

About Tracing Problems in Dynamic Geometry

Britta Denner-Broser

Received: 21 November 2011 / Revised: 3 October 2012 / Accepted: 11 October 2012 /
Published online: 27 November 2012
© Springer Science+Business Media New York 2012

Abstract Dynamic Geometry is the field of interactively performing geometric construction on a computer. In addition to simulating ruler-and-compass constructions we allow a *drag mode*. This drag mode allows to move geometric objects that have at least one degree of freedom. The remaining part of the construction should adjust automatically. Thus, during the motion, we have to trace the resulting paths of all geometric objects. This path tracking problem is known as the Tracing Problem from Dynamic Geometry. It combines the step-by-step procedure of doing geometric constructions with the continuous concept of motions. This study is based on the model for Dynamic Geometry used in the interactive geometry software *Cinderella*. We give a numerical solution to the Tracing Problem based on continuation methods and a reliable algorithm based on real and complex interval arithmetic. Degenerate situations like the intersection of two identical lines lead to critical points in the configuration space and are treated separately.

Keywords Dynamic Geometry · Tracing Problem · Geometric Straight-Line Program · Critical point · Continuation methods · Interval arithmetic

Mathematics Subject Classification (2000) 51M15 · 65D17 · 65G20 · 68U07

1 Introduction

Dynamic Geometry is the field of interactively performing geometric construction on a computer. It combines the discrete step-by-step procedure of doing geometric constructions with the continuous concept of motions. Usually, the classical

B. Denner-Broser (✉)
Institut für Informatik, Freie Universität Berlin, Takustr. 9, 14195 Berlin, Germany
e-mail: broser@inf.fu-berlin.de

ruler-and-compass constructions are considered. But also conic sections or higher algebraic curves may be admitted. 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 geometric objects involved during the motion.

The field of Dynamic Geometry has been founded in the late 1980s and 1990s. 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 DG have been designed. They all have the functionality to do geometric constructions and provide a drag mode; additionally, they have different features. However, their drag modes are implemented in different ways and have different properties that arise from the underlying mathematical models. In this study, we focus on the model by Kortenkamp and Richter-Gebert [15,22] that is the foundation of the geometry software *Cinderella* [23] and that leads to a continuous behavior of the drag mode.

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 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. We expect that there are no “jumps” in the motions of the geometric objects and prefer a continuous behavior of the Dynamic Geometry System.

In fact, to implement 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. In addition, 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. The Tracing Problem is to trace this implicitly given motion.

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 Fig. 1. 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. GSPs are closely related to Straight-Line Programs.

In some situations, considering *complex coordinates* of the objects turned out to be useful. Depending on the motion p of the free variables, tracing along p might force us to consider degenerate situations; see Fig. 1: when the free point P crosses the y -axis- l_y , then the dependent point 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 $S = p(t_0)$ at time $t = t_0$, we might bypass it

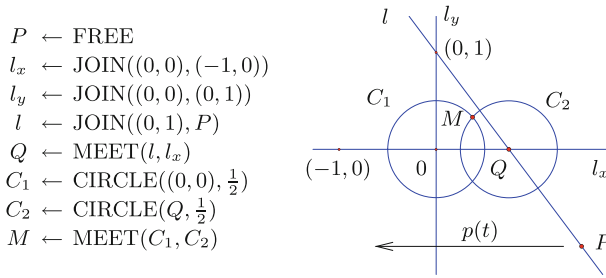


Fig. 1 A GSP Γ and an instance of Γ , i.e., a concrete drawing of the geometric construction, are shown. The point Q has to stay on the line l_x . A singularity occurs if P moves on the path $p(t)$ across the y axis l_y

with a detour by modifying p in a neighborhood of t_0 . Unfortunately, this might be impossible as our example shows; see Fig. 1: by construction, the dependent point Q is 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 to solve this problem is to consider detours for which the coordinates of the free points may have non-real coordinates. Now the line l_x of our example becomes a 2D object, and the point Q might bypass S without leaving l_x . In addition, 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). The idea of detouring around critical points as introduced by Kortenkamp and Richter-Gebert [15, 17] leads to a *consistent* treatment of degeneracies. A detailed discussion of this fact is given in [9, Sects. 2.6 and 7.2].

The geometric situation translates directly into an algebraic model where the objects are numbers (real or complex) and the operations are addition, subtraction, multiplication, division, and taking square roots [9, 15]. We discuss algorithms for the Tracing Problem in this algebraic model. We assume that each arithmetic operation and each interval operation is executed exactly and in constant time.

The basic notions from Dynamic Geometry that are used in our model are explained in Sect. 2 including the new concepts of the derivative of a GSP and of using intervals in GSPs [8, 9]. The interpretation of continuous evaluations as implicitly defined curves in Sect. 3 enables the use of numerical continuation methods [9, Chap. 5]. We adapt these methods to the discrete structure of GSPs to obtain 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. The aim of Sect. 4 is to give an algorithm for the Tracing Problem in the algebraic context. We have to deal with two problems occurring in this setup:

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

The idea is to avoid these problems in advance using interval arithmetic [8, 9]. Our algorithm proceeds stepwise; in each step, the chosen step length guarantees that Problems 1 and 2 can be handled. It turns out that the algorithm is robust [9, Sect. 6.5]. The treatment of critical points is discussed in Sect. 5; see [9, Chap. 7].

2 Some Basic Concepts from Dynamic Geometry

We explain basic notions from Dynamic Geometry introduced by Kortenkamp and Richter-Gebert [15,22] such as Geometric Straight-Line Programs, critical points, continuous evaluations, and the Tracing Problem. The new concept of the derivative GSP $\dot{\Gamma}$ of a GSP Γ with respect to time is introduced [8,9]. We show how we can use intervals in GSPs [8,9]. 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.

Geometric Straight-Line Programs Geometric constructions can be described by Geometric Straight-Line Programs (GSP); see [9,22,16]. We consider an algebraic situation: the objects are real or complex numbers with the operations addition, subtraction, multiplication, division, and square root. In this context, GSPs describe algebraic expressions like $\sqrt{z^2 - 1}$ instead of geometric constructions.

Definition 1 [9, Def. 3.1.1] *GSP 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

$$\begin{aligned} \gamma_j: v_j &\leftarrow \text{FREE} \\ \gamma_j: v_j &\leftarrow v_a + v_b && (+) \\ \gamma_j: v_j &\leftarrow v_a - v_b && (-) \\ \gamma_j: v_j &\leftarrow v_a \cdot v_b && (\cdot) \\ \gamma_j: v_j &\leftarrow v_a/v_b && (/) \\ \gamma_j: v_j &\leftarrow \sqrt{v_a} && (\sqrt{\quad}) \end{aligned}$$

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 $v_j \leftarrow \text{FREE}$ 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 $\sqrt{\quad}$ 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 $\sqrt{\quad}$ 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 $\sqrt{\quad}$ -operation. For $\mathbb{K} = \mathbb{R}$, only positive numbers are valid inputs for the $\sqrt{\quad}$ -operation. The divisor of a division operation must not be zero.

Definition 1 is a practical description of the notion of GSPs, a formal and detailed definition is given by Kortenkamp and Richter-Gebert and can be found in [15,22]. The notion of GSPs is derived from the notion of Straight-Line Programs described by Bürgisser et al. in [4, Sect. 4]. The main difference is that the concept of GSPs allows non-deterministic operations like the square root operation. Straight-Line Programs are not designed for dealing with non-determinism. 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. Note that every dependent variable of a GSP Γ describes an algebraic function in the free variables z_{-k+1}, \dots, z_0 of Γ [9, Sect. 3.1].

