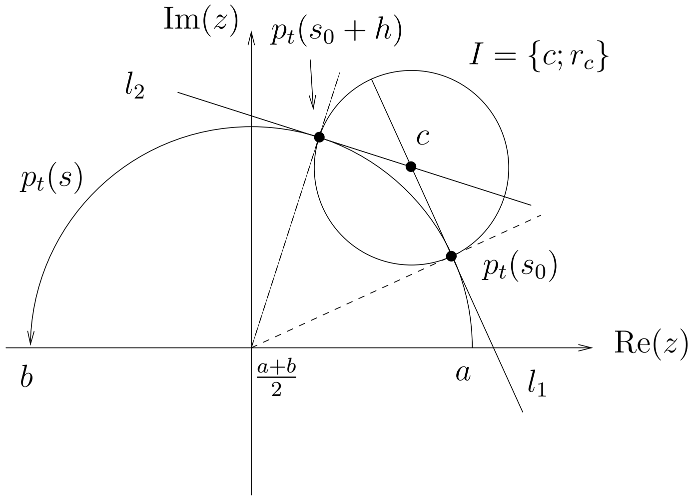


Figure 7.3: The construction of the interval $I = \{c; r_c\}$ is shown.

1. The circle $\{c; r_c\}$ contains the set $\{p_t(s) \mid s \in [s_0, s_0+h]\}$; and
2. the midpoint c is the intersection point of the two lines l_1 and l_2. Here, l_1 is the line that passes through $p_t(s_0)$ and that is orthogonal to the line connecting the points $p_t(s_0)$ and $\frac{a+b}{2}$. The point $\frac{a+b}{2}$ is the midpoint of the semi circle of p_t. The line l_2 passes through the point $p_t(s_0+h)$ and is orthogonal to the line that connects the points $p_t(s_0+h)$ and $\frac{a+b}{2}$.

To compute the intervals $\dot{I}_l$ for Algorithm 4 (Cone Algorithm), we determine the derivative $\dot{p}_t$ of p_t: We have $\dot{p}_t(s) = (b-a)\dot{t}(s) = -\frac{i\pi(b-a)}{2}e^{i\pi s}$, which again describes a semi circle. Hence, the intervals $\dot{I}_l$ can be computed in the same way as the intervals I_l.

Since our algorithms Algorithm 2, Algorithm 3, Algorithm 4 and Algorithm 5 become quite inefficient if the starting instance is chosen close to a critical point, we propose to use a buffer zone around a critical point; see Figure 7.4. We define the intervals $I = I_l$ and $I = \dot{I}_l$ such that they do not hit this buffer zone. This approach ensures that the intervals I_l and $\dot{I}_l$ have a larger distance to the critical point. For this reason, we expect that $0 \in I_c$ in step 4 and 5 of Algorithm 2 and Algorithm 4 occurs more rarely. Hence, we expect that the number of restarts in step 5 of Algorithm 4 or step 5 of Algorithm 4 is reduced that way. To sum up, the buffer zone enlarges the detour p_l around the critical point, but therefor the step length h might become larger as well.

Remark 7.2.1. An open problem is to determine whether a path p_t "catches" a singularity. One possibility for this could be to determine whether a complex time

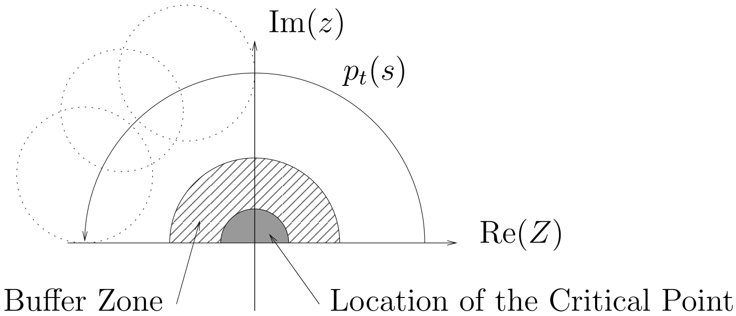


Figure 7.4: The construction of the buffer zone is illustrated. The dotted circles are candidates for the interval I: They do not intersect the buffer zone.

interval contains a critical point. This could be done with an Interval Newton Method; see Section 7.4 and Appendix A.5.

7.3 Consequences of Detouring

If we trace a detour around a (possible) critical point, then the final instance depends in many situations on the detour. Thus, the choice of the detour heavily influences the properties of the Dynamic Geometry system. We discuss the different situations and describe Kortenkamp's approach [40] used in the Dynamic Geometry Software *Cinderella* [42, 41] that leads to local and global consistency [40, Sect. 6.3]. In the following discussion, we assume that the given GSP only has one free variable.

In most applications in Dynamic Geometry, the restriction to GSPs with only one free variable is not a serious constraint for solving the Tracing Problem. Here, the paths p_l of the free variables z_l of a GSP Γ are usually linear paths and can be described by a GSP Γ_l having the time t as single free variable; $l = -k+1, \ldots, 0$. We replace the operations $z_l \leftarrow$ FREE of Γ by the operations of Γ_l. The resulting GSP is a GSP with only one free variable t. The same argument is used in the proof of Lemma 3.2.10 on page 37.

Let Γ be a GSP with only one free variable. Since we consider complex detours, we assume that Γ is a GSP over $\mathbb{C}$. At a critical point, a division by zero or a root of zero occurs; see Definition 3.2.3 from page 34. We have seen in Section 3.2 that critical points of Γ are either isolated or an entire connected component of the configuration space consists of critical points. An isolated critical point might lead to a removable singularity, a pole, or a branch point. Since the dependent variables of a GSP Γ describe algebraic functions, we can exclude essential singularities: If the free variable z of Γ moves on a polynomial path $p(t)$, then the dependent variables v_j of Γ are algebraic functions $\hat{v}_j(t)$ in t. These algebraic functions can be seen as meromorphic functions that are defined on the covering space of a branched covering; see Section 8.3, Theorem 8.3.3, and Section 8.4 for a detailed discussion of examples.

Algorithms 3 and 5 proceed stepwise. In each step, Algorithms 2 and 4, respectively, compute the final instance coordinate by coordinate. Thus, we consider the paths $v_j(t)$ of the dependent variables v_j separately. Doing this, we have to keep in mind that the described effects might accumulate: At a certain time $\tilde{t}$, two (or even more) dependent variables could cause a critical point simultaneously; see example 7.3.1 and Section 1.3.

Example 7.3.1. We consider the path $p(t)\colon [0,1] \to \mathbb{C}$, $t \mapsto 1-2t$, and the GSP Γ:

$$\begin{array}{rrcl} \Gamma\colon & z & \leftarrow & \text{FREE} \\ & v_1 & \leftarrow & \sqrt{z} \\ & v_2 & \leftarrow & \frac{1}{z} \\ & v_3 & \leftarrow & z \cdot v_2 \end{array}$$

The dependent variable v_1 describes the function $z \mapsto \sqrt{z}$, which has a branch point at $z = 0$. The dependent variable v_2 describes the reciprocal $z \mapsto 1/z$, which has a pole of order 1 at $z = 0$. Finally, the dependent variable v_3 describes the function $z \mapsto z/z$, which has a removable singularity at $z = 0$. By Definition 3.2.2, the point $z = 0$ is a 1-critical point, the other dependent variables are not considered. However, detouring around the origin influences *all* coordinates of the final instance. A detour around the origin is obtained by leaving the path $p(t)$ of the free variable z in a neighborhood of the time $\tilde{t} = 1/2$ and bypasses the singularities of *all* dependent variables simultaneously.

In Kortenkamp's approach [40] described in Section 7.2, critical points are avoided by modifying the path p_j of the free variables. This change of the paths p_l induces a *simultaneous* modification of *all* paths $v_j(t)$ of the dependent variables v_j in the continuous evaluation.

Example 7.3.2. We consider the algebraic expression $\sqrt{z} + \sqrt{z}$, which is described by the following GSP Γ:

$$\begin{array}{rrcl} \Gamma\colon & z & \leftarrow & \text{FREE} \\ & v_1 & \leftarrow & \sqrt{z} \\ & v_2 & \leftarrow & \sqrt{z} \\ & v_3 & \leftarrow & v_1 + v_2 \end{array}$$

The point $z_0 = 0$ is the only critical value of Γ. We observe

$$v_3 = v_1 + v_2 = \sqrt{z} + \sqrt{z} = \begin{cases} 2\sqrt{z}, & \text{if } v_1 = v_2 \qquad (1) \\ 0, & \text{if } v_1 = -v_2. \qquad (2) \end{cases}$$

At a given instance $A = [a_0, a_1, a_2, a_3]$, we either have $a_1 = a_2$ or $a_1 = -a_2$ leading to $v_3 = 2a_1 = 2\sqrt{a_0}$ (case 1) or $v_3 = 0$ (case 2), respectively. Thus, the starting instance A fixes the case in the definition of the function of v_3. Tracing along a path $p(t)$ of the free variable without hitting the point $z_0 = 0$ does not change the case in the definition of the function of v_3.

If we trace along the path $p(t) = 2t-1$ of the free variable z, then both dependent variables v_1 and v_2 induce a critical point. Here, the critical point is the branch point of the covering map $z \mapsto z^2$. To avoid the critical point, we consider the path $p_t(s) = -e^{i\pi s}$ as in Section 7.2. Since this change of the path p affects both dependent variables v_1 and v_2 in the same way, we still remain in the starting case of the variable v_3, hence we observe a consistent behavior. In contrast to this, an independent treatment of the variables v_1 and v_2 can lead to a change of the case for the variable v_3.

Formally, this observation can be treated using the theory of Riemann surfaces of algebraic function as described in Chapter 8. Switching from the path $p(t)$ to the path $p_t(s)$ of the free variable z enforces that the variables v_1 and v_2 stay on their sheets of the covering $z \mapsto z^2$. An independent treatment of v_1 and v_2 may lead to a change of the sheets of only one of the two variables v_1 and v_2.

To investigate the influences of detouring around critical points, we explain when a modified path "catches" a critical point; see Figure 7.5. We assume that v_j is a dependent variable that is defined by a root or division operation. Let v_c be the radicand or divisor variable of v_j. Let $v_c(t)$ be the path of v_c in the continuous evaluation induced by the (polynomial) path $p(t)$ of the free variable, let $a_c = v_c(0)$ and $b_c = v_c(1)$ be the coordinates of v_c in the starting and final instances. As above, let $\hat{v}_c(z)$ and $\hat{v}_j(z)$ be the algebraic functions of v_c and v_j, respectively, induced by the (polynomial) path $p(t)$ of the free variable. The zeros $z_0, \ldots, z_{k_c}$ of $\hat{v}_c$ are the critical points caused by the variable v_j. Let $\tilde{v}_c(t)$ be another path with $\tilde{v}_c(0) = a_c$ and $\tilde{v}_c(1) = b_c$. Let $U \subset \mathbb{C}$ be a simply connected open subset that contains the images of the paths $v_c(t)$ and $\tilde{v}_c(t)$. We say that the path $\tilde{v}_c(t)$ catches a critical point if the closed path $v_c(t) - \tilde{v}_c(t)$ obtained by first following $v_c(t)$ and afterwards following $\tilde{v}_c(t)$ backwards is not null-homotopic in the set $U \setminus \{z_0, \ldots, z_{k_c}\}$; see Fig 7.5 and Definition 8.6.10 of Section 8.6. A closed path is null-homotopic if it is homotopic to a constant path.

If we surround a critical point that leads to a removable singularity of a dependent variable v_j without catching other singularities of this variable, then the final position b_j of v_j does not depend on the chosen path. This fact is a consequence of Cauchy's integral theorem.

If we surround a critical point that leads to a pole z_0 of the function $\hat{v}_j$, then the final point b_j depends on the chosen detour. Let two paths $\tilde{v}_c$ and $\tilde{\tilde{v}}_c$ from a_c to b_c

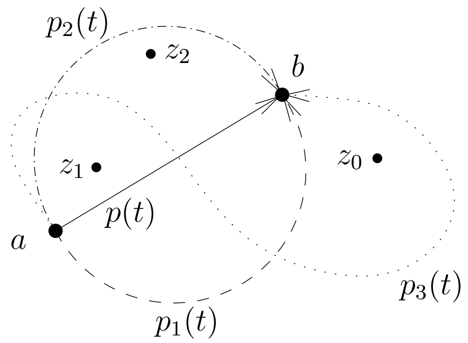


Figure 7.5: The path $p(t)$ together with the path $p_1(t)$ does not catch one of the points z_0, z_1, z_2; the path $p(t)$ together with the path $p_2(t)$ catches the points z_1 and z_2; the path $p(t)$ together with the path $p_3(t)$ catches the points z_0 and z_1.

lead to the final points $\tilde{b}_j$ and $\tilde{\tilde{b}}_j$ of the variable v_j. Then the difference $\tilde{b}_j - \tilde{\tilde{b}}_j$ of the two final positions depends on the winding number of the concatenated path $-\tilde{\tilde{v}}_c + \tilde{v}_c$ around the point z_0 and on the residue of the function $\hat{v}'_j$ in the point z_0. This is a consequence of the residue theorem.

If we surround a critical point that leads to a branch point, then following a detour results in a change of the sheets of the corresponding Riemann Surface. This change of the sheets depends on the choice of the modified path. Thus, the final point b_j of the variable v_j depends on the chosen detour.

7.4 How to Approximate all Critical Points of a GSP with one free variable in a Box Using Interval Analysis

We indicate how all critical points of a GSP with only one free variable that are contained in a given box can be approximated using methods from interval analysis. In Appendix A, we give an introduction to interval analysis including the solution of square systems of equations in Section A.5. We remark that this idea is not worked out completely, yet. At this moment, we have neither experiences concerning the quality of the method nor on the efficiency. Since it could be a promising approach, we sketch this work in progress. As explained in Section 7.3, the restriction to GSPs with only one free variable is not a serious constraint for solving the Tracing Problem.

Let Γ be a GSP over $\mathbb{K}$ with one free variable z and n dependent ones $v_1, \ldots, v_n$; $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$. Let $v_{j_1}, v_{j_2}, \ldots, v_{j_d}$ be the dependent variables that are defined by a division or square root operation. Only these variables can cause critical points. Let $v_{c_1}, v_{c_2}, \ldots, v_{c_m}$ be the corresponding divisor or radicand variables. The aim is to describe the set of critical points as the union of the zero sets of certain functions $F_m \colon \mathbb{K} \times \mathbb{K}^{c_m} \to \mathbb{K}^{c_m} \times \mathbb{K}$; $m = 1, 2, \ldots, d$. To define the functions F_m, we assign to every dependent variable $a = v_j$ of Γ a multivariate polynomial $\mathcal{P}_a = \mathcal{P}_{v_j}$ as in Section 5.1:

Dependent Variable a	Polynomial $\mathcal{P}_a$
$a \leftarrow b + c$	$a - (b + c)$
$a \leftarrow b - c$	$a - (b - c)$
$a \leftarrow b \cdot c$	$a - b \cdot c$
$a \leftarrow b/c$	$a \cdot c - b$
$a \leftarrow \sqrt{b}$	$a^2 - b$

The polynomials $\mathcal{P}_a$ are polynomials in at most three variables; we have $\mathcal{P}_a = 0$ if and only if the relation given by the dependent variable a is fulfilled. Using the polynomials $\mathcal{P}_a = \mathcal{P}_{v_j}$, we define the functions

$$F_m = F_m(\Gamma) \colon \quad \begin{array}{ccc} \mathbb{K} \times \mathbb{K}^{c_m} & \longrightarrow & \mathbb{K}^{c_m} \times \mathbb{K} \\ (z, \; v_1, \ldots, v_{c_m}) & \mapsto & (\mathcal{P}_{v_1}, \ldots, \mathcal{P}_{v_{c_m}}, \; v_{c_m}) \end{array}$$

for $m = 1, \ldots, d$. A point $(a_0, a_1, \ldots, a_{c_m}) \in \mathbb{K} \times \mathbb{K}^{c_m}$ fulfills the relations of the c_m-head $\Gamma^{(c_m)}$ of Γ and $v_{c_m} = a_{c_m} = 0$ if and only of $F_m(a_0, a_1, \ldots, a_{c_m}) = (0, \ldots, 0, 0)$ holds. We remark that $(a_0, a_1, \ldots, a_{c_m})$ could be a critical point of $\Gamma^{(c_m)}$. Then formally, the variable v_{d_m} does not cause the critical point although its radicand or divisor variable v_{c_m} is zero; compare with Example 7.3.1.

Now we assume that Γ is a GSP over $\mathbb{K} = \mathbb{R}$ and show how the interval methods for solving square systems of equations explained in Appendix A.5 could be applied to approximate all critical points of Γ that are contained in a given box $\mathbf{A} = (A_0, A_1, \ldots, A_n) \in (\mathrm{I}(\mathbb{R}))^{n+1}$. We refer to the complex situation ($\mathbb{K} = \mathbb{C}$) at the end of this section.

The idea is to apply the Krawczyk method, which is treated in Appendix A.5, to the functions $F_1, \ldots, F_d$. The Krawczyk method applied to a function F_m outputs a list of boxes $\mathbf{X}$ with small widths that are contained in the initial box, where each box contains exactly one root of F_m or has an unknown status; every root of F_m is contained in one of these boxes. As mentioned in [35, Chap. 4], a box of unknown status usually occurs if it contains a zero that is a singular point of the function. To apply the Krawczyk method, we have to ensure that every box $\mathbf{A} = (A_0, A_1, \ldots, A_{c_m})$ with $\mathrm{w}(A_i) > 0$ for all $i = 0, 1, \ldots, c_m$ contains a regular point $(a_0, a_1, \ldots, a_{c_m}) \in \mathbf{A}$. Here, $\mathrm{w}(A_i) = [\min A_i, \max A_i]$ is the width of the interval A_i. We recall that $(a_0, a_1, \ldots, a_{c_m})$ is a regular point of F_m if and only if the Jacobian F_m' of F_m in the point $(a_0, a_1, \ldots, a_{c_m})$ has full rank. We have to compute or to approximate the inverse of $F_m'(a_0, a_1, \ldots, a_{c_m})$.

Claim 7.4.1. Let $\mathbf{A} = (A_0, A_1, \ldots, A_{c_m})$ be a box with $\mathrm{w}(A_i) > 0$ for all $i = 0, 1, \ldots, c_m$. Then $\mathbf{A}$ contains a point $(a_0, a_1, \ldots, a_{c_m})$ such that the Jacobian F'_m of F_m in the point $(a_0, a_1, \ldots, a_{c_m})$ has full rank.

Proof. We give an outline, the elaboration of a detailed proof will be subject to future work.

The Jacobian F'_m of F_m in an arbitrary point $(z, v_1, \ldots, v_n)$ is a square matrix whose entries are the partial derivatives of the coordinate functions of F_m. For the dependent variables v_i, the coordinate functions are the polynomials $\mathcal{P}_{v_i}$, which are polynomials in v_i and in at most two other variables of Γ. Thus, the Jacobian F'_m of F_m has the following structure

$$F'_m(z, v_1, \ldots, v_{c_m}) = \begin{pmatrix} * & * & 0 & \cdots & \cdots & \cdots & 0 \\ * & * & * & 0 & \cdots & \cdots & 0 \\ \vdots & & \ddots & \ddots & \ddots & & \vdots \\ * & \cdots & \cdots & \cdots & * & * & 0 \\ * & \cdots & \cdots & \cdots & \cdots & * & * \\ 0 & 0 & \cdots & \cdots & 0 & 0 & 1 \end{pmatrix},$$

where $*$ denotes a possibly nonzero entry. The entries on the upper right secondary diagonal are either 1, the divisor variable v_{c_m} of a division variable $v_i = v_{d_m}$, or $2v_{d_m}$ if $v_i = v_{d_m}$ is a square root variable. We have to show that there are values for z, $v_1, \ldots, v_n$ in the given box $\mathbf{A}$ such that F'_m is a regular matrix. Furthermore, it should be possible to determine these values easily. □

Now, we consider complex GSPs with one free variable. To use the Krawczyk method in this situation, we split every variable of Γ into its real and imaginary part. In the functions F_m, we split every coordinate function into its real and imaginary part, as well. Let $\tilde{F}_m = \tilde{F}_m(\Gamma)\colon \mathbb{R}^2 \times \mathbb{R}^{2c_m} \to \mathbb{R}^{2c_m} \times \mathbb{R}^2$ be the resulting function. By switching to the complex situation in this way, we lose a lot of the simplicity of the functions F_m. However, $\mathbb{K} = \mathbb{C}$ is the more interesting case, since we would like to use this method for the treatment of critical points. Here, it would be nice to have a method to determine whether a modified path of the free variable catches further singularities; see Sections 7.2 and 7.3.

Chapter 8

Algebraic Functions

Geometric Straight-Line Programs are closely related to algebraic functions. The dependent variables of a GSP Γ are algebraic functions in the free variables of Γ. Formally, this correlation can be shown using resultants. The paths of the dependent variables in a continuous evaluation are basically liftings of the paths of the free variables to the corresponding Riemann surfaces.

In this chapter, we give an introduction to algebraic functions in one variable and its Riemann surfaces; see Sections 8.1-8.3. We point out the connection of Geometric Straight-Line Programs and continuous evaluations to algebraic functions in Sections 8.4 and 8.5. Finally, we recall the notion of covering maps in Section 8.6 and show the uniqueness and existence of continuous evaluations. Here, we only require that the paths of the free variables are continuous paths, we do not assume differentiability.

8.1 A Brief Introduction

We give a brief introduction to algebraic functions in one variable. A detailed description can be found in [32, 22, 47]. We only give a sketch of many beautiful old theorems and constructions. The aim is to explain the structure of algebraic functions in one complex variable without using too many formal concepts.

Definition 8.1.1. *An* algebraic function $w = w(z)$ *is a multivalued analytic function that fulfills a polynomial equation* $P(z, w) = 0$ *with* $\deg_w(P) \geq 1$*. The coefficients of* P *are complex numbers.*

We rewrite the polynomial equation $P(z, w) = 0$ as

$$P(z, w) = a_n(z)w^n + a_{n-1}(z)w^{n-1} + \cdots + a_1(z)w + a_0(z) = 0$$

where the coefficients $a_i(z)$ are complex polynomials in the variable z. If P is irreducible and if $a_n(z) \not\equiv 0$, then n is the degree of the algebraic function [61, p. 91]. If we allow rational functions as coefficients, we can assume $a_n(z) = 1$. We remark that using polynomials or rational functions as coefficients is equivalent, since we can multiply the equation $P(z,w) = 0$ with the least common denominator of the coefficients $a_i(z)$. Moreover, the rational functions on $\mathbb{C}$ are the meromorphic functions on the Riemann sphere $\hat{\mathbb{C}} = \mathbb{C} \cup \{\infty\}$; see [47, Sect. 1.6.5]. If $P(z,w)$ has polynomials as coefficients and if $a_n(z) \equiv 1$, then the corresponding algebraic function is called an *entire algebraic function*.

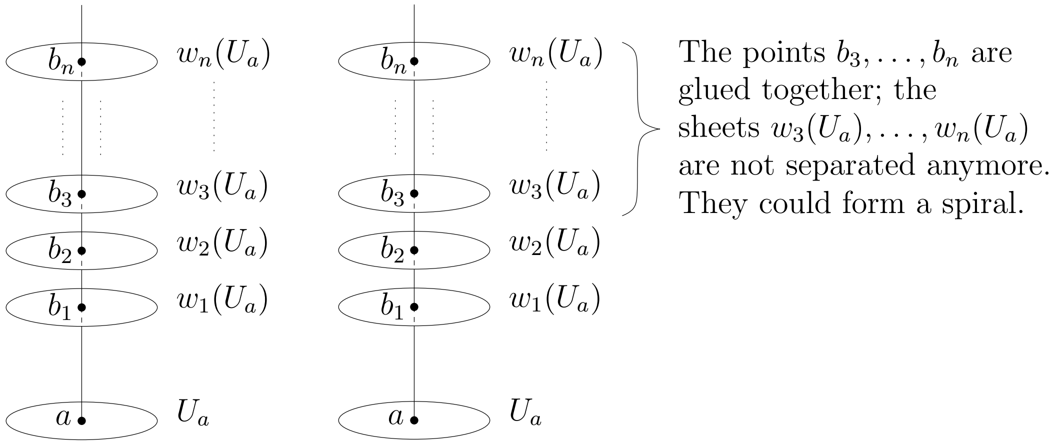
Example 8.1.2.

1. The square root function $w(z) = \sqrt{z}$ is algebraic since it fulfills the polynomial equation $P(z,w) = w^2 - z = 0$. This function is multivalued since it takes two different values for $z \neq 0$: $\sqrt{z}$ or $-\sqrt{z}$.
2. Rational functions are algebraic since a rational function $w(z) = \frac{f(z)}{g(z)}$ with $f(z),\ g(z) \in \mathbb{C}[z]$ and $g(z) \neq 0$ fulfills the polynomial equation $P(z,w) = g(z)w - f(z) = 0$.

If $z = z_0$ is fixed with $a_n(z_0) \neq 0$, then $P(z_0,w)$ is a polynomial in w with degree n and coefficients in $\mathbb{C}$. Thus, $P(z_0,w)$ has at most n distinct zeros. We are interested in the set of all solutions of the equation $P(z,w)$, $z \in \mathbb{C}$. In Section 8.3, we describe how these solutions form the Riemann surface of the algebraic function defined by $P(z,w) = 0$. Using this Riemann surface, the algebraic function can be defined globally without ambiguities.

Locally, an algebraic function defined by an irreducible polynomial $P(z,w) = a_n(z)w^n + a_{n-1}(z)w^{n-1} + \cdots + a_1(z)w + a_0(z) = 0$ can be described via the implicit function theorem [51, Sect. 1]: Let $a \in \mathbb{C}$ with $a_n(a) \neq 0$ such that there is no $b \in \mathbb{C}$ with $P(a,b) = 0 = \frac{\partial P}{\partial w}(a,b)$. Hence, the polynomial $P(a,w)$ has exactly n distinct roots $b_1,\ldots,b_n$. By the implicit function theorem, there is a neighborhood $U_a \subset \mathbb{C}$ of a such that there is a holomorphic function $w_i(z)$ on U_a with $w_i(a) = b_i$ and $P(z,w_i(z)) = 0$ for all $z \in U_a$; $i = 1,\ldots,n$. If U_a is chosen small enough, then $w_i(z) \neq w_j(z')$ holds for all $z,\ z' \in U_a$ and $i \neq j$. Since the equation $P(z,w) = 0$ has at most n solutions, all solutions (z,w) with $z \in U_a$ lie on the graph of one of the functions w_i. In other words, if $(z,w) \in U_a \times \mathbb{C}$ with $P(z,w) = 0$, then $w = w_i(z)$ holds for an $i = 1,\ldots,n$; see Figure 8.1.

A point $a \in \mathbb{C}$ with $a_n(a) \neq 0$ and $P(a,w) = 0 = \frac{\partial P}{\partial w}(a,w)$ for $w \in \mathbb{C}$ might lead to a *branch point* of the algebraic function; see page 122. The condition $P(a,w) = 0 = \frac{\partial P}{\partial w}(a,w)$ implies that the equation $P(a,w) = 0$ has multiple roots. Let $w_1,\ ,w_2,\ldots,\ w_l$ be the roots of $P(a,w) = 0$, let λ_i be the multiplicity of the root w_i. Thus $\lambda_i \geq 2$ holds for an $i = 1,\ldots,l$. Since $a_n(a) \neq 0$, the polynomial $P(a,w) \in \mathbb{C}[w]$ has degree n and $\lambda_1 + \ldots, \lambda_l = n$. Thus at the point $z = a$, λ_i solution functions $w_j(z)$ meet in the solution w_i, in other words, $w_j(a) = w_i$ for λ_i solutions $w_j(z)$; $i = 1,\ldots,l$.



(a) All roots of $P(a, w)$ are simple roots.

(b) $P(a, w)$ has a multiple root $b_3 = \cdots = b_n$.

Figure 8.1: The local situation of the algebraic function defined by the polynomial $P(z, w)$ is shown.

If the polynomial $P(z, w) = a_n(z)w^n + a_{n-1}(z)w^{n-1} + \cdots + a_1(z)w + a_0(z) = 0$ is irreducible, then there are only finitely many $z \in \mathbb{C}$ such that the equations $P(z, w) = 0$ and $\frac{\partial P}{\partial w}(z, w) = 0$ have a simultaneous solution $w \in \mathbb{C}$; see [51, Prop. 1, p. 4]. This claim also holds for reduced polynomials $P(z, w) \in \mathbb{C}[z][w]$; a reduced polynomial is a polynomial without multiple roots (over an extension field). This generalization can be proved using the discriminant of P; see [47, p. 119], Definition 8.2.3 and Lemma 8.2.4.

The functions $w_i(z)$ are called branches of the algebraic function defined by P. If their domains are chosen properly and as large as possible, then the branches can be glued together at so-called branch cuts. This process forms an n-fold covering space called Riemann surface of the algebraic function defined by P; in [32, 7. Vorlesung], Hensel and Landsberg give a beautiful and descriptive deduction of the construction. The images of the functions w_i in the Riemann surface are called sheets. The Riemann surface of an algebraic function defined by a polynomial $P(z, w)$ is connected if and only if P is irreducible [32, p. 110]. More generally, the following holds [32, p. 110]: The number of irreducible factors of a polynomial $P(z, w)$ equals the number of connected components of the Riemann surface of the corresponding algebraic function. The number of sheets of each component equals the w-degree of the corresponding irreducible factor of P. This theorem enables us to investigate the irreducible factors of a polynomial P and the corresponding Riemann surfaces, separately.

