A Robust and Efficient Method for Solving Point Distance Problems by Homotopy

Rémi Imbach, Pascal Mathis, Pascal Schreck

informatics / *mathematics*

Inría

A Robust and Efficient Method for Solving Point Distance Problems by Homotopy

Rémi Imbach, Pascal Mathis, Pascal Schreck

A Robust and Efficient Method for Solving Point Distance Problems by Homotopy

Rémi Imbach[*], Pascal Mathis[†], Pascal Schreck[†]

Project-Team VEGAS

Abstract: The goal of Point Distance Solving Problems is to find 2D or 3D placements of points knowing distances between some pairs of points. The common guideline is to solve them by a numerical iterative method (*e.g.* Newton-Raphson method). A sole solution is obtained whereas many exist. However the number of solutions can be exponential and methods should provide solutions close to a sketch drawn by the user. Geometric reasoning can help to simplify the underlying system of equations by changing a few equations and triangularizing it. This triangularization is a geometric construction of solutions, called construction plan. We aim at finding several solutions close to the sketch on a one-dimensional path defined by a global parameter-homotopy using a construction plan. Some numerical instabilities may be encountered due to specific geometric configurations. We address this problem by changing on-the-fly the construction plan. Numerical results show that this hybrid method is efficient and robust.

Key-words: Point Distance Solving Problems, Reparameterization, Curve Tracking, Symbolic-Numeric Algorithm

---

[*] Travail en partie effectué au laboratoire ICube.

[†] Laboratoire ICube - UMR 7357 CNRS Université de Strasbourg, bd Sébastien Brant, F-67412 Illkirch Cedex, France

Une méthode robuste et efficace pour résoudre des systèmes de contraintes de distances entre points par homotopie

Résumé : Le but de la résolution de problèmes de contraintes de distances entre points est de placer en 2D ou 3D un ensemble de points connaissant certaines distaces entre paires de points. De tels problèmes sont en général résolu grâce à une méthode numérique, souvent Newton-Raphson, qui ne produit qu'une solution alors qu'il en existe un nombre exponentiel. Celles ressemblant à l'esquisse sont d'un intéret particulier. Le raisonnement géométrique peut cependant aider à simplifier les sytèmes d'équations correspondant en remplaçant quelques équations, ce qui permet de les trianguariser. Une telle triangularisation est une construction géométrique des solutions et est appellée plan de construction. On se propose dans ce rapport de trouver plusieurs solutions, proches de l'esquisse sur une courbe définie par une homotopie utilisant le plan de construction pour réduire son coût. L'utilisation d'un plan de construction induit des instabilités numériques à proximité de certains points; ces instabilités sont évités en changeant le plan de construction pendant le suivi de la courbe. La méthode décrites ici a été implémentée, et les résultats obtenus montrent son efficacité et sa robustesse.

Mots-clés : Problèmes de Contraintes de Distances entre Points, Re-paramétrisation, Suivi de courbes, Algorithme symbolique-numérique

1   Introduction

Geometric Constraints Solving Problems arise in many fields such as CAD, robotics or molecular modeling. The problem is to determine the positions of geometric elements (points, lines, planes, circles, etc.) that must satisfy a set of constraints such as distances, angles, tangencies and so on. Commercial solvers generally rely on numerical methods such as Newton or quasi-Newton that provide a single solution. Even when problems are well-constrained the number of solutions may grow exponentially with the number of constraints. Usually the user is not interested in all solutions but only to those whose shape is close to a sketch that he provided.

Here the restrained class of problems involving only points and distance constraints is considered. Our aim is precisely to design a method that uses the sketch to guide the research of several solutions. We assume that the considered problems are structurally well-constrained. Roughly speaking, this means that there exist some assignments for dimensions leading to finitely many solutions. We also consider problems resisting to decomposition-recombination methods.

Several methods can yield several or even all the solutions. Subdivision methods [15, 6] provide all the solutions but the number of boxes to be explored can be huge. In algebraic approaches, homotopy methods have been successfully studied in this area [3, 11] but only for small size problems. Indeed, the number of homotopy paths to follow grows exponentially with the number of constraints. In [11] it is proposed to use the sketch to define a parameter-homotopy. A sole path is followed but a sole solution is obtained.

Another way to get several solutions comes from geometric methods. A construction plan is first derived by applying some geometric construction rules. It consists in a sequence of basic construction steps. Next, such a plan is numerically evaluated to yield different solutions. However, no construction plan can be easily found for some 2D problems and for most 3D problems. To circumvent this, [7] proposes a new approach that performs a *reparameterization*. In this approach, a geometric constraint system, say S_1, is modified by adding and removing some constraints to obtain a system S_2 similar to the original one and from which a construction plan can be easily derived. In turn, this construction plan is used to define a reduced system R from the constraints removed from S_1. R is then solved by a numerical solver in order to meet the removed constraints while still satisfying the constraints of S_2. For point distance problems, the number of constraints that have to be swapped to obtain S_2 from S_1 is much lower than the number of constraints of S_1 hence the size of the system to be numerically solved is drastically reduced.

The drawback is that the equations of system R are much harder to deal with due to irregular configurations. So the choice of the numerical method is crucial to provide several solutions. In [7] one or two constraints are removed and the solutions are found by a sampling method. In [4], this idea is extended for more than two constraints, Newton-Raphson method allows to get some of the sought solutions. In [2], the reparameterization is used at a low-level to simplify linear algebra involved by numerical methods. Finally [8] presents a first attempt for using a homotopy method along with reparameterization. The idea is to follow a homotopy path to which belongs the sketch.

All these work can quickly find some of the solutions desired by the user. However some solutions are often missed because of numerical inaccuracies. In this paper we provide an effective and original method to face it. This work is based on tracking homotopy paths defined by a construction plan obtained after reparameterization of the problem (system S_2). The central idea is to detect ill-conditioned configurations induced by the interpretation of the construction plan, and to change on-the-fly the construction plan to get away from such configurations. We justify this approach by showing that these changes of construction plans during paths tracking do not change the path that is followed. More precisely,

- we show that the paths followed with a homotopy method applied to the reduced system using a construction plan can be glued together around the singular points. The whole path is exactly the path which would be followed by continuation on the original system. Thus, it is independent of the reconstruction plan obtained with the reparameterization phase.

- we use geometric criteria to detect in advance the singular points caused by a particular construction plan, and we design a way to modify on-the-fly the reparameterization in order to avoid singular points.

- putting everything together, we marry a homotopy method with a reparameterization to have a new algorithm to solve point distance satisfaction problems. We prove that this algorithm terminates and is correct. We compare the results obtained with our new method with another homotopy method and we find that this is algorithm is more than three time faster than our previous algorithm without reparameterization.

The rest of the paper is organized as follows. Focusing on 2D problems, Sec. 2 gives definitions on construction plans and homotopy. Sec. 3 gives results about homotopy paths tracking on construction plans that justify our approach. Sec. 4 explains how to change a construction plan on-the-fly to avoid critical situations. The soundness of the approach is justified in Sec. 5. Sec. 6 gives tracks to extend our method to 3D problems, and Sec. 7 presents some experimental results.

2   Notations and Definitions

A Point Distance Satisfaction Problem (PDSP) is a constraint satisfaction problem where constraints are imposed distances between points. Unknown points are sought either in the Euclidean plane for 2D PDSP or in the Euclidean space for 3D PDSP. We focus here on the 2D case.

The method presented in this paper uses symbolic manipulations on PDSP to ease their numerical solving. For the sake of clarity in the description of this symbolic-numeric approach, we will use different typefaces to denote a variable and its numeric value. A boldface lowercase letter as $\mathbf{x}$ will denote a variable, and an uppercase boldface letter as $\mathbf{X}$ will denote a set of variables. A value for $\mathbf{x}$, in general a real number, will be denoted by a lowercase italic letter x, and a value for $\mathbf{X}$, in general a real vector, will be denoted by a uppercase italic letter X. We make an exception for variables associated with geometric objects: if $\mathbf{p}$ is a point, we will note p a value for $\mathbf{p}$ whereas it refers to a vector of real values (its coordinates) in a geometric context.

2.1   Point Distance Satisfaction Problems

A PDSP G is denoted by $G = C[\mathbf{P}, \mathbf{A}]$ where $\mathbf{P}$ is a set of unknown points, $\mathbf{A}$ is a set of length parameters, and C is a set of m constraints of distance. A distance constraint of parameter $\mathbf{a}_1 \in \mathbf{A}$ between points $\mathbf{p}_2, \mathbf{p}_3 \in \mathbf{P}$ is written $distance(\mathbf{p}_2, \mathbf{p}_3) = \mathbf{a}_1$. A PDSP that consists in constructing 6 points $\mathbf{p}_1, \ldots, \mathbf{p}_6$ in a plane knowing 9 distances is given in Fig. 1. We call it $K_{3,3}$[1]. Numerical values A^{so} for $\mathbf{A}$ are usually given by a user. The aim is to find the solutions that respect these dimensions. We suppose in addition that a *sketch*, *i.e.* a geometric placement of points of $\mathbf{P}$, with possibly a representation of constraints, is available. Right part of Fig. 1 shows a sketch of the PDSP $K_{3,3}$.