Definition 2 [9, 15, 22] *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 $A = (a_{-k+1}, \dots, a_0, a_1, \dots, a_n)$ with $a_{-k+1} = \tilde{a}_{-k+1}, \dots, a_0 = \tilde{a}_0$ is called the *fiber* of the point $(\tilde{a}_{-k+1}, \dots, \tilde{a}_0)$. The instance A lies above the position $(a_{-k+1}, \dots, a_0) = (\tilde{a}_{-k+1}, \dots, \tilde{a}_0)$ of the free variables.

In an instance, the “signs” for the square root operations are fixed in contrast to the underlying GSP Γ . Definitions 1 and 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 1 The expression $\sqrt{z^2 - 1}$ can be described by the GSP Γ :

$$\begin{aligned} \Gamma: \quad & z \leftarrow \text{FREE} \\ & v_1 \leftarrow z \cdot z \\ & v_2 \leftarrow v_1 - 1 \\ & v_3 \leftarrow \sqrt{v_2}; \quad \text{described by } v_3^2 = v_2 \text{ and } v_2 \geq 0. \end{aligned}$$

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

Critical Points We discuss the notion of critical points. 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 for 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 coincide. This causes problems since we consider dynamic constructions, where we keep track of the motions of *all* objects. A division by zero cannot be executed. It arises in the case of the intersection of two parallel lines, for example. We use the definition of a critical point from [9] which is less restrictive than the definition from [7]. 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 [9, Def. 3.2.1] *m -Head $\Gamma^{(m)}$ of a GSP Γ*

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

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

Definition 3 implies that $\Gamma^{(m)}$ has k free variables z_{-k+1}, \dots, z_0 like Γ , and m dependent variables v_1, v_2, \dots, v_m . Each dependent variable of $\Gamma^{(m)}$ is defined by the same operation as the corresponding dependent variable of Γ .

Definition 4 [9, Def. 3.2.2] m -Critical Point of a GSP Γ

Let Γ be a GSP with k free variables z_{-k+1}, \dots, z_0 , and n dependent ones v_1, \dots, v_n . Let $m \in \{1, \dots, n\}$, and let $\tilde{C} = (c_{-k+1}, \dots, c_0, c_1, \dots, c_{m-1}) \in \mathbb{K}^{k+m-1}$ 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 $(c_{-k+1}, \dots, c_{m-1}, c_m)$ 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 5 [9, Def. 3.2.3] Critical Point of Γ

A point $C = (c_{-k+1}, \dots, c_0, c_1, c_2, \dots, c_n)$ 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 6 [9, Def. 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 observe that the set of critical values of Γ is obtained by projecting the set of critical points to the k coordinates of the free variables. Hence, the set of critical points of a GSP Γ is finite if and only if the set of critical values of Γ is finite [9, Lem. 3.2.5]. 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 [9, Lem. 3.2.9]. Consequently, along a polynomial path of the free variables, we either have at most a finite number of critical points, or every point on the path can be extended to a critical point [9, Lem. 3.2.10, Cor. 3.2.11].

Example 2 In Example 1, $(\pm 1, 0, 0)$ are the 3-critical points, $(\pm 1, 0, 0, 0)$ are the critical points, and ± 1 are the critical values. We describe $\sqrt{z - \sqrt{z^2}}$ by the GSP Γ_1 .

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

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 $z \neq 0$.

Continuity and the Tracing Problem 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 [22]. As before, we consider the algebraic situation.

Definition 7 [9, Def. 3.3.1] *Continuous Evaluation*

Let Γ be a GSP having k free variables and n -dependent elements; without loss of generality, let the first k variables z_{-k+1}, \dots, z_0 be the free variables and v_1, \dots, v_n be the dependent ones. Furthermore, for each free variable z_l , we are given a continuous path $p_l(t) : [0, 1] \rightarrow \mathbb{K}$. A *continuous evaluation* of the GSP Γ under the movement $\{p_l(t)\}$ is an assignment of continuous functions $v_i(t) : [0, 1] \rightarrow \mathbb{K}$ to the dependent variables, i.e., for each dependent variable v_i , there is a function $v_i(t)$ such that for all $t \in [0, 1]$ the point $(p_{-k+1}(t), \dots, p_0(t), v_1(t), \dots, v_n(t))$ is an instance of the GSP Γ .

If a 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 [7,9] using the theory of coverings. If a starting instance is fixed (i.e. values for $v_1(0), \dots, v_n(0)$) 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. If the GSP has one free variable z , only, then the path $v_i(t)$ of a dependent variable v_i in a continuous evaluation along a path $p(t)$ is essentially a lifting of $p(t)$ to the covering space of the Riemann surface of the algebraic function that belongs to v_i ; see [9, Sect. 8.5] for a detailed derivation.

We come back to the drag mode in a Dynamic Geometry System: Here, we are given a starting configuration, which is an instance of the underlying GSP Γ , and a continuous path of the free points. The resulting motion of the entire construction is the continuous evaluation that is implicitly defined by the GSP Γ , the starting instance, and the path of the free variables. The task is to trace this continuous evaluation, and we call this problem the *Tracing Problem from Dynamic Geometry* [7,9,22]. Let Γ be a GSP with k free and n -dependent variables.

Problem 1 *Tracing Problem*

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 $p_{-k+1}, \dots, p_0 : [0, 1] \rightarrow \mathbb{K}$ of the free variables of Γ with $(p_{-k+1}(0), \dots, p_0(0)) = z_A$ and $(p_{-k+1}(1), \dots, p_0(1)) = z_B$ are given. We assume that the resulting continuous evaluation starting at A exists. Does this continuous evaluation end at the given instance B ?

In [22], Kortenkamp and Richter-Gebert show by a reduction of 3-SAT that the Tracing Problem is NP-hard; it is decidable for piecewise polynomial paths [7, 9]. The Tracing Problem is formulated 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}, \dots, p_0 and the instance A ends at B . 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 have to determine the values of the dependent variables at some intermediate times $t \in (0, 1)$. Recall that critical points of a GSP Γ cannot be instances 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.

The Derivative of a GSP with Respect to Time For the Tracing Problem, we are given a GSP Γ , paths p_{-k+1}, \dots, p_0 of the free variables, and a starting instance A . They define the corresponding continuous evaluation $(v_1(t), \dots, v_n(t))$ 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 $(\dot{v}_1(t), \dots, \dot{v}_n(t))$ of the continuous evaluation $(v_1(t), \dots, v_n(t))$ with respect to time; see [8] and [9, Sect. 3.5]. We assume that the paths p_{-k+1}, \dots, p_0 of the free variables of Γ are continuously differentiable. Before explaining the general case, we look at an example.

Example 3 We consider the GSP Γ from Example 1 describing $\sqrt{z^2 - 1}$. If z moves on $p(t)$, then the dependent variables become functions in t . By the chain rule, we get the derivatives $\dot{v}_1(t)$, $\dot{v}_2(t)$, and $\dot{v}_3(t)$ of these functions with respect to time.

$$\begin{aligned} z &\leftarrow \text{FREE} \quad \backslash\backslash z = p(t), & \dot{z} &= \dot{p}(t) \\ v_1 &\leftarrow z \cdot z \quad \backslash\backslash v_1 = z^2, & \dot{v}_1(t) &= 2p(t)\dot{p}(t) = 2z\dot{z} \\ v_2 &\leftarrow v_1 - 1 \quad \backslash\backslash v_2 = z^2 - 1, & \dot{v}_2(t) &= \dot{v}_1(t) \\ v_3 &\leftarrow \sqrt{v_2} \quad \backslash\backslash v_3 = \sqrt{z^2 - 1}, & \dot{v}_3(t) &= \frac{\dot{v}_2(t)}{2v_3(t)} = \frac{\dot{v}_2}{2v_3} \end{aligned}$$

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 the GSP Γ , the path $p(t)$, the instance A , and the variables $\dot{z}, \dot{v}_1, \dot{v}_2, \dot{v}_3$. For the definition of \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 8 and Lemma 1.