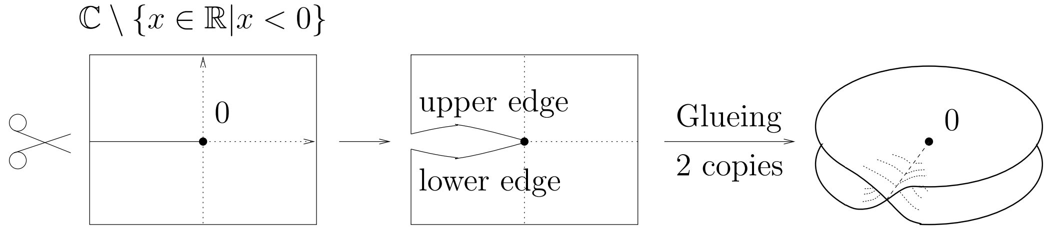


Figure 8.2: Construction of the Riemann surface of the square root function $z \mapsto \sqrt{z}$ with "scissors and glue".

Example 8.1.3. We consider the square root function, which is defined by $P(z, w) = w^2 - z$; see Example 8.1.2. On the simply connected region $G := \mathbb{C} \setminus \{x \in \mathbb{R} \mid x \leq 0\}$ we have two different branches of the root function, which can be defined using the branches of the complex logarithm function [63, Sect. II.3]. These two branches reflect the two possible choices for the root function, for example $\sqrt{4} = \pm 2$. More generally, if $z = re^{i\phi} \in \mathbb{C} \setminus \{0\}$ is a complex number written in polar coordinates, then we either have $\sqrt{z} = \sqrt{r}e^{i\phi/2}$ or $\sqrt{z} = -\sqrt{r}e^{i\phi/2} = \sqrt{r}e^{i\phi}e^{i\pi} = \sqrt{r}e^{i(\phi/2+\pi)}$. The two branches do not match at the negative real line. Instead, cycling once around the unit circle from -1 to -1 enforces a change of the branch. If a point $z = re^{i\phi}$ moves once around the unit circle, then its argument ϕ increases (or decreases) by 2π. This implies that the argument of "$\sqrt{z}$" changes by π, and the final point of the motion differs from the starting point.

This observation leads to a descriptive construction of the Riemann surface of the square root function. We take two copies of $G = \mathbb{C} \setminus \{x \in \mathbb{R} \mid x \leq 0\}$, one copy for each branch. We glue these copies along the negative real line in the following way: We cut both copies of G along the negative real line. Each cut leads to a double edge; see Figure 8.2. One of these two edges of G delimits the upper left quadrant of $\mathbb{C}$, the other one delimits the lower left quadrant of $\mathbb{C}$. These edges are called upper and lower edge, respectively. The lower edge of the first copy of G is glued with the upper edge of the second copy of G, and the lower edge of the second copy of G is glued with the upper edge of the first copy of G. The resulting surface is shown in Figure 8.2, it is the Riemann surface of the square root function. The self-intersection is due to the embedding into $\mathbb{R}^3$. We observe that cycling once around the unit circle results in a change of the sheets of the surface; compare with Figure 8.8 and Figure 8.9 from Example 8.6.2 from page 142. On this Riemann surface, the square root function can be defined uniquely; see Section 8.3.

Having the intuition of Riemann surfaces in mind as explained in Example 8.1.3, we come back to the notion of *branch points*. Let z_0 be a point with $a_n(z_0) \neq 0$

and $P(z_0, w) = 0 = \frac{\partial P}{\partial w}(z_0, w)$ for some $w \in \mathbb{C}$. In a neighborhood of z_0, the Riemann surface looks like stacked "spirals". Let s be the number of spirals, and let $a_1, \dots, a_s$ be the numbers of their sheets. Clearly we have $a_1 + \cdots + a_s = n$. If at least one of the numbers a_i is greater than one, then z_0 is an (algebraic) branch point of the algebraic function defined by P. This descriptive approach can be formalized using the notion of analytic continuation. The Encyclopaedia [61, p. 91] gives an overview of common notions from the area of algebraic functions.

We remark that, locally, algebraic functions can be described by Puiseux expansions; see [47, Sect. 6.3] and [32, 4. Vorlesung]. In [32, 5. Vorlesung], some important basic properties of algebraic functions are summarized. The results are based on the notion of an "Ordnungszahl" (order) of a rational function $f(z) = \frac{g(z)}{h(z)}$ in a point $\alpha \in \mathbb{C} \cup \{\infty\}$, which is the multiplicity m of the root $z = \alpha$ or the negative of the multiplicity of the pole $z = \alpha$ of $f(z)$, respectively. Here, we only consider finite points $\alpha \in \mathbb{C}$. This simplifies the presentation. Moreover, we assume that the coefficient polynomials $a_i(z)$ of P have no common zeros. Otherwise, we divide the equation $P(z, w) = 0$ by the corresponding factor.

We fix a point $\alpha \in \mathbb{C}$, and consider the equation $P(\alpha, w) = 0$. Since the polynomials $a_i(z)$ do not have a common zero, there must be an index $i \in \{1, \dots, n\}$ with $a_i(\alpha) \neq 0$. Let $g \in \{0, \dots, n\}$ be the smallest index with $a_g(\alpha) \neq 0$, and let $k \in \{0, \dots, n\}$ be the largest index with $a_k(\alpha) \neq 0$. A generalization of Theorem 8.1.4 can be found in [32, p. 74].

Theorem 8.1.4. *Let $w_1(z), \dots, w_n(z)$ be the n solutions of the equation $P(z, w) = 0$ taking values in the Riemann sphere $\hat{\mathbb{C}} = \mathbb{C} \cup \{\infty\}$, and let $z = \alpha \in \mathbb{C}$. Then, with the notation from above,*

- *for $n - k$ solutions $w_i(z)$ we have $w_i(\alpha) = \infty$,*
- *for g solutions $w_i(z)$ we have $w_i(\alpha) = 0$, and*
- *the remaining $k - g$ solutions take nonzero and finite values at the point $z = \alpha$. Moreover, these values are the solutions w of the equation*

$$a_k(\alpha) w^{k-g} + \cdots + a_g(\alpha) = 0.$$

Proof. Since we investigate the solutions at infinity as well, we compute the homogenization $\tilde{P}(\alpha, w, \tilde{w})$ of the polynomial $P(\alpha, w) = a_n(\alpha) w^n + a_{n-1}(\alpha) w^{n-1} + \cdots + a_1(\alpha) w + a_0(\alpha) \in \mathbb{C}[w]$ in the variable w:

$$\tilde{P}(\alpha, w, \tilde{w}) = a_n(\alpha) w^n + a_{n-1}(\alpha) w^{n-1} \tilde{w} + \cdots + a_1(\alpha) w \tilde{w}^{n-1} + a_0(\alpha) \tilde{w}^n$$

Thus, instead of considering solutions $w \in \mathbb{C}$ of $P(\alpha, w) = 0$, we consider solutions $(w, \tilde{w}) \in \mathbb{CP}^1$ of the homogeneous equation $\tilde{P}(\alpha, w, \tilde{w}) = 0$, where $\mathbb{CP}^1$

is the complex projective line. We can apply this approach since the complex projective line $\mathbb{CP}^1$ is in fact the Riemann sphere $\hat{\mathbb{C}}$.

By construction of g and k, we have $a_0(\alpha) = ... = a_{g-1}(\alpha) = 0$ and $a_{k+1}(\alpha) = \cdots = a_n(\alpha) = 0$, hence

$$\begin{aligned} 0 &= \tilde{P}(\alpha, w, \tilde{w}) \\ &= \underbrace{a_n(\alpha)w^n + \dots}_{=0} + a_k(\alpha)w^k\tilde{w}^{n-k} + \cdots + a_g(\alpha)w^g\tilde{w}^{n-g} + \underbrace{\cdots + a_0(\alpha)\tilde{w}^n}_{=0} \\ &= a_k(\alpha)w^k\tilde{w}^{n-k} + \cdots + a_g(\alpha)w^g\tilde{w}^{n-g} \\ &= w^g\tilde{w}^{n-k}\left(a_k(\alpha)w^{k-g} + \cdots + a_g(\alpha)\tilde{w}^{k-g}\right). \end{aligned}$$

We can read off that g solutions are 0, $n-k$ solutions are ∞, and the remaining solutions are the zeros of the polynomial $\tilde{Q}(w, \tilde{w}) := a_k(\alpha)w^{k-g} + \cdots + a_g(\alpha)\tilde{w}^{k-g}$. The finite solutions of $\tilde{Q}(w, \tilde{w}) = 0$ are the $k-g$ solution of the non-homogeneous equation $a_k(\alpha)w^{k-g} + \cdots + a_g(\alpha) = 0$. Since $a_g(\alpha) \neq 0$, these solutions are non-zero. Moreover, the equation $\tilde{Q}(w, \tilde{w}) = 0$ has no solutions at infinity since $a_k(\alpha) \neq 0$. □

Example 8.1.5. We consider the polynomial $P(z, w) = w^2 - 2zw + (z^2 - z)$ describing the function $w(z) = z \pm \sqrt{z}$. Thus we have $n = \deg_w(P) = 2$, and $a_0(z) = z^2 - z$, $a_1(z) = -2z$, and $a_2(z) = 1$. We investigate the following points:

$z = 0$**:** We have $a_2(0) = 1 \neq 0$, $a_1(0) = 0$ and $a_0(0) = 0$, hence we get $g = k = 2$. Theorem 8.1.4 implies that both solutions $w_1(z)$ and $w_2(z)$ are zero at the point $z = 0$.

$z = 1$**:** Here, we observe $a_1(1) = -2$, $a_2(1) = 1$, and $a_0(1) = 0$, thus $g = 1$ and $k = 2$. By Theorem 8.1.4, one function $w_i(z)$ is zero for $z = 1$. The other function $w_i(z)$ takes a nonzero finite value. To compute this value, we consider the equation

$$\begin{aligned} 0 &= a_k(1)w^{k-g} + \cdots + a_g(1) \\ &= a_2(1)w + a_1(1) \\ &= w - 2 \end{aligned}$$

having the solution $w = 2$. Indeed, the functions $w_1(z)$ and $w_2(z)$ take the values 0 and 2 for $z = 1$.

$z = 2$**:** We compute $a_0(2) = 2$, $a_1(2) = -4$ and $a_2(2) = 1$ and get $g = 0$ and $k = 2$. Thus the two solutions $w_1(z)$ and $w_2(z)$ take non-zero finite values. These values are the solutions of

$$\begin{aligned} 0 &= a_k(2)w^{k-g} + \cdots + a_g(2) \\ &= a_2(2)w^2 + a_1(2)w + a_0(2) \\ &= w^2 - 4w + 2, \end{aligned}$$

hence $w_{1/2} = 2 \pm \sqrt{2}$ as expected.

By Theorem 8.1.4, the algebraic function defined by $P(z,w) = 0$ has a zero if and only if $g > 0$. Since $g > 0$ is equivalent to $a_0(\alpha) = 0$, the zeros of the algebraic function occur at the zeros of the coefficient polynomial $a_0(z)$. Similarly, the algebraic function defined by $P(z,w) = 0$ has a pole at $z = \alpha$ if and only if $k < n$. Since $k < n$ is equivalent to $a_n(\alpha) = 0$, the algebraic function defined by $P(z,w) = 0$ has a pole at $z = \alpha$ if and only if $z = \alpha$ is a zero of the coefficient polynomial $a_n(z)$. We summarize these observations:

Corollary 8.1.6. *Let $w_1(z), \ldots, w_n(z)$ be the n solutions of $P(z,w) = 0$. There is only a finite number of points $\alpha \in \mathbb{C}$ where one of the n solutions $w_i(z)$ is zero or infinite. These points are the zeros of $a_0(z)$ or $a_n(z)$, respectively, or we have $\alpha = \infty$.*

8.2 Resultants, Discriminants, and their Applications to Algebraic Functions

We briefly introduce the resultant of two polynomials and the discriminant of a polynomial. A detailed introduction to this well known topic of mathematics can be found in [66, 12]. We use the discriminant to determine the branch points of the algebraic function defined by $P(z,w) = 0$. Via the resultant, we examine the basic arithmetic operations on the field of algebraic functions over $\mathbb{C}$.

Definition 8.2.1. Resultant of two Polynomials
Let $f(x) = a_m x^m + \cdots + a_i x + a_0$ and $g(x) = b_n x^n + \cdots + b_1 x + b_0$ be polynomials of degree m and n. Then the resultant $\operatorname{Res}(f,g) = \operatorname{Res}_x(f,g)$ is the determinant of the $(m+n) \times (m+n)$ Sylvester matrix of f and g:

$$\operatorname{Res}(f,g) = \det \begin{pmatrix} a_m & a_{m-1} & \cdots & \cdots & a_0 & & & \\ & a_m & a_{m-1} & \cdots & \cdots & a_0 & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & & a_m & a_{m-1} & \cdots & \cdots & a_0 \\ b_n & b_{n-1} & \cdots & \cdots & b_0 & & & \\ & b_n & a_{n-1} & \cdots & \cdots & b_0 & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & & b_n & b_{n-1} & \cdots & \cdots & b_0 \end{pmatrix}$$

The missing entries in the matrix are 0. If both polynomials are constants ($m = n = 0$), then the resultant can be defined as follows [12, p. 156]:

$$\operatorname{Res}(f,g) = \operatorname{Res}(a_0,b_0) = \begin{cases} 0 & \textit{if either } a_0 = 0 \textit{ or } b_0 = 0 \\ 1 & \textit{if } a_0 \neq 0 \textit{ and } b_0 \neq 0. \end{cases}$$

By definition, the resultant of two non-constant polynomials f and g is an integer polynomial in the coefficients a_i and b_j of f and g; $i = 0, \dots, m$, $j = 0, \dots, n$. Plugging in the values for a_i and b_j defined by the concrete polynomials f and g leads to $\mathrm{Res}(f, g) \in k$, if k is the underlying field. In our situation, we have $k = \mathbb{C}$, $k = \mathbb{C}(z)$, the field of rational functions on $\mathbb{C}$, or we choose k to be the field of algebraic functions in one variable, which is the algebraic closure of $\mathbb{C}(z)$. The resultant has several nice properties and applications, for example in elimination theory. Via the resultant of two polynomials f and g, we can determine whether f and g have a common root [66, 12]:

Lemma 8.2.2. *[12, Prop. 8, p. 151] Let k be a field, and let $f, g \in k[x]$ be polynomials of positive degree. Then f and g have a common factor in $k[x]$ if and only if* $\mathrm{Res}(f, g) = 0$.

We mentioned on page 121 that the discriminant of a polynomial is useful to investigate the branch points of the Riemann surface of the algebraic function defined by a polynomial equation $P(z, w) = 0$.

Definition 8.2.3. *[66, Sect. 6.6]* Discriminant of a Polynomial
Let k be a field and $\bar{k}$ its algebraic closure. Let $f(x) = a_m x^m + \cdots + a_i x + a_0 \in k[x]$ be a polynomial with $m := \deg f \geq 1$. The discriminant of $f(x)$ is defined as

$$\mathrm{disc}(f) := a_m^{2m-2} \Pi_{1 \leq i < j \leq m} (\alpha_i - \alpha_j)^2$$

where $\alpha_1, \dots, \alpha_m \in \bar{k}$ are the (not necessarily distinct) roots of f.

We observe that $\mathrm{disc}(f) = 0$ holds if and only if f has a multiple root. Yap [66] shows $\mathrm{disc}(f) \in k$ using the theory of symmetric functions. We recall that the discriminant of f equals the resultant of f and its derivative f' up to a constant factor:

Lemma 8.2.4. Discriminant and Resultant
Let $f(x) = a_m x^m + \cdots + a_1 x + a_0 \in k[x]$ be a polynomial of degree $m \geq 1$. Then the following relation holds:

$$a_m \cdot \mathrm{disc}(f) = -1^{\binom{m}{2}} \mathrm{Res}(f, f').$$

If $P(z, w) \in \mathbb{C}[z][w]$ is irreducible, then the branch points of the algebraic function defined by $P(z, w) = 0$ are zeros of the discriminant of the polynomial $P(z, w) \in \mathbb{C}[z][w] \subset \mathbb{C}(z)[w]$. This is a necessary condition for the existence of branch points, only; see page 122. The point at infinity might be a branch point, as well. Let $w_i(z), \dots, w_n(z)$ be the n solution functions of $P(z, w) = 0$. Then the discriminant $\mathrm{disc}_w(P)$ of P is the polynomial

$$\mathrm{disc}_w(P) := a_n(z)^{2n-2} \Pi_{1 \leq i < j \leq n} \left(w_i(z) - w_j(z)\right)^2 \in \mathbb{C}[z].$$

By definition, the discriminant of P vanishes if and only if two solutions $w_i(z)$ and $w_j(z)$ with $i \neq j$ coincide. To compute the discriminant we use Lemma 8.2.4:

$$a_n(z)\mathrm{disc}_w(P) = (-1)^{\binom{n}{2}}\mathrm{Res}_w\left(P(z,w), \frac{\partial P(z,w)}{\partial w}\right).$$

The discriminant $\mathrm{disc}_w(P(z,w))$ of P is a polynomial in the coefficients $a_i(z)$ of $P(z,w) = a_n(z)w^n + \cdots + a_1(z)w + a_0(z)$. Thus, $\mathrm{disc}_w(P)$ is a polynomial in the variable z, itself. At the zeros of the polynomial $\mathrm{disc}_w(P)$, two or more solutions $w_i(z)$ of $P(z,w)$ meet.

Example 8.2.5. We consider our favorite example $w = \pm\sqrt{z^2-1}$. This algebraic function is the solution of the polynomial equation $P(z,w) = w^2 - z^2 + 1 = 0$. To examine the branch points, we compute the discriminant of P via the resultant. Note that $m = \deg_w P = 2$, $a_m(z) = a_2(z) = 1$, and $\binom{m}{2} = 1$.

$$\begin{aligned}
\mathrm{disc}_w(P) &= -\mathrm{Res}_w\left(P(z,w), \frac{\partial P(z,w)}{\partial w}\right) = -\mathrm{Res}_w(w^2 - z^2 + 1,\ 2w) \\
&= \det\begin{pmatrix} 1 & 0 & -z^2+1 \\ 2 & 0 & 0 \\ 0 & 2 & 0 \end{pmatrix} \\
&= 4(-z^2+1) = 4(1-z)(1+z)
\end{aligned}$$

As expected, the branch points can occur at $z = \pm 1$, only.

Now we use the resultant to examine the basic arithmetic operations of algebraic functions. In [66], this approach is used to investigate algebraic numbers. Yap's description [66] is quite general and can be applied to algebraic functions as well.

Theorem 8.2.6. *Let $P(z,w)$ and $Q(z,w)$ be polynomials with $\deg_w(P) = m$ and $\deg_w(Q) = n$ defining the algebraic functions $p(z)$ and $q(z)$. Then the following holds:*

1. *The function $(q \pm p)(z)$ is algebraic; it is a zero of the polynomial $R(z,w) := \mathrm{Res}_y\left(P(z,y), Q(z, w \mp y)\right)$, and $\deg_w(R) \leq mn$.*
2. *The function $(pq)(z)$ is algebraic; it is a zero of the polynomial*

$$R(z,w) := \mathrm{Res}_y\left(P(z,y), y^n Q(z, \frac{w}{y})\right),$$

   and $\deg_w(R) \leq mn$.
3. *The function $1/p(z)$ is algebraic; it is a zero of the polynomial $R(z,w) := w^m P(z, \frac{1}{w})$, and $\deg_w R \leq m$.*

4. *The function* $\sqrt[k]{p(z)}$ *is algebraic; it is a zero of the polynomial* $R(z,w) := P(z,w^k)$*, and* $\deg_w R = m^k$.

5. *The inverse function of an algebraic function is an algebraic function.*

Proof. To see (1), we observe that $y = p(z)$ is a common zero of the polynomials $P(z,y)$ and $Q(z,(q(z)+p(z))-y)$. Hence, $q(z)+p(z)$ is a zero of $R(z,w)$. We can treat (2) in a similar way. To consider the reciprocal of an algebraic function as in (3), we replace the second variable w by $\frac{1}{w}$. Multiplying the resulting equation by w^m leads to an equivalent polynomial equation. (4) follows immediately. To prove (5), we interchange the roles of z and w. The resulting polynomial equation $\tilde{P}(z,w) = 0$ defines the inverse of the function of $p(z)$. □

Note that Theorem 8.2.6 holds for multivariate algebraic functions, as well. Multivariate algebraic functions are algebraic functions in more than one variable and fulfill a polynomial equation $P(z_1, \ldots, z_k, w) = 0$; see [61, p. 91].

8.3 Formal Definition of the Riemann Surface of an Algebraic Function

The previous treatment of algebraic functions was based on Hensel's and Landsberg's descriptive approach [32]. Now, we give a formal definition of the Riemann surface of an algebraic function and try to explain it. A detailed description is given by Lamotke in [47]. Again, we consider an equation $P(z,w) = 0$. We assume that $P(z,w) = w^n + a_{n-1}(z)w^{n-1} + \cdots + a_1(z)w + a_0(z)$ is a monic polynomial (i.e., $a_n(z) = 1$) where the coefficients $a_i(z)$ are meromorphic functions on the Riemann sphere $\hat{\mathbb{C}} = \mathbb{C} \cup \{\infty\}$. We denote the field of meromorphic functions on $\hat{\mathbb{C}}$ by $\mathcal{M}(\hat{\mathbb{C}})$. Thus $P(z,w)$ can be regarded as a polynomial $P \in \mathcal{M}(\hat{\mathbb{C}})[w]$. The polynomial P decomposes into linear factors over an extension field L, and we have $P = (w-\lambda_1)^{n_1} \cdot \cdots \cdot (w-\lambda_r)^{n_r}$ with $\lambda_1, \ldots, \lambda_r \in L$ and $n_1 + \cdots + n_r = n = \deg P = \deg_w P(z,w)$. If P has multiple roots, we consider its reduced polynomial $Q = (w-\lambda_1) \cdot \cdots \cdot (w-\lambda_r)$. The reduced polynomial Q has the same roots as P, and all roots of Q are simple roots. Again, Q is a polynomial with coefficients in $Q \in \mathcal{M}(\hat{\mathbb{C}})[w]$; see [12, pp. 178-179].

The main idea of the Riemann surface of an algebraic function is to transform the multivalued function $\tilde{f} : \hat{\mathbb{C}} \to \hat{\mathbb{C}}$ (or $\tilde{f} : \mathbb{C} \to \mathbb{C}$), $z \mapsto \{w \in \hat{\mathbb{C}} \mid P(z,w) = 0\}$, into an "ordinary" function $f : X \to \hat{\mathbb{C}}$ by fanning out the domain. To achieve this, all points in the image set $\tilde{f}(z) = \{w \in \mathbb{C} \mid P(z,w) = 0\}$ need a separate preimage in X. First, we give an informal overview of the construction. We define $X := \{(z,w) \in \hat{\mathbb{C}} \times \hat{\mathbb{C}} \mid P(z,w) = 0\}$. The projection $\eta : X \to \hat{\mathbb{C}}$, $(z,w) \mapsto z$ to the

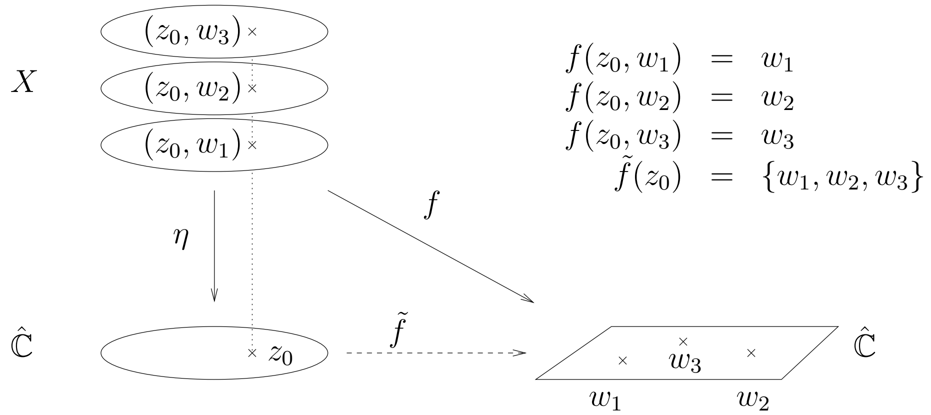


Figure 8.3: Illustration of the construction of X, η, and f.

first coordinate maps all points (z, w) with $w \in \tilde{f}(z)$ to z, and we have $\eta^{-1}(z) = \{(z, w) \,|\, w \in \hat{\mathbb{C}} \text{ and } P(z, w) = 0\} = \{(z, w) \,|\, w \in \tilde{f}(z)\} = \{z\} \times \tilde{f}(z)$. The projection η is used to fan out the domain of $\tilde{f}$. The map $f \colon X \to \hat{\mathbb{C}}$, $(z, w) \mapsto w$, is the projection to the second coordinate and we have $\tilde{f}(z) = f(\eta^{-1}(z))$ as desired; see Figure 8.3.

We will see that the solution set X is a Riemann surface, the projection η is a covering map of Riemann surfaces, and the function f is a meromorphic function on X. The triple (X, η, f) is called the Riemann surface of $\tilde{f}$. In German textbooks, the terms *Riemannsches Gebilde* or *algebraisches Gebilde* are used, as well. These notions distinguish the generic definition of Riemann surfaces and the Riemann surfaces of algebraic functions more clearly. A *Riemann surface* is a Hausdorff space X together with a holomorphic structure on X; see [47, Sect. 1.1]. Due to the holomorphic structure, a Riemann surface is a twodimensional topological manifold. The aim of this section is to understand that every reduced, monic polynomial $P \in \mathcal{M}(\hat{\mathbb{C}})[w]$ of degree $n \geq 1$ defines a "Riemannsches Gebilde"; see Theorem 8.3.3.

Let X be a Riemann surface and $\eta : X \to \hat{\mathbb{C}}$ be an $n$fold covering of Riemann surfaces. This implies that η is an open and holomorphic function. Since $\hat{\mathbb{C}}$ is compact, η only has a finite number of branch points [47, p. 17]. Let $f \colon X \to \hat{\mathbb{C}}$ be a meromorphic function. The projection B of the poles of f along η is finite. We cite the following theorem, which traces back to Riemann.

Theorem 8.3.1. *[47, Sect. 6.1.2]* Characteristic Polynomial
There is exactly one polynomial

$$\chi(z, w) = w^n - s_1(z)w^{n-1} + \cdots + (-1)^n s_n(z) \in \mathcal{M}(\hat{\mathbb{C}})[w]$$

with

$$\chi(z, w) = \Pi_{x \in \eta^{-1}(z)} \left(w - f(x)\right)^{v(\eta, x)}$$

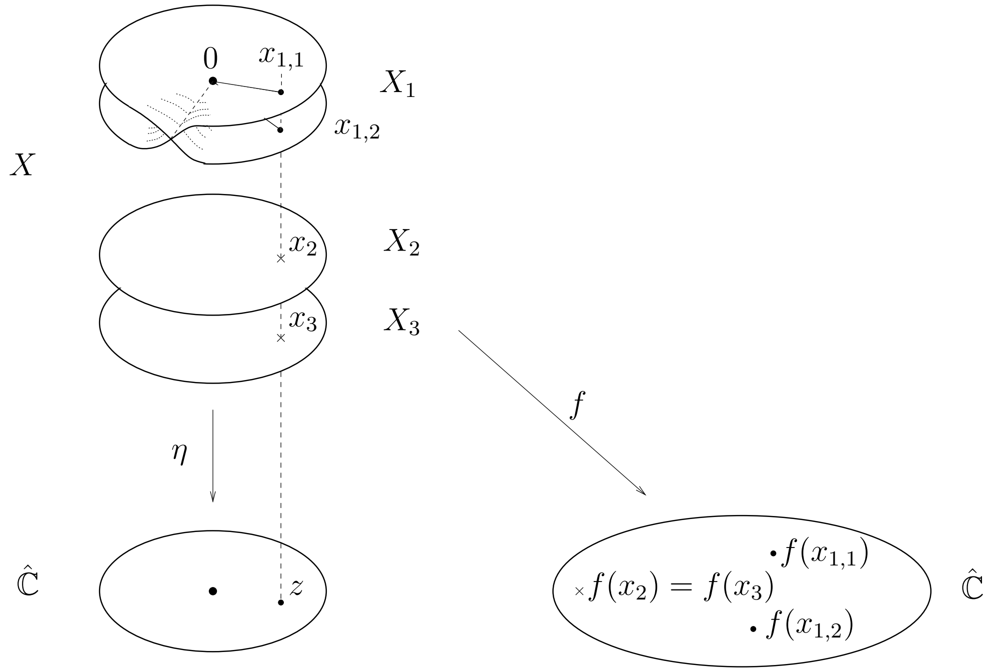


Figure 8.4: Illustration of Example 8.3.2; f maps the components X_2 and X_3 of X in the same way to $\hat{\mathbb{C}}$.

for every point $z \in \hat{\mathbb{C}} \setminus B$. Here, the function $v(\eta, x)$ denotes the winding number of η in the point x. The polynomial χ is called the characteristic polynomial *of f with respect to the covering map η.*

Definition 8.3.1 implies $\chi(\eta(x), f(x)) = 0$. The polynomials in $\mathcal{M}(\hat{\mathbb{C}})[w]$ that cancel f form a principal ideal. This ideal is generated by a unique monic and reduced polynomial P. The polynomial P is called *minimal polynomial* [47, Sect. 6.1.5]. If X is connected, then P is irreducible. The characteristic polynomial χ is a multiple of P. Hence $\deg P \leq n$, and we have $\deg P = n$ if and only if P is the characteristic polynomial. It can be shown that the reduction of the characteristic polynomial of f is the minimal polynomial [47, p. 132]. Moreover, the minimal polynomial P has degree n if and only if the meromorphic function f takes n different finite values in $\mathbb{C}$ along at least one fiber of the $n$fold covering η. In this case, f maps every fiber of a point in $\hat{\mathbb{C}} \setminus E$ to n different finite complex values [47, Sect. 6.1.5], where E is the set of exception points; see page 131.