---

[1] when considering right part of Fig. 1 as a non-oriented graph, it is the complete bipartite graph with 3 vertices in each component

Unknowns:
point $\mathbf{p}_1, ..., \mathbf{p}_6$
Parameters:
length $\mathbf{a}_1, ..., \mathbf{a}_9$
Constraints:
$distance(\mathbf{p}_1, \mathbf{p}_2) = \mathbf{a}_1$
$distance(\mathbf{p}_2, \mathbf{p}_3) = \mathbf{a}_2$
...
$distance(\mathbf{p}_2, \mathbf{p}_5) = \mathbf{a}_8$
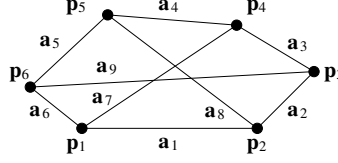$distance(\mathbf{p}_3, \mathbf{p}_6) = \mathbf{a}_9$

Figure 1: A symbolic statement (left part) and a dimensioned sketch (right part) of the PDSP $K_{3,3}$. Edges are distance constraints of parameters $\mathbf{a}_i$.

Unknown points $\mathbf{p}_j$ are sought in the Euclidean plane and are each associated with two algebraic unknowns corresponding to their coordinates. Let c_i be $distance(\mathbf{p}_j, \mathbf{p}_k) = \mathbf{a}_i$. It is associated with the numerical function $c_i(\mathbf{P}, \mathbf{A})$ called *numerical interpretation*:

$$c_i(\mathbf{P}, \mathbf{A}) = \mathbf{p}_j \mathbf{p}_k - \mathbf{a}_i \tag{1}$$

where $\mathbf{p}_j \mathbf{p}_k$ holds for the Euclidean distance between $\mathbf{p}_j$ and $\mathbf{p}_k$. Since constraints of distance are invariant up to rigid motions of the plane, placements of points in the plane fulfilling constraints are sought in a *reference, i.e.* the values of 3 unknown coordinates are fixed. For $K_{3,3}$ we could search values for points with $\mathbf{p}_1$ at the origin and $\mathbf{p}_2$ with null ordinate, and assign variables $\{\mathbf{x}_1, ..., \mathbf{x}_9\}$ to remaining free coordinates.

We denote by $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_m\}$ the set of free unknown coordinates of points in $\mathbf{P}$ and we define the system of equations $\mathcal{F}$ associated to G as

$$F(\mathbf{X}, \mathbf{A}) = 0 \tag{$\mathcal{F}$}$$

where $F : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}^m$ has as i-th component the numerical interpretation of the constraint c_i defined in Eq. (1).

We will call *figure* a set of real values X for $\mathbf{X}$. Given positive values A^{so} for $\mathbf{A}$, we call *solution* of G a figure X that is a solution of $\mathcal{F}_{so}$ defined as

$$F(\mathbf{X}, A^{so}) = 0 \tag{$\mathcal{F}_{so}$}$$

We highlight here that a sketch of G is a figure X^{sk}. In addition, by measuring on X^{sk} distances between appropriated points one can find positive values A^{sk} for $\mathbf{A}$ such that X^{sk} is a solution of the system $\mathcal{F}_{sk}$ defined as

$$F(\mathbf{X}, A^{sk}) = 0 \tag{$\mathcal{F}_{sk}$}$$

In the following we will consider structurally and generically well constrained PDSP. A structurally well constrained PDSP satisfies in particular $|\mathbf{X}| = |\mathbf{A}| = |C| = m$ if elements of $\mathbf{A}$ are algebraically independent (see the Koenig-Hall theorem [16]). A generically well constrained PDSP admits for generic values of parameters a not null and finite number of solutions. Here generic stands for the complementary of a set having a null Lebesgue measure in an open subset of the space of parameters.

2.2 Homotopy

Equations of $\mathcal{F}_{so}$ can be written as polynomials, hence $\mathcal{F}_{so}$ can be solved in $\mathbb{C}^m$ by a classical homotopy method (see [1] for an introduction to homotopy methods, and [3] for its application to our context). All the complex roots of $\mathcal{F}_{so}$ are searched and found, whereas in applications only the real solutions are relevant. Here we aim at obtaining only real solutions of $\mathcal{F}_{so}$ and we use the sketch to define a real homotopy (see [11, 9]) between $\mathcal{F}_{sk}$ and $\mathcal{F}_{so}$ using an interpolation of parameters defined as follows.

Definition 1 *Let $A^{sk} = \{a_1^{sk}, \ldots, a_m^{sk}\}$ and $A^{so} = \{a_1^{so}, \ldots, a_m^{so}\}$ be strictly positive real values for $\mathbf{A}$. We call interpolation function from A^{sk} to A^{so} a C^∞ function $a : \mathbb{R} \to \mathbb{R}^m$ that satisfies:*

- *for $1 \le i \le m, a_i(0) = a_i^{sk}$ and $a_i(1) = a_i^{so}$, and*

- *for $1 \le i \le m, \forall t \in [0,1], a_i(t) > 0$.*

We call positive support of an interpolation function a and we note it $supp_+(a)$ the subset of $\mathbb{R}$ where $a_i(t) \ge 0$ for $1 \le i \le m$.

Notice that $[0,1] \subset supp_+(a)$. Given an interpolation function a from A^{sk} to A^{so}, we define the homotopy system $\mathcal{H}$ as:

$$H(\mathbf{X}, \mathbf{t}) = F(\mathbf{X}, a(\mathbf{t})) = 0 \tag{$\mathcal{H}$}$$

where $H : \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^m$ is called the homotopy function associated to G.

The set of solutions of $\mathcal{H}$ denoted by $H^{-1}(0)$ can be partitioned into a set of connected components that are called *homotopy paths* of $\mathcal{H}$. It is worth mentioning here that a point (X, t) with real components belongs to a homotopy path of $\mathcal{H}$ only if $t \in supp_+(a)$, otherwise two points of X would be separated by a negative length. The following result (see [9]) underlines the influence of a and $supp_+(a)$ on the topology of homotopy paths of $\mathcal{H}$. It is here stated in the general case where constraints are not necessarily distances but also angles, collinearities and so on.

Theorem 1 ([9]) *Let H define the homotopy $\mathcal{H}$, J_H be its Jacobian matrix, $\mathcal{D}H \subseteq \mathbb{R}^m \times \mathbb{R}$ be its domain of definition, and $\mathcal{D}H^c$ be its complementary. Under assumptions*

(h0) $\mathcal{D}H$ is open,

(h1) J_H has full rank on each point of $H^{-1}(0)$,

(h2) $supp_+(a)$ is compact,

homotopy paths of $\mathcal{H}$ are 1-dimensional manifolds diffeomorphic either to circles, or to open intervals. If an homotopy path $\mathcal{S}$ is diffeomorphic to an open interval then the extremities of $\mathcal{S}$ converge either to a point in $\mathcal{D}H^c$ or to a solution with an infinite norm.

We make here two remarks to adapt this result in the framework of PDSP. First, H is clearly C^∞ in $\mathbb{R}^m \times \mathbb{R}$, hence paths cannot converge to a point of $\mathcal{D}H^c$. Secondly, if $supp_+(a)$ is compact, $a(supp_+(a))$ is compact and all components of $a(t)$ are bounded if $t \in supp_+(a)$. Hence if (X, t) belongs to an homotopy path of $\mathcal{H}$, the components of X are coordinates of a set of points lying in a compact. In a PDSP context, Thm. 1 can be restated as follows:

Corollary 1 *Let G be a PDSP and H define the homotopy $\mathcal{H}$ satisfying assumptions (h1) and (h2) of Thm. 1. Homotopy paths of $\mathcal{H}$ are 1-dimensional manifolds diffeomorphic to circles.*

Denoting by $\mathcal{S}$ the homotopy path to which belongs $(X^{sk}, 0)$, $\mathcal{S}$ can be followed with a numerical path tracker until it loops on $(X^{sk}, 0)$. It allows to find points (X, t) of $\mathcal{S}$ with $t = 1$ that are real solutions of $\mathcal{F}_{so}$.