Definition 8 *Derivative GSP $\dot{\Gamma}$ of a GSP Γ with respect to time*

Let Γ be a GSP with k free variables z_{-k+1}, \dots, z_0 and n -dependent variables v_1, \dots, v_n . Then the *derivative GSP $\dot{\Gamma}$* consists of $2k$ free variables $\dot{z}_{-k+1}, \dots, \dot{z}_0, z_{-k+1}, \dots, z_0$ and at most $6n$ -dependent ones including $\dot{v}_1, \dots, \dot{v}_n$ and v_1, \dots, 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 $\dot{\Gamma}$
$v_a \leftarrow v_b \pm v_c$	$\dot{v}_a(t) = \dot{v}_b(t) \pm \dot{v}_c(t)$	$v_a \leftarrow v_b \pm v_c, \dot{v}_a \leftarrow \dot{v}_b \pm \dot{v}_c$
$v_a \leftarrow v_b \cdot v_c$	$\dot{v}_a(t) = \dot{v}_b(t)v_c(t) + v_b(t)\dot{v}_c(t)$	$v_a \leftarrow v_b \cdot v_c, v_a^{(1)} \leftarrow \dot{v}_b \cdot v_c$ $v_a^{(2)} \leftarrow v_b \cdot \dot{v}_c, \dot{v}_a \leftarrow v_a^{(1)} + v_a^{(2)}$
$v_a \leftarrow v_b/v_c$	$\dot{v}_a(t) = \frac{\dot{v}_b(t)v_c(t) - v_b(t)\dot{v}_c(t)}{(v_c(t))^2}$	$v_a \leftarrow v_b/v_c, v_a^{(1)} \leftarrow \dot{v}_b \cdot v_c$ $v_a^{(2)} \leftarrow v_b \cdot \dot{v}_c, v_a^{(3)} \leftarrow v_a^{(1)} - v_a^{(2)}$ $v_a^{(4)} \leftarrow v_c \cdot v_c, \dot{v}_a \leftarrow v_a^{(3)}/v_a^{(4)}$
$v_a \leftarrow \sqrt{v_c}$	$\dot{v}_a(t) = \frac{\dot{v}_c(t)}{2v_a(t)}$	$v_a \leftarrow \sqrt{v_c}, v_a^{(1)} \leftarrow 2 \cdot v_a$ $\dot{v}_a \leftarrow \dot{v}_c/v_a^{(1)}$

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 $v_a^{(1)}, \dots, v_a^{(4)}$ are auxiliary variables, they are needed to translate the differentiation rules to a GSP. They are not important for our applications and will be omitted.

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

Let p_{-k+1}, \dots, p_0 be continuously differentiable paths of the free variables z_{-k+1}, \dots, z_0 of Γ , and let $A = (p_{-k+1}(0), \dots, p_0(0), a_1, \dots, a_n)$ be an instance of Γ . There is a continuous evaluation of Γ along the paths p_{-k+1}, \dots, p_0 starting at A if and only if there is a continuous evaluation of $\dot{\Gamma}$ along the paths $\dot{p}_{-k+1}, \dots, \dot{p}_0, p_{-k+1}, \dots, p_0$ starting at \dot{A} with $\dot{a}_l = \dot{p}(0); l = -k + 1, \dots, 0$.

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, $i = 1, \dots, n$.

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; see [8] and [9, Sect. 3.6]. 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 [25, Chap. 1].

We consider real and complex interval arithmetic [1]. As complex intervals we choose closed circles $\{c; r\}$ with $r \geq 0$ and $c \in \mathbb{C}$ or axis parallel rectangles $A = A_1 + iA_2$. Here, A_1 and A_2 are real closed intervals. In rectangular arithmetic, the operations $+, -, \cdot, /$ are defined in a similar way as the operations on \mathbb{C} . For circular arithmetic, we choose the centered multiplication $\{a; r_a\} \cdot \{b; r_b\} = \{a \cdot b; |a|r_b + |b|r_a + r_a r_b\}$ [1, 21]. We denote the set of real intervals by $I(\mathbb{R})$, the set of rectangular intervals by $R(\mathbb{C})$, and the set of circular intervals by $K(\mathbb{C})$. Let $I(\mathbb{K}) \in \{I(\mathbb{R}), R(\mathbb{C}), K(\mathbb{C})\}$. The mentioned interval arithmetics fulfill the inclusion monotonicity property [1], which is crucial for our algorithm:

Theorem 1 *Let $A^{(1)}, A^{(2)}, B^{(1)}, B^{(2)} \in I(\mathbb{K})$ with $A^{(1)} \subset B^{(1)}$ and $A^{(2)} \subset B^{(2)}$. Then we have $A^{(1)} \circ A^{(2)} \subset B^{(1)} \circ B^{(2)}$ for all operations $\circ \in \{+, -, \cdot, /, \sqrt{\cdot}\}$.*

We define a square root operation on the set of intervals. The root operation on $I(\mathbb{R})$ is defined for intervals $A = [a_1, a_2]$ with $a_1 > 0$ by $\sqrt{A} = [\sqrt{a_1}, \sqrt{a_2}]$ or $\sqrt{A} = [-\sqrt{a_2}, -\sqrt{a_1}]$. We assign to each rectangle $A \in R(\mathbb{C})$ with $0 \notin A$ the smallest rectangle in $R(\mathbb{C})$ containing $\{\sqrt{a} \mid a \in A\}$. Here, $\sqrt{\cdot}$ is a branch of the square root function that is defined on A . In fact, the square root \sqrt{A} of a rectangle $A = [a_1, a_2] + i[b_1, b_2] \not\ni 0$ is the smallest axis parallel rectangle containing the five points $\sqrt{a_1 + ib_1}, \sqrt{a_1 + ib_2}, \sqrt{a_2 + ib_1}, \sqrt{a_2 + ib_2}$, and \sqrt{x} . Here, x is the point of A with the smallest distance to the origin [9, Sect. A.2.1]. As in [20, 21], we define $\sqrt{A} := \{\sqrt{a}; \sqrt{|a|} - \sqrt{|a| - r_a}\}$ for a circular interval $A = \{a; r_a\}$ with $0 \notin A$. In [20], the case $0 \in A$ is discussed as well. These root operations fulfill the inclusion monotonicity property. If $A \subset B$ for two intervals A and B , then $\sqrt{A} \subset \sqrt{B}$ holds if the same branch of the root function is chosen. Furthermore, $0 \notin A$ implies $\sqrt{A} \cap (-\sqrt{A}) = \emptyset$. We mention that the distributive law does not hold for the presented interval arithmetics.

There are many other range-based models that seem to be useful for our algorithms, as well. The investigation of these models is an interesting future project. In affine arithmetic, quantities are represented by affine forms [6, 25]. 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 [14], complex affine arithmetic is defined for rectangular ranges by deviding a complex quantity into its real and imaginary part; see [9, Sect. 6.7] for a discussion of the usage of affine arithmetic in the context of Dynamic Geometry. In Taylor models, the quantities are represented by higher order polynomials [19]. The Hermite-Obreschkoff method can deal with algebraic functions as explained in [18].