Example 8.3.2. Let $\eta\colon X \to \hat{\mathbb{C}}$ be the covering from Figure 8.4. The Riemann surface X consists of the connected components X_1, X_2 and X_3. Let $f\colon X \to \hat{\mathbb{C}}$ be a meromorphic function that is defined on X_2 in the same way as on X_3: If $x_2 \in X_2$ and $x_3 \in X_3$ with $\eta(x_2) = \eta(x_3)$, then we have $f(x_2) = f(x_3)$. The characteristic polynomial of f is

$$\begin{aligned}\chi(z,w) &= (w-f(x_{1,1}))\cdot(w-f(x_{1,2}))\cdot(w-f(x_2))\cdot(w-f(x_3))\\ &= (w-f(x_{1,1}))\cdot(w-f(x_{1,2}))\cdot(w-f(x_2))^2\,,\end{aligned}$$

and the minimal polynomial is

$$P(w)=(w-f(x_{1,1}))\cdot(w-f(x_{1,2}))\cdot(w-f(x_2))\,.$$

Since the components X_2 and X_3 of X are treated in the same way, they can be "collapsed" to a new component $\tilde{X}$ without losing information concerning the meromorphic function f. In many applications, the two components X_2 and X_3 result from the fact that zeros of multiplicity r are considered as r distinct zeros. Then, X_2 and X_3 can be seen as one component having multiplicity 2.

The described situation occurs if we consider the polynomial equation $w^2(w^2-2z)=0$. The polynomial $\chi(z,w)=w^2(w^2+2z)$ is the characteristic polynomial of the function $f(z)=\pm\sqrt{\pm\sqrt{z^2}-z}$. If the inner root $\sqrt{z^2}$ is chosen to be $+z$, then the outer root is constantly zero. Thus f is defined on the corresponding two sheets (resp. one sheet with "multiplicity" 2) to be zero. If we choose $\sqrt{z^2}=-z$, then the outer root can take two different values. Hence, we have the same situation as above shown in Figure 8.4. The minimal polynomial of f is $P(z,w)=w(w^2+2z)$.

Let $P(z,w)=w^n+a_{n-1}(z)w^{n-1}+\cdots+a_1(z)w+a_0(z)\in\mathcal{M}(\hat{\mathbb{C}})$ be a reduced monic polynomial. The set E of exception points of P consists of the poles of the meromorphic coefficient functions $a_i(z)\in\mathcal{M}(\hat{\mathbb{C}})$ and the zeros of the discriminant of $P\in\mathcal{M}(\hat{\mathbb{C}})[w]$. Since the functions $a_i(z)\colon\hat{\mathbb{C}}\to\hat{\mathbb{C}}$ are meromorphic, they have a finite number of poles. Since the polynomial P is reduced, the discriminant $\operatorname{disc}(P)\colon\hat{\mathbb{C}}\to\hat{\mathbb{C}}$ only has a finite number of zeros, and the set of exception points E is finite. The exception points are called critical values of the algebraic function defined by P, as well [61, p. 91].

Now, we have all ingredients to formulate and understand the promised theorem.

Theorem 8.3.3. *[47, Sect. 6.2.5]* Existence of "Riemannsche Gebilde"
Every reduced, monic polynomial $P\in\mathcal{M}(\hat{\mathbb{C}})[w]$ of degree $n\geq 1$ is the minimal polynomial of a "Riemannsches Gebilde" (X,η,f) over $\hat{\mathbb{C}}$.

Proof. We follow the proof of [47] and explain the intermediate steps. Let $P\in\mathcal{M}(\hat{\mathbb{C}})[w]$ be a reduced monic polynomial of degree $n\geq 1$.

1. Let E be the finite set of exception points of P. We consider the solution set

$$M:=\{(z,w)\in(\hat{\mathbb{C}}\setminus E)\times\mathbb{C}\mid P(z,w)=0\}$$

and the projection $\pi\colon M \to \hat{\mathbb{C}} \setminus E$, $(z,w) \mapsto z$. Let $(a,b) \in M$; thus we have $P(a,b) = 0$ and $\mathrm{disc}_w P(a,b) \neq 0$ implying $\frac{\partial P}{\partial w}(a,b) \neq 0$. As described on page 120, we apply the implicit function theorem and get a neighborhood U_a of a and W_b of b with:

- For all $z \in U_a$, the function $P(z,w)$ has exactly one zero $g(z)$ in W_b.
- The function $g\colon U_a \to W_b$, $z \mapsto g(z)$, is holomorphic.

The set $(U_a \times W_b) \cap M$ is a neighborhood of the point (a,b). The projection $\pi_{|(U_a \times W_b)\cap M}\colon (U_a \times W_b) \cap M \to U_a$, $(z, g(z)) \mapsto z$, is continuous. By construction, it is bijective. Since the function g is continuous, the inverse function $z \mapsto (z, g(z))$ of $\pi_{|(U_a \times W_b)\cap M}$ is continuous as well. Thus, $\pi_{|(U_a \times W_b)\cap M}$ is a homeomorphism, and the projection $\pi\colon M \to \hat{\mathbb{C}} \setminus E$, $(z,w) \mapsto z$, is a locally topological function between the Hausdorff space M and the Riemann surface $\hat{\mathbb{C}} \setminus E$. This observation shows that M is a two-dimensional topological manifold. Gluing all neighborhoods W_b to the entire surface M corresponds to the descriptive approach of [32, 7. Vorlesung]. This process leads to the common geometric intuition concerning Riemann surfaces of algebraic functions.

2. The holomorphic structure of the Riemann surface $\hat{\mathbb{C}} \setminus E$ is lifted uniquely to M by the local homeomorphisms from step (1); see [47, Sect. 1.2.1]. This lifting process uniquely transforms M into a Riemann surface and the projection $\pi\colon M \to \hat{\mathbb{C}} \setminus E$, $(z,w) \mapsto z$ into a locally biholomorphic function. Since the preimage $\pi^{-1}(z)$ of a point $z \in \hat{\mathbb{C}} \setminus E$ consists of $n = \deg_w P$ points, π is an unbranched $n$fold covering of Riemann surfaces.

3. We consider the function $f\colon M \to \mathbb{C}$, $(z,w) \mapsto w$. By (1), we have $(z,w) = (z, g(z))$ at least locally. Thus $f(z,w) = w = g(z) = g(\pi(z,w)) = g \circ \pi(z,w)$. Since g and π are holomorphic functions, f is holomorphic as well. By construction, f maps every fiber $\pi^{-1}(z)$ one-to-one onto the zeros of $P(z,w)$.

4. Let $X := \{(z,w) \in \hat{\mathbb{C}} \times \hat{\mathbb{C}} \mid P(z,w) = 0\}$ be the set of all solutions. The covering $\pi\colon M \to \hat{\mathbb{C}} \setminus E$ can be extended to a covering $\eta\colon X \to \hat{\mathbb{C}}$, such that the branch points of η are contained in the set E. In this process, the branch points are "bent up". The function $f\colon M \to \mathbb{C}$ can be extended to a meromorphic function $f\colon X \to \hat{\mathbb{C}}$; see [47].

5. By construction, we have

$$P(\eta(z,w), f(z,w)) = P(\pi(z,w), f(z,w)) = P(z,w) = 0$$

for all points $(z,w) \in M$. Since the set E of exception points of P is finite, the set $X \setminus M$ is finite as well. Thus the equation $P(\eta(z,w), f(z,w)) = 0$

holds for all points $(z, w) \in X$, and P is a multiple of the minimal polynomial of f with respect to the covering η. We have seen in (3) that f is injective on the fibers $\pi^{-1}(z)$; $z \in \hat{\mathbb{C}} \setminus E$. Hence the minimal polynomial of f has degree n, and P must be the minimal polynomial of f. □

Example 8.3.4. We consider the square root function that is defined by the polynomial equation $w^2 - z = 0$. The polynomial $P(z, w) = w^2 - z$, seen as a polynomial in $\mathcal{M}(\hat{\mathbb{C}})[w]$, has leading coefficient 1, hence it is monic. In the field of algebraic functions, $P(z, w) = (w + \sqrt{z})(w - \sqrt{z})$ decomposes into two different linear factors, hence P is reduced. The discriminant is $\operatorname{disc}(P) = -\operatorname{Res}_w(w^2 - z, 2w) = 4z$ and has a zero at $z = 0$. The coefficient polynomial $a_0(z) = -z$ has a pole at ∞. Thus the set E of exception points is $E = \{0, \infty\}$. The solution set M is

$$M = \{(z, w) \in (\mathbb{C} \setminus 0) \times \mathbb{C} \mid w^2 = z\} = \{(w^2, w) \mid w \in \mathbb{C} \setminus 0\} \approx \mathbb{C} \setminus 0.$$

The projection $\pi \colon M \to \mathbb{C} \setminus 0$, $(z, w) \mapsto z$ is in fact the map $w \mapsto w^2$. The function $f \colon M \to \mathbb{C}$, $(z, w) \mapsto w$ can be seen as the identity on $\mathbb{C} \setminus 0$. For every point $z \in \mathbb{C} \setminus 0$ we have $f(\pi^{-1}(z)) = f(\{\pm\sqrt{z}\}) = \{\pm\sqrt{z}\}$.

8.4 Algebraic Functions and GSPs

We investigate the connection of algebraic functions and Geometric Straight-Line Programs. Theorem 8.2.6 combined with an inductive argument shows that every dependent variable v_j of a GSP Γ describes an algebraic function $\hat{v}_j$ in the free variables of Γ. As before, we consider the unary situation and investigate GSPs with one free variable. These GSPs correspond to unary algebraic functions. We give two detailed examples and relate some notions from Dynamic Geometry like critical points and instances to the language of algebraic functions. From these two examples, we arise more general observations and conjectures.

Example 8.4.1. We consider the GSP Γ describing the algebraic expression $\sqrt{\sqrt{z^2} - z}$ from Example 8.3.2:

$$\begin{array}{lllll}
\Gamma: & z & \leftarrow & \text{FREE} & \\
 & v_1 & \leftarrow & z^2 & // \; v_1 = z^2 \\
 & v_2 & \leftarrow & \sqrt{v_1} & // \; v_2 = \sqrt{z^2} \\
 & v_3 & \leftarrow & v_2 - z & // \; v_3 = \sqrt{z^2} - z \\
 & v_4 & \leftarrow & \sqrt{v_3} & // \; v_4 = \sqrt{\sqrt{z^2} - z}
\end{array}$$

Every dependent variable v_i of Γ describes an algebraic function $\hat{v}_i(z)$.

$\hat{v}_1(z)$: We have $\hat{v}_1(z) = z^2$, this is the solution of the polynomial equation $w - z^2 = 0$. The polynomial $P_1(z, w) = w - z^2$ is the characteristic polynomial and the minimal polynomial of the algebraic function $\hat{v}_i(z) = z^2$. The polynomial P_1 can also be obtained by Theorem 8.2.6(2), since the free variable z is the algebraic function defined by $P_0(z, w) = w - z = 0$. In this situation we have

$$\begin{aligned} \hat{v}_1(z) &= \operatorname{Res}_y\left(P_0(z, y), y\, P_0\left(z, \frac{w}{y}\right)\right) = \operatorname{Res}_y(y - z, w - yz) \\ &= \det\begin{pmatrix} 1 & -z \\ -z & w \end{pmatrix} = w - z^2. \end{aligned}$$

The algebraic function $\hat{v}_1(z) = z^2$ is single valued with the zero $z_0 = 0$ of multiplicity 2. It does not have a pole in $\mathbb{C}$. Since we have $\operatorname{disc}_w(P_1(z, w)) = \operatorname{Res}_w(w - z^2, 1) = 1$, $\hat{v}_1(z)$ does not have branch points, as expected.

$\hat{v}_2(z)$: The dependent variable v_2 describes the algebraic expression $\pm\sqrt{z^2} = \pm z$. The corresponding algebraic function is described by the polynomial equation $P_2(z, w) = P_1(z, w^2) = w^2 - z^2 = 0$. We observe, that the polynomial $P_2(z, w) = w^2 - z^2 = (w - z)(w + z)$ decomposes into the irreducible factors $w - z$ and $w + z$ representing the choice $\sqrt{z^2} = z$ or $\sqrt{z^2} = -z$. The polynomial P_2 is the characteristic and the minimal polynomial of $\hat{v}_2(z)$. The Riemann surface (X_2, η_2, f_2) of $\hat{v}_2(z)$ decomposes into two Riemann surfaces $(X_{2,1}, \eta_{2,1}, f_{2,1})$ and $(X_{2,2}, \eta_{2,2}, f_{2,2})$ of the algebraic functions $\hat{v}_{2,1}(z) = z$ and $\hat{v}_{2,2}(z) = -z$; see [47, Sect. 6.2.2]:

$$\begin{array}{ccc} X_{2,1} = \{(z, w) | w = z\} \approx \mathbb{C} & & \\ {\scriptstyle \eta_{2,1}\,:\,(z,w)\mapsto z} \downarrow & \searrow {\scriptstyle f_{2,1}\,:\,(z,w)\mapsto w=z} & \\ \mathbb{C} & & \mathbb{C} \end{array} \qquad \begin{array}{ccc} X_{2,2} = \{(z, w) | w = -z\} \approx \mathbb{C} & & \\ {\scriptstyle \eta_{2,2}\,:\,(z,w)\mapsto z} \downarrow & \searrow {\scriptstyle f_{2,2}\,:\,(z,w)\mapsto w=-z} & \\ \mathbb{C} & & \mathbb{C} \end{array}$$

The functions $f_{2,1}$ and $f_{-2,2}$ have a zero at $z_0 = 0$ and no poles in $\mathbb{C}$. Both coverings $\eta_{2,1}$ and $\eta_{2,2}$ are unbranched onefold coverings. In contrast to this observation, the point $z_0 = 0$ is a critical value of Γ, since the radicand $v_1 = z^2$ of the dependent variable v_2 vanishes at $z_0 = 0$. This is important in connection with GSPs and the Tracing Problem, since close to the point $z_0 = 0$, the corresponding instances are arbitrary close (in the Euclidean metric). This fact must be carefully treated by algorithms for the Tracing Problem. Note that $\operatorname{disc}_w(P_2(z, w)) = 4z^2$ has $z_0 = 0$ as zero.

If $A^{(2)} = (a_0, a_1, a_2)$ with $a_0 \neq 0$ is an instance of the 2-head of the GSP Γ, then (a_0, a_2) is contained in exactly one of the two covering spaces $X_{2,1}$ and $X_{2,2}$. Starting from $A^{(2)}$, we can only reach instances $B^{(2)} = (b_0, b_1, b_2)$, where (b_0, b_2) lies in the same connected component as (a_0, a_2).

$\hat{v}_3(z)$: The dependent variable v_3 describes the algebraic expression

$$\pm\sqrt{z^2} - z = \pm z - z = \begin{cases} 0 \\ -2z. \end{cases}$$

We compute the characteristic polynomial $P_3(z,w)$ of $\hat{v}_3(z)$ using Theorem 8.2.6:

$$\begin{aligned} P_3(z,w) &= \mathrm{Res}_y(P_0(z,y), yP_2(z,w+y)) \\ &= \mathrm{Res}_y\left(y - z, (w+y)^2 - z^2\right) \\ &= \det\begin{pmatrix} 1 & -z & 0 \\ 0 & 1 & -z \\ 1 & 2w & w^2 - z^2 \end{pmatrix} = w^2 + 2wz = w(w+2z) \end{aligned}$$

The two irreducible factors w and $w + 2z$ correspond to the two possible choices $\sqrt{z^2} - z = 0$ or $\sqrt{z^2} - z = -2z$. The Riemann surface (X_3, η_3, f_3) decomposes into two Riemann surfaces $(X_{3,1}, \eta_{3,1}, f_{3,1})$ and $(X_{3,2}, \eta_{3,2}, f_{3,2})$ of the algebraic functions $\hat{v}_{3,1}(z) = 0$ and $\hat{v}_{3,2}(z) = -2z$.

$$\begin{array}{ccc} X_{3,1} = \{(z,w) | w = 0\} \approx \mathbb{C} & & \\ \downarrow \eta_{3,1}:\ (z,w)=(z,0)\mapsto z & \searrow f_{3,1}:\ (z,w)\mapsto w=0 & \\ \mathbb{C} & & \mathbb{C} \end{array} \qquad \begin{array}{ccc} X_{3,2} = \{(z,w) | w = -2z\} \approx \mathbb{C} & & \\ \downarrow \eta_{3,2}:\ (z,w)=(z,-2z)\mapsto z & \searrow f_{3,2}:\ (z,w)\mapsto w=-2z & \\ \mathbb{C} & & \mathbb{C} \end{array}$$

If $A^{(3)} = (a_0, a_1, a_2, a_3)$ is an instance of the 3-head of Γ, then (a_0, a_3) is contained either in the covering space $X_{3,1}$ or in $X_{3,2}$. This choice is already determined by a_2.

$\hat{v}_4(z)$: The dependent variable v_4 describes the algebraic expression

$$\sqrt{\sqrt{z^2} - z} = \begin{cases} \pm 0 \\ \pm\sqrt{-2z}. \end{cases}$$

The characteristic polynomial of $\hat{v}_4(z)$ is $P_4(z,w) = P_3(z,w^2) = w^2(w^2 + 2z)$. This polynomial is not reduced, its reduction is $Q_4(z,w) = w(w^2+2z)$; compare with Example 8.3.2. Let $(X_4; \eta_4, f_4)$ be the Riemann surface of $\hat{v}_4(z)$. Since P_4 is composed by three irreducible factors, the Riemann surface $(X_4; \eta_4, f_4)$ decomposes into three Riemann surfaces $(X_{4,1}; \eta_{4,1}, f_{4,1})$, $(X_{4,2}; \eta_{4,2}, f_{4,2})$, and $(X_{4,3}; \eta_{4,3}, f_{4,3})$, which belong to the three factors w, w and w^2+2z. Since the factor w has "multiplicity" 2, we have $(X_{4,1}; \eta_{4,1}, f_{4,1}) = (X_{4,2}; \eta_{4,2}, f_{4,2})$.

$$\begin{array}{ccc} X_{4,1} = X_{4,2} = \{(z,w) | w = 0\} \approx \mathbb{C} & & \\ \downarrow \eta_{4,1}=\eta_{4,2}:\ (z,w)=(z,0)\mapsto z & \searrow f_{4,1}=f_{4,2}:\ (z,w)\mapsto w=0 & \\ \mathbb{C} & & \mathbb{C} \end{array}$$

$$\begin{array}{ccc} X_{4,3} = \{(z,w) | w^2+2z = 0\} = \{(-\frac{w^2}{2}, w) | w \in \mathbb{C}\} \approx \mathbb{C} & & \\ \downarrow \eta_{4,3}:\ (z,w)\mapsto z=-\frac{w^2}{2} & \searrow f_{4,3}:\ (z,w)\mapsto w & \\ \mathbb{C} & & \mathbb{C} \end{array}$$

The covering $\eta_{4,1} = \eta_{4,2}$ is unbranched, the function $f_{4,1} = f_{4,2}$ is the zero function. Instead, the covering $\eta_{4,3}$ has a branch point at $z_0 = 0$. To determine the branch point, we compute the discriminant of the polynomial $P_{4,3}(z,w) = w^2 + 2z \in \mathbb{C}[z][w]$:

$$\begin{aligned} \operatorname{disc}(P_{4,3}) &= -\operatorname{Res}_w \left(P_{4,3}(z,w), \frac{\partial P_{4,3}(z,w)}{\partial w} \right) = -\operatorname{Res}_w(w^2+2z, 2w) \\ &= -\det \begin{pmatrix} 1 & 0 & 2z \\ 2 & 0 & 0 \\ 0 & 2 & 0 \end{pmatrix} = -8z \end{aligned}$$

The zero $z_0 = 0$ of $\operatorname{disc}(P_{4,3})$ determines the branch point of the algebraic function defined by $P_{4,3}(z,w) = w^2 + 2z = 0$. The function $f_{4,3}$ has a zero at $z_0 = 0$, and it does not have poles in $\mathbb{C}$.

The discriminant of P_4 mirrors the fact $(X_{4,1}; \eta_{4,1}, f_{4,1}) = (X_{4,2}; \eta_{4,2}, f_{4,2})$. Since $w \equiv 0$ is a zero of $P_4(z,w)$ of multiplicity two, the discriminant of P_4 is the zero polynomial.

The square root operation that defines v_4 transforms all points of $\mathbb{C} \setminus \{0\}$ to critical values since every position $z_0 \in \mathbb{C} \setminus \{0\}$ of the free variable z can be extended to a critical point $(z_0, z_0^2, z_0, 0, 0)$; see Definition 3.2.2, Definition 3.2.3, and Definition 3.2.4. The point $z_0 = 0$ is a critical point caused by the dependent variable v_2. Unfortunately, there does not seem to be an algorithm to decide whether a square root operation erases a component of the Riemann surface, yet.

Example 8.4.2. We consider the algebraic expression $\sqrt{1-\sqrt{z}} \cdot (3+\sqrt{z})$. We describe this expression with two GSPs Γ_1 and Γ_2. They differ in the treatment of the term $\sqrt{z}$ that occurs twice. In Γ_1, there is only one dependent variable that represents both terms $\sqrt{z}$, whereas in Γ_2, we have two dependent variables for $\sqrt{z}$, so that the two terms $\sqrt{z}$ are treated independently. We use Theorem 8.2.6 to compute polynomials $P_i(z,w)$ that nullify the algebraic functions of the dependent variables v_i.

$$\begin{array}{lllll}
\Gamma_1: & z \leftarrow \text{FREE} & // & \Rightarrow P_0(z,w) & = w - z \\
& v_1 \leftarrow \sqrt{z} & // \; v_1 = \sqrt{z} & \Rightarrow P_1(z,w) & = P_0(z,w^2) = w^2 - z \\
& v_2 \leftarrow 3 + v_1 & // \; v_2 = 3 + \sqrt{z} & \Rightarrow P_2(z,w) & = \text{Res}_y(P_1(z,y),(w-y)-3) \\
& & & & = (w-3)^2 - z \\
& v_3 \leftarrow 1 - v_1 & // \; v_3 = 1 - \sqrt{z} & \Rightarrow P_3(z,w) & = \text{Res}_y(P_1(z,y),(w+y)-1) \\
& & & & = (w-1)^2 - z \\
& v_4 \leftarrow \sqrt{v_3} & // \; v_4 = \sqrt{1-\sqrt{z}} & \Rightarrow P_4(z,w) & = P_3(z,w^2) = (w^2-1)^2 - z \\
& v_5 \leftarrow v_4 \cdot v_2 & // & \Rightarrow P_5(z,w) & = \text{Res}_y(P_4(z,y), y^2 P_2(z,\frac{w}{y}))
\end{array}$$

$$\begin{array}{lllll}
\Gamma_2: & z \leftarrow \text{FREE} & // & \Rightarrow P_0(z,w) & = w - z \\
& v_{1,1} \leftarrow \sqrt{z} & // \; v_{1,1} = \sqrt{z} & \Rightarrow P_{1,1}(z,w) & = P_0(z,w^2) = w^2 - z \\
& v_{1,2} \leftarrow \sqrt{z} & // \; v_{1,2} = \sqrt{z} & \Rightarrow P_{1,2}(z,w) & = P_0(z,w^2) = w^2 - z \\
& v_2 \leftarrow 3 + v_{1,1} & // \; v_2 = 3 + \sqrt{z} & \Rightarrow P_2(z,w) & = \text{Res}_y(P_{1,1}(z,y),(w-y)-3) \\
& & & & = (w-3)^2 - z \\
& v_3 \leftarrow 1 - v_{1,2} & // \; v_3 = 1 - \sqrt{z} & \Rightarrow P_3(z,w) & = \text{Res}_y(P_{1,2}(z,y),(w+y)-1) \\
& & & & = (w-1)^2 - z \\
& v_4 \leftarrow \sqrt{v_3} & // \; v_4 = \sqrt{1-\sqrt{z}} & \Rightarrow P_4(z,w) & = P_3(z,w^2) = (w^2-1)^2 - z \\
& v_5 \leftarrow v_4 \cdot v_2 & // & \Rightarrow P_5(z,w) & = \text{Res}_y(P_4(z,y), y^2 P_2(z,\frac{w}{y}))
\end{array}$$

In Γ_2, the dependent variable v_1 is replaced by two dependent variables $v_{1,1}$ and $v_{1,2}$. The rest, even the polynomials P_i, remains unchanged. We compute $P_5(z,w)$ with the computer algebra software Maple [30]:

$$\begin{aligned}
P_5(z,w) &= \text{Res}_y(P_4(z,y), y^2 P_2(z,\tfrac{w}{y})) \\
&= (-w^4 - 10zw^2 + 18w^2 - 81 + z^3 - 19z^2 + 99z) \\
&\quad \cdot (-w^4 + 14zw^2 + 18w^2 - 81 + z^3 - 19z^2 + 99z)
\end{aligned}$$

The polynomial P_5 decomposes into two irreducible factors. The Riemann surface of the algebraic function that is defined by the entire polynomial P_5 decomposes into two Riemann surfaces $(X_{5,1}, \eta_{5,1}, f_{5,1})$ and $(X_{5,2}, \eta_{5,2}, f_{5,2})$ that belong to the two irreducible factors. The covering spaces $X_{5,1}$ and $X_{5,2}$ are connected, and the covering maps $\eta_{5,1}$ and $\eta_{5,2}$ are fourfold coverings.

We rearrange the expression $\sqrt{1-\sqrt{z}} \cdot (3+\sqrt{z})$. We distinguish the two situations, where both occurrences of $\sqrt{z}$ are identical as in Γ_1 or where they are different as in Γ_2 if $v_{1,1} \neq v_{1,2}$.

(=) We assume that both terms $\sqrt{z}$ are equal, hence $\sqrt{v_1} = v_{1,1} = v_{1,2}$ and $v_1^2 = z$. This leads to

$$\begin{aligned}
\sqrt{1-\sqrt{z}} \cdot \left(3+\sqrt{z}\right) &= \sqrt{1-v_1}(3+v_1) = \sqrt{(1-v_1)(3+v_1)^2} \\
&= \sqrt{(1-v_1)(9+6v_1+z)} \\
&= \sqrt{9 + 6v_1 + z - 9v_1 - 6\underbrace{v_1^2}_{=z} - zv_1}
\end{aligned}$$

$$
\begin{aligned}
&= \sqrt{9 - 3v_1 - 5z - zv_1} \\
&= \sqrt{9 - 3\sqrt{z} - 5z - z\sqrt{z}}
\end{aligned}
$$

Since

$$
\begin{aligned}
w &= \sqrt{1-\sqrt{z}} \cdot (3+\sqrt{z}) \\
&= \sqrt{9 - 3\sqrt{z} - 5z - z\sqrt{z}} = \sqrt{9 - 5z - \sqrt{z}(3+z)} \\
\Rightarrow \quad w^2 - 9 + 5z &= -\sqrt{z}(3+z) \\
\Rightarrow \quad (w^2 - 9 + 5z)^2 &= z(3+z)^2 \\
\Rightarrow \quad z^3 + 6z^2 + 9z &= w^4 + 81 + 25z^2 - 18w^2 + 10w^2z - 90z \\
\Rightarrow \quad 0 &= -w^4 - 10w^2z + 18w^2 + z^3 - 19z^2 + 99z - 81,
\end{aligned}
$$

the case $v_{1,1} = v_{1,2} = \sqrt{z}$ belongs to the first factor of the polynomial $P_5(z,w)$. The Riemann surface of the function $v_5(z)$ for Γ_1 is defined by the first factor of P_5. More precisely, the first factor is the minimal polynomial (and the characteristic polynomial) of the function $v_5(z)$ in the GSP Γ_1. The second factor of P_5 can be neglected, since the dependent variable v_5 of the GSP Γ_1 cannot reach the corresponding connected component. The polynomial P_5 that has been determined by Theorem 8.2.6, is not the minimal polynomial of the function $\hat{v}_5(z)$ from Γ_1.

($\neq$) We assume that both terms $\sqrt{z} = v_{1,1} \neq v_{1,2} = \sqrt{z}$ differ. This implies $v_{1,1} = -v_{1,2}$ and $v_{1,1}v_{1,2} = -z$.

$$
\begin{aligned}
\sqrt{1-\sqrt{z}} \cdot \left(3+\sqrt{z}\right) &= \sqrt{1 - v_{1,2}}(3 + v_{1,1}) \\
&= \sqrt{(1 - v_{1,2})(9 + 6v_{1,1} + z)} \\
&= \sqrt{9 + 6v_{1,1} + z - 9\underbrace{v_{1,2}}_{=-v_{1,1}} - 6\underbrace{v_{1,1}v_{1,2}}_{=-z} - z\underbrace{v_{1,2}}_{=-v_{1,1}}} \\
&= \sqrt{9 + 15v_{1,1} + 7z + zv_{1,1}} \\
&= \sqrt{9 + 15\sqrt{z} + 7z + z\sqrt{z}}
\end{aligned}
$$

Since

$$
\begin{aligned}
w &= \sqrt{1-\sqrt{z}} \cdot (3+\sqrt{z}) \\
&= \sqrt{9 + 15\sqrt{z} + 7z + z\sqrt{z}} \\
&= \sqrt{9 + 7z + \sqrt{z}(15+z)} \\
\Rightarrow \quad w^2 - 9 - 7z &= z(15+z)^2 = z(225 + 30z + z^2) \\
\Rightarrow \quad z^3 + 30z^2 + 225z &= w^4 + 81 + 49z^2 - 18w^2 - 14w^2z + 126z \\
\Rightarrow \quad 0 &= -w^4 + 14w^2z + 18w^2 + z^3 - 19z^2 + 99z - 81,
\end{aligned}
$$

the second factor of $P_5(z,w)$ belongs to the case $v_{1,1} = -v_{1,2}$. If we have a starting instance $= (a_0, a_{1,1}, a_{1,2}, \ldots, a_5)$ that lies in the second connected

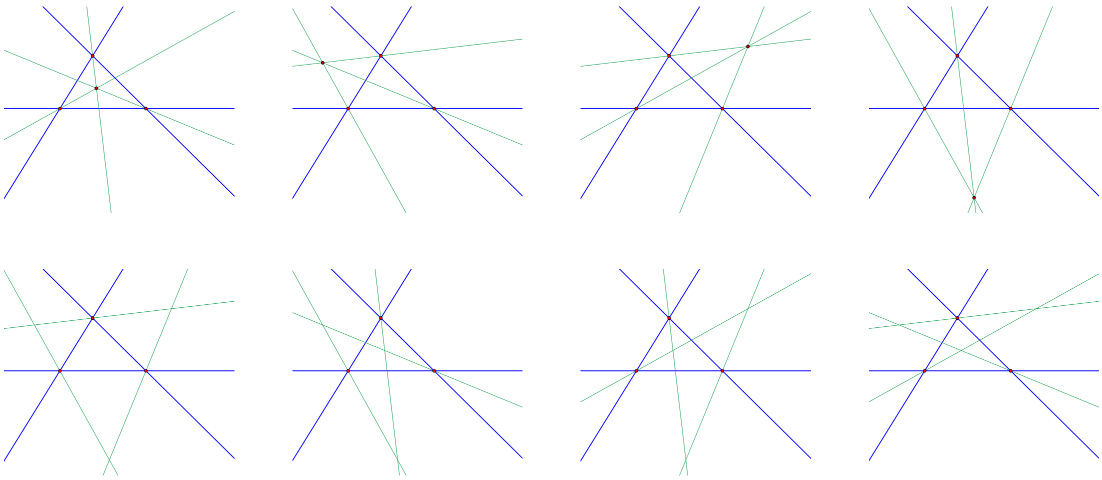


Figure 8.5: A triangle and its angular bisectors are considered. In the four configurations of the first row, the three angular bisectors meet in a common point; these "good" configurations are connected via the line at infinity. In the configurations of the second row, the angular bisectors do not meet; again, all "bad" configurations form a connected component of the configuration space (over $\mathbb{R}$).

component (i.e. $a_{1,1} = -a_{1,2} \neq 0$), then we cannot reach an instance B from the first connected component. This observation shows that the complex Reachability Problem is an interesting problem.