Unknowns:
point $\mathbf{p}_1, ..., \mathbf{p}_6$
Parameters:
length $\mathbf{a}_1, ..., \mathbf{a}_8, \mathbf{k}$
Constraints:
$distance(\mathbf{p}_1, \mathbf{p}_2) = \mathbf{a}_1$
$distance(\mathbf{p}_2, \mathbf{p}_3) = \mathbf{a}_2$
$distance(\mathbf{p}_1, \mathbf{p}_3) = \mathbf{k}$
...
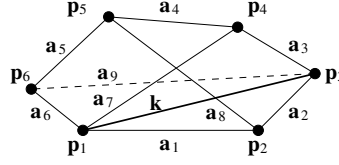$distance(\mathbf{p}_2, \mathbf{p}_5) = \mathbf{a}_8$

Figure 2: A symbolic statement (left) and a dimensioned sketch (right) of the PDSP $qK_{3,3}$.

2.3 Reparameterization

A PDSP can be solved very easily when its associated system $\mathcal{F}$ can be organized in a triangular form. From a geometric point of view, solving such a system is done by constructing points iteratively as intersections of two circles (three spheres in a 3D context) while making choices between possible intersections. The formal statement of the latter geometric construction is called a construction plan. When a PDSP G cannot be solved with this approach, an idea called reparameterization (see [7]) is to introduce d new constraints called *added constraints* with unknown parameters called *driving parameters*, in such a way that a construction plan parameterized by driving parameters constructs figures fulfilling all constraints but d that are called *removed constraints*. G is then solved by finding values of driving parameters such that the figures constructed by the construction plan satisfy the removed constraints.

2.3.1 Construction Plans

Consider the PDSP $qK_{3,3}$ depicted in Fig. 2 (q holds for "quasi") that has been obtained from $K_{3,3}$ by substituting the constraint $distance(\mathbf{p}_3, \mathbf{p}_6) = \mathbf{a}_9$ by $distance(\mathbf{p}_1, \mathbf{p}_3) = \mathbf{k}$. Knowing values for $\{\mathbf{a}_1, \ldots, \mathbf{a}_8, \mathbf{k}\}$, its solutions are all found by the simple ruler and compass construction given in the leftmost part of Fig. 3.

The instruction $\mathbf{p}_2 = InterCL(\mathbf{p}_1, \mathbf{a}_1, \mathbf{l}_1)$ holds for the construction of $\mathbf{p}_2$ as one of the intersections of the line $\mathbf{l}_1$ with a circle of center $\mathbf{p}_1$ and radius $\mathbf{a}_1$. Here $\mathbf{p}_1$ and $\mathbf{l}_1$ are objects of the reference that are fixed to construct solutions up to rigid motions. The instruction $\mathbf{p}_i = InterCC(\mathbf{p}_{i^1}, \mathbf{a}_{i^2}, \mathbf{p}_{i^3}, \mathbf{a}_{i^4})$ holds for the construction of $\mathbf{p}_i$ as one of the intersections of the circles respectively centered in $\mathbf{p}_{i^1}$ and $\mathbf{p}_{i^3}$ of radius $\mathbf{a}_{i^2}$ and $\mathbf{a}_{i^4}$.

We will note $I_i[\mathbf{p}_{i+1}, \mathbf{A}_i]$ the instruction that constructs $\mathbf{p}_{i+1}$ from objects $\mathbf{A}_i$ (after a possible re-indexing of points). Notice that objects of $\mathbf{A}_i$ are not only length parameters but also geometric objects of the reference or objects constructed by previous instructions.

Definition 2 (CP) *A Construction Plan (CP) of objects $\mathbf{P}$ and parameters $\mathbf{A}$ with reference $\mathbf{A}_0$ is a finite sequence $I = (I_i)_{i=1}^{l}$ of terms $I_i[\mathbf{p}_{i+1}, \mathbf{A}_i]$ in a triangular form,* i.e.

  (i) *each $\mathbf{p} \in \mathbf{P}$ is either in $\mathbf{A}_0$ or is constructed by an instruction $I_i \in I$,*

  (ii) *for each instruction $I_i[\mathbf{p}_{i+1}, \mathbf{A}_i] \in I$, each $\mathbf{a} \in \mathbf{A}_i$ is either in $\mathbf{A}_0$, or in $\mathbf{A}$, or is constructed by a term $I_j \in I$ with $j < i$.*

We note it $I[\mathbf{P}, \mathbf{A}, \mathbf{A}_0]$, or more simply I.

A CP can be seen as a symbolic solution of a set of constraints. Consider for instance an instruction $\mathbf{p}_{i+1} = InterCC(\mathbf{p}_{i^1}, \mathbf{a}_{i^2}, \mathbf{p}_{i^3}, \mathbf{a}_{i^4})$, it gives a symbolic solution to the constraints

Unknowns:
point $\mathbf{p}_2, ..., \mathbf{p}_6$
Parameters:
point $\mathbf{p}_1$, *line* $\mathbf{l}_1$
length $\mathbf{a}_1, ..., \mathbf{a}_8, \mathbf{k}$
Terms:
$\mathbf{p}_2 = InterCL(\mathbf{p}_1, \mathbf{a}_1, \mathbf{l}_1)$
$\mathbf{p}_3 = InterCC(\mathbf{p}_1, \mathbf{k}, \mathbf{p}_2, \mathbf{a}_2)$
$\mathbf{p}_4 = InterCC(\mathbf{p}_1, \mathbf{a}_7, \mathbf{p}_3, \mathbf{a}_3)$
$\mathbf{p}_5 = InterCC(\mathbf{p}_2, \mathbf{a}_8, \mathbf{p}_4, \mathbf{a}_4)$
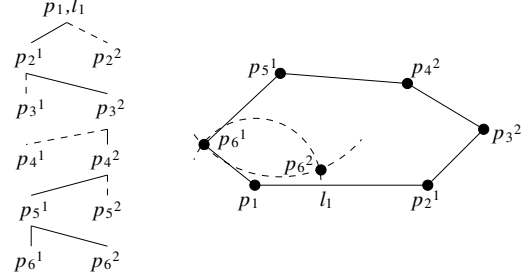$\mathbf{p}_6 = InterCC(\mathbf{p}_5, \mathbf{a}_5, \mathbf{p}_1, \mathbf{a}_6)$

Figure 3: Left: a construction plan of solutions of the PDSP $qK_{3,3}$. Middle: a sub-tree of the associated interpretation tree. Right: two figures obtained when evaluating the CP on the two branches in solid lines.

$distance(\mathbf{p}_{i+1}, \mathbf{p}_{i^1}) = \mathbf{a}_{i^2}$ and $distance(\mathbf{p}_{i+1}, \mathbf{p}_{i^3}) = \mathbf{a}_{i^4}$. We associate in such a way a set C_I of constraints with a CP I, and we say that $I[\mathbf{P}, \mathbf{A}, \mathbf{A}_0]$ is a CP of a PDSP $G = C[\mathbf{P}, \mathbf{A}]$ if $C_I = C$.

2.3.2   Evaluation of a Construction Plan

Given values A and A_0 for $\mathbf{A}$ and $\mathbf{A}_0$, a CP $I[\mathbf{P}, \mathbf{A}, \mathbf{A}_0]$ is evaluated to obtain numerical values of the solutions of constraints $C_I[\mathbf{P}, \mathbf{A}]$ by sequentially applying its instructions. At each step a choice between two intersections is done and considering all the possible intersections leads to construct an interpretation tree; its branches bring numerical values for $\mathbf{P}$. Middle part of Fig. 3 shows a sub-tree of the interpretation tree associated to the CP presented in the left part, and in its rightmost part it shows the two figures brought by the two branches in solid line.

Let $I_i[\mathbf{p}_{i+1}, \mathbf{A}_i]$ be an instruction of I. It is interpreted by a multi-function that maps to a value A_i of $\mathbf{A}_i$ the two possible intersection locii of objects defined by A_i. We index these locii by an integer and for a given index b_i we note $^{[b_i]}I_i(A_i)$ the function that maps to A_i the locus of index b_i. We consider that $^{[b_i]}I_i$ is not defined when the number of intersections is zero or infinite. We assume that indexation of intersections is continuous, *i.e* for each $1 \leq i \leq l$ and for each index b_i, $^{[b_i]}I_i(A_i)$ is C^∞ on the interior of its domain of definition.

We call *branch* of I a sequence $b = (b_i)_{i=1}^l$ of indexes and, for a given branch b, we call *evaluation of I on its branch* b the numerical function $^{[b]}I(\mathbf{A}_0, \mathbf{A})$ that maps to values (A_0, A) the composition of functions $^{[b_i]}I_i$. On a given branch b, $^{[b]}I$ is C^∞ in the interior of its domain of definition as a combination of C^∞ functions.