Definition 9 *Interval-GSP Γ_{int}*

Let Γ be a GSP over $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$ having the free variables z_{-k+1}, \dots, z_0 and the dependent variables v_1, \dots, v_n . Then, Γ_{int} is a GSP-like structure over $I(\mathbb{K})$ with k free variables Z_{-k+1}, \dots, Z_0 and n -dependent variables V_1, \dots, V_n . Every dependent variable V_i of $\Gamma_{\text{int}}, i \in \{1, \dots, n\}$, 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}
$v_a \leftarrow v_b \pm v_c$	$V_a \leftarrow V_b \pm V_c$
$v_a \leftarrow v_b \cdot v_c$	$V_a \leftarrow V_b \cdot V_c$
$v_a \leftarrow v_b / v_c$	$V_a \leftarrow V_b / V_c$
$v_a \leftarrow \sqrt{v_c}$	$V_a \leftarrow \sqrt{V_c}$

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. 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 2 [9, Lem. 3.6.3, 3.6.4] *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 $A = (a_{-k+1}, \dots, a_0, a_1, \dots, a_n)$ be an instance of Γ , and let p_{-k+1}, \dots, p_0 be paths of the free variables of Γ such that the corresponding continuous evaluation $v_1(t), \dots, v_n(t)$ 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
$v_a \leftarrow v_b \pm v_c$	$I_a = I_b \pm I_c$
$v_a \leftarrow v_b \cdot v_c$	$I_a = I_b \cdot I_c$
$v_a \leftarrow v_b / v_c$	$I_a = I_b / I_c$, if $0 \notin I_c$
$v_a \leftarrow \sqrt{v_c}$	$I_a = \sqrt{I_c} \ni v_a(t_1)$, if $0 \notin I_c$

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

Example 4 For interval-GSPs, we have to consider interval dependency [12, p. 4]. We describe the polynomial $f(x) = x^2 - x = x(x - 1)$ by two GSPs Γ and $\tilde{\Gamma}$:

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

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 interval $[0, 1]$ and the corresponding instances of Γ_{int} and of $\tilde{\Gamma}_{\text{int}}$; although $I_0 = \tilde{I}_0 = [0, 1]$, we have $I_2 \neq \tilde{I}_2$:

$$\begin{array}{ll}
 \Gamma_{\text{int}} : I_0 = [0, 1] & \tilde{\Gamma}_{\text{int}} : \tilde{I}_0 = [0, 1] \\
 I_1 = I_0 - 1 = [0, 1] - 1 = [-1, 0] & \tilde{I}_1 = \tilde{I}_0 \cdot \tilde{I}_0 = [0, 1] \cdot [0, 1] = [0, 1] \\
 I_2 = I_0 \cdot I_1 = [0, 1] \cdot [-1, 0] = [-1, 0] & \tilde{I}_2 = \tilde{I}_1 - \tilde{I}_0 = [0, 1] - [0, 1] = [-1, 1]
 \end{array}$$

3 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. Homotopy continuation methods have successfully been applied to solve *polynomial systems of equations* and have lead to the new area *Numerical Algebraic Geometry* [24]. 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. These paths are traced using numerical methods. In [11], a robust algorithm to trace curves that are implicitly defined by algebraic equations is given. The algorithm is based on interval analysis. We show that a continuous evaluation is an implicit curve defined by the function $H : \mathbb{R}^{k+n+1} \rightarrow \mathbb{R}^{k+n}$ that depends on the GSP Γ and on the paths of the free variables of Γ . Based on the general ideas of continuation methods and on the function H , we develop a numerical solution that captures the particular structure of the Tracing Problem. In the process, we maintain the parametrization of the given paths of the free variables. We follow the presentation from [9, Chap. 5].

Continuous Evaluations as Implicit Curves We discuss how a continuous evaluation along paths $p_{-k+1}(t), \dots, p_0(t)$ of the k free variables z_{-k+1}, \dots, 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 $H : \mathbb{K}^{k+n} \times [0, 1] \rightarrow \mathbb{K}^{k+n}$ 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
$a \leftarrow b \pm c$	$a - (b \pm c)$
$a \leftarrow b \cdot c$	$a - b \cdot c$
$a \leftarrow b/c$	$a \cdot c - b$
$a \leftarrow \sqrt{b}$	$a^2 - b$

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

$$F = F(\Gamma) : \mathbb{K}^{k+n} \rightarrow \mathbb{K}^{k+n},$$

$$(z_{-k+1}, \dots, z_0, v_1, \dots, v_n) \mapsto (z_{-k+1}, \dots, z_0, \mathcal{P}_{v_1}, \dots, \mathcal{P}_{v_n})$$

and $H = H(\Gamma, p_{-k+1}(t), \dots, p_0(t)) : \mathbb{K}^{k+n} \times [0, 1] \rightarrow \mathbb{K}^{k+n}, (z_{-k+1}, \dots, z_0, v_1, \dots, v_n, t) \mapsto F(z_{-k+1}, \dots, z_0, v_1, \dots, v_n) - (p_{-k+1}(t), \dots, p_0(t), 0, \dots, 0)$, where $p_l(t)$ is the path of the free variable z_l . We observe that a point $(a_{-k+1}, \dots, a_0, a_1, \dots, a_n) \in \mathbb{K}^{k+n}$ fulfills the relations of Γ if and only if $F(a_{-k+1}, \dots, a_0, a_1, \dots, a_n) - (a_{-k+1}, \dots, a_0, 0, \dots, 0) = (0, \dots, 0)$. Similarly, if a tuple of paths $(v_1(t), \dots, v_n(t))$ is a continuous evaluation along the given paths $p_{-k+1}(t), \dots, p_0(t)$ of the free variables z_{-k+1}, \dots, z_0 , then $H(p_{-k+1}(t), \dots, p_0(t), v_1(t), \dots, v_n(t), t) = (0, \dots, 0)$.

Lemma 3 [9, Lem. 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 $(a_{-k+1}, \dots, a_n) \in \mathbb{K}^{k+n}$ is an instance of the GSP Γ if and only if $F(a_{-k+1}, \dots, a_n) - (a_{-k+1}, \dots, a_0, 0, \dots, 0) = 0$ and $\det F'(a_{-k+1}, \dots, a_n) \neq 0$ hold.*
2. *If a point $(a_{-k+1}, \dots, a_n) \in \mathbb{K}^{k+n}$ is a critical point of the GSP Γ , then $\det F'(a_{-k+1}, \dots, a_n) = 0$ holds. A point $(a_{-k+1}, \dots, a_n) \in \mathbb{K}^{k+n}$ is an m -critical point of the GSP Γ if and only if $\det F(\Gamma^{(m)})'(a_{-k+1}, \dots, a_m) = 0$, $\det F(\Gamma^{(m-1)})'(a_{-k+1}, \dots, a_{m-1}) \neq 0$ and $F(\Gamma^{(m-1)})(a_{-k+1}, \dots, a_{m-1}) - (a_{-k+1}, \dots, a_0, 0, \dots, 0) = 0$ hold where $\Gamma^{(m)}$ and $\Gamma^{(m-1)}$ are the m -head and the $(m - 1)$ -head of Γ .*
3. *Let $p_{-k+1}(t), \dots, p_0(t)$ be paths of the free variables z_{-k+1}, \dots, z_0 of the GSP Γ that are continuously differentiable on a neighborhood $U_{[0,1]}$ of the time interval $[0, 1]$. 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 $\det F'(A) = \det \frac{\partial H}{\partial A} \neq 0$, then the corresponding continuous evaluation $(v_1(t), \dots, v_n(t))$ exists locally, i.e., there is an $\varepsilon > 0$ such that the paths $v_j(t)$ are defined for $t \in (-\varepsilon, \varepsilon)$. Moreover, the paths $v_j(t)$ are continuously differentiable over $(-\varepsilon, \varepsilon)$; $j = 1, \dots, n$.*