We remark that for the function $\hat{v}_5(z)$ of the second GSP Γ_2 the entire polynomial $P_5(z, w)$ is the minimal polynomial, since both situations $v_{1,1} = v_{1,2}$ and $v_{1,1} = -v_{1,2}$ can occur. For both GSPs Γ_1 and Γ_2, the number m of instances that lie over a regular value, equals the degree of the minimal polynomial of the variable v_5, and the corresponding covering is an $m$fold covering. For Γ_1, we have $m = 4$, and for Γ_2, we have $m = 8$.

A similar situation occurs in the geometric setting if we consider a triangle and the angular bisectors of the edges. At every vertex, we can choose either the bisector of the internal angle or of the external angle. If we restrict to real coordinates of the objects, then the corresponding covering decomposes into two connected components; see Figure 8.5.

The examples 8.4.1 and 8.4.2 seem to describe coherences that hold for general GSPs as well. Intuitively, we conjecture a correlation like "The number of instances of a GSP Γ equals the degree of the characteristic polynomial of the corresponding algebraic function". This correlation is difficult to formulate since there might be variables of Γ that are not involved in the definition of the algebraic function. Methods from Algebraic Geometry might help to get this problem under control.

We have seen in Example 8.4.1 that a connected component "disappears" if the characteristic polynomial is reducible. In this case, the discriminant is zero.

Theorem 8.2.6 suggests, that division free GSPs lead to entire algebraic functions. If the leading coefficients of P and Q are 1, then in 8.2.6 (1), (2) and (4), the polynomial R has leading coefficient 1 as well. In division free GSPs, all critical values are caused by square root operations. We expect that the set of critical values of a division free GSP with only one free variable is the union of the zeros of the discriminants of the characteristic polynomials of the algebraic functions that belong to the dependent variables of the GSP; compare with page 127 and page 136.

The polynomial P_i computed according to Theorem 8.2.6 might not be the characteristic polynomial as seen in Example 8.4.2.

In Theorem 8.1.4 we have seen, that for the roots and poles of an algebraic function is sufficient to know the coefficient polynomials $a_0(z)$ and $a_n(z)$. Unfortunately, it seems that these two coefficient polynomials cannot be determined easily from the GSP Γ using Theorem 8.2.6.

A detailed treatment of these observations could be the subject of future work and might establish another point of view to the Tracing Problem and the Reachability Problem from Dynamic Geometry.

8.5 Algebraic Functions and Continuous Evaluations

We relate the notion of continuous evaluations of a complex GSP Γ with one free variable to the Riemann surfaces of the algebraic functions of the dependent variables of Γ. For simplicity of notation, we only consider the last dependent variable v_n of Γ. To treat an arbitrary dependent variable v_j with $j \in \{1, \ldots, n\}$, we switch to the j-head $\Gamma^{(j)}$ of Γ.

Let $p(t)\colon [0,1] \to \mathbb{C}$ be a continuous path of the free variable z of Γ, and let $A = (a_0 = p(0), a_1, \ldots, a_n)$ be an instance of Γ. We assume that the continuous evaluation $(v_1(t), \ldots, v_n(t))$ along $p(t)$ starting at A does not pass through a critical point.

We consider the Riemann surface (X_n, η_n, f_n) of the algebraic function $\hat{v}_n(z)$ of the dependent variable v_n of Γ. In Section 8.3, we have seen $\hat{v}_n(z) = f_n(\eta_n^{-1}(z))$. Let $\tilde{a}_n \in X_n$ be the point in the fiber $\eta_n^{-1}(p(0)) = \eta_n^{-1}(a_0)$ with $f_n(\tilde{a}_n) = a_n$.

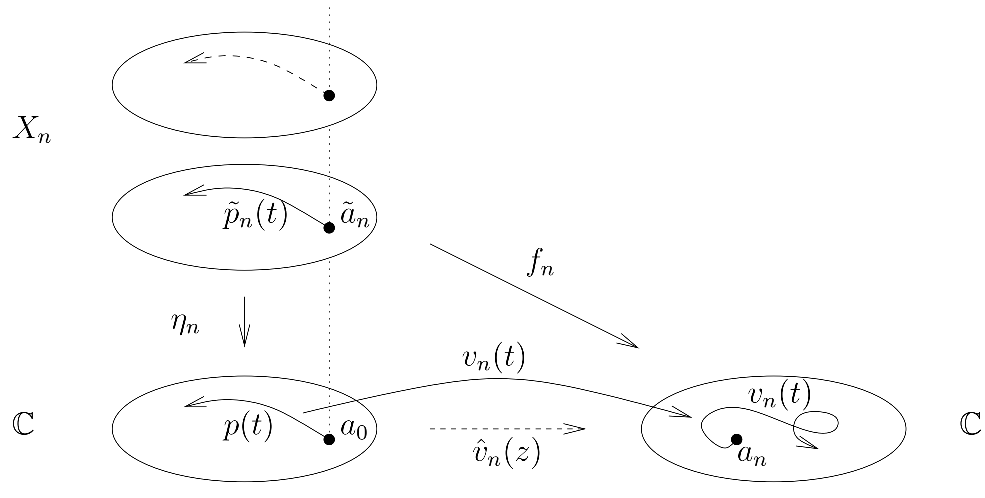


Figure 8.6: The Riemann surface (X_n, η_n, f_n) of the dependent variable v_n, the path $v_n(t)$ of the continuous evaluation, and the lifting $\tilde{p}_n(t)$ are shown.

Since the function $f_n \colon X_n \to \mathbb{C}$ is injective on the fiber $\eta_n^{-1}(a_0)$ and $a_n \in \hat{v}_n(a_0) = f_n(\eta_n^{-1}(a_0))$, the point $\tilde{a}_n$ is unique. Let $\tilde{p}_n(t)$ be the lifting of the path $p(t)$ of the free variable z with respect to the covering map $\eta_n \colon X_n \to \mathbb{C}$ with $\tilde{p}_n(0) = \tilde{a}_n$; see Figure 8.6. A lifting of the path $p(t)$ with respect to η_n is a path $\tilde{p}_n(t)$ with $\eta_n(\tilde{p}_n(t)) = p(t)$; see Definition 8.6.4 of Section 8.6.

Lemma 8.5.1. *Let $\tilde{p}_n(t)$ be the lifting of $p(t)$ with $\tilde{p}_n(0) = \tilde{a}_n$. Then we have*

$$v_n(t) = f_n(\tilde{p}_n(t))$$

for all $t \in [0, 1]$.

Proof. Since $\tilde{p}_n(t)$ is the lifting of $p(t)$ with $\tilde{p}_n(0) = \tilde{a}_n$, we have $a_n = v_n(0) = f_n(\tilde{p}_n(0))$. The path $f_n(\tilde{p}_n(t)) \colon [0, 1] \to \mathbb{C}$ is continuous since $\tilde{p}_n$ is continuous and f_n is holomorphic and hence continuous as well. By definition of f_n, the point $(p(t), f_n(\tilde{p}_n(t)))$ is a zero of the characteristic polynomial P of the algebraic function $\hat{v}_n(z)$; see Section 8.3. Since the polynomial P and the GSP Γ describe the same algebraic function, there is an instance $(z_t, v_{1,t}, \ldots, v_{n,t})$ of Γ with $z_t = p(t)$ and $v_{n,t} = f_n(\tilde{p}_n(t))$. Like $v_{n,t} = f_n(\tilde{p}_n(t))$, the coordinates $v_{j,t}$ can be chosen as continuous functions of t with $v_{j,0} = a_j$ as well. Thus, $f_n(\tilde{p}_n(t))$ is the path of the variable v_n in a continuous evaluation of Γ along p starting at A. The uniqueness of continuous evaluations implies $v_n(t) = f_n(\tilde{p}_n(t))$ for all $t \in [0, 1]$; see Corollary 8.6.7 of Section 8.6. □

We observe that the path $v_n(t)$ in a continuous evaluation $(v_1(t), \ldots, v_n(t))$ is essentially the lifting $\tilde{p}_n(t)$ of the path $p(t)$ of the free variable z to the covering space X_n of the Riemann surface (X_n, η_n, f_n) of the algebraic function $\hat{v}_n(z)$ of the dependent variable v_n with $\tilde{p}_n(0) = \tilde{a}_n$.

8.6 Continuous Evaluations and Coverings

We briefly explain the notion of coverings as they are used in the theory of Riemann surfaces [22, 17]. They are a useful tool for proving the uniqueness and existence of continuous evaluations for $\mathbb{K} = \mathbb{C}$.

Definition 8.6.1. *[22, Def. 4.1]* Covering
Let X be a topological space. A topological space Y together with a map $\pi\colon Y \to X$ is called covering *of X, if π is continuous and open and if for every point $x \in X$ the inverse image $\pi^{-1}(x)$ is empty or discrete in Y. The set $\pi^{-1}(x)$ is called* fiber over *x; see [17, p. 18]. The function π is called* covering map*, and Y is the* covering space*.*

A point $y \in Y$ is called branch point *of the covering π, if there is no neighborhood V of y such that $\pi_{|V}$ is injective [22, Def. 4.3]. A covering that has branch points is called a* branched covering*. Similarly, a covering without branch points is called an* unbranched covering*.*

A function $\pi\colon Y \to X$ of topological spaces X and Y is an unbranched covering if and only if π is a local homeomorphism [22, Thm. 4.4]; see Figure 8.7. Per definition, $\pi\colon Y \to X$ is a local homeomorphism, if every point $y \in Y$ has an open neighborhood V such that $\pi(V) \subset X$ is an open set and $\pi_{|V}\colon V \to \pi(V)$ is a homeomorphism. We remark that Definition 8.6.1 is more general than the notion of coverings from topology; see e.g. [8, Def. 3.1] or [47, Sect. 3.2.2].

Example 8.6.2. The function $\pi\colon Y = \mathbb{C} \to X = \mathbb{C}$, $z \mapsto z^2$, is a covering map. Indeed, π is continuous and open, and $\pi^{-1}(x) = \{\pm\sqrt{x}\}$ consists of two elements if $x \neq 0$, and $\pi^{-1}(0) = \{0\}$ contains one element. The point $0 \in Y = \mathbb{C}$ is a branch point of the covering (Y, π) since for every neighborhood $V \subset Y$ of 0 the restriction $\pi|_V$ of π to V is not injective; ϵ and $-\epsilon$ are both mapped to ϵ^2. Figure 8.8 shows the common visualization of this covering, which is the Riemann surface of the function $\sqrt{\_}\colon \mathbb{C} \to \mathbb{C}$. Note that this surface does not have self-intersections. The "dashed line" is due to the embedding into $\mathbb{R}^3$. Observe that the map π is drawn as a projection.

The following lemma helps to understand the definition of coverings.

Lemma 8.6.3. *[22, Thm. 4.2] Let X, Y be Riemann surfaces and $p\colon Y \to X$ a non-constant holomorphic map. Then p is a covering.*

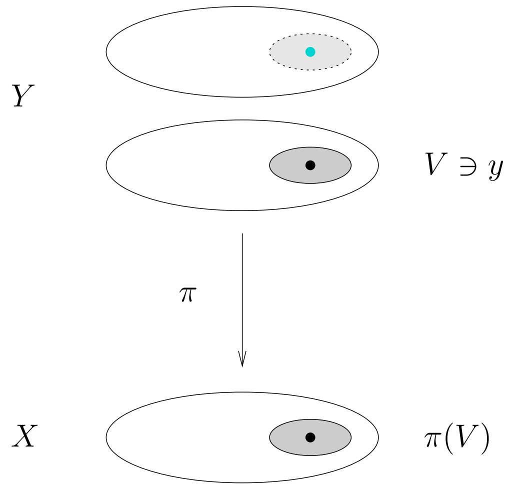
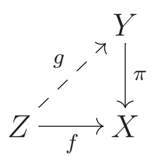


Figure 8.7: Visualization of an unbranched covering $\pi\colon Y \to X$.

Proof. Since p is a non-constant holomorphic function, it is continuous and open (i.e. the images of open sets are again open sets). A famous result from complex analysis implies that for every point $a \in X$, its inverse image $p^{-1}(a)$ is a discrete set. If $p^{-1}(a)$ were not discrete for an $a \in X$, then p would be the constant function $p(y) \equiv a$. □

We give some more standard definitions and properties of coverings [22] and apply them to the notion of continuous evaluations.

Definition 8.6.4. *[22, Def. 4.7]* Lifting of a Continuous Map
Let X, Y and Z be topological spaces, $\pi\colon Y \to X$ a covering and $f\colon Z \to X$ a continuous map. A lifting *of f (with respect to π) is a continuous map $g\colon Z \to Y$ with $f = \pi \circ g$, i.e., the following diagram commutes.*



Example 8.6.5. Again, we consider the covering $\pi\colon \mathbb{C} \to \mathbb{C}$, $z \mapsto z^2$. In Figure 8.8, all liftings of the curves $\gamma(t) = \frac{1}{2}e^{2\pi i t}$ and $\delta(t) = \frac{1}{2}e^{2\pi i(t-\frac{1}{2})} + 1$, $t \in [0,1]$, are shown. The curve γ cycles once around the point $0 \in X = \mathbb{C}$, which is the basepoint of the branch point $0 \in Y = \mathbb{C}$. The liftings $\tilde{\gamma}_1$ and $\tilde{\gamma}_2$ of γ change the sheets of the covering π whereas the liftings $\tilde{\delta}_1$ and $\tilde{\delta}_2$ of δ don't. Figure 8.9 shows the liftings $\tilde{\gamma}_1$, $\tilde{\gamma}_2$, $\tilde{\delta}_1$ and $\tilde{\delta}_2$ drawn in the complex plane $\mathbb{C}$.

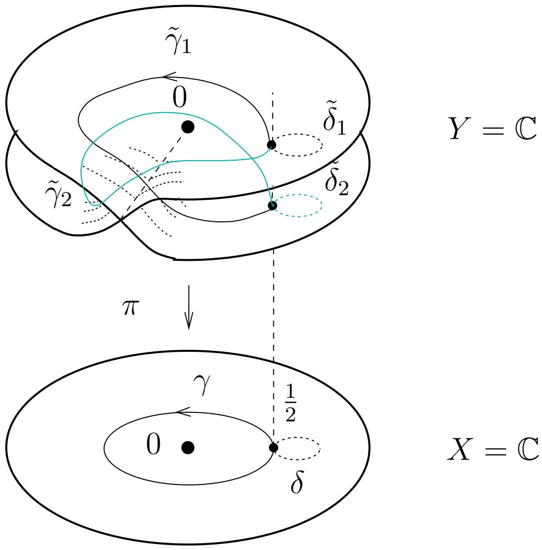


Figure 8.8: Two-fold covering of $\mathbb{C}$ defined by $Y = \mathbb{C}$ and $\pi(z) = z^2$ with the liftings $\tilde{\gamma}_1$, $\tilde{\gamma}_2$, $\tilde{\delta}_1$ and $\tilde{\delta}_2$ of the closed curves γ and δ.

Theorem 8.6.6. *[22, Thm. 4.8]* Uniqueness of Liftings
Let X, Y be Hausdorff spaces and $\pi\colon Y \to X$ a covering without *branch points. Let Z be a connected topological space and $f\colon Z \to X$ a continuous map. If g_1, $g_2\colon Z \to Y$ are two liftings of f and if $g_1(z_0) = g_2(z_0)$ holds for one point $z_0 \in Z$, then $g_1 = g_2$.*

To lift curves, we consider the time interval $[0, 1] =: Z$. We assume that no basepoint of a branch point of the covering π lies on the curves we want to lift. Then Theorem 8.6.6 implies that liftings of curves are unique if the starting point is specified, for example. Corollary 8.6.7 of Theorem 8.6.6 is the exact statement of an obvious fact.

Corollary 8.6.7. Uniqueness of Continuous Evaluations
Let Γ be a complex GSP with k free variables $z_{-k+1}, \dots, z_0$ and n dependent ones $v_1, \dots, v_n$. Let $p_{-k+1}, \dots, p_0\colon [0, 1] \to \mathbb{C}$ be continuous paths of the free variables of Γ, and let $A = (a_{-k+1} = p_{-k+1}(0), \dots, a_0 = p_0(0), a_1, \dots, a_n)$ be an instance of Γ. If $(v_1(t), \dots, v_n(t))$ and $(\tilde{v}_1(t), \dots, \tilde{v}_n(t))$ are continuous evaluations along the paths $p_{-k+1}, \dots, p_0$ starting at A, then $(v_1(t), \dots, v_n(t)) = (\tilde{v}_1(t), \dots, \tilde{v}_n(t))$.

Proof. (Induction on the number of dependent variables of Γ)
W.l.o.g. let the free variables $z_{-k+1}, \dots, z_0$ be the first k variables of Γ. If $n = 0$, then Γ consists of free variables, only, and the claim of Corollary 8.6.7 is fulfilled.

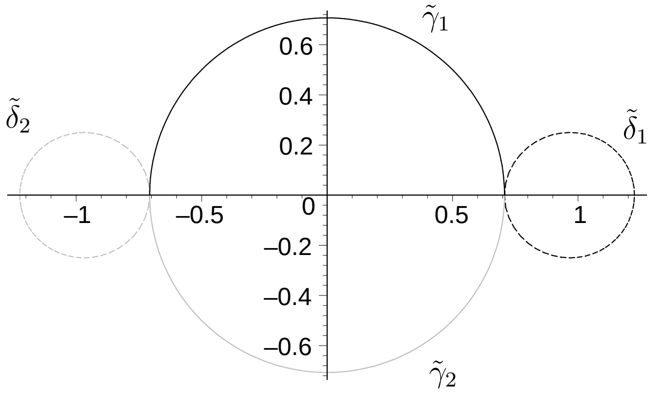


Figure 8.9: The curves $\tilde{\gamma}_1$, $\tilde{\gamma}_2$, $\tilde{\delta}_1$ and $\tilde{\delta}_2$ drawn in the complex plane $\mathbb{C} = Y$.

We assume that the claim of Corollary 8.6.7 holds for GSPs with $n-1$ dependent variables. Let Γ be a GSP with n dependent variables $v_1, \dots, v_n$, let $A = (a_{-k+1}, \dots, a_0, a_1, \dots, a_n)$ be an instance of Γ, and let $p_{-k+1}, \dots, p_0$ be continuous paths of the free variables of Γ with $p_l(0) = a_l$; $l = -k+1, \dots, 0$. Let $(v_1(t), \dots, v_{n-1}(t), v_n(t))$ and $(\tilde{v}_1(t), \dots, \tilde{v}_{n-1}(t), \tilde{v}_n(t))$ be continuous evaluations of Γ along the paths $p_{-k+1}, \dots, p_0$ starting at A. We consider the $n-1$-head $\Gamma^{(n-1)}$ of Γ. By definition, $\Gamma^{(n-1)}$ is a GSP with $n-1$ dependent variables, and the paths $(v_1(t), \dots, v_{n-1}(t))$ and $(\tilde{v}_1(t), \dots, \tilde{v}_{n-1}(t))$ are continuous evaluations of $\Gamma^{(n-1)}$ along the paths $p_{-k+1}, \dots, p_0$ starting at $A^{(n-1)} = (a_{-k+1}, \dots, a_0, a_1, \dots, a_{n-1})$. By induction, we have $v_i(t) = \tilde{v}_i(t)$ for all $t \in [0,1]$; $i = 1, \dots, n-1$. It remains to show $v_n(t) = \tilde{v}_n(t)$ for all $t \in [0,1]$.

If the dependent variable v_n is defined by one of the deterministic operations addition, subtraction, multiplication, or division, then $v_n(t) = \tilde{v}_n(t)$ holds for all $t \in [0,1]$, since these operations are determined by their input variables.

If $v_n \leftarrow \sqrt{v_c}$ is defined to be a square root of a free or dependent variable v_c, then we consider the path $v_c(t) = \tilde{v}_c(t)$ in the continuous evaluation of $\Gamma^{(n-1)}$. Since $(v_1(t), \dots, v_n(t))$ and $(\tilde{v}_1(t), \dots, \tilde{v}_n(t))$ are continuous evaluations of the original GSP Γ, the path $v_c(t) = \tilde{v}_c(t)\colon [0,1] \to \mathbb{C}$ does not pass through $0 \in \mathbb{C}$. Hence, the paths $v_n(t)$ and $\tilde{v}_n(t)$ are liftings of the path $v_c(t) = \tilde{v}_c(t)$ with $v_n(0) = \tilde{v}_n(0) = a_n$ with respect to the covering $\pi\colon \mathbb{C} \setminus \{0\} \to \mathbb{C} \setminus \{0\}$, $z \mapsto z^2$, and Theorem 8.6.6 implies $v_n(t) = \tilde{v}_n(t)$ for all $t \in [0,1]$. □

Now we discuss the existence of continuous evaluations. We need the following lemma that is a consequence of Definition 4.11 and Theorem 4.14 from [22].

Lemma 8.6.8. *The covering*

$$\begin{array}{rccc} \pi\colon & \mathbb{C} \setminus \{0\} & \to & \mathbb{C} \setminus \{0\} \\ & z & \mapsto & z^2 \end{array}$$

has the curve-lifting-property, i.e.: For each continuous function $u\colon [0,1] \to \mathbb{C} \setminus \{0\}$ *and each point* $y_0 \in \mathbb{C} \setminus \{0\}$ *with* $\pi(y_0) = u(0)$ *there is a lifting* $\tilde{u}\colon [0,1] \to \mathbb{C} \setminus \{0\}$ *of* u *with* $\tilde{u}(0) = y_0$.

Corollary 8.6.9. Existence of Continuous Evaluations
Let $S \subset \mathbb{C}^k$ *be the set of critical values of a GSP* Γ *having* k *free variables* $z_{-k+1}, \ldots, z_0$ *and* n *dependent ones* $v_1, \ldots, v_n$. *Let* $p = (p_{-k+1}, \ldots, p_0)\colon [0,1] \to \mathbb{C}^k \setminus S$ *be a continuous path of the free variables* $z_{-k+1}, \ldots, z_0$ *of the GSP* Γ, *and let* $A = (a_{-k+1}, \ldots a_0, a_1, \ldots, a_n) \in \mathbb{C}^{k+n}$ *be an instance of* Γ *with* $a_{-k+1} = p_{-k+1}(0), \ldots, a_0 = p_0(0)$. *Then there is a continuous evaluation of* Γ *along the paths* $p_{-k+1}, \ldots, p_0$ *starting at* A.

Proof. Corollary 8.6.9 can be proven by induction on the number of dependent variables like Corollary 8.6.7 using Lemma 8.6.8.

For $n = 0$, the GSP Γ does not have dependent variables and the claim of Corollary 8.6.9 holds. We assume, that Corollary 8.6.9 holds for GSPs with $n-1$ dependent variables. To prove the inductive step, let Γ be a GSP with n dependent variables $v_1, \ldots, v_{n-1}, v_n$. Then, the $n-1$-head $\Gamma^{(n-1)}$ is a GSP with the $n-1$ dependent variables $v_1, \ldots, v_{n-1}$. By induction, there is a continuous evaluation $(v_1(t), \ldots, v_{n-1}(t))$ of $\Gamma^{(n-1)}$ along the paths $p_{-k+1}, \ldots, p_0$ starting at the instance $A^{(n-1)} = (a_{-k+1}, \ldots, a_0, a_1, \ldots, a_{n-1})$. We consider the dependent variable v_n. If $v_n \leftarrow v_i \circ v_j$, $\circ \in \{+, -, \cdot, /\}$, is defined by one of the four elementary arithmetic operations, then $(v_1(t), \ldots, v_{n-1}(t), v_n(t) = v_i(t) \circ v_j(t))$ is a continuous evaluation of Γ along the paths $p_{-k+1}, \ldots, p_0$ starting at the instance A. Since critical values are excluded, divisions by zero cannot occur.

If $v_n \leftarrow \sqrt{v_c}$ is defined by a square root operation, we consider the path $v_c(t)$ in the continuous evaluation of $\Gamma^{(n-1)}$ of the radicand variable v_c. Since critical values of Γ are excluded for the paths $p_{-k+1}, \ldots, p_0$, the path $v_c(t)$ does not pass through zero. By Lemma 8.6.8, there is a lifting $v_n(t) := \tilde{v}_c(t)$ of the path $v_c(t)\colon [0,1] \to \mathbb{C}$ with $v_n(0) = a_n$ with respect to the covering $\pi\colon \mathbb{C} \setminus \{0\} \to \mathbb{C} \setminus \{0\}$. By construction, $(v_1(t), \ldots, v_{n-1}(t), v_n(t) = \tilde{v}_c(t))$ is a continuous evaluation of Γ along the paths $p_{-k+1}, \ldots, p_0$ starting at the instance A. □

A closer look at the proof of Corollary 8.6.9 shows that Corollary 8.6.9 holds in a more general situation: A continuous evaluation exists if and only if the paths of the radicand or divisor variables of the root or division operations do not pass through zero. In this case, we can find the continuous evaluation along given paths $p_{-k+1}, \ldots, p_0$ starting at a given instance A, inductively. This approach is used in the algorithms for the Tracing Problem from Chapter 6.

In addition to the lifting of curves, we can lift continuous functions $f\colon Z \to X$ defined on a simply connected, path-connected and locally path-connected topological space Z with respect to unbranched coverings $\pi\colon Y \to X$ that have the curve lifting property; see [22, Section 1, pp. 22-26]. We need this generalization to lift analytic functions in Section 8.6.1.

To investigate the Reachability Problem, the notion of homotopic paths is helpful as seen in 4.3. Roughly speaking, two paths with the same endpoints a and b are called homotopic, if one of them can be continuously transformed into the other one, and the transformation leaves the endpoints a and b of the curves fixed.

Definition 8.6.10. *[22, Def. 3.1]* Homotopic Paths
Let X be a topological space and a, $b \in X$. Let $c_1, c_2\colon [0,1] \to X$ be continuous paths from a to b. The two curves c_1 and c_2 are called homotopic if there is a map $H\colon [0,1] \times [0,1] \to X$, called homotopy, with the following properties:

1. *$\forall t \in [0,1] : \; H(t,0) = c_1(t)$, i.e., for $s = 0$ we get the path c_1.*

2. *$\forall t \in [0,1] : \; H(t,1) = c_2(t)$, i.e., for $s = 1$ we get the path c_2.*

3. *$\forall s \in [0,1] : \; H(0,s) = a$ and $H(1,s) = b$, i.e., the endpoints a and b of c_1 and c_2 are fixed.*

This defines an equivalence relation on the set of continuous paths $[0,1] \to X$; two paths $c_1, c_2\colon [0,1] \to X$ are equivalent if and only if they have the same endpoints and are homotopic [22, Thm. 3.2]. The equivalence classes are called homotopy classes. Theorem 8.6.11 implies that the continuous evaluations of homotopic paths starting at the same instance have the same final instance.

Theorem 8.6.11. *[22, Thm. 4.10]* Lifting of Homotopic Curves
Let X and Y be Hausdorff spaces and $\pi\colon Y \to X$ an unbranched covering. Let a, $b \in X$ and $\hat{a} \in Y$ with $\pi(\hat{a}) = a$. Let $H\colon [0,1] \times [0,1] \to X$ be a continuous map with $H(0,s) = a$ and $H(1,s) = b$ for all $s \in [0,1]$. We define

$$u_s(t) := H(t,s)$$

and assume that each curve u_s can be lifted to a curve $\hat{u}_s$ with starting point $\hat{a}$. Then $\hat{u}_0$ and $\hat{u}_1$ have the same endpoint, and they are homotopic.