2.3.3   Reparameterized Construction Plans

Reparameterized construction plans (RCP) are central objects in the method presented in this paper. They appear in two steps. First, each problem must be derived in a CP. We are interested in this article in problems whose construction is not known, it is then necessary to transform the problem by adding and removing constraints as explained above. Constraints are added so that a CP is easy to establish. We do not detail here the way a RCP with a sole driving parameter in each instruction is obtained, see [7], [14], [4] for different approaches. This new CP is called a RCP and is completely characterized by: the CP itself, the removed constraints that must be satisfied by all solutions and the driving parameters that are the added dimensions. For instance, a RCP for $K_{3,3}$ could be given by the CP I given in fig. 3, the removed constraint $dist(\mathbf{p}_3, \mathbf{p}_6) = \mathbf{a}_9$ and the driving parameter $\mathbf{k}$. Secondly, RCP take also place during the homotopy process.

For stability reasons, some distances are removed and others are added on-the-fly according to numerical considerations. The elements of the reference of the CP are also modified during the homotopy process and in the following definition we put forward the reference as one of the characterizing elements of a RCP:

Definition 3 (RCP) *Let $G = C[\mathbf{P}, \mathbf{A}]$ a PDSP. A RCP R of G is a quadruplet $(I, C_-, \mathbf{A}_+, \mathbf{A}_0)$, where:*

- *C_- is a subset of C involving parameters $\mathbf{A}_-$,*

- *$\mathbf{A}_+$ is a set of parameters called driving parameters,*

- *$I[\mathbf{P}, \mathbf{A}', \mathbf{A}_0]$ is the CP of $C_I[\mathbf{P}, \mathbf{A}']$,*

- *$\mathbf{A}' = \mathbf{A} \setminus \mathbf{A}_- \cup \mathbf{A}_+$,*

- *$C_I \setminus C$ are distance constraints called added constraints,*

and $(I, C_-, \mathbf{A}_+, \mathbf{A}_0)$ is such that $C_- = C \setminus C_I$.

In a RCP, the situation where a point is the intersection of two added constraints could not occur. This would create a new point which is not given in the initial statement. So, we will consider RCP that meet the following conditions: for each circle-circle intersection $\mathbf{p}_{i+1} = interCC(\mathbf{p}_{i^1}, \mathbf{a}_{i^2}, \mathbf{p}_{i^3}, \mathbf{a}_{i^4})$, (i) $\mathbf{a}_{i^2}$ is not a driving parameter (*i.e.* $\mathbf{a}_{i^2} \in \mathbf{A}$), and (ii) if $\mathbf{a}_{i^4}$ is a driving parameter (*i.e.* $\mathbf{a}_{i^4} \in \mathbf{A}_+$), $\mathbf{p}_{i^3}$ is a reference point (*i.e.* $\mathbf{p}_{i^3} \in \mathbf{A}_0$) and is not used as a reference for another instruction.

Given a RCP with a sole driving parameter in each instruction, it can always be modified to meet the conditions (i) and (ii) as follows: (i) is satisfied by rewriting instructions, and (ii) is satisfied by creating a new reference point for each driving parameter. A RCP for $K_{3,3}$ satisfying (i) and (ii) is $R' = (I', \{dist(\mathbf{p}_3, \mathbf{p}_6) = \mathbf{a}_9\}, \{\mathbf{k}\}, \{\mathbf{p}_1, \mathbf{l}_1, \mathbf{p}'_1\})$ where I' is obtained from I by substituting $\mathbf{p}_3 = InterCC(\mathbf{p}_1, \mathbf{k}, \mathbf{p}_2, \mathbf{a}_2)$ by $\mathbf{p}_3 = InterCC(\mathbf{p}_2, \mathbf{a}_2, \mathbf{p}'_1, \mathbf{k})$.

We focus here on the numerical step of the reparameterization method, that consists in finding values for $\mathbf{A}_+$ such that figures constructed by I fulfill constraints of C_-.

Let $R = (I, C_-, \mathbf{A}_+, \mathbf{A}_0)$ be a RCP of $G = C[\mathbf{P}, \mathbf{A}]$, $\mathbf{A}_-$ be the set of parameters of C_- and $\mathbf{A}' = (\mathbf{A} \setminus \mathbf{A}_-) \uplus \mathbf{A}_+$, where $\uplus$ holds for a disjoint union. Given a branch b, we recall that $^{[b]}I(\mathbf{A}_0, (\mathbf{A} \setminus \mathbf{A}_-) \uplus \mathbf{A}_+)$ is the evaluation of I on b. Both for the sake of readability and to make appear the different roles played by driving parameters and other parameters, we will note it $^{[b]}I(\mathbf{A}_0, \mathbf{A} \uplus \mathbf{A}_+)$ even if elements of $\mathbf{A}_-$ are not involved in I. When values A_0 for $\mathbf{A}_0$ are explicitly fixed, we will note $^{[b]}I(\mathbf{A} \uplus \mathbf{A}_+)$ for $^{[b]}I(A_0, \mathbf{A} \uplus \mathbf{A}_+)$.

Let $C_- = \{c_1^-, \ldots, c_d^-\}$. We associate with the RCP $R = (I, C_-, \mathbf{A}_+, \mathbf{A}_0)$ the numerical functions

$$
\begin{aligned}
^{[b]}R: \quad \mathbb{R}^d \times \mathbb{R}^m \quad &\to \quad \mathbb{R}^d \\
\mathbf{A}^+, \mathbf{A} \quad &\mapsto \quad \begin{pmatrix} c_1^-(^{[b]}I(\mathbf{A} \uplus \mathbf{A}_+), \mathbf{A}) \\ \ldots \\ c_d^-(^{[b]}I(\mathbf{A} \uplus \mathbf{A}_+), \mathbf{A}) \end{pmatrix}
\end{aligned}
\tag{2}
$$

where b is a branch of I and numerical interpretations c_i^- are defined as in Eq. (1). Since numerical interpretations c_i^- are C^∞, functions $^{[b]}R$ are C^∞ on the interiors of domains of definition of functions $^{[b]}I$.

Given values A^{so} for $\mathbf{A}$, there is a one to one correspondence between real solutions of $^{[b]}R(\mathbf{A}^+, A^{so}) = 0$ for all branches b and real solutions of $\mathcal{F}_{so}$.

3   Leading Homotopy by Reparameterization

We aim at finding solutions of $\mathcal{F}_{so}$ lying on the path $\mathcal{S}$ of $\mathcal{H}$ to which belongs the sketch. Instead of using a path tracker to follow $\mathcal{S}$ in $\mathbb{R}^m \times \mathbb{R}$, we propose to compute it indirectly by following a sequence of paths defined by homotopy functions constructed with a RCP. These paths are tracked in $\mathbb{R}^d \times \mathbb{R}$ where d is the number of driving parameters of the RCP with $d << m$, what makes cheaper the path tracking. We give here a justification of this approach by showing that such paths are diffeomorphic to connected subsets of $\mathcal{S}$.

Sec. 3.1 enumerates assumptions that are required to make our approach valid. We define in Sec. 3.2 homotopy functions using RCP and characterize their domains of definition and boundary configurations in Sec. 3.3. We establish the link between their paths and $\mathcal{S}$ in Sec. 3.4.

3.1   Assumptions

Let $G = C[\mathbf{P}, \mathbf{A}]$ be a PDSP, and H the homotopy function with interpolation function a. Let $R = (I, C_-, \mathbf{A}_+, \mathbf{A}_0)$ be a RCP of G, where $I = (I_i)_{i=1}^l$, I_1 is the instruction $\mathbf{p}_2 = interCL(\mathbf{p}_1, \mathbf{a}_1, \mathbf{l}_1)$ and for $i \geq 2$, I_i is the instruction $\mathbf{p}_{i+1} = interCC(\mathbf{p}_{i^1}, \mathbf{a}_{i^2}, \mathbf{p}_{i^3}, \mathbf{a}_{i^4})$, where $\mathbf{a}_{i^4}$ is either in $\mathbf{A}$ or in $\mathbf{A}_+$. Up to a re-indexing, the component a_i of a interpolates values of $\mathbf{a}_i \in \mathbf{A}$.

The method presented here is valid under the following hypothesis on the interpolation function a and the Jacobian matrix J_H of H.

(h1) J_H has full rank on each point of $H^{-1}(0)$,

(h2) $supp_+(a)$ is compact,

(h3) $\forall i, j$, $a_i(t) = a_j(t)$ has a finite number of solutions,

(h4) if $t \in supp_+(a)$, $a_1(t) > 0$,

(h5) if $i \geq 2$ and I_i is s.t. $a_{i^4} \in \mathbf{A}$, then $a_{i^2}(t)a_{i^4}(t) = 0 \Rightarrow a_{i^2}(t) \neq 0$ or $a_{i^4}(t) \neq 0$ for $t \in supp_+(a)$,

(h6) if $i \geq 2$ and I_i is s.t. $a_{i^4} \in \mathbf{A}_+$, then $a_{i^2}(t) > 0$ for $t \in supp_+(a)$.