Proof We consider the Jacobian $F'(a_{-k+1}, \dots, a_n)$ of F in the point (a_{-k+1}, \dots, 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 $\frac{\partial \mathcal{P}_{v_j}}{\partial v_j} = \frac{\partial \mathcal{P}_a}{\partial a}$. Hence, $\det F'(a_{-k+1}, \dots, a_n) = \frac{\partial \mathcal{P}_{v_1}}{\partial v_1} \dots \frac{\partial \mathcal{P}_{v_n}}{\partial v_n}$, and $\det F'(a_{-k+1}, \dots, a_n) = 0$ 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}, \dots, a_n) fulfills all relations of Γ if and only if $F(a_{-k+1}, \dots, a_n) - (a_{-k+1}, \dots, a_0, 0, \dots, 0) = 0$ holds. Since a tuple (a_{-k+1}, \dots, 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 $H(A, 0) = 0$.

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 $\alpha : J \rightarrow \mathbb{K}^{k+n}$ 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 show that the curve $\alpha(t)$ is the wanted continuous evaluation $(p_{-k+1}(t), \dots, p_0(t), v_1(t), \dots, v_n(t))$. 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 $(v_1(t), \dots, v_n(t))$ by the following initial value problem: Let $(u(t), 1) = (u, 1) \in \ker H'(\alpha(t), t)$, then α is described by $\dot{\alpha} = u, \dot{t} = 1$, and $\alpha(0) = A, t(0) = 0$. 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 $\text{rank}(M) = k + n$; see [2, Def. 2.1.7, p. 9]; if $\mathbb{K} = \mathbb{C}$ then $\tau(M) \in \mathbb{C}^{k+n} \times \mathbb{R} \cong \mathbb{R}^{2(k+n)+1}$. 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.

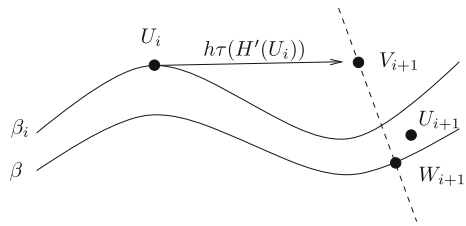
The previous observation shows that the Tracing Problem can be interpreted as an initial value problem that has the corresponding continuous evaluation as solution curve. In addition, this fact implies the existence and uniqueness of continuous evaluations. The approach via initial value problems illuminates the Tracing Problem from the viewpoint of Dynamical Systems.

About Numerical Solutions for the Tracing Problem Various numerical methods have been developed to trace implicitly defined curves [2, 10, 11, 24] and to solve initial value problems [10] 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 [24, p. 10]. Allgower and Georg give a well-founded introduction to Predictor–Corrector methods in [2]. They investigate a general situation where $H: \mathbb{R}^{N+1} \rightarrow \mathbb{R}^N$ 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 $\beta: \mathbb{R} \supset J \rightarrow \mathbb{R}^{N+1}$ with $\beta(0) = u_0$ and $H(\beta(s)) = 0$ for all $s \in J$ as long as we have $\text{rank}(H'(\beta(s))) = N$ and $\beta'(s) \neq 0$.

In our context of Dynamic Geometry, the paths $p_l(t)$ of the free variables z_l are part of the definition of the function $H: \mathbb{R}^{k+n} \times [0, 1] \rightarrow \mathbb{R}^{k+n}$. 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 above. Recall that the curve $\alpha(t)$ is the desired continuous evaluation $(p_{-k+1}(t), \dots, p_0(t), v_1(t), \dots, v_n(t))$. To harmonize the notations, we set $N := k + n$.

The aim of Predictor–Corrector methods in general is to determine iteratively a sequence of points $U_1, U_2, U_3, \dots \in \mathbb{R}^{N+1}$ beginning at a starting point U_0 alongside the curve β that fulfill a certain tolerance criterion like $\|H(U_i)\| \leq \epsilon$ 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 $\beta_i: \mathbb{R} \supset J \rightarrow \mathbb{R}^{k+n+1} = \mathbb{R}^{N+1}$ of the initial value problem $\dot{\beta} = \tau(H'(\beta)), \beta(0) = U_i$. To obtain a new point U_{i+1} , we first make a *predictor step*; see Fig. 2. Like Allgower and Georg in [2], we choose an Euler predictor

Fig. 2 The predictor point V_{i+1} and the corrector point U_{i+1} are shown



$V_{i+1} := U_i + h\tau(H'(U_i))$, where $h > 0$ represents a step length and $\tau(H'(U_i))$ is the tangent vector [2, Def. 2.1.7] defined on page 13. 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 Fig. 2. The Predictor–Corrector continuation method for approximating β consists of repeatedly performing predictor and corrector steps. We adapt the general predictor and corrector steps to the Tracing Problem from Dynamic Geometry and to the function H from page 12. 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 choose $U_0 := (A_0, 0) \in \mathbb{R}^{N+1}$ as starting point. 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 $A_i = (a_{i,-k+1}, \dots, a_{i,0}, a_{i,1}, \dots, a_{i,n})$ is an instance of the underlying GSP Γ with $a_{i,-k+1} = p_{-k+1}(t_i), \dots, a_{i,0} = p_0(t_i)$. 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 $\beta(t) = (\alpha(t), t)$. In addition, we have $\alpha(t) = (p_{-k+1}(t), \dots, p_0(t), v_1(t), \dots, v_n(t))$, where $(v_1(t), \dots, v_n(t))$ is the wanted continuous evaluation.

To keep track of the parametrization induced by the paths $p_l(t)$ of the free variables of 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 $\alpha(t) = (p_{-k+1}(t), \dots, p_0(t), v_1(t), \dots, v_n(t))$, where $(v_1(t), \dots, v_n(t))$ is the wanted continuous evaluation. Thus, to determine $\dot{\alpha}(t_i)$ we compute the derivatives $\dot{p}_l(t_i)$ of the paths p_l and the derivative $(\dot{v}_1(t_i), \dots, \dot{v}_n(t_i))$ of the continuous evaluation $(v_1(t), \dots, v_n(t))$ 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 $(\dot{v}_1(t_i), \dots, \dot{v}_n(t_i))$ using the derivative GSP $\dot{\Gamma}$ defined on page 8.

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 $A_{i+1} = (p_{-k+1}(t_{i+1}), \dots, p_0(t_{i+1}), v_1(t_{i+1}), \dots, v_n(t_{i+1})) = \alpha(t_{i+1})$ lying on the wanted continuous evaluation $(p_{-k+1}(t), \dots, p_0(t), v_1(t), \dots, v_n(t))$.

Now, 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 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 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 method 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 develop the algorithms of Sect. 4, they are based on a step length adaptation that guarantees the correctness of the solution curve. Critical points are surrounded by a detour; see Sect. 5.

Kearfott and Xing [13] 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 contains a single solution. This approach seems not to be efficient for solving the Tracing Problem: 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 Sect. 4.

Blum, Cucker, Shub, and Smale [3, 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.