Corollary 8.6.12. *Let $S \subset \mathbb{C}^k$ be the set of critical values of a GSP Γ having k free variables $z_{-k+1}, \ldots, z_0$ and n dependent variables $v_1, \ldots, v_n$. Let $p = (p_{-k+1}, \ldots, p_0)\colon [0,1] \to \mathbb{C}^k \setminus S$ and $p' = (p'_{-k+1}, \ldots, p'_0)\colon [0,1] \to \mathbb{C}^k \setminus S$ be homotopic continuous paths of the free variables $z_{-k+1}, \ldots, z_0$ of the GSP Γ, and let $A = (a_{-k+1}, \ldots, a_0, a_1, \ldots, a_n) \in \mathbb{C}^{k+n}$ be an instance of Γ with $a_{-k+1} = p_{-k+1}(0) = p'_{-k+1}(0), \ldots, a_0 = p_0(0) = p'_0(0)$. Then the continuous evaluations of p and p' starting at A end at the same instance.*

Proof. Corollary 8.6.12 can be proven in a similar way as Corollary 8.6.9 using Theorem 8.6.11. Since the paths p and p' are homotopic in the set $\mathbb{C}^k \setminus S$, the functions $u_s(t)$ are paths in $\mathbb{C}^k \setminus S$. By Corollary 8.6.9, these paths can be extended to continuous evaluations starting at A. □

8.6.1 Coverings of Riemann Surfaces

We consider coverings of Riemann surfaces [22, 47]. As mentioned in Section 8.3, a *Riemann surface* is a Hausdorff space X together with a holomorphic structure on X [47, Sect. 1.1]. Let $\eta\colon X \to Y$ be a covering, where X and Y are Riemann surfaces. If η is holomorphic, then η is locally biholomorphic [22, p. 20]. In [47], branched coverings of Riemann surfaces are defined using winding maps [47, Sect. 1.4.1]. This definition is stronger than Definition 8.6.1 and adapts to the special strucure of Riemann surfaces. In [47, 1.4.5] is shown that every finite holomorphic function $\eta\colon X \to Y$ of Riemann surfaces is a covering of Riemann surfaces. The function η is called finite if every fiber $\eta^{-1}(y)$, $y \in Y$, is finite.

If $\eta\colon X \to Y$ is a holomorphic unbranched covering of Riemann surfaces, then the lifting of a holomorphic function is again holomorphic [22, p. 21, Thm. 4.9]. Since continuous evaluations do not pass through branch points, this fact implies that the paths in a continuous evaluation are smooth if the paths of the free variables are smooth. This observation is fundamental for defining the derivative of a GSP in Section 3.5 and for considering the derivatives of the paths in a continuous evaluation. Compare Lemma 8.6.13 with Corollary 8.6.9.

Lemma 8.6.13. *Let Γ be a GSP over $\mathbb{C}$ with k free variables and n dependent variables $v_1, \dots, v_n$. Let A be an instance of Γ, and let $p_{-k+1}, \dots, p_0\colon [0,1] \to \mathbb{C}$ be paths of the free variables of Γ that can be extended to analytic functions on a connected neighborhood $U_{[0,1]} \subset \mathbb{C}$ of the time interval $[0,1]$ and for which the corresponding continuous evaluation along $p_{-k+1}, \dots, p_0$ starting at A exists. Then there is a connected neighborhood $\hat{U}_{[0,1]} \subset \mathbb{C}$ of the time interval $[0,1]$ such that the paths of the dependent variables $v_i(t)$ in the continuous evaluation of Γ can be extended to analytic functions $\hat{v}_i$ on $\hat{U}_{[0,1]}$, and at each point $t \in \hat{U}_{[0,1]}$ the relations of Γ are fulfilled.*

Proof. Induction on the number of dependent variables of Γ.
For $n = 0$, we choose $\hat{U}_{[0,1]} := U_{[0,1]}$. Now we assume that the claim is shown for GSPs with n dependent variables, and let Γ be a GSP with $n+1$ dependent variables. The n-head $\Gamma^{(n)}$ of Γ is a GSP with n dependent variables. Let $U_{[0,1],n} := \hat{U}_{[0,1]} \subset \mathbb{C}$ be a connected neighborhood of the time interval $[0,1]$ for that all paths $v_i(t)$ in the continuous evaluation of $\Gamma^{(n)}$ can be extended to

analytic functions $\hat{v}_i$ such that at each point $t \in U_{[0,1],n}$ the relations of $\Gamma^{(n)}$ are fulfilled.

If $v_{n+1} \leftarrow v_i \circ v_j$, $\circ \in \{+, -, \cdot\}$, is defined by an addition, subtraction or multiplication, then the path $v_{n+1}(t)$ in the continuous evaluation can be extended to an analytic function on $\hat{U}_{[0,1]} := U_{[0,1],n}$ since the functions $\hat{v}_i$ and $\hat{v}_j$ of the input variables v_i and v_j are analytic functions on $U_{[0,1],n}$ by the induction hypothesis.

If $v_{n+1} \leftarrow v_i/v_c$ is defined by a division, then we consider the divisor variable v_c and its analytic function $\hat{v}_c \colon U_{[0,1],n} \to \mathbb{C}$. We set $\hat{U}_{[0,1]} := U_{[0,1],n+1} := \hat{v}_c^{-1}(\mathbb{C} \setminus \{0\})$. Since $\hat{v}_c$ is an analytic function on $U_{[0,1],n}$, the set of zeros $\hat{v}_c^{-1}(\{0\})$ of $\hat{v}_c$ is a discrete set, and $\hat{U}_{[0,1]}$ remains a connected neighborhood of $[0,1]$.

If $v_{n+1} \leftarrow \sqrt{v_c}$ is defined by a square root operation, we consider the radicand v_c and choose $\hat{U}_{[0,1]} := U_{[0,1],n+1} := \hat{v}_c^{-1}(\mathbb{C} \setminus \{0\})$ as for the division operation. The analytic function $\hat{v}_c \colon \hat{U}_{[0,1]} \to \mathbb{C}$ can be lifted to the covering space of the covering $\eta \colon \mathbb{C} \setminus \{0\} \to \mathbb{C} \setminus \{0\}$; see page 148 and [22, p. 21, Thm. 4.9]. This process leads to the analytic function $\hat{v}_{n+1} \colon \hat{U}_{[0,1]} \to \mathbb{C}$ that extends the path $v_{n+1}(t)$ in the continuous evaluation. Clearly, the relations of the GSP Γ remain fulfilled. $\square$

Appendix A

Interval Arithmetic

Interval arithmetic is a computation model for self-validated numerics and uses intervals as approximate values. The "correct" value is contained in the corresponding interval. In self-validated numerics, computation models are discussed, in which approximate results are automatically provided with guaranteed error bounds [60, Chap. 1].

We give a brief introduction to interval analysis that is based on the description of Alefeld and Herzberger [1]. In Section A.1 we consider Interval Analysis over $\mathbb{R}$, whereas in Section A.2 complex interval analysis is discussed. As complex intervals, we choose axis-parallel rectangles or circles. We state the inclusion monotonicity property [1], which is crucial for the correctness of our algorithms. Additionally, we give some estimates that are needed for showing that Algorithm 2 and Algorithm 3 from Chapter 6 terminate. These estimates are based on the continuity of the four elementary arithmetic operations and the sqaure root function. We mention the problem of interval dependency [35]. In Section A.5, we introduce the interval Newton method [53], which is a variant of the usual Newton method using interval arithmetic. This method might be useful for the treatment of critical points in Chapter 7. Finally, we discuss affine arithmetic in Section A.6, which is like interval arithmetic a range based model for self-validated computing. Here, quantities are represented by affine forms.

There are many other range based models that seem to be useful for our algorithms from Chapter 6, as well. The investigation of these models is an interesting future project. In Taylor models, the quantities are represented by higher order polynomials; see [54]. The Hermite-Obreschkoff method can deal with algebraic functions as explained in [52].

A.1 Real Interval Arithmetic

We briefly introduce real interval analysis. The main idea is that we do all computations with real closed intervals instead of real numbers. We define the arithmetic interval operations and mention the computation rules as in [1]. Lemma A.1.10 contains some estimates that are based on the continuity of the considered functions and that we use in Chapter 6 as mentioned in the introduction of this chapter.

As usual, a real closed interval $A = [a_1, a_2]$ with $a_1 \leq a_2$, $a_1, a_2 \in \mathbb{R}$, is the set of all real numbers t with $a_1 \leq t \leq a_2$. We denote the set of all real closed intervals by $\mathrm{I}(\mathbb{R})$. A real number $x \in \mathbb{R}$ can be regarded as the interval $[x, x]$. We omit the addition of real and closed since we only consider real closed intervals. Let $A = [a_1, a_2]$ and $B = [b_1, b_2]$ be intervals. The intervals A and B are equal if and only if $a_1 = b_1$ and $a_2 = b_2$; see [1, Def. 1, p. 1].

We define the arithmetic operations on $\mathrm{I}(\mathbb{R})$ as in [1, Abschnitt 1]:

Definition A.1.1. *[1, Def. 2, p. 2]* Arithmetic Operations on $\mathrm{I}(\mathbb{R})$
Let $\circ$ be one of the arithmetic operations $+$, $-$, $\cdot$ and $:$ on $\mathbb{R}$. Let A, $B \in \mathrm{I}(\mathbb{R})$. We set

$$A \circ B := \{a \circ b \mid a \in A,\ b \in B\}.$$

The division $A : B$ is only defined if $0 \notin B$.

For every operation $\circ \in \{+, -, \cdot, :\}$, the set $\{a \circ b \mid a \in A,\ b \in B\}$ is an interval: The function $\circ\colon A \times B \to \mathbb{R}$ is a continuous function on the compact set $A \times B$. Hence, it takes its minimum and maximum and all intermediate values. The intervals $A \circ B$ can be calculated explicitly using the interval bounds of A and B. The formulas are listed in Lemma A.1.2 below.

Lemma A.1.2. *[1, p. 2]* Explicit Computation of $A \circ B$
Let A, $B \in \mathrm{I}(\mathbb{R})$. Then we have

$$\begin{aligned}
A + B &= [a_1 + b_1,\ a_2 + b_2];\\
A - B &= [a_1 - b_2,\ a_2 - b_1];\\
A \cdot B &= [\min\{a_1 b_1, a_1 b_2, a_2 b_1, a_2, b_2\},\ \max\{a_1 b_1, a_1 b_2, a_2 b_1, a_2, b_2\}];\\
A : B &= [a_1, a_2] \cdot \left[\frac{1}{b_2}, \frac{1}{b_1}\right] \qquad \textit{if } 0 \notin B.
\end{aligned}$$

Remark A.1.3. We can also define a division $A : B$ if $0 \in B = [b_1, b_2]$; see for example [35]. One possibility is to define

$$\begin{aligned}
\frac{1}{[b_1, 0]} &:= \left[-\infty, \frac{1}{b_1}\right],\\
\frac{1}{[0, b_2]} &:= \left[\frac{1}{b_2}, \infty\right],
\end{aligned}$$

and for $b_1 < 0 < b_2$

$$\frac{1}{[b_1, b_2]} := \left[-\infty, \frac{1}{b_1}\right] \cup \left[\frac{1}{b_2}, \infty\right].$$

A division $A : B$ with $0 \in B$ can be defined as $A \cdot \frac{1}{B}$. If a division by an interval B with $0 \in B$ occurs in a sequence of computations, then the two resulting intervals $\left[-\infty, \frac{1}{b_1}\right]$ and $\left[\frac{1}{b_2}, \infty\right]$ are usually treated separately in further computation steps.

The following example is taken from [35, p. 4]. It motivates the problem of interval dependency.

Example A.1.4. [35, p. 4] We consider the expression $f(x) = x^2 - x$ over the interval $[0, 1]$. The range of f over $[0, 1]$ is $f([0, 1]) = \{f(x) \,|\, x \in [0, 1]\} = [-1/4, 0]$. Evaluating f with interval arithmetic leads to

$$[0, 1]^2 - [0, 1] = [0, 1] \cdot [0, 1] - [0, 1] = [0, 1] - [0, 1] = [-1, 1].$$

We observe an overestimation of the range of f since $f([0, 1]) = [-1/4, 0] \subsetneqq [-1, 1]$. The intervals of the term x^2 and of the term $-x$ both depend on x, but by performing the operations in interval arithmetic, we neglect this dependency. The resulting interval $[-1, 1]$ includes e.g. numbers like $1^2 - 0 = 1$ and $0^2 - 1 = -1$ in addition to $1^2 - 1 = 0$ and $0^2 - 0 = 0$. This phenomenon is called *interval dependency.* Expressions that are equivalent in real arithmetic may lead to different results in interval arithmetic: We rewrite $f(x) = x^2 - x = x(x - 1)$ and evaluate the expression $x(x - 1)$ in interval arithmetic over the interval $[0, 1]$ and observe

$$[0, 1] \cdot ([0, 1] - 1) = [0, 1] \cdot ([0, 1] - [1, 1]) = [0, 1] \cdot [-1, 0] = [-1, 0] \subsetneqq [-1, 1].$$

Example A.1.4 shows that the distributive law does not hold in interval arithmetic. We give an overview of calculation rules in $\mathrm{I}(\mathbb{R})$. They are proved in [1, Abschnitt 1].

Lemma A.1.5. *[1, Thm. 4, p. 3]* Calculation Rules in $\mathrm{I}(\mathbb{R})$
Let A, B, $C \in \mathrm{I}(\mathbb{R})$. Then the following calculation rules hold:

1. *Commutativity: $A + B = B + A$, and $A \cdot B = B \cdot A$;*
2. *Associativity: $(A + B) + C = A + (B + C)$, and $(A \cdot B) \cdot C = A \cdot (B \cdot C)$;*
3. *Neutral Elements:*
   $X = [0, 0]$ is the unique neutral element for the addition, i.e.,
   $$A = X + A = A + X \text{ for all } A \in \mathrm{I}(\mathbb{R}) \iff X = [0, 0];$$
   $Y = [1, 1]$ is the unique neutral element for the multiplication, i.e.,
   $$A = Y \cdot A = A \cdot Y \text{ for all } A \in \mathrm{I}(\mathbb{R}) \iff Y = [1, 1];$$

4. *Subdistributivity:* $A \cdot (B + C) \subset A \cdot B + A \cdot C$, *but* $a \cdot (B + C) = a \cdot B + a \cdot C$ *holds for* $a \in \mathbb{R}$;

5. $\mathrm{I}(\mathbb{R})$ *has no zero divisors;*

6. *An interval* $A = [a_1, a_2] \in \mathrm{I}(\mathbb{R})$ *with* $a_1 \neq a_2$ *has no inverse for addition and multiplication. However, we have* $0 \in A - A$ *and* $1 \in A : A$.

A very important property of real interval arithmetic is the *inclusion monotonicity property*, which is proved in [1]. It is also denoted as inclusion isotonicity property; see [35, 57]. It states that if we combine two intervals $B^{(1)}$ and $B^{(2)}$ by an operation $\circ \in \{+, -, \cdot, /\}$ and if we shrink the input intervals $B^{(1)}$ and $B^{(2)}$, then the new result must be contained in $B^{(1)} \circ B^{(2)}$. This natural property is important for many algorithms that use interval analysis. However, there are (complex) interval arithmetics that do not fulfill the inclusion monotonicity property; see [2, 57] and Section A.2.

Theorem A.1.6. *[1, Thm. 5, p. 7]* Inclusion Monotonicity Property
Let $A^{(1)}$, $A^{(2)}$, $B^{(1)}$ *and* $B^{(2)} \in \mathrm{I}(\mathbb{R})$ *with* $A^{(1)} \subset B^{(1)}$ *and* $A^{(2)} \subset B^{(2)}$. *Then*

$$A^{(1)} \circ A^{(2)} \subset B^{(1)} \circ B^{(2)}$$

holds for all operations $\circ \in \{+, -, \cdot, :\}$.

Corollary A.1.7. *[1, Cor. 6, p. 8]* *Let* A, $B \in \mathrm{I}(\mathbb{R})$ *and* $a \in A$, $b \in B$. *Then*

$$a \circ b \in A \circ B$$

holds for all operations $\circ \in \{+, -, \cdot, :\}$.

In addition to the arithmetic operations, we consider continuous monadic functions like $x \mapsto x^k$ or $0 < x \mapsto \sqrt[k]{x}$.

Definition A.1.8. *[1, Def. 3, p. 3]* *Let* $r \colon \mathbb{R} \supset X \to \mathbb{R}$, $x \mapsto r(x)$, *be a continuous function, let* $A \in \mathrm{I}(\mathbb{R})$. *We define by*

$$r(A) := [\min_{x \in A} r(x)\,,\ \max_{x \in A} r(x)]$$

the corresponding operation on $\mathrm{I}(\mathbb{R})$.

If the function r is monotonic like the functions $x \mapsto x^k$ or $x \mapsto \sqrt[k]{x}$ over the positive real numbers, then the corresponding interval operation can be computed explicitly using the interval bounds of the argument $A \in \mathrm{I}(\mathbb{R})$. Let $A = [a_1, a_2]$ be an interval with $0 < a_1$. Then we have $A^2 = [a_1^2, a_2^2]$, and $\sqrt{A} = \left[\sqrt{a_1}, \sqrt{a_2}\right]$ or $\sqrt{A} = -\left[\sqrt{a_1}, \sqrt{a_2}\right] = \left[-\sqrt{a_2}, -\sqrt{a_1}\right]$. The inclusion monotonicity property also holds for the monadic operations from Definition A.1.8.

Lemma A.1.9. *[1, p. 8] Let r be a function as in Definition A.1.8 and A, $B \in \mathrm{I}(\mathbb{R})$ with $A \subset B$. Then we have $r(A) \subset r(B)$. Consequently, $r(a) \in r(A)$ holds for $a \in A$.*

The estimates of Lemma A.1.10 are a consequence of the continuity of the functions $+, -, \cdot, / \colon \mathbb{R}^2 \to \mathbb{R}$ and $\pm\sqrt{\_} \colon \mathbb{R}_{>0} \to \mathbb{R}$. We use Lemma A.1.10 to show that Algorithm 2 and Algorithm 3 from Chapter 6 terminate.

Lemma A.1.10. *Let $\circ \in \{+, -, \cdot, :\}$ and $\epsilon > 0$. Let a, $b \in \mathbb{R}$ and $c := a \circ b$. Then there exist $\delta_a > 0$ and $\delta_b > 0$ such that*

$$[a - \delta_a,\, a + \delta_a] \circ [b - \delta_b,\, b + \delta_b] \subset [c - \epsilon,\, c + \epsilon] = c + [-\epsilon, \epsilon],$$

and for $c = \pm\sqrt{a}$ we have

$$\sqrt{[a - \delta_a,\, a + \delta_a]} \subset [c - \epsilon,\, c + \epsilon].$$

For the division operation we assume $b \neq 0$, and for the square root operation we assume $a > 0$. If $a \in [\min_a, \max_a]$ and $b \in [\min_b, \max_b]$ hold, then δ_a and δ_b can be chosen, such that they only depend on ϵ and on the interval bounds $\min_a$, $\max_a$, $\min_b$, *and* $\max_b$.

Proof. Lemma A.1.10 is a consequence of the continuity of the functions

$$+,\ -,\ \cdot,\ \_/\_ \colon \mathbb{R}^2 \to \mathbb{R}$$

and of the functions $\pm\sqrt{\_} \colon \mathbb{R}_{>0} \to \mathbb{R}$. On $\mathbb{R}^2$ we choose the metric which is induced by the maximum norm on $\mathbb{R}^2$. The supplement of Lemma A.1.10 holds since the considered functions are uniformly continuous on the compact sets $[\min_a, \max_a] \times [\min_b, \max_b]$ and $[\min_a, \max_a]$. □

A.2 Complex Interval Arithmetic

This introduction to complex interval analysis summarizes [1, Abschnitt 5]. Additionally, the computation of square roots in complex interval arithmetic is treated; see [56] for roots in circular arithmetic. We reformulate Lemma A.1.10 for complex intervals in Lemma A.2.12 and Lemma A.2.25.

As complex intervals, axis-parallel rectangles or circles are used. We can transfer the computation rules from real interval arithmetic to complex intervals, but here, overestimation becomes worse: In real interval arithmetic, the range of a single interval operation is computed correctly. Overestimation can only occur if interval operations are combined, or if the represented quantities depend on each other. In complex interval arithmetic, we have to deal with overestimation in a single multiplication or division operation. In the sequel, we denote by $\mathrm{I}(\mathbb{C})$ the set of either rectangular or circular intervals [1].

Petković and Petković mention in [57, p. 25] that circular sectors and circular rings (annuli) are used as complex intervals as well. This kind of complex intervals only has a little practical importance.

A.2.1 Rectangular Interval Arithmetic

First, we define complex rectangular intervals:

Definition A.2.1. *[1, Def. 1, p. 55]* Complex Rectangular Interval
Let A_1 and $A_2 \in \mathrm{I}(\mathbb{R})$. Then the set

$$A := A_1 + iA_2 := \{a_1 + ia_2 \,|\, a_1 \in A_1,\ a_2 \in A_2\}$$

is termed complex rectangular interval. *We denote the set of all rectangles $A = A_1 + iA_2$ with $A_1,\ A_2 \in \mathrm{I}(\mathbb{R})$ by* $\mathrm{R}(\mathbb{C})$.

A complex interval $A = A_1 + iA_2 \in \mathrm{R}(\mathbb{C})$ describes an axis parallel rectangle in the complex plane. The interval A_1 is the real part, the interval A_2 the imaginary part. A complex number $a_1 + ia_2$ can be considered as the complex interval $[a_1, a_1] + i[a_2, a_2]$. The set $\mathrm{I}(\mathbb{R})$ is embedded in $\mathrm{R}(\mathbb{C})$ in the same way as $\mathbb{R}$ is embedded in $\mathbb{C}$: An interval $A_1 \in \mathrm{I}(\mathbb{R})$ can be treated as the interval $A_1 + i[0,0] \in \mathrm{R}(\mathbb{C})$. Let $A = A_1 + iA_2$ and $B_1 + iB_2$ complex intervals. Then A and B are equal if and only if $A_1 = B_1$ and $A_2 = B_2$ hold.

We define the arithmetic operations on $\mathrm{R}(\mathbb{C})$ as in [1, Abschnitt 5]. They are defined in a similar way as the arithmetic operations on $\mathbb{C}$. Instead of real numbers for the real part and the imaginary part we have real intervals.

Definition A.2.2. *[1, Def. 3, p. 56]* Arithmetic Operations on $\mathrm{R}(\mathbb{C})$
Let $A = A_1 + iA_2$ and $B = B_1 + iB_2 \in \mathrm{R}(\mathbb{C})$, and $\circ$ one of the operations $+$, $-$, $\cdot$ and $:$. Then $A \circ B$ is defined as follows:

$$\begin{aligned}
A + B &:= A_1 + B_1 + i(A_2 + B_2) \\
A - B &:= A_1 - B_1 + i(A_2 - B_2) \\
A \cdot B &:= A_1 \cdot B_1 - A_2 \cdot B_2 + i(A_1 \cdot B_2 + A_2 \cdot B_1) \\
A : B &:= \frac{A_1 \cdot B_1 + A_2 \cdot B_2}{B_1^2 + B_2^2} + i\frac{A_2 \cdot B_1 - A_1 \cdot B_2}{B_1^2 + B_2^2}
\end{aligned}$$

The division $A : B$ is only defined if $0 \notin B$. The denominator $B_1^2 + B_2^2$ must be computed as $B_1^2 + B_2^2 = \{b_1^2 |\, b_1 \in B_1\} + \{b_2^2 |\, b_2 \in B_2\}$.

Remark A.2.3. If $B_1^2 + B_2^2$ is computed as $B_1^2 + B_2^2 = B_1 \cdot B_1 + B_2 \cdot B_2$, it might be $0 \notin B = B_1 + iB_2$ and $0 \in B_1^2 + B_2^2$ as the example taken from [1, p. 56] shows: For $B = [-1, 1] + i[1, 3]$ we have $0 \notin B$ and

$$B_1 \cdot B_1 + B_2 \cdot B_2 = [-1, 1] + [1, 9] = [0, 10] \ni 0.$$

In contrast to this, we have

$$B_1^2 + B_2^2 = [0, 1] + [1, 9] = [1, 10] \not\ni 0;$$

see Definition A.1.8.

The following lemma states that there is an overestimation of the range of a single multiplication or division in R(ℂ). Compare Lemma A.2.4 with Definition A.1.1 and Lemma A.1.2.

Lemma A.2.4. *Let* A, $B \in \mathrm{R}(\mathbb{C})$. *Then the following holds for the arithmetic operations in* R(ℂ)*; see [1, Abschnitt 5]:*

1. $A \pm B = \{a \pm b | \, a \in A, \, b \in B\}$;
2. $A \cdot B \supset \{a \cdot b | \, a \in A, \, b \in B\}$,
   indeed, $A \cdot B$ *is the smallest rectangle in* R(ℂ) *that contains the set* $\{a \cdot b | \, a \in A, \, b \in B\}$;
3. $A : B \supset \{a : b | \, a \in A, \, b \in B\}$, *if* $0 \notin B$.

As mentioned in [1, p. 59], we might achieve an improvement for the division if we define $A : B := A \cdot \frac{1}{B}$ and if $\frac{1}{B}$ is chosen as the smallest rectangle in R(ℂ) containing the set $\{a : b | \, a \in A, \, b \in B\}$. This has larger computational costs.

In the set R(ℂ), we have almost the same computation rules as in I(ℝ); see Lemma A.1.5 and [1, Thm. 8, p. 62]. The differences are stated in Lemma A.2.5 below.

Lemma A.2.5. Calculation Rules in R(ℂ)
In R(ℂ)*, we have almost the same calculation rules as in* I(ℝ). *We formulate the differences to Lemma A.1.5:*

Associativity: *The associative law holds for the addition in* R(ℂ)*, only.*

Neutral Elements: *The unique neutral element for addition is* $X = [0, 0] + i[0, 0]$, *the unique neutral element for multiplication is* $Y = [1, 1] + i[0, 0]$.

The inclusion monotonicity property holds on R(ℂ) as well. We state it as a theorem since this property is crucial for our algorithms.

Theorem A.2.6. *[1, Thm. 9, p. 64]* Inclusion Monotonicity Property for $\mathrm{R}(\mathbb{C})$
Let $A^{(1)}$, $A^{(2)}$, $B^{(1)}$ and $B^{(2)} \in \mathrm{R}(\mathbb{C})$ with $A^{(1)} \subset B^{(1)}$ and $A^{(2)} \subset B^{(2)}$.
Then

$$A^{(1)} \circ A^{(2)} \subset B^{(1)} \circ B^{(2)}$$

holds for all interval operations $\circ \in \{+, -, \cdot, :\}$.

Corollary A.2.7. *[1, Cor. 10, p. 66] Let A, $B \in \mathrm{R}(\mathbb{C})$ and $a \in A$, $b \in B$. Then*

$$a \circ b \in A \circ B$$

holds for all operations $\circ \in \{+, -, \cdot, :\}$.

We define a square root operation on $\mathrm{R}(\mathbb{C})$. It assigns to each rectangle $A \in \mathrm{R}(\mathbb{C})$ with $0 \notin A$ the smallest rectangle in $\mathrm{R}(\mathbb{C})$ containing $\{\sqrt{a}\,|\,a \in A\}$. Here, $\sqrt{\_}$ is a branch of the square root function that is defined on A.

Definition A.2.8. Square Roots in Rectangular Interval Arithmetic
Let $A \in \mathrm{R}(\mathbb{C})$ with $0 \notin A$. Let $\sqrt{\_}$ be a branch of the square root function that is defined on A. Then $\sqrt{A}$ is defined as the smallest rectangle in $\mathrm{R}(\mathbb{C})$ containing the set $\{\sqrt{a}\,|\,a \in A\}$:

$$\sqrt{A} := \bigcap \big\{B \in R(\mathbb{C}) \,|\, \{\sqrt{a}\,|\,a \in A\} \subset B\big\}$$

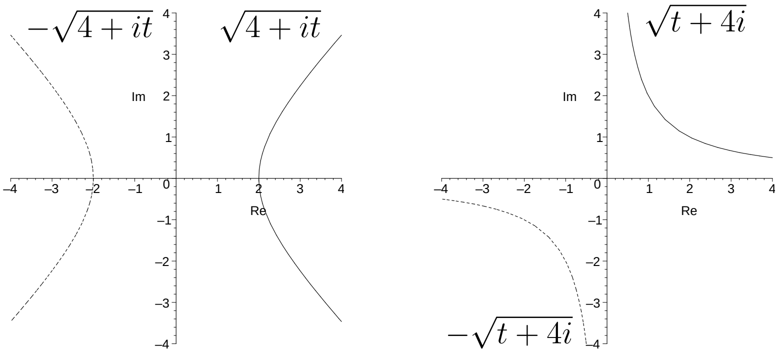
Lemma A.2.9 shows how $\sqrt{A}$ can be computed.

Lemma A.2.9. Computation of Square Roots in Rectangular Arithmetic
Let $A = [a_1, a_2] + i[b_1, b_2] \in \mathrm{R}(\mathbb{C})$ with $0 \notin A$, and let $x \in A$ be the point that is closest to $0 \in \mathbb{C}$. Then, $\sqrt{A}$ is the smallest rectangle in $\mathrm{R}(\mathbb{C})$ that contains the square roots of the four vertices of A and of the point x:

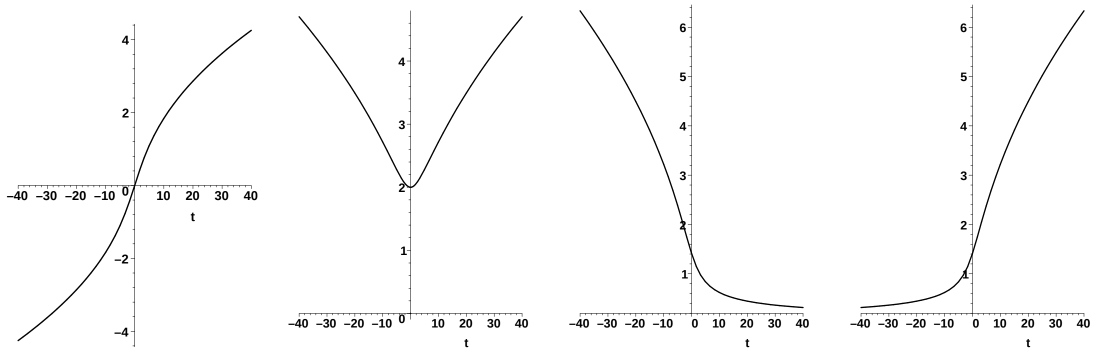
$$\sqrt{A} = \bigcap \Big\{B \in R(\mathbb{C}) \,\Big|\, \sqrt{x}, \sqrt{a_1 + ib_1}, \sqrt{a_1 + ib_2}, \sqrt{a_2 + ib_1}, \sqrt{a_2 + ib_2} \in B\Big\}.$$

Furthermore, we have $0 \notin \sqrt{A}$.