In what follows, some instructions of the RCP will be changed in such a way that *(h6)* is satisfied only on a subset of $supp_+(a)$, and we will say that *(h6)* is satisfied on a given subset U of $supp_+(a)$ if *(h6)* holds for each t in U.

Let us explain these hypothesis. *(h1)* and *(h2)* are the hypothesis of Cor. 1 and they guarantee that paths of $\mathcal{H}$ are diffeomorphic to circles. When *(h1)* is satisfied, each point of $H^{-1}(0)$ admits a tangent and paths can be tracked with a numerical path-tracker. Here $H(\mathbf{X}, \mathbf{t}) = F(\mathbf{X}, a(\mathbf{t}))$ hence *(h1)* is strongly related to the rank of J_F, the Jacobian matrix of H. Since G is generically well-constrained, J_F has full rank on each point of $F^{-1}(0)$ for generic values of $\mathbf{A}$ (see Subsec. 2.1). Verifying that G is generically well-constrained and characterizing the interpolation function a to satisfy *(h1)* are both challenging problems and are beyond the scope of this paper. [12] justifies real homotopies thanks to the theorem of Sard. *(h2)* ensures that values t such that $H(\mathbf{X}, t) = 0$ has real solutions are in a compact interval (see Subsec. 2.2). *(h2)* can be satisfied by setting a component a_i of a to $-ct^2 + (a_i^{so} - a_i^{sk} + c)t + a_i^{sk}$ with $c > 0$, and a_j for $j \neq i$ to linear interpolations.

Beside its influence on the topology of homotopy paths, a has an impact on the geometric configurations of the figures encountered in such paths. Here, we are using RCP to build numerical functions to track homotopy paths of H. The obtained functions are not defined in the

whole space of parameters, and the borders of their domains of definition are characterized in terms of geometric configurations. *(h4), (h5)* and *(h6)* restrain the geometrical configurations a path can pass trough. Configurations that can be encountered are detailed in Subsec. 3.3. *(h3)* is used to prove the termination of our algorithm.

3.2 R-reduced Homotopy Functions

Let $R = (I, C_-, \mathbf{A}_+, \mathbf{A}_0)$ be a RCP and suppose values A_0 for $\mathbf{A}_0$ are fixed. We call *R-reduced homotopy* the homotopies defined by

$$^{[b]}H_R(\mathbf{A}^+, \mathbf{t}) = {}^{[b]}R(\mathbf{A}^+, a(\mathbf{t})) = 0 \qquad (\mathcal{H}_R)$$

where b is a branch of R. The functions $^{[b]}H_R : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$ are called *R-reduced homotopy functions* and the connected components of solutions of $\mathcal{H}_R$ are called *R-reduced paths*.

Considering Eq. (2), the i-th component of $^{[b]}H_R$ is $c_i^-(^{[b]}I(a(\mathbf{t}) \uplus \mathbf{A}^+), a(\mathbf{t}))$. Noting $\mathcal{D}^{[b]}I \subset \mathbb{R}^d \times \mathbb{R}$ the domain of definition of the function defined as

$$\mathbf{A}^+, \mathbf{t} \mapsto {}^{[b]}I(a(\mathbf{t}) \uplus \mathbf{A}^+)$$

we state that the domain of definition of $^{[b]}H_R$ is $\mathcal{D}^{[b]}I$, and that $^{[b]}H_R$ is C^∞ in the interior of $\mathcal{D}^{[b]}I$.

We are now interested in characterizing sets $\mathcal{D}^{[b]}I$ and their borders in terms of geometric configurations of objects constructed by I. A point $(A^+, t) \in \mathbb{R}^d \times \mathbb{R}$ belongs to the border of $\mathcal{D}^{[b]}I$ if its neighborhoods contain points of $\mathcal{D}^{[b]}I$ and points where $^{[b]}I$ is not defined. In general, $\mathcal{D}^{[b]}I$ is neither open nor close, and contains only a possibly empty subset of its border. We will call *boundary* of $\mathcal{D}^{[b]}I$ the subset of the border of $\mathcal{D}^{[b]}I$ that is in $\mathcal{D}^{[b]}I$.

3.3 Domain of Definition and Boundary Configurations

Since $^{[b]}I$ is the combination of functions $^{[b_i]}I_i$ we first characterize the domain of definition of the latter functions.

Let $\mathbf{p}_2 = interCL(\mathbf{p}_1, \mathbf{a}_1, \mathbf{l}_1)$ be the instruction I_1. Since $\mathbf{p}_1, \mathbf{l}_1$ are part of the reference their values p_1, l_1 are fixed s.t. $p_1 \in l_1$. Hence $^{[b_1]}I_1$ maps to the value a_1 of $\mathbf{a}_1$ one of the intersections p_2, p_2' of l_1 with a circle of radius a_1 which center belongs to l_1 (see the configuration (c1) on fig. 4). When $a_1 > 0$, there is an open neighborhood of a_1 where $^{[b_1]}I_1$ is C^∞. Here we have $a_1 = a_1(t)$ and from assumption *(h4)*, $a_1(t) > 0$ on $supp_+(a)$.

Let $i \geq 2$ and $\mathbf{p}_{i+1} = interCC(\mathbf{p}_{i^1}, \mathbf{a}_{i^2}, \mathbf{p}_{i^3}, \mathbf{a}_{i^4})$ be the instruction I_i of I. $^{[b_i]}I_i$ maps to $(p_{i^1}, a_{i^2}, p_{i^3}, a_{i^4})$ one of the two intersections p, p' of two circles. We make a disjunction on the number of intersections of the two circles.

When the two circles are disjoint or coincident (with non zero radius), $^{[b]}I$ is not defined. When the two circles have exactly two intersections, $^{[b_i]}I_i$ is clearly C^∞ (see the configuration (c2) on fig. 4).

Two configurations can lead the two circles to have exactly one intersection. The first one is when the latter circles are concentric with null radii, and $a_{i^2} = a_{i^4} = 0$. Recall that either $a_{i^4} \in \mathbf{A}$ or $a_{i^4} \in \mathbf{A}_+$. Suppose first $a_{i^4} \in \mathbf{A}$. Hence $a_{i^2} = a_{i^2}(t)$ and $a_{i^4} = a_{i^4}(t)$, and assumption *(h5)* forbids the situation $a_{i^2}(t) = a_{i^4}(t) = 0$ when $t \in supp_+(a)$. Suppose now $a_{i^4} \in \mathbf{A}_+$. Hence $a_{i^2} = a_{i^2}(t)$ and assumption *(h6)* forbids the situation $a_{i^2}(t) = 0$ while t is in a subset for which it holds. As a consequence, $a_{i^2} = a_{i^4} = 0$ does not happen when *(h5)* and *(h6)* hold.

The second configuration is when the two circles are tangent, and at least one circle has a strictly positive radius. The two centers of circles and their intersection are collinear (see the
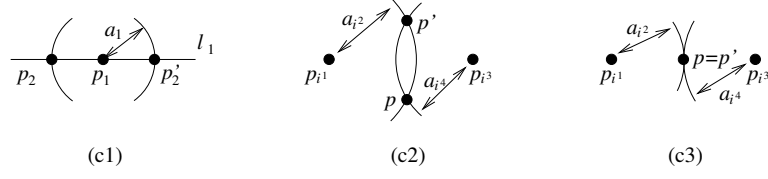
(c1)   (c2)   (c3)

Figure 4: Three geometric configurations for $InterCL$ and $InterCC$ instructions.

configuration (c3) on fig. 4). Clearly, this configuration characterizes the boundary of the domain of definition of the mapping $^{[b]}I(\mathbf{p}_{i^1}, \mathbf{a}_{i^2}, \mathbf{p}_{i^3}, \mathbf{a}_{i^4})$ and is called a *boundary configuration* of I_i.

Since I is the combination of its instructions, a point (A_+, t) is in the boundary of $\mathcal{D}^{[b]}I$ only if a boundary configuration holds for $(A_+, a(t))$ for at least one instruction I_i with $i \geq 2$. We will say in this case that the figure $^{[b]}I(a(t) \uplus A_+)$ presents a boundary configuration of I or R, or that (A_+, t) leads to a boundary configuration of I or R.