4 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 [8, 9]. We assume that there are no critical points on the paths p_{-k+1}, \dots, p_0 of the free variables z_{-k+1}, \dots, z_0 , the treatment of

critical points is discussed in Sect. 5. Furthermore, we restrict ourselves to linear paths p_{-k+1}, \dots, p_0 which simplifies Step 2 of Algorithm 1. The used interval arithmetic has to fulfill the inclusion monotonicity property. Algorithm 2 traces the continuous evaluation $v_1(t), \dots, v_n(t)$ 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 1 performs a single tracing step, Algorithm 2 uses Algorithm 1 and traces the whole continuous evaluation as long as there occur no critical points. Algorithm 1 can be extended by an improved steplength adaptation using the derivative GSP $\dot{\Gamma}$; see [9, Sect. 6.3]. The given algorithm is robust as shown in [9, Sect. 6.5]; the problem of overestimation is discussed in [9, Sect. 6.4]. The extension of the algorithms to GSPs with cubic and higher roots is addressed in [9, Sect. 6.6].

Before we give the algorithm, we describe the main idea of our reliable algorithm 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 $v_m = d_1/d_2, d_1, d_2 \in \{z_{-k+1}, \dots, z_0, v_1, \dots, v_{m-1}\}$, be a division operation, and let $d_1(t), d_2(t) : [0, 1] \rightarrow \mathbb{K}$ be the paths of the variables d_1 and d_2 in the continuous evaluation. 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 in the computation of $d_1(t)$ and $d_2(t)$. To execute 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.

Root Let $v_m = \sqrt{r}$ with $r \in \{z_{-k+1}, \dots, z_0, v_1, \dots, v_{m-1}\}$ be a root instruction. Let $r(t) : [0, 1] \rightarrow \mathbb{C}$ and $v_m(t) : [0, 1] \rightarrow \mathbb{C}$ be the paths of the variables r and v_m in the continuous evaluation. To point 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. The main challenge is to resolve 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

$$\forall t \in [0, t_0] : |v_m(t) - a_m| < |v_m(t) - (-a_m)|. \tag{1}$$

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

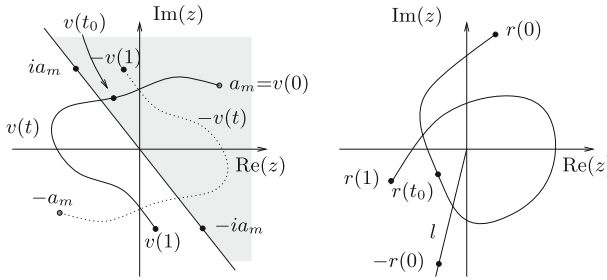


Fig. 3 The left figure shows the path $v(t) := v_m(t)$, the bisector of $a_m = v(0)$ and $-a_m$, and the point $v(t_0)$. On the right hand side, the path $r(t)$ of the radicand and the ray l is shown

evaluation at time t_0 , and $h = t_0$ is a proper step length for the first root-instruction of the first step. Condition (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 Fig. 3. 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 Fig. 3.

If we choose $t_0 \in [0, 1]$ such that the path $r(t) : [0, t_0] \rightarrow \mathbb{K}$ 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 . 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 be done over the field of real numbers \mathbb{R} . We summarize the previous observations and note that Lemma 4 directly generalizes to higher order roots; see [9, Sect. 6.6].

Lemma 4 [9, Lem. 6.1.1, Cor. 6.1.2] *Radicand Lemma*

Let $r : [0, 1] \rightarrow \mathbb{C}^* = \mathbb{C} \setminus \{0\}$ be a continuous path and $r_0 := r(0)$ its starting point. Let $v : [0, 1] \rightarrow \mathbb{C}^*$ be the unique continuous path with $(v(t))^2 = r(t)$ for all $t \in [0, 1]$ and $v(0) = v_0$ for a given $v_0 \in \{\pm\sqrt{r_0}\}$. 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

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

Fig. 4 Algorithm 1 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

$|v(t) - v_0| < |v(t) + v_0|$ at any time $t \in [0, 1]$. The endpoint $v(1)$ is the unique point $v_1 \in \{\pm\sqrt{r(1)}\}$ with $|v_1 - v_0| < |v_1 + v_0|$. Moreover, 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 $t \in [0, 1]$.

If there is no critical point on the paths p_{-k+1}, \dots, p_0 , then we have exactly one continuous evaluation $v_1(t), \dots, v_n(t)$ starting at a given instance A . We trace this continuous evaluation stepwise. The previous considerations lead to an algorithm for a single tracing step; see Fig. 4 for an example.

Algorithm 1 [9, Sect. 6.2]

- Input:** GSP Γ , linear paths p_{-k+1}, \dots, p_0 , starting time $t_0 \in [0, 1]$, starting instance $A = (p_{-k+1}(t_0), \dots, p_0(t_0), a_1, \dots, a_n)$.
 - Output:** Step length h , final instance $B = (p_{-k+1}(t_0 + h), \dots, p_0(t_0 + h), b_1, \dots, b_n)$.
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 $p_l([t_0, t_0 + h]) \subset I_l, l = -k+1, \dots, 0$. Since p_{-k+1}, \dots, 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 $l \in \{-k+1, \dots, 0\}$. 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 $l \in \{-k+1, \dots, 0\}$.
 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
$v_i \leftarrow v_a \pm v_b$	$b_i := b_a \pm b_b$	$I_i := I_a \pm I_b$
$v_i \leftarrow v_a \cdot v_b$	$b_i := b_a \cdot b_b$	$I_i := I_a \cdot I_b$

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

This algorithm is based on the Radicand Lemma 4. 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 [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 $0 \notin I_c$. Equivalently, we could require $a_d \in \sqrt{I_c}$.

We use the inclusion monotonicity property of the chosen interval arithmetic to show that Algorithm 1 terminates. For this reason, the intervals I_l in Step 2 must be

constructed such that the inclusion monotonicity property holds; see [9, p. 75]. If this condition is fulfilled, then Algorithm 1 also works for nonlinear paths p_l . Close to a critical point, the step length computed by Algorithm 1 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 Sect. 5, we discuss how critical points can be treated.

Lemma 5 *Algorithm 1 terminates, and the correct final instance B after the step of length h is computed.*

Proof The continuity of the functions $+$, $-$, \cdot , $:$, and $\sqrt{\quad}$ combined with an inductive argument implies that for every division or square root operation there are finitely many restarts in Step 5; see [9, p. 75]. Hence, Algorithm 1 terminates.

Algorithm 2 [9, Sect. 6.2] Trace this continuous evaluation stepwise using Algorithm 1. In the process, the final instance of a previous step is the starting instance of the next step.

Note that Algorithm 2 does not terminate if there is a critical point on the traced continuous evaluation. In this situation, the step length computed by Algorithm 1 in Algorithm 2 becomes arbitrary small before the critical point is reached.

Theorem 2 *Algorithm 2 terminates and computes the correct final instance B as long as no critical points occur.*

Proof Using Lemmas 2 and 4, we can show $B = (p_{-k+1}(t_0 + h), \dots, p_0(t_0 + h), v_1(t_0 + h), \dots, v_n(t_0 + h))$. The functions $p_l(t)$, $v_i(t)$ and $+$, $-$, \cdot , $:$, $\sqrt{\quad}$ are continuous on a compact set. Thus, they attain their minimum and maximum, and they are uniformly continuous. Therefore, there is a lower bound $h_0 > 0$ for the step length [9, p. 77].