Proof. We have to show that the functions $\mathrm{Re}\sqrt{z}$ and $\mathrm{Im}\sqrt{z}\colon A \to \mathbb{R}$ take there minimum and there maximum at one of the four vertices of the rectangle A or the point $x \in A$ that is closest to $0 \in \mathbb{C}$. For this purpose, we consider the four lines l_{h1}, l_{h2}, l_{v1} and l_{v2} containing the four edges of the rectangle A. We extend the branch $\sqrt{\_}$ of the square root function such that it is also defined on these four lines. The two vertical lines l_{v1} and l_{v2} have the structure $l_v\colon \mathbb{R} \to \mathbb{C}$, $t \mapsto a+ti$, and the horizontal lines l_{h1} and l_{h2} have the form $l_h\colon \mathbb{R} \to \mathbb{C}$, $t \mapsto t+ib$. We investigate the functions $\sqrt{l_v(t)}$ and $\sqrt{l_h(t)}$ by analyzing the real functions $\mathrm{Im}(\sqrt{l_v(t)}\,)$, $\mathrm{Re}(\sqrt{l_v(t)}\,)$, $\mathrm{Im}(\sqrt{l_h(t)}\,)$, and $\mathrm{Re}(\sqrt{l_h(t)}\,)$; see Figure A.1.



(a) The left figure shows the graph of $\sqrt{l_v(t)} = \sqrt{4+it}$, the right one shows the graph of $\sqrt{l_h(t)} = \sqrt{t+4i}$. The horizontal axis is the real axis, the vertical axis is the imaginary axis.



(b) $\mathrm{Im}(\sqrt{4+it}\,)$ (c) $\mathrm{Re}(\sqrt{4+it}\,)$ (d) $\mathrm{Im}(\sqrt{t+4i}\,)$ (e) $\mathrm{Re}(\sqrt{t+4i}\,)$

Figure A.1: The functions $\sqrt{l_v(t)}$ and $\sqrt{l_h(t)}$ are plotted for $l_v(t) = 4 + it$ and $l_h(t) = t + 4i$; see Figure A.1(a). In Figure A.1(b)-A.1(e), the graphs of the functions $\mathrm{Im}(\sqrt{l_v(t)}\,)$, $\mathrm{Re}(\sqrt{l_v(t)}\,)$, $\mathrm{Im}(\sqrt{l_h(t)}\,)$, and $\mathrm{Re}(\sqrt{l_h(t)}\,)$ are shown. The horizontal axis is the t-axis.

First, we consider the vertical line $l_v(t)$ and the function $\sqrt{l_v(t)}$. The derivative of $\sqrt{l_v(t)}$ is

$$\left(\sqrt{l_v(t)}\right)' = \frac{i}{2\sqrt{l_v(t)}} = \frac{i \cdot \overline{\sqrt{l_v(t)}}}{2|l_v(t)|}.$$

We consider the case $a > 0$ and show that the function $\mathrm{Im}(\sqrt{l_v(t)}\,)\colon \mathbb{R} \to \mathbb{R}$ is strictly monotonic as well as the functions $\mathrm{Re}(\sqrt{l_v(t)}\,)\colon \mathbb{R}_{>0} \to \mathbb{R}$ and $\mathrm{Re}(\sqrt{l_v(t)}\,)\colon \mathbb{R}_{<0} \to \mathbb{R}$. The Cauchy-Riemann differential equations imply

$$\left(\mathrm{Im}\left(\sqrt{l_v(t)}\right)\right)' = \mathrm{Im}\left(\sqrt{l_v(t)}\right)' = \underbrace{\frac{1}{2|l_v(t)|}}_{>0}\mathrm{Re}\left(\sqrt{l_v(t)}\right)$$

and

$$\left(\mathrm{Re}\left(\sqrt{l_v(t)}\right)\right)' = \mathrm{Re}\left(\sqrt{l_v(t)}\right)' = \underbrace{\frac{1}{2|l_v(t)|}}_{>0}\mathrm{Im}\left(\sqrt{l_v(t)}\right).$$

We consider the function $\mathrm{Im}(\sqrt{l_v(t)}\,)$ and its derivative $(\mathrm{Im}(\sqrt{l_v(t)}\,))'$. For this purpose, we examine $\mathrm{Re}(\sqrt{l_v(t)}\,)$. Let $t_0 \in \mathbb{R}$ with $\mathrm{Re}(\sqrt{l_v(t_0)}\,) = 0$, and let r, ϕ with $\sqrt{l_v(t_0)} = re^{i\phi}$. We have $\mathrm{Re}(\sqrt{l_v(t_0)}\,) = 0$ if and only if $\phi = \frac{\pi}{2} + k\pi$ holds for an integer $k \in \mathbb{Z}$. Hence, we have $l_v(t_0) = r^2e^{i2\phi} = r^2e^{i(\pi+2k\pi)} = -r^2$, and l_v intersects the negative real line which is not true; we assumed $a > 0$. Hence, $(\mathrm{Im}(\sqrt{l_v(t)}\,))'$ is a continuous real function without zeros. Consequently, this derivative does not change its sign, and $\mathrm{Im}(\sqrt{l_v(t)}\,)$ is strictly monotonic.

We analyze the function $\mathrm{Re}(\sqrt{l_v(t)}\,)$. We have already shown that $\mathrm{Re}(\sqrt{l_v(t)}\,)$ does not change its sign. We consider the derivative $(\mathrm{Re}(\sqrt{l_v(t)}\,))'$ and examine $\mathrm{Im}(\sqrt{l_v(t)}\,)$. Let $t_0 \in \mathbb{R}$ with $\mathrm{Im}(\sqrt{l_v(t_0)}\,) = 0$, and let r, ϕ with $\sqrt{l_v(t_0)} = re^{i\phi}$. We have $\mathrm{Im}(\sqrt{l_v(t_0)}\,) = 0$ if and only if $\phi = k\pi$ holds for a $k \in \mathbb{Z}$. This notation implies $l_v(t_0) = r^2e^{i2\phi} = r^2e^{i2k\pi} = r^2$, and (t_0, r^2) is an intersection point of l_v and the positive real line. Therefore, we have $t_0 = 0$, and $(\mathrm{Re}(\sqrt{l_v(t)}\,))'$ is a continuous real function having the unique zero $t_0 = 0$. Consequently, this derivative does neither change its sign on $\mathbb{R}_{<0}$ nor on $\mathbb{R}_{>0}$, and $\mathrm{Re}(\sqrt{l_v(t)})$ is strictly monotonic on $\mathbb{R}_{<0}$ and on $\mathbb{R}_{>0}$.

The case $a < 0$ can be reduced to the case $a > 0$ by converting $\sqrt{l_v(t)} = \sqrt{a + ti} = \sqrt{-1}\sqrt{-a + i(-t)}$. For $a = 0$, the functions $\mathrm{Im}(\sqrt{l_v(t)}\,)$, $\mathrm{Re}(\sqrt{l_v(t)}\,)\colon \mathbb{R}_{>0} \to \mathbb{R}$ and $\mathrm{Im}(\sqrt{l_v(t)}\,)$, $\mathrm{Re}(\sqrt{l_v(t)}\,)\colon \mathbb{R}_{<0} \to \mathbb{R}$ are monotonic as well.

Second, we treat the horizontal line $l_h(t) = t + ib$ and the function $\sqrt{l_h(t)}$. The derivative of $\sqrt{l_h(t)}$ is

$$\left(\sqrt{l_h(t)}\right)' = \frac{1}{2\sqrt{l_h(t)}} = \frac{\overline{\sqrt{l_h(t)}}}{2|l_h(t)|}.$$

We show by similar arguments as above that the functions $\operatorname{Im}(\sqrt{l_h(t)}\,)$ and $\operatorname{Re}(\sqrt{l_h(t)}\,)\colon \mathbb{R} \to \mathbb{R}$ are strictly monotonic for $b \neq 0$. The Cauchy-Riemann differential equations imply

$$\left(\operatorname{Im}\left(\sqrt{l_h(t)}\right)\right)' = \operatorname{Im}\left(\sqrt{l_h(t)}\right)' = -\underbrace{\frac{1}{2|l_h(t)|}}_{>0}\operatorname{Im}\left(\sqrt{l_h(t)}\right)$$

and

$$\left(\operatorname{Re}\left(\sqrt{l_h(t)}\right)\right)' = \operatorname{Re}\left(\sqrt{l_h(t)}\right)' = \underbrace{\frac{1}{2|l_h(t)|}}_{>0}\operatorname{Re}\left(\sqrt{l_h(t)}\right).$$

To investigate the monotonicity of the functions $\operatorname{Im}(\sqrt{l_h(t)}\,)$ and $\operatorname{Re}(\sqrt{l_h(t)}\,)$, we consider their derivatives $(\operatorname{Im}(\sqrt{l_h(t)}\,))'$ and $(\operatorname{Re}(\sqrt{l_h(t)}\,))'$. Thus, we examine $\operatorname{Im}(\sqrt{l_h(t)}\,)$ and $\operatorname{Re}(\sqrt{l_h(t)}\,)$: Let $t_0 \in \mathbb{R}$ with $\operatorname{Im}(\sqrt{l_h(t_0)}\,) = 0$ or $\operatorname{Re}(\sqrt{l_h(t_0)}\,) = 0$. Let r, ϕ such that $\sqrt{l_h(t_0)} = re^{i\phi}$. We have $\operatorname{Im}(\sqrt{l_h(t_0)}\,) = 0$ if and only if $\phi = k\pi$ for an integer $k \in \mathbb{Z}$ holds, and we have $\operatorname{Re}(\sqrt{l_h(t_0)}\,) = 0$ if and only if $\phi = \frac{\pi}{2} + \tilde{k}\pi$ for an integer $\tilde{k} \in \mathbb{Z}$ holds. So, $l_h(t_0) = r^2 e^{i2\phi} = \pm r^2$, and the horizontal line l_h intersects the real line which is not true; we have $b \neq 0$. Thus, both real functions $(\operatorname{Im}(\sqrt{l_h(t)}\,))'$ and $(\operatorname{Re}(\sqrt{l_h(t)}\,))'$ are continuous functions without zeros. Hence they cannot change there signs. Consequently, $\operatorname{Im}(\sqrt{l_h(t)}\,)$ and $\operatorname{Re}(\sqrt{l_h(t)}\,)$ are strictly monotonic functions. Each of these functions is either positive or negative.
For $b = 0$, the functions $\operatorname{Im}(\sqrt{l_h(t)}\,)$, $\operatorname{Re}(\sqrt{l_h(t)}\,)\colon \mathbb{R}_{>0} \to \mathbb{R}$ and $\operatorname{Im}(\sqrt{l_h(t)}\,)$, $\operatorname{Re}(\sqrt{l_h(t)}\,)\colon \mathbb{R}_{<0} \to \mathbb{R}$ are monotonic as well.

These observations combined with the injectivity of the root function $\sqrt{\_}\colon \mathbb{C}_{\neq 0} \to \mathbb{C}$ imply that the functions $\operatorname{Re}\sqrt{z}$ and $\operatorname{Im}\sqrt{z}\colon A \to \mathbb{R}$ take there minimum and there maximum at one of the four vertices of the rectangle A or at the point $x \in A$ that is closest to $0 \in \mathbb{C}$.

The observations also imply $0 \notin \sqrt{A}$: Since $0 \notin A$, either the two horizontal lines l_{h1} and l_{h2} or the two vertical lines l_{v1} and l_{v2} are contained in one of the coordinate half planes. If the two horizontal lines l_{h1} and l_{h2} lie on the same side of the real axis, then $\sqrt{A}$ is contained in one open quadrant of the coordinate system. This fact holds since both functions $\operatorname{Im}(\sqrt{l_h(t)})$ and $\operatorname{Re}(\sqrt{l_h(t)})$ do not change the sign. For both horizontal lines l_{h1} and l_{h2}, these signs are identical, since they lie on the same side of the real axis. If the two vertical lines l_{v1} and l_{v2} lie on the same side of the imaginary axis, then $\sqrt{A}$ is contained in an open half plane of the coordinate system. This holds since either the function $\operatorname{Re}(\sqrt{l_v(t)})$ or the function $\operatorname{Im}(\sqrt{l_v(t)})$ does not change the sign. For both vertical lines l_{v1} and l_{v2} this sign of $\operatorname{Re}(\sqrt{l_v(t)})$ or of $\operatorname{Im}(\sqrt{l_v(t)})$ is the same, since they lie on the same side of the imaginary axis. □

Corollary A.2.10. *Let $A = [a_1, a_2] + i[b_1, b_2] \in \mathrm{R}(\mathbb{C})$ with $0 \notin A$, and let $\sqrt[+]{\_}$ and $\sqrt[-]{\_}$ be the two branches of the root function that are defined on A. Then we have*

1. $\sqrt[-]{A} = -\sqrt[+]{A}$; *and*
2. $\sqrt[+]{A} \cap \sqrt[-]{A} = \emptyset$.

Proof. We observe that $\sqrt[+]{A}$ and $\sqrt[-]{A}$ are symmetric axis-parallel rectangles. Since $0 \notin \sqrt[+]{A}$, we have $\sqrt[+]{A} \cap \sqrt[-]{A} = \emptyset$. □

The inclusion monotonicity property also holds for the square root operation in rectangular interval arithmetic.

Lemma A.2.11. Inclusion Monotonicity of $\sqrt{\_}$ in $\mathrm{R}(\mathbb{C})$
Let A, $B \in \mathrm{R}(\mathbb{C})$ with $A \subset B$, $0 \notin B$, and let $\sqrt{\_}$ be a branch of the square root function defined on B. Then $\sqrt{A} \subset \sqrt{B}$ holds.

Proof. The intervals $\sqrt{A}$ and $\sqrt{B}$ are the smallest axis parallel rectangles containing the sets $\{\sqrt{a} \,|\, a \in A\}$ and $\{\sqrt{b} \,|\, b \in B\}$. Clearly, $A \subset B$ implies $\{\sqrt{a} \,|\, a \in A\} \subset \{\sqrt{b} \,|\, b \in B\}$. Since rectangular complex intervals are axis parallel rectangles in the complex plane, we have $\sqrt{A} \subset \sqrt{B}$. □

According to Definition A.2.8, we could also define higher order roots in rectangular arithmetic. If $\sqrt[n]{\_}$ is a branch of the nth root function and if $A \in \mathrm{R}(\mathbb{C})$ with $0 \notin A$, then $\sqrt[n]{A}$ is defined as the smallest rectangle in $\mathrm{R}(\mathbb{C})$ containing the set $\{\sqrt[n]{a} \,|\, a \in A\}$:

$$\sqrt[n]{A} := \bigcap \big\{B \in \mathrm{R}(\mathbb{C}) \,|\, \{\sqrt[n]{a} \,|\, a \in A\} \subset B\big\}$$

Then, the nth root function fulfills the inclusion monotonicity property. This fact can be proven like Lemma A.2.11. Unfortunately, we cannot compute $\sqrt[n]{A}$ in the simple way described in Lemma A.2.9; see Figure A.2 for an example with cubic roots.

We transfer Lemma A.1.10 to complex rectangular interval arithmetic.

Lemma A.2.12. *Let $\circ \in \{+, -, \cdot, :\}$ and ϵ_1, $\epsilon_2 > 0$. Let $a = a_1 + ia_2$, $b = b_1 + ib_2 \in \mathbb{R}$, and $c = c_1 + ic_2 := a \circ b$. Then there exist δ_{a1}, $\delta_{a2} > 0$ and δ_{b1}, $\delta_{b2} > 0$ with*

$$\begin{aligned}(a + [-\delta_{a1}, \delta_{a1}] + i[-\delta_{a2}, \delta_{a2}]) \circ (b + [-\delta_{b1}, \delta_{b1}] + i[-\delta_{b2}, \delta_{b2}]) \\ \subset c + [-\epsilon_1, \epsilon_1] + i[-\epsilon_2, \epsilon_2].\end{aligned}$$

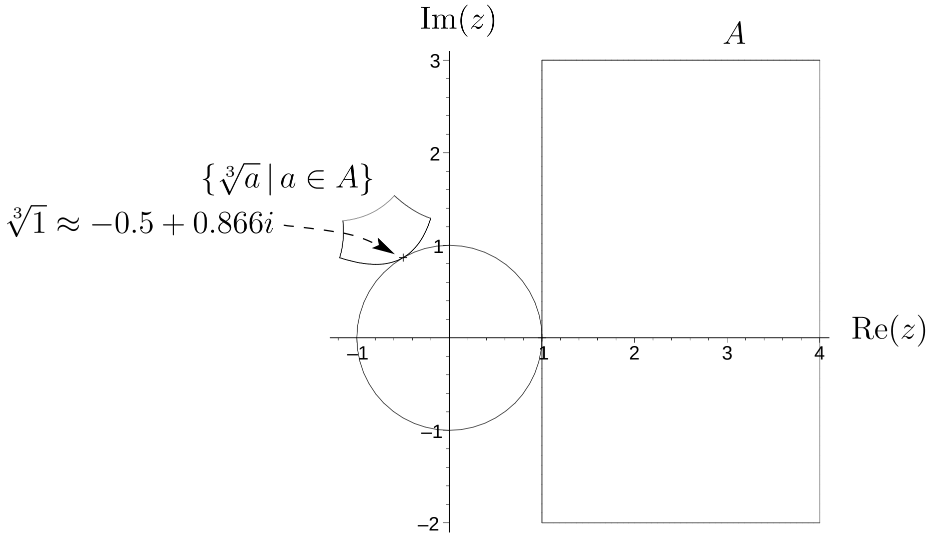


Figure A.2: An example of a cubic root of a rectangular interval A is shown. The circle is the unit circle, it passes through the point $\sqrt[3]{1} = 1/2 + i/2\sqrt{3}$. The set $\{\sqrt[3]{a} \mid a \in A\}$ is **not** contained in the smallest rectangle containing the cubic roots of the vertices of the rectangle and the cubic root of the closest point to the origin, which is 1 in this example; compare with Lemma A.2.9.

For $c = \pm\sqrt{a}$, we have

$$\sqrt{a + [-\delta_{a1}, \delta_{a1}] + i[-\delta_{a2}, \delta_{a2}]} \subset c + [-\epsilon_1, \epsilon_1] + i[-\epsilon_2, \epsilon_2].$$

For the division operation we assume $b \neq 0$, and for the square root operation we assume $a \neq 0$. If is known in advance that the points a and b stay in fixed intervals A, $B \in \mathrm{R}(\mathbb{C})$, then δ_{a1}, $\delta_{a2} > 0$ and δ_{b1}, $\delta_{b2} > 0$ can be chosen such that they only depend on ϵ_1, ϵ_2, A and B.

Proof. The arithmetic operations on $\mathrm{R}(\mathbb{C})$ are defined by the operations of $\mathrm{I}(\mathbb{R})$. Hence we can apply Lemma A.1.10, and it remains to show the bounds of Lemma A.2.12 for the denominator of the division operation and for the root function. For the denominator we observe that the function $x \mapsto x^2$ is continuous on $\mathbb{R}$. Taking square roots is a continuous operation as well. The claimed bounds follow by considering the metric induced by the maximum norm on $\mathbb{C} \cong \mathbb{R}^2$. The supplement of Lemma A.2.12 holds since the considered functions are uniformly continuous on the compact sets $A \times B$ and A. □

A.2.2 Circular Interval Arithmetic

We define circular complex intervals. Again, we follow the presentation of [1, Abschnitt 5]. The notation for circular intervals is taken from [57].

Definition A.2.13. *[1, Def. 5, p. 59], [57, p. 20]* Circular Complex Interval
Let $a \in \mathbb{C}$ and $0 \le r \in \mathbb{R}$. We define

$$Z := \{a; r\} := \{z \in \mathbb{C} \mid |z - a| \le r\}$$

and term $Z = \{a; r\}$ a circular complex interval. *We denote the set of all circular intervals $\{a; r\}$ with $a \in \mathbb{C}$ and $r \ge 0$ by* $\mathrm{K}(\mathbb{C})$.

The letter "K" in $\mathrm{K}(\mathbb{C})$ comes from "Kreis", which is the German word for circle. The set $A = \{a; r\}$ is a closed circle in the complex plane. A complex number $a \in \mathbb{C}$ can be considered as the circular interval $\{a; 0\}$. Hence $\mathbb{C} \subset \mathrm{K}(\mathbb{C})$. In contrast to rectangular complex intervals, a real interval $[a_1, a_2] \in \mathrm{I}(\mathbb{R})$ cannot be regarded as a circular interval, unless it consists of a single point. Hence we have $\mathrm{I}(\mathbb{R}) \not\subset \mathrm{K}(\mathbb{C})$. In general, the intersection of two circular intervals is not a circular interval. This fact is a disadvantage of circular arithmetic over rectangular or real arithmetic. Let $A = \{a; r_a\}$ and $B = \{b; r_b\}$ be complex intervals. Then $A = B$ holds if and only if $a = b$ and $r_a = r_b$; see [1, p. 60].

We define the arithmetic operations on $\mathrm{K}(\mathbb{C})$ as a generalization of the corresponding arithmetic operations on $\mathbb{C}$ as in [1, Def. 7, p. 60]:

Definition A.2.14. *[1, Def. 7, p. 60]* Arithmetic Operations on $\mathrm{K}(\mathbb{C})$
Let $A = \{a; r_a\}$ and $B = \{b; r_b\} \in \mathrm{K}(\mathbb{C})$, and let $\circ$ be one of the operations $+$, $-$, $\cdot$ and $:$. Then $A \circ B$ is defined as follows:

$$\begin{aligned}
A + B &:= \{a + b\,;\, r_a + r_b\};\\
A - B &:= \{a - b\,;\, r_a + r_b\};\\
A \cdot B &:= \{a \cdot b\,;\, |a| r_b + |b| r_a + r_a r_b\};\\
\frac{1}{B} &:= \left\{\frac{\bar{b}}{b\bar{b} - r_b^2}\,;\, \frac{r_b}{b\bar{b} - r_b^2}\right\};\\
A : B &:= A \cdot \frac{1}{B}.
\end{aligned}$$

The inversion $\frac{1}{B}$ and hence the division $A : B$ is only defined if $0 \notin B$. As usual, $|b| = \sqrt{b_1^2 + b_2^2}$ is the modulus of $b = b_1 + ib_2 \in \mathbb{C}$ and $\bar{b} = b_1 - ib_2$ is the complex conjugate of b.

If $0 \notin B$, then $r_b < |b|$ holds and vice versa. This fact implies $b\bar{b} - r_b^2 = |b|^2 - r_b^2 > 0$ if and only if $0 \notin B$. Hence, the inversion and the division operation is defined for all $B = \{b; r_b\} \in \mathrm{K}(\mathbb{C})$ with $0 \notin B$. As in $\mathrm{R}(\mathbb{C})$, there is an overestimation of the range of each multiplication and hence of each division in $\mathrm{K}(\mathbb{C})$. We remark in advance that there is no overestimation for an inversion. Compare Lemma A.2.15 with Definition A.1.1, Lemma A.1.2, and Lemma A.2.4.

Lemma A.2.15. *Let A, $B \in \mathrm{K}(\mathbb{C})$. Then the following holds for the arithmetic operations in $\mathrm{K}(\mathbb{C})$; see [1, Abschnitt 5].*

1. $A \pm B = \{a \pm b |\, a \in A,\, b \in B\}$;
2. $\frac{1}{B} = \left\{\frac{1}{b} \,|\, b \in B\right\}$ *if* $0 \notin B$;
3. $A \cdot B \supset \{a \cdot b |\, a \in A,\, b \in B\}$,
   and $A \cdot B$ is the smallest circle in $\mathrm{R}(\mathbb{C})$ with $a \cdot b$ as center that contains the set $\{a \cdot b |\, a \in A,\, b \in B\}$;
4. $A : B \supset \{a : b |\, a \in A,\, b \in B\}$ *if* $0 \notin B$.

Proof. The proof of Lemma A.2.15 can be found in [1, Abschnitt 5]. We give the proofs of 2 and 3, since here surprising facts are stated.

ad (2): The inversion $z \mapsto \frac{1}{z}$ is a Möbius transform [55, Abschnitt 2, Sect. 6.10]. Since $0 \notin B = \{b; r_b\}$, the image of the circle B under the inversion is the circle $\left\{\frac{\bar{b}}{b\bar{b}-r_b^2}\,;\, \frac{r_b}{b\bar{b}-r_b^2}\right\}$. Differing from [1], we present the calculations of this fact from complex analysis.

We are given the circle $B = \{b; r_b\}$ with $b = b_1 + ib_2$. Since $0 \notin B$, we have $|b| > r_b$. We are interested in the set

$$
\begin{aligned}
\left\{\frac{1}{z}\,\middle|\, z \in \{b; r_b\}\right\} &= \left\{\frac{1}{z}\,\middle|\, |z-b| \le r_b\right\} \\
&= \left\{\frac{1}{z}\,\middle|\, (b_1 - z_1)^2 + (b_2 - z_2)^2 - r_b^2 \le 0,\ z = z_1 + iz_2 \in \mathbb{C}\right\} \\
&= \left\{\frac{1}{z}\,\middle|\, z_1^2 + z_2^2 - 2b_1 z_1 - 2b_2 z_2 + b_1^2 + b_2^2 - r_b^2 \le 0\right\} \\
&= \left\{\frac{1}{z}\,\middle|\, \underbrace{z\bar{z}}_{z_1^2+z_2^2} + \underbrace{(-\bar{b})z}_{\substack{-b_1z_1-b_2z_2 \\ +i(-b_1z_2+b_2z_1)}} + \underbrace{(-b)\bar{z}}_{\substack{-b_1z_1-b_2z_2 \\ +i(b_1z_2-b_2z_1)}} + \underbrace{b\bar{b}}_{b_1^2+b_2^2} - r_b^2 \le 0\right\} \\
&= \left\{z\,\middle|\, \frac{1}{z\bar{z}} + (-\bar{b})\frac{1}{z} - b\frac{1}{\bar{z}} + b\bar{b} - r_b^2 \le 0\right\} \qquad \textit{(Inversion } z \mapsto \frac{1}{z}\textit{)} \\
&= \left\{z \,|\, 1 - \bar{b}\bar{z} - bz + (b\bar{b} - r_b^2)z\bar{z} \le 0\right\} \qquad \textit{(Multiplication with } z\bar{z}\textit{)}
\end{aligned}
$$

$$= \left\{ z \,\middle|\, z\bar{z} + \underbrace{\frac{-bz}{b\bar{b} - r_b^2}}_{\substack{\frac{1}{b\bar{b}-r_b^2}(-b_1z_1+b_2z_2 \\ +i(-b_1z_2-b_2z_1))}} + \underbrace{\frac{-\bar{b}\bar{z}}{b\bar{b} - r_b^2}}_{\substack{\frac{1}{b\bar{b}-r_b^2}(-b_1z_1+b_2z_2 \\ +i(b_1z_2+b_2z_1))}} + \frac{1}{b\bar{b} - r_b^2} \leq 0 \right\}$$

(Division by $b\bar{b} - r_b^2 > 0$)

$$= \left\{ z \,\middle|\, (z_1 - \frac{b_1}{b\bar{b} - r_b^2})^2 + (z_2 - \frac{-b_2}{b\bar{b} - r_b^2})^2 - \frac{r_b^2}{(b\bar{b} - r_b^2)^2} \leq 0 \right\}$$

$$= \left\{ \frac{\bar{b}}{b\bar{b} - r_b^2} \,;\, \frac{r_b}{b\bar{b} - r_b^2} \right\}.$$

ad (3): Let $A = \{a; r_a\}$, $B = \{b; r_b\} \in \mathrm{K}(\mathbb{C})$, let $z_a \in A$ and $z_b \in B$. Claim 3 follows by the inequalities (see [1, p. 61])

$$\begin{aligned} |z_a z_b - ab| &= |\underbrace{a(z_b - b)}_{az_b - ab} + \underbrace{b(z_a - a)}_{bz_a - ab} + \underbrace{(z_a - a)(z_b - b)}_{z_a z_b - bz_a - az_b + ab}| \\ &\leq |a||z_b - b| + |b||z_a - a| + |z_a - a||z_b - b| \\ &\leq |a| r_b + |b| r_a + r_a r_b. \end{aligned}$$

□

In the set $\mathrm{K}(\mathbb{C})$, we have almost the same calculation rules as in $\mathrm{I}(\mathbb{R})$ and in $\mathrm{R}(\mathbb{C})$; see Lemma A.1.5, Lemma A.2.5, and [1, Thm. 8, p. 62]. The differences are stated in Lemma A.2.16 below.

Lemma A.2.16. Calculation Rules in $\mathrm{K}(\mathbb{C})$
In $\mathrm{K}(\mathbb{C})$, we have almost the same calculation rules as in $\mathrm{I}(\mathbb{R})$ and in $\mathrm{R}(\mathbb{C})$. We formulate the differences to Lemma A.1.5 and to Lemma A.2.5:

Associativity: *The associative law holds for addition and multiplication in* $\mathrm{K}(\mathbb{C})$.

Neutral Elements: $X = \{0; 0\}$ *is the unique neutral element for the addition,* $Y = \{1; 0\}$ *is the unique neutral element for the multiplication.*

The inclusion monotonicity property holds on $\mathrm{K}(\mathbb{C})$ as well. We state it as a theorem since this property is crucial for our algorithms.

Theorem A.2.17. *[1, Thm. 9, p. 64]* Inclusion Monotonicity Property for $\mathrm{K}(\mathbb{C})$
Let $A^{(1)}$, $A^{(2)}$, $B^{(1)}$ and $B^{(2)} \in \mathrm{K}(\mathbb{C})$ with $A^{(1)} \subset B^{(1)}$ and $A^{(2)} \subset B^{(2)}$. Then

$$A^{(1)} \circ A^{(2)} \subset B^{(1)} \circ B^{(2)}$$

holds for all operations $\circ \in \{+, -, \cdot, :\}$.