3.4   R-Reduced Paths

The point here is to characterize R-reduced paths, and to link them with paths of $\mathcal{H}$. To achieve this, let us define the mappings $^{[b]}\varphi : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^m \times \mathbb{R}$ as

$$^{[b]}\varphi(\mathbf{A}_+, \mathbf{t}) = \begin{pmatrix} ^{[b]}I(a(\mathbf{t}) \uplus \mathbf{A}_+) \\ \mathbf{t} \end{pmatrix} \tag{3}$$

and $\varphi' : \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}^d \times \mathbb{R}$ as

$$\varphi'(\mathbf{X}, \mathbf{t}) = \begin{pmatrix} c_1^+(\mathbf{X}) \\ \dots \\ c_d^+(\mathbf{X}) \\ \mathbf{t} \end{pmatrix} \tag{4}$$

where $c_1^+, \dots, c_d^+$ are the d added constraints of R. $^{[b]}\varphi$ are C^∞ on $\mathcal{D}^{[b]}I$ and φ' is C^∞ on $\mathbb{R}^m \times \mathbb{R}$. Consider the following remark, that is a consequence of the characterization of a boundary configuration.

Remark 1 *Let (X, t) be a point of $H^{-1}(0)$ and R a RCP. X does not present any boundary configuration if and only if it exists a unique branch b and a unique point $(A_+, t) = \varphi'(X, t)$ s.t. $(X, t) = {}^{[b]}\varphi(A_+, t)$. X presents a boundary configuration if and only if there exist at least two branches $b1$ and $b2$ and a unique point $(A_+, t) = \varphi'(X, t)$ s.t. $(X, t) = {}^{[b1]}\varphi(A_+, t) = {}^{[b2]}\varphi(A_+, t)$.*

Let us justify Rem. 1. If (X, t) is a point of $H^{-1}(0)$, $(A_+, t) = \varphi'(X, t)$ is unique by definition of φ'. Suppose X does not present any boundary configurations: each point of X is constructed by R by intersecting two circles having two different intersections. If $b1 \neq b2$ are two branches of R, they correspond to different choices of intersections and $^{[b1]}\varphi(A_+, t) \neq {}^{[b2]}\varphi(A_+, t)$. If X has a boundary configuration, at least one point $\mathbf{p}$ of $\mathbf{X}$ is the intersection of two circles in configuration (c3). The two branches $b1 \neq b2$ corresponding to the same choices for each point but for $\mathbf{p}$ are such that $(X, t) = {}^{[b1]}\varphi(A_+, t) = {}^{[b2]}\varphi(A_+, t)$.

We show now that R-reduced paths are locally diffeomorphic to paths of $\mathcal{H}$, then we extend the latter diffeomorphism to a global diffeomorphism between pieces of R-reduced paths and pieces of paths of $\mathcal{H}$.

Lemma 1 *Let (A^+, t) be s.t. $^{[b]}H_R(A^+, t) = 0$, and $X = {}^{[b]}I(a(t) \uplus A^+)$. If (A^+, t) does not lead to a boundary configuration of R, it exists a neighborhood $\mathcal{U}$ of (A^+, t) and a neighborhood $\mathcal{V}$ of (X, t) such that $^{[b]}H_R^{-1}(0) \cap \mathcal{U}$ is diffeomorphic to $H^{-1}(0) \cap \mathcal{V}$.*

Proof of Lem. 1: Let $X = {}^{[b]}I(a(t) \uplus A^+)$. Then $^{[b]}H_R(A^+, t) = 0 \Rightarrow H(X, t) = 0$, and it exists a homotopy path $\mathcal{S}$ of $\mathcal{H}$ to which belongs (X, t). Let $\mathcal{H}_-$ be the system of equations having all the equations of $\mathcal{H}$ but the ones corresponding to constraints of C_-, and $\mathcal{S}_-$ be the set of its solutions. $\mathcal{S}_-$ is a $d + 1$-dimensional smooth manifold, and $(X, t) \in \mathcal{S}_-$ and $\mathcal{S} \subseteq \mathcal{S}_-$ hold. As a consequence, $\mathcal{S}$ is a 1-dimensional smooth submanifold of $\mathcal{S}_-$, and in any open neighborhood of (X, t) in $\mathcal{S}_-$, $\mathcal{S}$ is a 1-dimensional smooth manifold.

Since (A^+, t) does not lead to a boundary configuration, $^{[b]}\varphi$ is C^∞ on an open neighborhood $\mathcal{U} \subset \mathbb{R}^d \times \mathbb{R}$ of (A_+, t). Let $\mathcal{V} = {}^{[b]}\varphi(\mathcal{U})$. Clearly $\mathcal{V} \subset \mathcal{S}_-$ and $(X, t) \in \mathcal{V}$ hold. We show that $\mathcal{V}$ is an open neighborhood of (X, t) in $\mathcal{S}_-$. Let $\varphi'_{|\mathcal{S}_-} : \mathcal{S}_- \to \mathbb{R}^d \times \mathbb{R}$ be the restriction of φ' to $\mathcal{S}_-$. φ' is C^∞ on $\mathcal{S}_-$ and $\varphi'(\mathcal{V}) = \mathcal{U}$. Hence $\mathcal{V}$ is an open neighborhood of (X, t) in $\mathcal{S}_-$ as the inverse image of an open neighborhood.

We finish the proof by remarking that the mapping $^{[b]}\varphi_{|\mathcal{U}} : \mathcal{U} \to \mathcal{V}$, which inverse is $\varphi'_{|\mathcal{S}_-}$, is a diffeomorphism. $\square$

The following Prop. extends the property of Lem. 1 to a global property.

Proposition 1 *Let $\mathcal{S}$ be a homotopy path of $\mathcal{H}$ and R be a RCP. Let $\mathcal{S}' \subseteq \mathcal{S}$ be a connected subset of $\mathcal{S}$ that does not contain any figure with a boundary configuration of R. Then it exists a unique branch b, a unique R-reduced path $^{[b]}\mathcal{S}$ and a subset $^{[b]}\mathcal{S}' \subseteq {}^{[b]}\mathcal{S}$ such that $^{[b]}\mathcal{S}'$ is diffeomorphic to $\mathcal{S}'$ by $^{[b]}\varphi$.*

Proof of Prop. 1: Let (X, t) be a point of $\mathcal{S}'$. Since X does not present a boundary configuration of R, it exists (see Rem. 1) a unique branch b and a unique point $(A_+, t) = \varphi'(X, t)$ such that $^{[b]}\varphi(A_+, t) = (X, t)$. Hence $^{[b]}H_R(A_+, t) = 0$ and (A_+, t) belongs to an homotopy path $^{[b]}\mathcal{S}$ of $^{[b]}H_R^{-1}(0)$.

Since $\mathcal{S}'$ is connected and φ' is C^∞, $\varphi'(\mathcal{S}')$ is a connected subset of $\mathbb{R}^d \times \mathbb{R}$. Moreover, $\varphi'(\mathcal{S}')$ belongs to the interior of $\mathcal{D}^{[b]}I$ otherwise a point of $\varphi'(\mathcal{S}')$ would belong to the boundary of $\mathcal{D}^{[b]}I$ and would lead to a boundary configuration of R. As a consequence, $^{[b]}\varphi$ is well defined on $\varphi'(\mathcal{S}')$.

We show now that $\varphi'(\mathcal{S}') \subseteq {}^{[b]}\mathcal{S}$. If it is not the case, it exists at least another branch $b2$ and another R-reduced path $^{[b2]}\mathcal{S}$ with $\varphi'(\mathcal{S}') \subseteq ({}^{[b]}\mathcal{S} \cup {}^{[b2]}\mathcal{S})$, and $^{[b]}\mathcal{S} \cap {}^{[b2]}\mathcal{S} \neq \emptyset$ holds from the local diffeomorphism property. Let $(A'_+, t') \in ({}^{[b]}\mathcal{S} \cap {}^{[b2]}\mathcal{S})$. Then (A'_+, t') is such that $(X', t') = {}^{[b]}\varphi(A'_+, t') = {}^{[b2]}\varphi(A'_+, t')$ and $(X', t') \in \mathcal{S}'$ presents a boundary configuration of R from Rem. 1.

Let us defined $^{[b]}\mathcal{S}'$ as $\varphi'(\mathcal{S}')$. It is a connected subset of $^{[b]}\mathcal{S}$, and $^{[b]}\varphi({}^{[b]}\mathcal{S}') = \mathcal{S}'$. Finally $^{[b]}\varphi$ is injective and is a global diffeomorphism from $^{[b]}\mathcal{S}'$ to $\mathcal{S}'$. $\square$

Prop. 1 states that it is possible to compute a part of $\mathcal{S}$ that does not contain any boundary configurations by following a path of $^{[b]}H_R^{-1}(0)$.