Step Length Adaptation Algorithm 1 does not give an indication for a “good” choice for the step length h , the number of restarts in Step 5 is not known in advance. We supplement Algorithm 1 such that there is at most one restart for every division or root operation. This improvement uses the derivative GSP \dot{I} . A detailed derivation is given in [9, Sect. 6.3]. As in Step 4 of Algorithm 1, we consider a division or root variable v_d and its radicant or divisor $v_c \in \{z_{-k+1}, \dots, v_{d-1}\}$ and its path $v_c(t)$ in the continuous evaluation. The time interval is $[t_0, t_0 + h]$. Let \dot{I}_c be an interval with $\dot{v}_c(t) \in \dot{I}_c$ for all $t \in [t_0, t_0 + h]$. The mean value theorem implies $v_c(t) \in a_c + (t - t_0)\dot{I}_c$ where $a_c = v_c(t_0)$ is the coordinate of v_c in the starting instance A ; see Fig. 5.

If $0 \in \dot{I}_c$, then $v_c(t) \in a_c + h\dot{I}_c$ for all $t \in [t_0, t_0 + h]$, and we set $I_c := a_c + h\dot{I}_c$. If $0 \notin \dot{I}_c$, we know $v_c(t) \in a_c + (t - t_0)\dot{I}_c \subset a_c + T_h\dot{I}_c$ where T_h is the smallest interval in the used interval arithmetic that contains the real interval $[0, h]$. In this case, we set $I_c := a_c + T_h\dot{I}_c$. We consider the cone C with apex (t_0, a_c) and base $(t_0 + h, I_c) = \{(t_0 + h, z) \mid z \in I_c\}$; the graph of $v_c(t)$ is contained in C . If C does not intersect the t axis, then the path $v_c(t)$ does not pass through zero, and the variable v_d does not cause a critical point. Otherwise, let $t' \in (t_0, t_0 + h]$ be the earliest time for which C intersects the t -axis. Then, $h := \frac{2}{3}(t' - t_0)$ is a suitable step length; see Fig. 5.

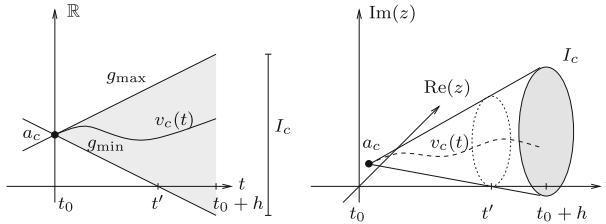


Fig. 5 The cone C and the definition of t' are shown for $I(\mathbb{R})$ on the left and for $K(\mathbb{C})$ on the right

The interval \hat{I}_i can be computed by the interval derivative GSP \hat{I}_{int} . The needed interval operations can be executed since $0 \in I_c$ holds if and only if the cone C intersects the t axis. Hence, a division or root operation is executed only if the divisor or radicant interval does not contain 0. Since we use an interval arithmetic with the inclusion monotonicity property, every division or root variable causes at most one adaptation for the step length.

5 Detection and Treatment of Critical Points

The detection and treatment of critical points in the tracing process described in Sect. 4 is discussed. A division or root variable causes a critical point if the path $v_c(t)$ of its divisor or radicant passes through zero. 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 [15, 17]. We remark that a similar idea is used for solving polynomial systems of equations via homotopy methods [24]. 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.

Detection of Critical Points 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 1 proceeds only 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 1 is restarted with a smaller step length. Thus, a first indication of a critical point is that the step length computed by Algorithm 1 becomes arbitrarily small. Hence, we should stop Algorithm 2 if a suitable lower bound for the step length is reached. How to choose h_l properly and in advance 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 radicant or divisor variable v_c in a continuous evaluation of Γ . We consider linear or circular paths p_l of the free variables z_l of Γ . These paths can be extended to analytic functions $\hat{p}_l: U_{[0,1]} \rightarrow \mathbb{C}$ on a neighborhood $U_{[0,1]} \subset \mathbb{C}$ of the time interval $[0, 1] \subset \mathbb{R}$. 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 $v_j(t): [0, 1] \rightarrow \mathbb{K}$ can be extended to analytic functions $\hat{v}_j: U_{[0,1]} \rightarrow \mathbb{C}$, as well [9, Sect. 8.6.1]. The identity theorem from complex analysis 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 Γ is a GSP over \mathbb{C} , then the real part $\operatorname{Re}(v_j(t))$ and the imaginary part $\operatorname{Im}(v_j(t))$ of $v_j(t)$ are either constantly zero on the interval $[0, 1] \subset \mathbb{R}$ or have a finite number of zeros in $[0, 1]$; see [9, Lem. 7.1.1]. In the starting instance A at time t_0 , 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 $\operatorname{Re}(v_c(t))$ or $\operatorname{Im}(v_c(t))$ 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.

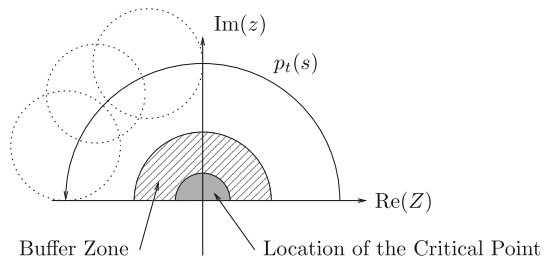
We consider the product $a_c b_c = v_c(t_0)v_c(t_0 + h)$, where b_c is the value of the variable v_c in the final instance B of the current step. By the intermediate value theorem, if $a_c b_c < 0$, then the continuous function $v_c(t)$ has a zero in the time interval $[t_0, t_0 + h]$.

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. Note that $0 \in \dot{I}_c$ does not imply $0 \in \{\dot{v}_c(t) \mid t \in [t_0, t_0 + h]\}$. Thus, 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 $v_c(t_0)v_c(t_0 + h) < 0$ fails. 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. 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 1. The algorithm could be used backwards, as well. As usual, let a_c be the coordinate of v_c in the starting instance A , and b_c the value of v_c in the final instance B of the current step; 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 1 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 Definition 8 and by considering the corresponding value of $\dot{v}_c(t)$ in the instance of $\dot{\Gamma}$ at a fixed time t .

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 . 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 1. 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 μ th derivative differs from zero. Then, $\mu - 1$ is the multiplicity of the zero of v_c that causes the m -critical point.

Treatment of Critical Points After the detection of a critical point, we have to deal with it. We choose Kortenkamp's and Richter-Gebert's approach [15, 17] of detouring around degeneracies in the complex plane, which is used in the Dynamic Geometry

Fig. 6 The construction of the buffer zone is illustrated. The dotted circles are candidates for the interval I : they do not intersect the buffer zone



Software *Cinderella* [23]. This approach leads to a locally and globally consistent treatment of critical points; see [15, Sect. 6.3] and [9, Sects. 1.3, 7.3].

As in our algorithms (Algorithms 1 and 2), we assume that the given paths of the free variables of the GSP Γ are linear paths $p = p_t$ with range \mathbb{C} , for example $p: [0, 1] \rightarrow \mathbb{C}, t \mapsto a + t(b - a)$ 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 $[0, 1]$.

If a candidate 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 in the complex plane \mathbb{C} having the line segment \overline{ab} as diameter [15]. In [9, Sect. 7.2], the necessary changes to Algorithms 1 and 2 are discussed, as well.