Corollary A.2.18. *[1, Cor. 10, p. 66] Let A, $B \in \mathrm{K}(\mathbb{C})$ and $a \in A$, $b \in B$. Then*

$$a \circ b \in A \circ B$$

holds for all operations $\circ \in \{+, -, \cdot, :\}$.

Remark A.2.19. Some effort has been made to decrease the overestimation of the multiplication in $\mathrm{K}(\mathbb{C})$. One possibility is to define $\{a; r_a\} \cdot \{b; r_b\}$ as the smallest disc containing the set $\{z_a z_b \,|\, z_a \in \{a; r_a\},\ z_b \in \{b; r_b\}\}$. The decrease of the radius is achieved by moving the midpoint apart from the point ab. This multiplication is called minimal circular arithmetic; it is e.g. mentioned in [1, 57]. Unfortunately, it does not have the inclusion monotonicity property, and we cannot use it for our algorithms.

In [29], Hauenschild discusses an alternative circular arithmetic, which is called optimal circular arithmetic. Here, $\{a; r_a\} \cdot \{b; r_b\}$ is defined as the circle with the smallest radius that contains the set $\{z_a z_b \,|\, z_a \in \{a; r_a\},\ z_b \in \{b; r_b\}\}$ and that is contained in the circle $\{0; \sup\{|z_a z_b| \,|\, z_a \in \{a; r_a\},\ z_b \in \{b; r_b\}\}$. The product circle in this arithmetic has in general a smaller radius than the product circle defined in Definition A.2.14. Additionally, it fulfills the inclusion monotonicity property. Thus, this arithmetic could be a good choice for our algorithms.

Now, we define a square root operation on $\mathrm{K}(\mathbb{C})$ like Petković and Petković in [56]. We are only interested in square roots of circular intervals $A = \{a; r_a\}$ with $0 \notin A$. In [56], the case $0 \in A$ is discussed as well.

Definition A.2.20. *(see [56])* Square Roots in Circular Interval Arithmetic
Let $A = \{a; r_a\} \in \mathrm{K}(\mathbb{C})$ with $0 \notin A$. Let $\sqrt{\_}$ be a branch of the square root function that is defined on A. We define

$$\sqrt{A} := \{\sqrt{a}\,;\ \sqrt{|a|} - \sqrt{|a| - r_a}\,\}.$$

Lemma A.2.21. *(see [56]) Let $A = \{a; r_a\} \in \mathrm{K}(\mathbb{C})$ with $0 \notin A$ as in Definition A.2.20. Then $\{\sqrt{z} \,|\, z \in A\} \subset \sqrt{A}$ holds.*

The square root operation on $\mathrm{K}(\mathbb{C})$ from Definition A.2.20 fulfills the inclusion monotonicity property; see [57, Chap. 2]:

Lemma A.2.22. Inclusion Monotonicity of $\sqrt{\_}$ in $\mathrm{K}(\mathbb{C})$
Let A, $B \in \mathrm{K}(\mathbb{C})$ with $A \subset B$, $0 \notin B$, and let $\sqrt{\_}$ be a branch of the square root function defined on B. Then $\sqrt{A} \subset \sqrt{B}$ holds.

Proof. Lemma A.2.22 can be proved by combining Claim 6 from [57, p. 32] and Theorem 2.8 from [57, p. 48]. □

For Algorithm 2, step 5, and Algorithm 4, step 5, is important that $\sqrt{A} \cap -\sqrt{A} = \emptyset$ holds. According to Lemma A.2.10, we formulate Lemma A.2.23.

Lemma A.2.23. *Let $A = \{a; r_a\} \in \mathrm{K}(\mathbb{C})$ with $0 \notin A$ and let $\sqrt{\_}$ be a branch of the root function that is defined on A. Then $0 \notin \sqrt{A}$ holds.*

Let $\sqrt[-]{\_}$ be the other branch of the root function defined on A. Then we have

1. $\sqrt[-]{A} = -\sqrt{A}$*; and*
2. $\sqrt{A} \cap \sqrt[-]{A} = \emptyset$.

Proof. To proof the first claim of Lemma A.2.23, we remark that $0 \notin A$ implies $|a| > r_a$. Hence $\sqrt{|a| - r_a} > 0$ holds. Consequently, we have

$$|\sqrt{a}\,| = \sqrt{|a|} > \sqrt{|a|} - \sqrt{|a| - r_a},$$

which implies $0 \notin \sqrt{A}$. For the proof of the second claim, we observe that $\sqrt[-]{A}$ is the reflection of $\sqrt{A}$ at the point $0 \in \mathbb{C}$. Since $0 \notin \sqrt{A}$ by the first claim, $\sqrt{A} \cap \sqrt[-]{A} = \emptyset$ holds. □

Remark A.2.24. The presented form of the computation of square roots in circular arithmetic is called centered form. As shown in [56, p. 31], the disc $\sqrt{A}$ contacts the set $\{\sqrt{z} \mid z \in A\}$ only in a single point. Thus, the diameter of $\{\sqrt{z} \mid z \in A\}$ is smaller than the diameter of $\sqrt{A}$. For this reason, in [57, 56] the diametrical inclusive disc of the set $\{\sqrt{z} \mid z \in A\}$ is considered. This disc is the smallest disc containing the set $\{\sqrt{z} \mid z \in A\}$. Unfortunately, this definition might not fulfill the inclusion monotonicity property; compare with Remark A.2.19, see [57, p. 52]. An other improvement might be achieved by transferring Hauenschild's ideas [29] for the optimal circular arithmetic to the square root operation; see Remark A.2.19.

More generally, Petković and Petković [56, 57] investigated the computation of nth roots in circular arithmetic. If $A := \{c; r\} \in \mathrm{K}(\mathbb{C})$ with $0 \notin \{c; r\}$, and if $\sqrt[n]{\_}$ is a branch of the nth root function defined on $\{c; r\}$, then we set

$$\sqrt[n]{\{c; r\}} := \left\{ \sqrt[n]{c};\ \sqrt[n]{|c|} \left(1 - \left(1 - \frac{r}{|c|} \right)^{1/n} \right) \right\}.$$

Note that $0 \notin \{c; r\}$ implies $r/|c| < 1$. As mentioned in [56, p. 30], the inclusion $\{\sqrt[n]{a} \mid a \in A = \{c; r\}\} \subset \sqrt[n]{\{c; r\}}$ holds. Furthermore, the inclusion monotonicity property is fulfilled. This fact can be proved like Lemma A.2.22. If $+\sqrt[n]{\_}$ and $-\sqrt[n]{\_}$ are two branches of the nth root, then the discs $+\sqrt[n]{\{c; r\}}$ and $-\sqrt[n]{\{c; r\}}$ might overlap if the radius r is too large and if $n > 2$; see [56, Thm. 1, p. 30].

We adapt Lemma A.1.10 and Lemma A.2.12 to circular interval arithmetic.

Lemma A.2.25. *Let* $\circ \in \{+, -, \cdot, :\}$ *and* $\epsilon > 0$. *Let* $a, b \in \mathbb{C}$ *and* $c := a \circ b$. *Then there are* $\delta_a > 0$ *and* $\delta_b > 0$ *with*

$$\{a; \delta_a\} \circ \{b; \delta_b\} \subset \{c; \epsilon\}.$$

For $c = \pm\sqrt{a}$, *we have*

$$\sqrt{\{a; \delta_a\}} \subset \{c; \epsilon\}.$$

For the division operation we assume $b \neq 0$, *and for the square root operation we assume* $a \neq 0$. *If is known in advance that* $a \in A$ *and that* $b \in B \in \mathrm{K}(\mathbb{C})$, *then* δ_a *and* δ_b *can be chosen such that they only depend on* ϵ, A, *and* B.

Proof. We consider the space $\mathbb{C}^2 \cong \mathbb{R}^4$ with the metric $\mathrm{d}(a + ib, c + id) := \max\{|a - c|, |b - d|\}$. On $\mathbb{C} \cong \mathbb{R}^2$ we choose the Euclidean metric. The functions $+, -, \cdot \colon \mathbb{C}^2 \to \mathbb{C}$ and $1/\_, \sqrt{\_} \colon \mathbb{C}_{\neq 0} \to \mathbb{C}$ are continuous, which implies the bounds of Lemma A.2.25. Note that $\{a; \delta_a\} \cdot \{b; \delta_b\}$ is the smallest disc with ab as midpoint that contains the set $\{z_a z_b \mid z_a \in \{a; \delta_a\},\ z_b \in \{b; \delta_b\}\}$. Similarly, $\sqrt{\{a; \delta_a\}}$ is the smallest disc with $\sqrt{a}$ as midpoint that contains the set $\{\sqrt{z} \mid z \in \{a; \delta_a\}\}$. The supplement of Lemma A.2.25 holds since the considered functions are uniformly continuous on the compact sets $A \times B$, A and B. □

A.3 Rounded Interval Arithmetic

In Sections A.1 and A.2.1, we assumed exact computation for the interval bounds, and in Section A.2.2, we assumed exact computation for the centers and the radii of the circular intervals. This is a very powerful assumption since exact computation is time and space consuming; see [64, 65]. In real interval arithmetic, we can use outward rounding instead of exact computation. This leads to a slight overestimation that usually does not cause problems in applications. In outward rounding, the lower bound of an interval is rounded downwards and the upper bound is rounded upwards; see [1, Abschnitt 4]. Rounded real interval arithmetic still has the inclusion monotonicity property; see [1, Thm. 2, p. 50]. Since rectangular complex interval arithmetic is reduced to real interval arithmetic by treating the real and imaginary part separately, outward rounding preserves the inclusion monotonicity property for rectangular arithmetic, as well.

In circular complex arithmetic, the radius of the resulting circular interval has to be chosen large enough to compensate the rounding errors for the computation of the center, as well; see [46, p. 32]. In [46], aspects of inclusion monotonicity are not discussed for rounded circular arithmetic.

A.4 Interval Extensions of Real Functions

Interval arithmetic can be used to obtain bounds for the range of a real function that is defined by a sequence of elementary operations. We recall the notion of interval extensions of functions and follow the description from [35, Sect. 1.1]. We use interval extensions in Section A.5 for solving systems of equations and in Section 6.4 to discuss the overestimation of the algorithm for the Tracing Problem given in Chapter 6. Definition A.4.1 is taken from [35], a more detailed description can be found in [1].

Definition A.4.1. *[35, Defs. 1.3 and 1.5]* Interval Extension of a Function
A function $\mathbf{f}\colon \mathrm{I}(\mathbb{R}) \to \mathrm{I}(\mathbb{R})$ *is said to be an* interval extension *of* $f\colon \mathbb{R} \to \mathbb{R}$ *if*

$$\{f(x)|\, x \in A\} \subset \mathbf{f}(A)$$

holds for all intervals $A \in \mathrm{I}(\mathbb{R})$*. In higher dimensions, a function* $\mathbf{F}\colon (\mathrm{I}(\mathbb{R}))^n \to (\mathrm{I}(\mathbb{R}))^m$ *is said to be an* interval extension *of* $F\colon \mathbb{R}^n \to \mathbb{R}^m$ *if*

$$\{f(x_1, \ldots, x_n)|\, x_i \in A_i \text{ for } i = 1, \ldots, n\} \subset \mathbf{F}(A_1, \ldots, A_n)$$

holds for all intervals $A_1, \ldots, A_n \in \mathrm{I}(\mathbb{R})$*.*

Let $F\colon \mathbb{R}^n \to \mathbb{R}^m$*,* $(x_1, \ldots, x_n) \mapsto f(x_1, \ldots, x_n)$ *be a function computable as an expression, algorithm or computer program involving the four elementary arithmetic operations interspersed with evaluations of standard functions like e.g. the root functions. Then, a* natural interval extension *of* F*, whose value over a vector* $\mathbf{A} = (A_1, \ldots, A_n)$ *of intervals is denoted by* $\mathbf{F}(\mathbf{A})$*, is obtained by replacing each occurrence of the variable* x_i *by the corresponding interval* A_i *for* $i = 1, \ldots, n$*. Additionally, all operations are executed in real interval arithmetic, and the exact ranges of the standard functions are used as in Definition A.1.8.*

We define the order of an interval extension as in [35], which somehow describes the quality of the interval extension for intervals with a small width. The width $\mathrm{w}(A)$ of an interval $A = [a, b] \in \mathrm{I}(\mathbb{R})$ is defined as $\mathrm{w}(A) := b - a$. The width of a vector $\mathbf{A} = (A_1, \ldots, A_n) \in (\mathrm{I}(\mathbb{R}))^n$ of intervals $A_1, \ldots, A_n$ is defined as $\mathrm{w}(\mathbf{A}) := \max\{\mathrm{w}(A_1), \ldots, \mathrm{w}(A_n)\} = \|(\mathrm{w}(A_1), \ldots, \mathrm{w}(A_n))\|_\infty$.

Definition A.4.2. *[35, Def. 1.4]* Order of an Interval Extension
Let $\mathbf{F}$ *be an interval extension of a function* $F\colon \mathbb{R}^n \to \mathbb{R}^m$*. We assume that the image* $F(A_1, \ldots, A_n)$ *of* F *is again a vector of real intervals for all* $A_1, \ldots, A_n \in \mathrm{I}(\mathbb{R})$*. The function* $\mathbf{F}$ *is an* interval extension of order α *if there is a constant* K *with the following property: For all vectors* $\mathbf{A} = (A_1, \ldots, A_n)$ *of intervals* $A_1, \ldots, A_n \in \mathrm{I}(\mathbb{R})$ *having a sufficiently small width* $\mathrm{w}(\mathbf{A})$ *we have*

$$\mathrm{w}(\mathbf{F}(\mathbf{A})) - \mathrm{w}(F(\mathbf{A})) \le K(\mathrm{w}(\mathbf{A}))^\alpha.$$

When α is 1 or 2, we call the interval extension first order or second order, respectively.

Theorem A.4.3. *[35, Thm. 1.5] Natural interval extensions are first order.*

Depending on the particular application, second order extensions seem to be more desirable than first order extensions. One way to get a second order interval extension is to consider series expansions and to bound the range of the derivatives. The mean value extension is based on the mean value theorem.

Definition A.4.4. *[35, Def. 1.6]* Mean Value Extension
Let $F\colon D \subset \mathbb{R}^n \to \mathbb{R}$ be a continuously differentiable real valued function. Let $\mathbf{A} = (A_1, \ldots, A_n) \subset D$ be a vector of real intervals, and let $(a_1, \ldots, a_n) \in \mathbf{A}$. Let $\mathbf{F}'(\mathbf{A})$ be a componentwise interval enclosure for the range $F'(\mathbf{A}) = \{F'(x_1, \ldots, x_n) |\, x_i \in A_i$ for $i = 1, \ldots, n\}$ of the derivative F' of F over $\mathbf{A}$, i.e., $(F'(\mathbf{A}))_i \subset (\mathbf{F}'(\mathbf{A}))_i$ for $i = 1, \ldots, n$. Then the mean value extension *for F centered at $(a_1, \ldots, a_n)$ is defined by*

$$\mathbf{F_2}(\mathbf{A}, (a_1, \ldots, a_n)) := F(a_1, \ldots, a_n) + \mathbf{F}'(\mathbf{A})(\mathbf{A} - (a_1, \ldots, a_n)).$$

Theorem A.4.5. *[35, Thm. 1.6] Suppose that the components of $\mathbf{F}'$ are interval extensions of F' of order at least one. Then $\mathbf{F_2}$ is a second order interval extension of F.*

A.5 About Solving Square Systems of Equations Using Interval Analysis

Solving square systems of linear and nonlinear equations is an important topic in interval analysis [1, 35, 53, 28]. We give a brief introduction since these methods might be useful for the approximation of critical points; see Section 7.4. Throughout this section, the considered functions are assumed to be differentiable. We begin with the unary situation and investigate Newton's method, which is a common numerical method to approximate zeros of continuously differentiable functions [27].

We consider a continuously differentiable function $f\colon \mathbb{R} \to \mathbb{R}$. The aim of Newton's method is to approximate a zero ξ of f starting with an initial guess x_0. The next approximate value to ξ is the zero x_1 of the tangent to the graph of f in the point $(x_0, f(x_0))$; see Figure A.3. This approach leads to the iteration

$$x_{n+1} := x_n - \frac{f(x_n)}{f'(x_n)} = x_n - (f'(x_n))^{-1} f(x_n). \tag{A.1}$$

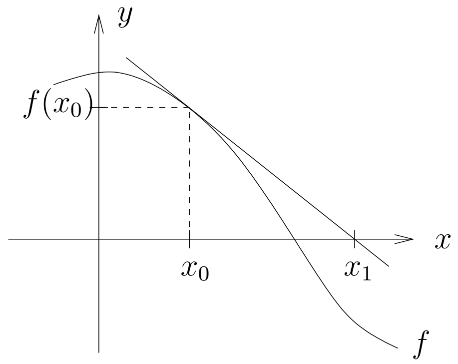


Figure A.3: Geometric interpretation of the first Newton step.

Now we consider the function f over an interval $[a, b] \in \mathrm{I}(\mathbb{R})$. Let $[c, d]$ be an interval that contains the set of all derivatives $\{f'(x) |\, x \in [a, b]\}$ over $[a, b]$. Plugging $[c, d]$ into equation A.1 leads to

$$x_{n+1} := x_n - \frac{f(x_n)}{f'(x_n)} \in x_n - \frac{f(x_n)}{[c, d]};$$

see Figure A.4. The mean value theorem implies that the graph of f over the interval $[a, b]$ is contained in the cone that is bounded by the two lines passing through the point $(x_n, f(x_n))$ and having c and d as slopes, respectively; compare with Lemma 6.3.1. Hence, the zeros of f from the interval $[a, b]$ are contained in the interval $x_n - \frac{f(x_n)}{[c,d]}$. In particular, if $[a, b] \cap \left(x_n - \frac{f(x_n)}{[c,d]}\right) = \emptyset$, then the function f does not have a zero in the interval $[a, b]$; see Figure A.4.

The previous observations lead to the interval Newton methods for unary functions. For a proper description, we use the common notion of interval extensions of functions that are defined in Section A.4. Let $\mathbf{f}'(X)$ be an interval extension of the derivative f' of the given function f over an interval $X \in \mathrm{I}(\mathbb{R})$. If $\xi \in X$ is a zero of f, then

$$\xi \in x - \frac{f(x)}{\mathbf{f}'(X)} =: \mathrm{N}(f; X, x)$$

holds for all $x \in X$. This leads to an iterative method called *univariate interval Newton method* [35]. Starting with an initial interval $X_0 = [a, b]$, we compute a set of intervals using the iteration step $X_{n+1} = X_n \cap \mathrm{N}(f; X_n, x_n)$ for an $x_n \in X_n$. Depending on the chosen interval arithmetic, the set $X_n \cap \mathrm{N}(f; X_n, x_n)$ might be disconnected and could consist of more than one intervals if for example $0 \in \mathbf{f}'(X_n)$ holds; see Remark A.1.3 from page 152. If the interval X_n contains two distinct zeros $\xi_1 < \xi_2$, then there is a point $\tilde{\xi} \in [\xi_1, \xi_2] \subset X_n$ with $0 = f'(\tilde{\xi}) \in \mathbf{f}'(X_n)$, and ξ_1 and ξ_2 are contained in different connected components of X_{n+1}. In this way,

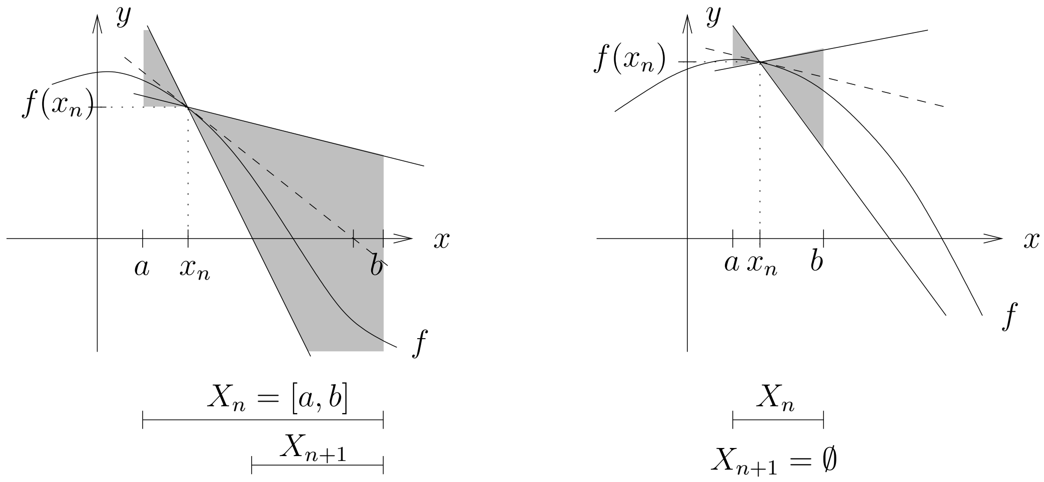


Figure A.4: Illustration of the univariate interval Newton method.

interval Newton methods can separate distinct zeros. If $X_n \cap \mathrm{N}(f; X_n, x_n) = \emptyset$, then the interval X_n does not contain a zero of f.

Example A.5.1. We consider the function $f(x) = x^2 - 4$ and the interval $X_0 = [-4, 4]$. The function f has the zeros $\pm 2 \in X_0$, and we have $f'(x) = 2x$ and $\mathbf{f}'(X_0) = 2[-4, 4] = [-8, 8]$. We choose $x_0 = 0 \in X_0$. By the first interval Newton step, we obtain

$$\begin{aligned} X_1 &= X_0 \cap \mathrm{N}(f; X_0, x_0) = X_0 \cap \left(x_0 - \frac{f(x_0)}{\mathbf{f}'(X_0)} \right) \\ &= [-4, 4] \cap \left(0 - \frac{-4}{[-8, 8]} \right) = [-4, 4] \cap \frac{1}{[-2, 2]} \\ &= [-4, 4] \cap \left(\left[-\infty, -\frac{1}{2} \right] \cup \left[\frac{1}{2}, \infty \right] \right) = \left[-4, -\frac{1}{2} \right] \cup \left[\frac{1}{2}, 4 \right]. \end{aligned}$$

This result implies that all zeros of f in the interval $X_0 = [-4, 4]$ are contained in $\left[-4, -\frac{1}{2}\right] \cup \left[\frac{1}{2}, 4\right]$. More interval Newton steps are used separately on the two intervals $\left[-4, -\frac{1}{2}\right]$ and $\left[\frac{1}{2}, 4\right]$. The intervals obtained by the interval Newton method converge to arbitrary small intervals around the solutions ± 2; see [35, Thm. 1.14, p. 52].

In the multivariate situation, the interval Newton methods have more variety. As in the univariate case, the basepoints $x_n \in X_n$ and the interval extension $\mathbf{F}'$ of the derivative of the function $F\colon \mathbb{R}^m \to \mathbb{R}^m$ can be chosen. But, there are more possibilities to bound the solution set of the resulting linear systems than in the univariate case. Since $\mathbf{F}'(\mathbf{X})$ with $\mathbf{X} \in (\mathrm{I}(\mathbb{R}))^m$ is a vector of intervals, it describes a convex set, and the mean value theorem for multivariate vectorial functions implies that for all x, $y \in \mathbf{X}$ we have

$$F(z) - F(x) = A(z - x) \text{ for an } A \in \mathbf{F}'(\mathbf{X}).$$

Thus we have

$$F(z) - F(x) \in \mathbf{F}'(\mathbf{X})(z - x) = \{A(z - x) |\ A \in \mathbf{F}'(\mathbf{X})\};$$

see [35] or [53, Sect. 3.1] for an introduction to interval arithmetic in matrices. If $z \in \mathbf{X}$ is a zero of F, we have to determine the solution set of the interval system $-F(x) \in \mathbf{F}'(\mathbf{X})(z - x)$.

One possibility is to choose the Krawczyk method [35, 53]. The aim is to approximate all solutions of F in a given interval vector $\mathbf{X}$. Let $x \in \mathbf{X}$, and let $A := F'(x)$ be the Jacobi matrix of F at x. We assume that A is a regular matrix. Let Y be an approximation to A^{-1}. Let

$$P(z) := z - YF(z), \tag{A.2}$$

and consider a mean value extension of P over $\mathbf{X}$:

$$\mathbf{P}(\mathbf{X}) = P(x) + \mathbf{P}'(\mathbf{X})(\mathbf{X} - x). \tag{A.3}$$

Since $P'(z) = I - YF'(x)$, where I is the $m \times m$ identity matrix, we can choose $\mathbf{P}'(\mathbf{X}) = I - Y\mathbf{F}'(\mathbf{X})$, where $\mathbf{F}'(\mathbf{X})$ is an interval extension of F' over $\mathbf{X}$. Combined with equations A.2 and A.3, this leads to

$$\begin{aligned} \mathbf{K}(F; \mathbf{X}, x) &:= \mathbf{P}(\mathbf{X}) = P(x) + \mathbf{P}'(\mathbf{X})(\mathbf{X} - x) \\ &= \underbrace{x - YF(x)}_{=P(x)} + \underbrace{I - Y\mathbf{F}'(\mathbf{X})}_{=\mathbf{P}'(\mathbf{X})}(\mathbf{X} - x). \end{aligned}$$

We show that all roots of F from the interval $\mathbf{X}$ are contained in $\mathbf{K}(F; \mathbf{X}, x)$. Let $\xi \in \mathbf{X}$ be a root of F. This implies

$$\xi = \xi - Y \underbrace{F(\xi)}_{=0} = P(\xi) \in \mathbf{P}(\mathbf{X}) = \mathbf{K}(F; \mathbf{X}, x).$$

In the next iteration step, we consider the set $\mathbf{X} \cap \mathbf{K}(F; \mathbf{X}, x)$. As in the univariate interval Newton method, the set $\mathbf{X} \cap \mathbf{K}(F; \mathbf{X}, x)$ could consist of more than one intervals.

Since we only need the inverse of the Jacobian of F at a properly chosen point $x \in \mathbf{X}$, the Krawczyk method could be used to determine all critical points a GSP Γ in a given box $\mathbf{X}$; see Section 7.4. Unfortunately, there is no straightforward analogue to the interval Newton method for complex interval arithmetic, since here the mean value theorem does not hold in the used form. One possibility to deal with complex systems of equations is to split all variables into their real and imaginary part. This doubles the number of variables and the number of equations, and the system becomes more complicated.

A.6 Affine Arithmetic

We give a brief introduction to affine arithmetic based on de Figueiredo's and Stolfi's presentation in [60, 20]. The advantage over interval arithmetic is that linear dependencies between quantities can be handled, and the computed ranges are usually smaller than in interval arithmetic. However, affine arithmetic has higher computational costs, is more complicated and needs more space in memory. Primarily, affine arithmetic is defined for real quantities ($\mathbb{K} = \mathbb{R}$), in [38] complex affine arithmetic is defined for rectangular ranges by deviding a complex quantity into its real and imaginary part.

In affine arithmetic, a partially unknown quantity x is represented by an affine form $\hat{x}$.

Definition A.6.1. *[60]* Affine Form
An affine form *is a first-degree polynomial*

$$\hat{x} = x_0 + x_1\epsilon_1 + x_2\epsilon_2 + \cdots + x_n\epsilon_n$$

in the variables $\epsilon_1, \epsilon_2, \ldots, \epsilon_n$ *that are called noise symbols. A* noise symbol *is a real variable that is allowed to take values in the closed interval* $[-1, 1]$. *The coefficients* $x_0, x_1, x_2, \ldots, x_n$ *are real (or floating point) numbers. The constant term* x_0 *is called* central value*, the coefficients of the linear part are called* partial deviations*.*

Different noise symbols ϵ_i represent different kinds of errors like measurement errors, rounding errors, or approximation errors. The corresponding coefficients x_i determine the magnitude with which the noise symbols ϵ_i contribute to the original quantity x. The reader should be aware of the fact that the noise symbols ϵ_i are variables and that the coefficients x_i are fixed constants. In many applications, the partial deviations $x_1, \ldots, x_n$ are nonnegative numbers at the beginning of a computation. During the computation, the quantities are combined by arithmetic operations, and the partial deviations of the (intermediate) results can become negative, as well.

Two affine forms $\hat{x}$ and $\hat{y}$ can share noise symbols. This indicates a partial dependency between the underlying quantities x and y.

Example A.6.2. We are given two quantities x and y that are represented by the affine forms $\hat{x}$ and $\hat{y}$.

$$\begin{aligned} \hat{x} &= 5 + 2\,\epsilon_1 - \epsilon_2 \\ \hat{y} &= 4 - \epsilon_1 + 3\,\epsilon_2 - \epsilon_3 \end{aligned}$$

A first observation is that $5 - 2 - 1 = 2$ is the smallest value x can take and $5 + 2 + 1 = 8$ is the largest value corresponding to the assignments $\epsilon_1 = -1$,

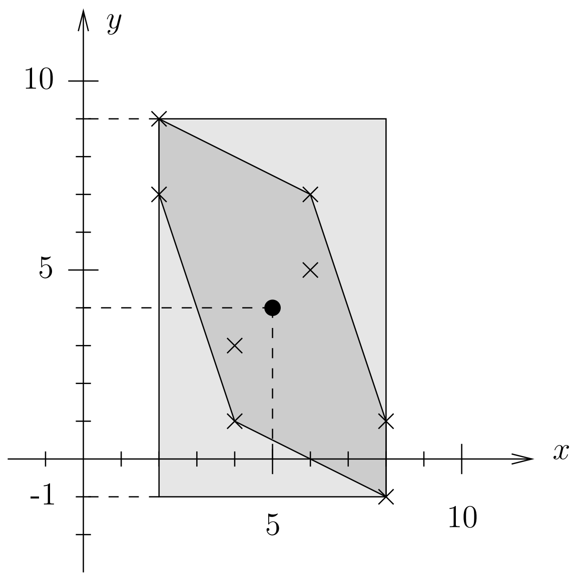


Figure A.5: The rectangle $[2,8] \times [-1,9]$ and the image $A([-1,1]^3)$ are shown for the affine forms $\hat{x} = 5 + 2\epsilon_1 - \epsilon_2$ and $\hat{y} = 4 - \epsilon_1 + 3\epsilon_2 - \epsilon_3$. The crosses mark the images of the vertices of the cube $[-1,1]^3$ under the affine mapping A, the circle marks the point $(x_0, y_0) = (5,4)$.