Now, assuming that the figures of $\mathcal{S}$ presenting a boundary configuration of R are in a finite number (boundary configurations of type (c3) can be each described by a polynomial equation, hence figures presenting a boundary configuration of R can be seen as the solutions of systems of $m + 1$ polynomials involving $m + 1$ variables) the set $\mathcal{S}$ can be written $\mathcal{S}^1 \cup (X^1, t^1) \cup \mathcal{S}^2 \cup \ldots \cup \mathcal{S}^n \cup (X^n, t^n)$, where (X^i, t^i) are such that X^i presents a boundary configuration of I, and $\mathcal{S}^i \subseteq \mathcal{S}$ are connected, pairwise disjoint and does not contain any figure presenting a boundary configuration. From Prop. 1 each path $\mathcal{S}^i$ can be computed by following a path $^{[bi]}\mathcal{S}$ of $^{[bi]}H_R^{-1}(0)$,
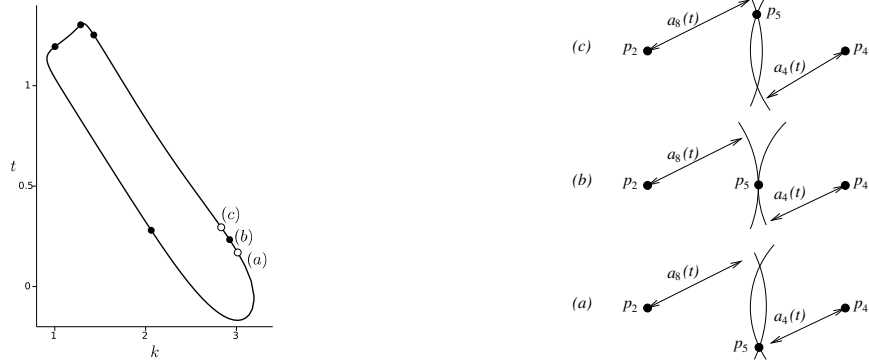
Figure 5: Left: the sequence of R-reduced paths corresponding to a path $\mathcal{S}$ of $\mathcal{H}$ for $K_{3,3}$. Black circles mark points leading to a a boundary configuration. Right: geometric configurations corresponding to points (a), (b), (c).

hence $\mathcal{S}$ can be computed by following the sequence of paths $^{[bi]}\mathcal{S}$ and making the appropriate branch changing in points (X^i, t^i).

The left part of fig. 5 shows the sequence of R-reduced paths $^{[bi]}\mathcal{S}$ corresponding to a path of $\mathcal{H}$ for the PDSP $K_{3,3}$. Its right part shows geometric configurations near a point leading to a boundary configuration.

4   On-The-Fly Change of the RCP

The method roughly depicted above hides a pitfall: it leads to follow R-reduced paths until a boundary configuration is reached. But R-reduced paths are numerically bad behaving near boundary configurations.

It can be easily seen by considering a function that maps to a positive real number k the positive y-coordinate of the intersections of the unit circle with a circle centered in $(0, 2)$ of radius equal to k. Noting y this function, we have $y(k) = \dfrac{\sqrt{-k^4 + 10k^2 - 9}}{4}$ and $y'(k) = \dfrac{-4k^3 + 20k}{8\sqrt{-k^4 + 10k^2 - 9}}$. The domain of definition of y is $[1, 3]$ which boundaries $1, 3$ corresponds to configurations where the circles are tangent; we have $lim_{k \to \{1,3\}} y'(k) = +\infty$, and it is not due to the chosen system of coordinates.

Such unbounded values of derivatives highly affect the efficiency of a numerical path tracking, that proceeds by approximating a path by its tangent.

We propose here to introduce a measure of the distance from a figure of $\mathcal{S}$, or from a point of a R-reduced path, to a boundary configuration, and to stop the tracking process of a R-reduced path when this distance is smaller than a real parameter α. Then we change either the RCP or the values of its references in a way that the distance to a boundary configuration is greater than α. The new R-reduced homotopy path is then followed to compute the path $\mathcal{S}$. This process is repeated each time the distance to a boundary configuration is smaller than α.

The distance to a boundary configuration is defined in Sec. 4.2, and our algorithm to change on-the-fly the RCP is described in Sec. 4.3. We first give an intuition of our approach on the example $K_{3,3}$ in Sec. 4.1.
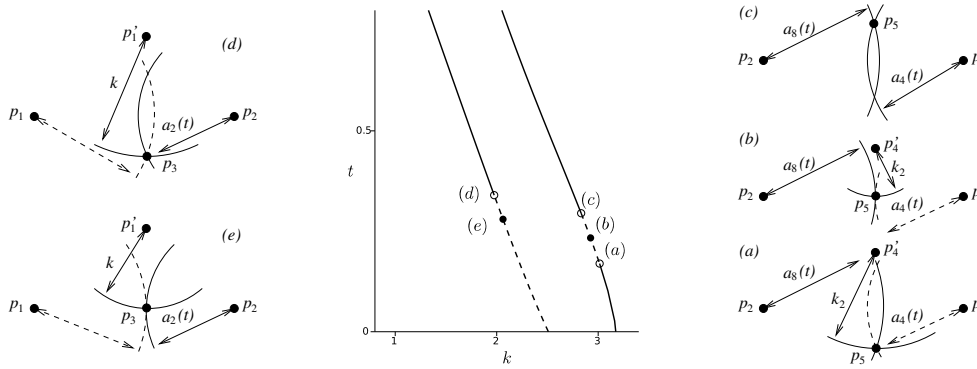
Figure 6: Center: a zoomed view of a sequence of R-reduced paths for $K_{3,3}$. Pieces in solid lines are followed by our algorithm. Black circles mark points leading to a a boundary configuration. Left and Right: geometric configurations corresponding to points $(a), \ldots (d)$. Dashed double arrows correspond to constraints removed by our algorithm.

4.1 Overview of our Method on an Example

The middle part of Fig. 6 shows a zoomed view of pieces of R-reduced paths (R is given in Sec. 2.3.3) for $K_{3,3}$. When following this union of paths from the sketch, the point (a) is first reached. The points p_5, p_4, p_2 in the figure X^a constructed in (a) is shown in the right part of Fig. 6. The two circles constructed when evaluating the instruction $\mathbf{p}_5 = interCC(\mathbf{p}_2, \mathbf{a}_8, \mathbf{p}_4, \mathbf{a}_4)$ are almost tangent, and X^a is "too close" to a boundary configuration (this notion will be detailed in Sec. 4.2).

To avoid the point (b) that leads to a boundary configuration, a new driving parameter and a new reference point is added to the RCP: the instruction $\mathbf{p}_5 = interCC(\mathbf{p}_2, \mathbf{a}_8, \mathbf{p}_4, \mathbf{a}_4)$ is replaced by $\mathbf{p}_5 = interCC(\mathbf{p}_2, \mathbf{a}_8, \mathbf{p}'_4, \mathbf{k}_2)$, where $\mathbf{p}'_4$ is a new reference point and $\mathbf{k}_2$ is a new driving parameter. The constraint $distance(\mathbf{p}_4, \mathbf{p}_5) = \mathbf{a}_4$ is added to the set of removed constraints to guarantee that it is fulfilled by constructed figures. For an appropriated placement of $\mathbf{p}'_4$, presented in Sec. 4.3.3, the figure constructed by the CP is not too close to a boundary configuration. The new RCP defines new paths, that can be followed while avoiding the boundary configuration of point (b) (see right part of Fig. 6). When reaching the point (c), the original instruction $\mathbf{p}_5 = interCC(\mathbf{p}_2, \mathbf{a}_8, \mathbf{p}_4, \mathbf{a}_4)$ can be restored while staying "far away" from a boundary configuration. The piece of path between (a) and (c) in the middle part of Fig. 6 is drawn in dashed line to underline that it is a projection of the path that is followed with our algorithm.

Suppose now that the point (d) (see central part of Fig. 6) is reached. The constructed figure is too close to a boundary configuration of the instruction $\mathbf{p}_3 = interCC(\mathbf{p}_2, \mathbf{a}_2, \mathbf{p}'_1, \mathbf{k})$ where $\mathbf{k}$ is a driving parameter. In that case, the point $\mathbf{p}'_1$ is moved in order to avoid the boundary configuration of point (e) (see left part of Fig. 6). The piece of path after (d) is drawn in dashed line to figure out that it is no longer the path that is followed by our algorithm.