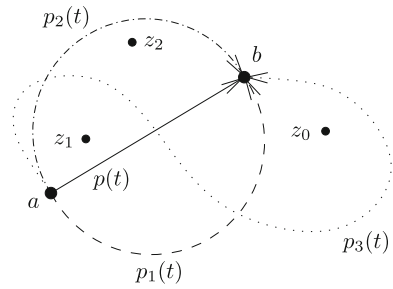
Since Algorithms 1 and 2 become quite inefficient if the starting instance is chosen close to a critical point, we propose to use a buffer zone around a critical point; see Fig. 6 and [8,9]. We define the intervals $I = I_l$ and $I = \hat{I}_l$ such that they do not hit this buffer zone. This approach ensures that the intervals I_l and \hat{I}_l have a larger distance to the critical point. For this reason, we expect that $0 \in I_c$ in Steps 4 and 5 of Algorithm 1 occurs more rarely. Hence, we expect that the number of restarts in Step 5 of Algorithm 1 is reduced that way. To sum up, the buffer zone enlarges the detour p_l around the critical point, but therefore the step length h might become larger.

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 interval contains a critical point. This could be done with an Interval Newton Method [9, Sect. 7.4, Sect. A.5].

Consequences of Detouring If we trace a detour around a candidate critical point, then the final instance depends on the detour. Thus, the choice of the detour heavily influences the properties of the Dynamic Geometry System.

Let Γ be a GSP with 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. 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 5. We have seen in Sect. 2, p. 6, that critical points of Γ are either isolated or form an entire connected component of the configuration space. An isolated critical point might lead

Fig. 7 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 $p(t)$ with the path $p_2(t)$ catches z_1 and z_2 ; the path $p(t)$ with the path $p_3(t)$ catches z_0 and z_1



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.

Algorithm 2 proceeds stepwise. In each step, Algorithm 1 computes 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. In Kortenkamp’s and Richter-Gebert’s approach [15, 17] described on page 22, critical points are avoided by modifying the paths p_l 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 in the continuous evaluation [9, Sect. 7.2].

To investigate the influences of detouring around critical points, we explain when a modified path “catches” a critical point; see Fig. 7 and [9, Sect. 7.3]. 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. Let $\hat{v}_c(z)$ be the algebraic function of v_c induced by the (polynomial) path $p(t)$ of the free variable. The zeros z_0, \dots, 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 $U \setminus \{z_0, \dots, z_{k_c}\}$; see Fig. 7. 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 lead to the final values \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.

6 Conclusion and Future Work

We gave a numerical solution for the Tracing Problem based on numerical continuation methods. By computing the derivative of a GSP, we can exploit the special structure of the Tracing Problem successfully. Moreover, we developed a reliable algorithm for the Tracing Problem from Dynamic Geometry. This algorithm uses interval arithmetic to guarantee the correctness by detecting (potential) critical points in advance and by handling the ambiguity of the root function, which is an advantage over our numerical approach. The problem of interval dependency leads to an overestimation of the range of the considered paths and thus to an underestimation of the step length. For dealing with critical points, we adapted the approach by Kortenkamp and Richter-Gebert [17, 15] and surrounded (potential) critical points by circular detours. An implementation is planned to verify the practical relevance.

Acknowledgments We would like to thank Ulrich Kortenkamp, Helmut Alt, and Dominique Michelucci for fruitful discussions and helpful ideas.

References

1. Alefeld, G., Herzberger, J.: Einführung in die Intervallrechnung, Reihe Informatik, vol. 12. Bibliographisches Institut. In: An English Translation Appeared in 1983, Academic Press, New York (1974)
2. Allgower, E.L., Georg, K.: Numerical Continuation Methods. Springer, Heidelberg (1990)
3. Blum, L., Cucker, F., Shub, M., Smale, S.: Complexity and Real Computation. Springer, New York (1998)
4. Bürgisser, P., Clausen, M., Shokrollahi, M.A.: Algebraic Complexity Theory. Springer, Berlin (1997)
5. Burnikel, C., Funke, S., Mehlhorn, K., Schirra, S., Schmitt, S.: A separation bound for real algebraic expressions. In: European Symposium on Algorithms 2001, LNCS, vol. 2161, pp. 254–265 (2001)
6. de Figueiredo, L.H., Stolfi, J.: Affine arithmetic: concepts and applications. Numer. Algorithm. **37**, 147–158 (2004)
7. Denner-Brosler, B.: On the decidability of tracing problems in dynamic geometry. In: Automated Deduction in Geometry 2004, LNAI, vol. 3763, pp. 111–129. Springer, Heidelberg (2006)
8. Denner-Brosler, B.: An algorithm for the tracing problem using interval analysis. In: ACM Symposium on Applied Computing 2008; Session Geometric Constraints and Reasoning, pp. 1832–1837 (2008)
9. Denner-Brosler, B.: Tracing-problems in dynamic geometry. Ph.D. thesis, Freie Universität, Berlin (2008)
10. Deuffhard, P.: Newton Methods for Nonlinear Problems, Springer Series in Computational Mathematics, vol. 35. Springer, Berlin (2004)
11. Faudot, D., Michelucci, D.: A new robust algorithm to trace curves. Reliab. Comput. **13**, 309–324 (2007)
12. Kearfott, R.B.: Rigorous Global Search: Continuous Problems. Kluwer Academic Publishers, Dordrecht (1996)
13. Kearfott, R.B., Xing, Z.: An interval step control for continuation methods. SIAM J. Numer. Anal. **3**(3), 892–914 (1994)
14. Ko, K., Sakkalis, T., Patrikalakis, N.: A reliable algorithm for computing the topological degree of a mapping in \mathbb{R}^2 . Appl. Math. Comput. **196**(2), 666–678 (2007)
15. Kortenkamp, U.: Foundations of dynamic geometry. Ph.D. thesis, ETH, Zürich (1999)

16. Kortenkamp, U., Richter-Gebert, J.: Decision complexity in Dynamic Geometry. In: Automated Deduction in Geometry, Workshop 2000, vol. 2061, pp. 193–199 (2001)
17. Kortenkamp, U., Richter-Gebert, J.: A dynamic setup for elementary geometry. In: Multimedia Tools in Communicating Mathematics. Springer, Lisboa (2002)
18. Nedialkov, N., Kreinovich, V., Starks, S.: Interval arithmetic, affine arithmetic, taylor series methods: Why, what next? Numer. Algorithm. **37**(1–4), 325–336 (2004)
19. Neumaier, A.: Taylor forms: use and limits. Reliab. Comput. **9**, 43–79 (2003)
20. Petković, L., Petković, M.: On the k -th root in circular arithmetic. Computing **33**, 27–35 (1984)
21. Petković, M., Petković, L.: Complex Interval Arithmetic and Its Applications. Wiley, Berlin (1998)
22. Richter-Gebert, J., Kortenkamp, U.: Complexity issues in dynamic geometry. In: Foundations of Computational Mathematics, Proceedings of the SMALEFEST 2000 World Scientific, Hongkong (2002)
23. Richter-Gebert, J., Kortenkamp, U.: The Cinderella. 2 Manual: Working with The Interactive Geometry Software. Springer, Berlin (2012). <http://www.cinderella.de>
24. Sommese, A.J., Verschelde, J., Wampler, W.C.: Introduction to numerical algebraic geometry. In: Solving Polynomial Equations, Algorithms and Computation in Mathematics, vol. 14. Springer, Berlin (2005)
25. Stolfi, J., de Figueiredo, L.H.: Self-validated numerical methods and applications. In: Brazilian Mathematics Colloquium Monograph, IMPA, Rio de Janeiro, Brazil. <http://www.ic.unicamp.br/~stolfi/EXPORT/bibliography/stolfi.html> (1997)