$\epsilon_2 = +1$, and $\epsilon_1 = +1$, $\epsilon_2 = -1$. Thus we know $x \in [2,8]$. The same argument shows $y \in [4-1-3-1, 4+1+3+1] = [-1,9]$. Hence the pair (x,y) is contained in the rectangle $[2,8] \times [-1,9]$; see Figure A.5. This information corresponds to an independent treatment of the quantities x and y as it would be given by interval arithmetic.

The affine forms $\hat{x}$ and $\hat{y}$ provide even more information about x and y. Recall that every noise symbol ϵ_i stands for a certain uncertainty. To get the correct values for x and y, there is **exactly one** assignment of the variables ϵ_1, ϵ_2, and ϵ_3 with values from the interval $[-1,1]$. This assignment gives the correct values for x and y, simultaneously. This observation implies that the pair (x,y) lies in the image $A([-1,1]^3)$ of the cube $[-1,1]^3$ under the affine mapping

$$\begin{array}{llll} A\colon & \mathbb{R}^3 & \to & \mathbb{R}^2 \\ & (\epsilon_1,\epsilon_2,\epsilon_3) & \mapsto & \begin{pmatrix}5\\4\end{pmatrix} + \epsilon_1\begin{pmatrix}2\\-1\end{pmatrix} + \epsilon_2\begin{pmatrix}-1\\3\end{pmatrix} + \epsilon_3\begin{pmatrix}0\\-1\end{pmatrix}. \end{array}$$

The image $A([-1,1]^3)$ is the convex hull of the images of the $2^3 = 8$ vertices of the cube $[-1,1]^3$; see Figure A.5. Thus, $A([-1,1]^3)$ is a polygon that is symmetric to the point $\binom{x_0}{y_0} = \binom{5}{4}$, and $A([-1,1]^3) \subset [2,8] \times [-1,9]$.

In arbitrary dimensions, k affine forms having n noise symbols define an affine mapping $A\colon \mathbb{R}^n \to \mathbb{R}^k$ and the image $A([-1,1]^n)$ is a k-dimensional convex polytope that restricts the values of the k underlying quantities.

A.6.1 Affine Forms versus Intervals

In Example A.6.2, we have seen how to convert an affine form to the interval that contains the range of the affine form. If $\hat{x} = x_0 + x_1\epsilon_1 + \cdots + x_n\epsilon_n$ is an affine form that belongs to a quantity x, then $X := [x_0 - \sum_{i=1}^n |x_i|, x_0 + \sum_{i=1}^n |x_i|]$ is the corresponding interval; see [11, 60, 20]. The sum $\mathrm{rad}(\hat{x}) := \sum_{i=1}^n |x_i|$ of the moduli of the partial deviations is the *total deviation* of $\hat{x}$; see [20]. Note that X is the smallest interval that contains all possible values of $\hat{x}$ since the noise symbols $\epsilon_1, \ldots, \epsilon_n$ take values in $[-1,1]$.

If we are given an interval $X = [a,b]$ describing a quantity x, then we can translate X to an equivalent affine form $\hat{x}$; see [11, 60, 20]. The central value x_0 is the midpoint of X, and we introduce a *new* noise symbol ϵ_k that represents the uncertainty of the value of x given by the fact $x \in X = [a,b]$. Thus the partial deviation x_k is half of the width of the given interval X, and we have

$$\hat{x} = x_0 + x_k\epsilon_k = \frac{a+b}{2} + \frac{b-a}{2}\,\epsilon_k.$$

The noise symbol ϵ_k must be distinct from all other noise symbols that have been used so far. This choice is important since the only information about the quantity x is $x \in X$. There is nothing known about dependencies between x and other quantities in the computation.

The correspondence between affine forms and intervals permits to replace interval arithmetic by affine arithmetic in many applications. Note that the properties of affine arithmetic and of interval arithmetic differ; compare with Section A.6.3.

A.6.2 Computing with Affine Arithmetic

In affine arithmetic, the operations are divided into affine and non-affine operations. Affine operations like addition and subtraction can be done exactly, if we assume exact computation. Non-affine operations are approximated by affine functions. An upper bound of the resulting approximation error is the partial deviation of a newly introduced noise symbol; see [11, 60, 20].

Affine Operations Let $\hat{x} = x_0 + x_1\epsilon_1 + \cdots + x_n\epsilon_n$ and $\hat{y} = y_0 + y_1\epsilon_1 + \cdots + y_n\epsilon_n$ be affine forms describing the two quantities x and y. The considerations can be generalized to an arbitrary number of operands in a straightforward way.

Let $\alpha,\ \beta$, and $\gamma \in \mathbb{R}$ be real numbers. Then the affine combination $\hat{z} = \alpha\hat{x}+\beta\hat{y}+\gamma$ defines the affine form

$$\hat{z} = \alpha\hat{x} + \beta\hat{y} + \gamma = \underbrace{\alpha x_0 + \beta y_0 + \gamma}_{z_0} + \underbrace{(\alpha x_1 + \beta y_1)}_{z_1}\epsilon_1 + \cdots + \underbrace{(\alpha x_n + \beta y_n)}_{z_n}\epsilon_n.$$

This formula assumes exact computation. For dealing with rounding errors, we introduce a new noise symbol $\epsilon_k \notin \{\epsilon_1, \ldots, \epsilon_n\}$ that models the rounding error of the single operation $\hat{z} = \alpha\hat{x} + \beta\hat{y} + \gamma$. The partial deviation z_k must be an upper bound of the total rounding error; see [60, Sect. 3.6]. Note that the noise symbol ϵ_k models the rounding error of this particular affine operation, only. For other affine operations, we have to introduce other new noise symbols.

Non-Affine Operations Again, let $\hat{x} = x_0 + x_1\epsilon_1 + \cdots + x_n\epsilon_n$ and $\hat{y} = y_0 + y_1\epsilon_1 + \cdots + y_n\epsilon_n$ be affine forms describing the two quantities x and y. The ideas easily generalize to an arbitrary number of operands. We consider the operation $z \leftarrow f(x, y)$, where f is a non-affine function. Plugging in the formulas for $\hat{x}$ and $\hat{y}$ leads to

$$\begin{aligned} z &= f(x, y) \\ &= f(x_0 + x_1\epsilon_1 + \cdots + x_n\epsilon_n,\ y_0 + y_1\epsilon_1 + \cdots + y_n\epsilon_n) \\ &=: f^*(\epsilon_1, \ldots, \epsilon_n), \end{aligned}$$

where f^* is a real valued function defined on the n-dimensional cube $[-1, 1]^n$. The task is to approximate f^* by an affine function f^a with an error bound δ. For simplicity and efficiency, we choose f^a as an affine combination of the affine forms $\hat{x}$ and $\hat{y}$, in other words we choose $f^a(\epsilon_1, \ldots, \epsilon_n) = \alpha\hat{x} + \beta\hat{y} + \gamma$ with $\alpha,\ \beta,\ \gamma \in \mathbb{R}$; see [60, Sect. 3.7]. Thus we get

$$\hat{z} = f^a(\epsilon_1, \ldots, \epsilon_n) + \delta\epsilon_k,$$

where ϵ_k is a new noise symbol. The partial deviation δ must be an upper bound of the approximation error, i.e.

$$\delta \geq \max_{\epsilon_i \in [-1,1]} |f^*(\epsilon_1, \ldots, \epsilon_n) - f^a(\epsilon_1, \ldots, \epsilon_n)|.$$

Possible rounding errors can be included in the term $\delta\epsilon_k$. Then the noise symbol ϵ_k represents both, the approximation error and the rounding error of the operation $z \leftarrow f(x, y)$. Additionally, we have to add an upper bound of the total rounding error to the partial deviation δ; see [60, Sect. 3.7].

We discuss the operations $z \leftarrow x \cdot y$, $z \leftarrow \frac{1}{x}$, and $z \leftarrow \sqrt{z}$ separately.

Multiplication. We consider the non-affine operation $z \leftarrow f(x,y) = x \cdot y$. The quantities x and y are given by the affine forms $\hat{x} = x_0 + x_1\epsilon_1 + \cdots + x_n\epsilon_n$ and $\hat{y} = y_0 + y_1\epsilon_1 + \cdots + y_n\epsilon_n$. Plugging in the formulas for $\hat{x}$ and $\hat{y}$ leads to

$$\begin{aligned} z &= x \cdot y \\ &= \left(x_0 + \sum_{i=1}^{n} x_i\epsilon_i\right) \cdot \left(y_0 + \sum_{i=1}^{n} y_i\epsilon_i\right) \\ &= \underbrace{x_0y_0 + \sum_{i=0}^{n}(x_0y_i + y_0x_i)\epsilon_i}_{\text{affine term}} + \underbrace{\left(\sum_{i=1}^{n} x_i\epsilon_i\right) \cdot \left(\sum_{i=1}^{n} y_i\epsilon_i\right)}_{\text{quadratic term}}. \end{aligned}$$

Thus we need an affine approximation for the quadratic term $q(\epsilon_1, \ldots, \epsilon_n) = (\sum_{i=1}^n x_i\epsilon_i) \cdot (\sum_{i=1}^n y_i\epsilon_i) = \sum_{i=1}^n \sum_{j=1}^n x_iy_j\epsilon_i\epsilon_j$. Since we have $q(-\epsilon_1, \ldots, -\epsilon_n) = q(\epsilon_1, \ldots, \epsilon_n)$, and $(-\epsilon_1, \ldots, -\epsilon_n) \in [-1,1]^n$ iff $(\epsilon_1, \ldots, \epsilon_n) \in [-1,1]^n$, we observe that the best affine approximation to q is constant. A simple but not best solution for the multiplication is

$$\hat{z} = \hat{x}\hat{y} = \underbrace{x_0y_0}_{z_0} + \sum_{i=0}^{n}(\underbrace{x_0y_i + y_0x_i}_{z_i})\epsilon_i + \delta\epsilon_k,$$

where ϵ_k is a new noise symbol and δ is chosen as

$$\delta := \left(\sum_{i=1}^{n} |x_i|\right) \cdot \left(\sum_{i=1}^{n} |y_i|\right) \geq \left|\left(\sum_{i=1}^{n} x_i\epsilon_i\right) \cdot \left(\sum_{i=1}^{n} y_i\epsilon_i\right)\right| \quad \forall\epsilon_i \in [-1,1]$$

Here, we have chosen $q(\epsilon_1, \ldots, \epsilon_n) = 0$ as affine approximation of the quadratic term. A more detailed description including the treatment of rounding errors can be found in [60, Sect. 3.13].

Example A.6.3. We consider the polynomial $f(x) = x^2 - x$ from Example A.1.4 from page 153 over the interval $[0,1]$. We assign to the interval $[0,1]$ the affine form $\hat{x} = \frac{1}{2} + \frac{1}{2}\epsilon_1$ as described in Section A.6.1. In affine arithmetic, plugging $\hat{x}$ into f leads to

$$\begin{aligned} \hat{x} \cdot \hat{x} - \hat{x} &= \left(\frac{1}{2} + \frac{1}{2}\epsilon_1\right)^2 - \left(\frac{1}{2} + \frac{1}{2}\epsilon_1\right) \\ &= \left(\frac{1}{4} + \frac{1}{2}\epsilon_1 + \frac{1}{4}\epsilon_2\right) - \left(\frac{1}{2} + \frac{1}{2}\epsilon_1\right) \\ &= -\frac{1}{4} + \frac{1}{4}\epsilon_2 \\ &\in \left[-\frac{1}{2}, 0\right]. \end{aligned}$$

We observe that the noise symbol ϵ_1 cancels out. This fact shows how affine arithmetic deals with dependencies.

As in Example A.1.4, we rewrite $f(x) = x^2 - x = x(x-1)$ and evaluate the new expression with affine arithmetic.

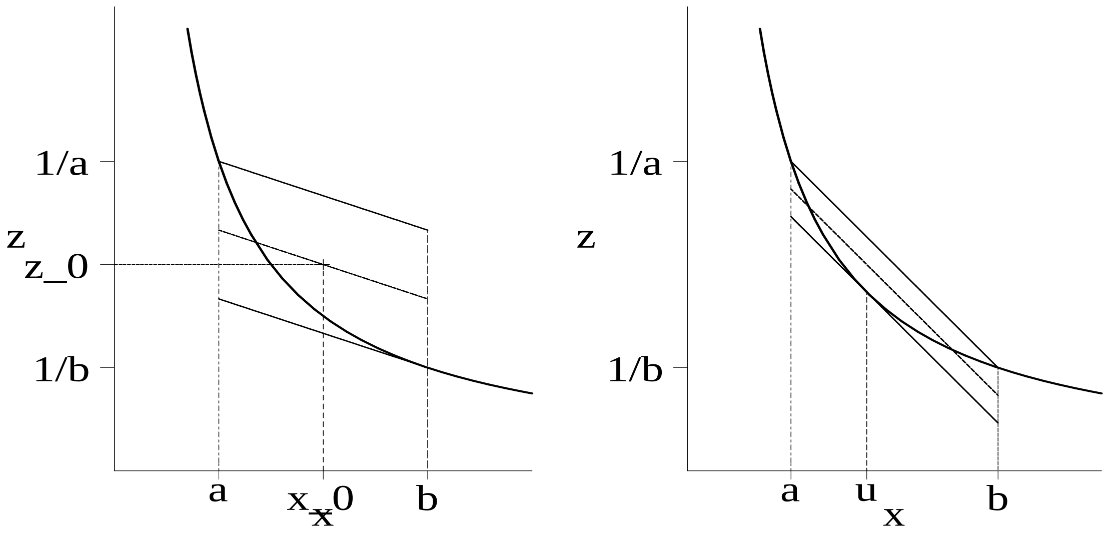
$$\begin{aligned}
\hat{x}(\hat{x}-1) &= \left(\frac{1}{2}+\frac{1}{2}\epsilon_1\right)\cdot\left(\frac{1}{2}+\frac{1}{2}\epsilon_1 - 1\right) \\
&= \left(\frac{1}{2}+\frac{1}{2}\epsilon_1\right)\cdot\left(-\frac{1}{2}+\frac{1}{2}\epsilon_1\right) \\
&= -\frac{1}{4}+\frac{1}{4}\epsilon_3 \\
&\in \left[-\frac{1}{2}, 0\right].
\end{aligned}$$

In both cases, the resulting range is much smaller than the ranges obtained with interval arithmetic. Recall that the exact range is $[-\frac{1}{4}, 0]$. At first view, both expressions lead to the same affine form, but the meaning of the noise symbols ϵ_2 and ϵ_3 differs: The noise symbol ϵ_2 models the approximation error of the quadratic term of the first multiplication, whereas the noise symbol ϵ_3 models the approximation error of the multiplication in the second case.

Due to the approximation of the quadratic terms, the associative law does not hold for multiplication. By the same reason, the distributive law does not hold, either.

Division. Divisions can be reduced to the multiplication with the reciprocal. The reciprocal defines a univariate non-affine function. In [60, Sect 3.12] two possibilities for defining the reciprocal in affine arithmetic are defined: The min-range and the Chebyshev approximation. We only give a very brief overview. Following the approach for non-affine operations, we have to determine an affine approximation $f^a(\epsilon_1, \ldots, \epsilon_n) = \alpha\hat{x} + \gamma$ to the function $f^*(\epsilon_1, \ldots, \epsilon_n) = f(x) = \frac{1}{x}$. Since the range corresponding to $\hat{x}$ is the interval $X = [a, b] = [x_0 - \sum_{i=1}^n |x_i|, x_0 + \sum_{i=1}^n |x_i|\,]$, we have to approximate $f(x) = \frac{1}{x}$ by a line over the interval X. In Figure A.6, we indicate how the function $f(x) = \frac{1}{x}$ could be approximated by an affine function.

From the approximations from Figure A.6 we can derive an affine form $\hat{z}$ of $z = \frac{1}{x}$ when x is given by the affine form $\hat{x} = x_0 + x_1\epsilon_1 + \cdots + x_n\epsilon_n$. Let $g(x) = mx + c$ be the equation defining a line that approximates $f(x) = \frac{1}{x}$. This is the middle line in Figures A.6(a) and A.6(b). The approximation error δ is half of the length of the vertical edges of the parallelogram. Then $\hat{z} = z_0 + z_1\epsilon_1 + \cdots + z_n\epsilon_n + z_k\epsilon_k$



(a) Min-range approximation (b) Chebyshew approximation

Figure A.6: The min-range and the Chebyshew approximation of the function $f(x) = \frac{1}{x}$ over the interval $[a, b]$ are shown. For the min-range approximation, the central value x_0 of the affine form $\hat{x}$ and the central value z_0 of the resulting affine form $\hat{z}$ are marked.

is defined by

$$
\begin{aligned}
z_0 &= mx_0 + c; \\
z_i &= mx_i \qquad (i = 1, \ldots, n); \\
z_k &= \delta,
\end{aligned}
$$

where ϵ_k is a new noise symbol. Note that the parallelograms of Figure A.6 describe all possible positions for the pair (x, z); see also Example A.6.2.

We observe that the area of the parallelogram from Figure A.6(a) is larger than the area of the parallelogram from Figure A.6(b). On the other hand, the corresponding range for $\hat{z}$ in the min-range approximation is the exact range of the function $f(x)$ over the interval X. In the Chebyshew approximation, the range of $\hat{z}$ is larger. Although the function $f(x) = \frac{1}{x}$ is strictly positive for $x > 0$, the range of $\hat{z}$ could contain zero or even negative numbers. Thus, it depends on the application whether the min-range approximation or the Chebyshew approximation is the better choice.

Square Root Operation The square root operation can be treated like the reciprocal. The interested reader is referred to [60, Sect. 3.9].

A.6.3 Problem: Inclusion Monotonicity

The construction of the operations in affine arithmetic ensures that if an input quantity x lies in the range described by an affine form $\hat{x}$, then the output z lies in the range of the computed affine form $\hat{z}$. This property is called the fundamental invariant of range analysis [60, Sect.1.2.2] and is equivalent to Corollary A.1.7 from page 154. However, it seems to be difficult to translate the inclusion monotonicity property stated in Theorem A.1.6 to affine arithmetic.

Example A.6.4. We consider the affine forms $\hat{x} = \hat{y} = \hat{a} = 2 + \epsilon_1 + \epsilon_2$ and $\hat{b} = 2 - \epsilon_1 - \epsilon_2$ with the corresponding intervals $X = Y = A = B = [0, 4]$. Thus we have $X \subset A$ and $Y \subset B$. Let $\hat{z} := \hat{x} + \hat{y} = 4 + 2\epsilon_1 + 2\epsilon_2$ and $\hat{c} := \hat{a} + \hat{b} = 4$. We observe $Z = [0, 8] \not\subset [4, 4] = C$ for the corresponding intervals Z and C of the affine forms $\hat{z}$ and $\hat{c}$, and the inclusion monotonicity property does not hold for the corresponding ranges. A closer look to the given affine forms shows that they are dependent: They all share the noise symbols ϵ_1 and ϵ_2. Additionally, $\hat{x}$, $\hat{y}$ and $\hat{a}$ have the same partial deviations. Hence, considering dependencies as in affine arithmetic contradicts the inclusion monotonicity property for the ranges. These observations indicate the difficulties for formulating a suitable inclusion monotonicity property for affine arithmetic.

Bibliography

[1] G. Alefeld and J. Herzberger. *Einführung in die Intervallrechnung*, volume 12 of *Reihe Informatik*. Bibliographisches Institut, 1974. an English translation appeared in 1983, Academic Press New York.

[2] E. L. Allgower and K. Georg. *Numerical Continuation Methods*. Springer-Verlag, 1990.

[3] S. Basu, R. Pollack, and M.-F. Roy. On the combinatorial and algebraic complexity of quantifier elimination. *Journal of the ACM*, 43(6):1002–1045, 1996.

[4] S. Basu, R. Pollack, and M.-F. Roy. *Algorithms in Real Algebraic Geometry*. Springer-Verlag, 2003.

[5] M. d. Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf. *Computational Geometry*. Springer-Verlag, 2000.

[6] L. Bieberbach. *Theorie der Geometrischen Konstruktionen*. Birkhäuser-Verlag, 1952.

[7] L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and Real Computation*. Springer-Verlag, 1998.

[8] G. E. Bredon. *Topology and Geometry*, volume 139 of *Graduate Texts in Mathematics*. Springer-Verlag, 2nd edition, 1993.

[9] P. Bürgisser, M. Clausen, and M. A. Shokrollahi. *Algebraic Complexity Theory*. Springer-Verlag, 1997.

[10] C. Burnikel, S. Funke, K. Mehlhorn, S. Schirra, and S. Schmitt. A separation bound for real algebraic expressions. In *ESA 2001*, volume 2161 of *LNCS*, pages 254–265, 2001.

[11] J. Comba and J. Stolfi. Affine arithmetic and its applications to computer graphics. In *Proceedings of VII SIBGRAPI (Brazilian Symposium on Computer Graphics and Image Processing)*, pages 9–18, 1993.

[12] D. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms*. Springer-Verlag, 2nd edition, 1997.

[13] B. Denner-Broser. On the decidability of tracing problems in dynamic geometry. In *proc ADG 2004*, volume 3763 of *LNAI*, pages 111–129. Springer-Verlag, 2006.

[14] B. Denner-Broser. An algorithm for the tracing problem using interval analysis. In *SAC 2008; session Geometric constraints and reasoning*, pages 1832–1837, 2008.

[15] P. Deuflhard. *Newton Methods for Nonlinear Problems*, volume 35 of *Springer Series in Computational Mathematics*. Springer-Verlag, 2004.

[16] A. Dolzmann, T. Sturm, and V. Weispfenning. A new approach for automated theorem proving in real geometry. *Journal of Automated Reasoning*, 12:357–380, 1998.

[17] W. Ebeling. *Funktionentheorie, Differentialtopologie und Singularitäten*. Verlag Vieweg, Braunschweig/Wiesbaden, 2001.

[18] A. Fabre and P. Schreck. Combining symbolic and numerical solvers to simplify indecomposable systems solving. In *SAC 2008; session Geometric constraints and reasoning*, pages 1838–1842, 2008.

[19] D. Faudot and D. Michelucci. A new robust algorithm to trace curves. *Reliable Computing*, 13:309–324, 2007.

[20] L. H. d. Figueiredo and J. Stolfi. Affine arithmetic: Concepts and applications. *Numerical Algorithms*, 37:147–158, 2004.

[21] W. Fischer and I. Lieb. *Ausgewählte Kapitel aus der Funktionentheorie*. Verlag Vieweg, 1988.

[22] O. Forster. *Riemannsche Flächen*. Springer-Verlag Berlin Heidelberg, 1977.

[23] W. Gellert, H. Kästner, and S. Neuber, editors. *Lexikon der Mathematik*. Bibliographisches Institut Leipzig, 1981.

[24] M. J. González-Lopéz. Using dynamic geometry software to simulate physical motion. *Interbational Journal of Computers for Mathematical Learning*, 6:127–142, 2001.

[25] M. Grimmer, K. Petras, and N. Revol. Multiple precision interval packages: Comparing different approaches. In *Numerical Software with Result Verification*, volume 2991 of *LNCS*, pages 64–91. Springer-Verlag, 2004.

[26] J. A. Grunert, editor. *Archiv der Mathematik und Physik*. Dreissigster Theil. C. A. Koch's Verlagsbuchhandlung, Th. Kunike, 1858. available at http://books.google.de.

[27] G. Hämmerlin and K. Hoffmann. *Numerische Mathematik*. Springer-Verlag, 1994.

[28] E. Hansen. A multidimensional interval newton method. *Reliable Computing*, 12:253–272, 2006.

[29] M. Hauenschild. Arithmetiken für komplexe Kreise. *Computing*, 13:299–312, 1974.

[30] A. Heck. *Introduction to Maple*. Springer-Verlag, 2003.

[31] J. Heintz, M.-F. Roy, and P. Solernó. Description of the connected components of a semialgebraic set. *Discrete Computational Geometry*, 11:121–140, 1994.

[32] K. Hensel and G. Landsberg. *Theorie der Algebraischen Funktionen einer Variablen*. B. G. Teubner, Leipzig, 1902.

[33] N. J. Higham. *Handbook of Writing for the Mathematical Sciences*. siam, 1993.

[34] S. Iyanaga and Y. Kawada, editors. *Encyclopedic Dictionary of Mathematics*. MIT Press, 1977. English translation.

[35] R. Kearfott. *Rigorous Global Search: Continuous Problems*. Kluwer Academic Publishers Dordrecht, 1996.

[36] R. B. Kearfott and Z. Xing. An interval step control for continuation methods. *SIAM Journal on Numerical Analysis*, 3(3):892–914, 1994.

[37] O. Kerner, J. Maurer, J. Steffens, T. Thode, and R. Voller, editors. *Vieweg Mathematik Lexikon*. Verlag Vieweg, 1995. 3. edition; a useful and small encyclopedia.

[38] K. Ko, T. Sakkalis, and N. Patrikalakis. A reliable algorithm for computing the topological degree of a mapping in $\mathbb{R}^2$. *Applied Mathematics and Computation*, 2007.

[39] H. Kopka. *Latex, Band 1: Einführung*. Pearson Studium, 2000.

[40] U. Kortenkamp. *Foundations of Dynamic Geometry*. PhD thesis, ETH Zürich, 1999.

[41] U. Kortenkamp and J. Richter-Gebert. The interactive geometry software Cinderella.2. CD-ROM, to appear at Springer in November 2008, URL: http://www.cinderella.de.

[42] U. Kortenkamp and J. Richter-Gebert. *The Interactive Geometry Software Cinderella*. Springer-Verlag, 1999. URL: http://www.cinderella.de.

[43] U. Kortenkamp and J. Richter-Gebert. Complexity issues in dynamic geometry. In *Proceedings of the Smale Fest 2000*, 2001.

[44] U. Kortenkamp and J. Richter-Gebert. Decision complexity in dynamic geometry. In *Automated Deduction in Geometry, Workshop 2000*, volume 2061, pages 193–199, 2001.

[45] U. Kortenkamp and J. Richter-Gebert. Using automatic theorem proving to improve the usability of geometry software. In P. Libbrecht, editor, *Proceedings of MathUI 2004*, 2004.

[46] W. Krämer and I. Geulig. *Interval Calculus in Maple (The Extension intpakX to the Package intpak of the Share-Library)*. University of Wuppertal, Germany, 2001. Available from http://www.math.uni-wuppertal.de/wrswt/literatur.html.

[47] K. Lamotke. *Riemannsche Flächen*. Springer-Verlag, 2005.

[48] H. Lamure and D. Michelucci. Solving geometric constraints by homotopy. *IEEE Transactions on Visualization and Computer Graphics*, 2(1):28–34, 1996.

[49] F. Levi. *Geometrische Konfigurationen*. Hirzel, Leipzig, 1929.

[50] R. E. Moore. *Interval Analysis*. Series in Automatic Computation. Prentice-Hall, Inc., 1966.

[51] R. Narasimhan. *Compact Riemann Surfaces*. Lectures in Mathematics. Birkhäuser-Verlag, Basel, 1992.

[52] N. Nedialkov, V. Kreinovich, and S. Starks. Interval arithmetic, affine arithmetic, taylor series methods: Why, what next? *Numerical Algorithms*, 37(1–4):325–336, 2004.

[53] A. Neumaier. *Interval Methods for Systems of Equations*. Cambridge University Press, 1990.

[54] A. Neumaier. Taylor forms – use and limits. *Reliable Computing*, 9:43–79, 2003.

[55] K. Osgood. *Lehrbuch der Funktionentheorie*. B. G. Teubner, 1912.

[56] L. Petković and M. Petković. On the k-th root in circular arithmetic. *Computing*, 33:27–35, 1984.

[57] M. Petković and L. Petković. *Complex Interval Arithmetic and Its Applications*. WILEY-VCH Berlin, 1998.

[58] F. Preparata and M. Shamos. *Computational Geometry*. Springer-Verlag, 1985.

[59] A. J. Sommese, J. Verschelde, and W. C. W. Introduction to numerical algebraic geometry. In *Solving Polynomial Equations*, volume 14 of *Algorithms and Computation in Mathematics*. Springer-Verlag, 2005.

[60] J. Stolfi and L. H. d. Figueiredo. Self-validated numerical methods and applications, 1997. available at http://www.ic.unicamp.br/˜stolfi/EXPORT/bibliography/stolfi.html.

[61] I. Vinogradov, editor. *Encyclopaedia of Mathematics*. Kluwer, 1988. English translation.

[62] D. Wang. Geometry machines: From ai to smc. In *Artificial Intelligence and Symbolic Mathematical Computation*, volume 1138 of *LNCS*, pages 213–239. Springer-Verlag, 1996.

[63] D. Werner. *Einführung in die Höhere Analysis*. Springer-Verlag, 2006.

[64] C. Yap and T. Dubé. The exact computation paradigm. In D.-Z. Du and F. Hwang, editors, *Computing in Euclidean Geometry*, volume 4 of *Lecture Notes Series on Computing*. World Scientific Press, 2nd edition edition, 1995.

[65] C.-K. Yap. Towards exact geometric computation. *Computational Geometry: Theory and Applications*, 7(1-2):3–23, 1997.

[66] C.-K. Yap. *Fundamental Problems of Algorithmic Algebra*. Oxford University Press, 2000.

Index

Curriculum Vitae

Der Lebenslauf ist in der Online-Version
aus Gründen des Datenschutzes nicht enthalten