4.2 Distance to a Boundary Configuration

Let I_i be $\mathbf{p}_{i+1} = interCC(\mathbf{p}_{i^1}, \mathbf{a}_{i^2}, \mathbf{p}_{i^3}, \mathbf{a}_{i^4})$. We associate with I_i the real function γ_i taking its values in $[0, 1]$:

$$\gamma_i(\mathbf{X}) = \frac{dist(\mathbf{p}_{i+1}, (\mathbf{p}_{i^1}, \mathbf{p}_{i^3}))}{max(\mathbf{p}_{i+1}\mathbf{p}_{i^1}, \mathbf{p}_{i+1}\mathbf{p}_{i^3})}$$

Algorithm 1 Path tracking with RCP swapping

Input: A RCP $R = (I, C_-, \mathbf{A}_+, \mathbf{A}_0)$, an interpolation function a, a sketch X^{sk}, α.
Output: a list of solutions $\mathcal{L}_{sol}$

1: Let $t = 0$, $X = X^{sk}$
2: Find A_+, A_0, b s.t. $^{[b]}I(A_0, a(t) \uplus A_+) = X$, fix $\mathbf{A}_0$ to A_0
3: **while** True **do**
4:      Follow the path $^{[b]}\mathcal{S} \subset {}^{[b]}H_R^{-1}(0)$ from (A_+, t) **while checking**
5:      **if** $^{[b]}\mathcal{S}$ passes trough the sketch **then**
6:          **return** $\mathcal{L}_{sol}$
7:      **else if** $^{[b]}\mathcal{S}$ passes trough hyperplane $\mathbf{t} = 1$ **then**
8:          Append current figure to $\mathcal{L}_{sol}$
9:          Set (A_+, t) to current point
10:      **else if** current point $(A_+^{cur}, t^{cur}) \in {}^{[b]}\mathcal{S}$ satisfies (sc1) **or** $^{[b]}\gamma_I(A_+^{cur}, t^{cur}) \leq \alpha$ **then**
11:          Let $X = {}^{[b]}I(a(t^{cur}) \uplus A_+^{cur})$
12:          Apply Algo. 2 to change $R = (I, C_-, \mathbf{A}_+, \mathbf{A}_0)$, and obtain A_+, A_0
13:          Find b s.t. $^{[b]}I(A_0, a(t) \uplus A_+) = X$, fix $\mathbf{A}_0$ to A_0

---

where $dist(\mathbf{p}_{i+1}, (\mathbf{p}_{i^1}, \mathbf{p}_{i^3}))$ is the distance between $\mathbf{p}_{i+1}$ and the line $(\mathbf{p}_{i^1}, \mathbf{p}_{i^3})$.

γ_i is defined and continuous on X when points $\{p_i, p_{i^1}, p_{i^3}\} \subset X$ are not coincident. Notice it never happens when I_i is an instruction of a RCP for which assumptions *(h5)* and *(h6)* hold. γ_i vanishes only on figures presenting a boundary configuration of I_i.

We associate to $(I_i)_{i=1}^l$ the function γ_I that measures the distance of a figure X to a boundary configuration of I defined as

$$\gamma_I(\mathbf{X}) := min_{2 \leq i \leq l} \gamma_i(\mathbf{X}).$$

γ_I is defined and continuous at least when assumptions *(h5)* and *(h6)* hold, and vanishes on figures presenting a boundary configuration of I. In the following, we will note $^{[b]}\gamma_I(\mathbf{A}_+, \mathbf{t})$ for $\gamma_I(^{[b]}I(a(\mathbf{t}) \uplus \mathbf{A}_+))$.

4.3 Path Tracking with On-The-Fly Change of the RCP.

Algo. 1 describes the main process of our method. We consider a PDSP $G = C[\mathbf{X}, \mathbf{A}]$, a RCP R, a sketch X^{sk} and an interpolation function a from A^{sk} to A^{so}. We assume that assumptions *(h1)* to *(h6)* are satisfied for a and R.

In the step 2, value A_0 for elements of $\mathbf{A}_0$ are read on X and fixed. Values A_+ for $\mathbf{A}_+$ are obtained by evaluating $\varphi'(X, t)$. In steps 2 and 13, the branch b is found by evaluating one by one instructions I_i on each branch, and keeping for each b_i the choice that leads to construct X.

When entering in the step 4, a point (X, t) of the path $\mathcal{S}$ of $\mathcal{H}$ to which belongs the sketch is known as well as a point (A_+, t) with $^{[b]}\varphi(A_+, t) = (X, t)$. We temporary assume *(h7)*: X is not closer than α to a boundary configuration.

In the step 4, an abstract path-tracker is used to follow in a given orientation the R-reduced path $^{[b]}\mathcal{S}$ s.t. $(A_+, t) \in {}^{[b]}\mathcal{S}$. We assume that it allows to compute in a finite number of iterations the connected subset $^{[b]}\mathcal{S}' \subseteq {}^{[b]}\mathcal{S}$ s.t. points of $^{[b]}\mathcal{S}'$ are not closer than α to a boundary configuration, and that it stops when a point at a distance α to a boundary configuration is reached or when the stopping condition (sc1) described below is satisfied. It is also assumed that it is possible to detect when $^{[b]}\mathcal{S}'$ passes trough hyperplanes $\mathbf{t} = 0$ and $\mathbf{t} = 1$, and to get exact intersections with latter hyperplanes.

Algorithm 2 Change RCP

Input: A RCP $(I, C_-, \mathbf{A}_+, \mathbf{A}_0)$, a table T of original instructions, a current figure X, α
Output: New values A_+, A_0

  1: **for** $2 \leq i \leq l$ **do**
  2:      Let $I_i \in I$ be $\mathbf{p}_{i+1} = interCC(\mathbf{p}_{i^1}, \mathbf{a}_{i^2}, \mathbf{p}_{i^3}, \mathbf{a}_{i^4})$
  3:      **if** $\gamma_i(X) \leq \alpha$ **and** $T[i] == \emptyset$ **then**
  4:          **if** $\mathbf{a}_{i^4} \in \mathbf{A}_+$ **then**
  5:              Apply Algo. 4 to obtain a_{i^4}, p_{i^3}
  6:          **else** $\triangleright$ assume $a_{i^2} \geq a_{i^4}$
  7:              Let $\mathbf{p}'_{i^3}$ be point and $\mathbf{a}_{d+1}$ a length parameter
  8:              Let $p'_{i^3} = p_{i^3}$ and $a_{d+1} = a_{i^4}$
  9:              Let I'_i be $\mathbf{p}_{i+1} = interCC(\mathbf{p}_{i^1}, \mathbf{a}_{i^2}, \mathbf{p}'_{i^3}, \mathbf{a}_{d+1})$
10:              Apply Algo. 3 with I_i and I'_i as inputs
11:              Apply Algo. 4 to obtain a_{d+1}, p'_{i^3}
12:      **else if not** $T[i] == \emptyset$ **then**
13:          Let $I'_i = T[i]$ associated with γ'_i
14:          Let I'_i be $\mathbf{p}_{i+1} = interCC(\mathbf{p}_{i^1}, \mathbf{a}_{i^2}, \mathbf{p}'_{i^3}, \mathbf{a}'_{i^4})$
15:          **if** $\gamma'_i(X) > \alpha$ **then** $\triangleright$ restore I'_i
16:              Apply Algo. 3 with I_i and I'_i as inputs
17:          **else if** $a_{i^2} < a'_{i^4}$ **then** $\triangleright$ change driving parameter
18:              Apply Algo. 3 with I_i and I'_i as inputs
19:              **goto** step 3
20:          **else if** $\gamma_i(X) \leq \alpha$ **then** $\triangleright$ move point
21:              Apply Algo. 4 to obtain a_{i^4}, p_{i^3}
22:      Actualize A_+, A_0 and $d = |\mathbf{A}_+|$
23: **return** A_+, A_0

The tracking process stops when a point (A_+, t) of $^{[b]}\mathcal{S}$ is at a distance α to a boundary configuration. In this case a new RCP, or at least new values A_0, A_+ for reference points and driving parameters, is computed thanks to Algo. 2 described in Sec 4.3.1, and the process re-enters in step 4 with a new RCP R, new values A_0, A_+ and a new branch b such that (A_+, t) is not closer than α to a boundary configuration. Hence assumption *(h7)* is satisfied when re-entering step 4. If *(h7)* is not satisfied when performing for the first time step 4, Algo. 2 is directly applied.

Stopping condition *(sc1)* is detailed in Sec. 4.3.2. It is satisfied when it is not possible to ensure that assumption *(h6)* holds. When *(sc1)* is satisfied, Algo. 2 changes the RCP in such a way *(h6)* holds on the computed path.

Sought solutions are found when $^{[b]}\mathcal{S}$ passes through the hyperplane $\mathbf{t} = 1$, and the overall process is stopped when a point (A_+, t) of $^{[b]}\mathcal{S}$ is s.t. $^{[b]}I(a(t) \uplus A_+) = X^{sk}$. Latter termination criterion is checked each time the the hyperplane $\mathbf{t} = 0$ is crossed.

The orientation used to follow the new R-reduced path is chosen in order to avoid backtrack on $\mathcal{S}$. We will discuss practical details of path-tracking in Sec. 7.1. We now focus on the description of the way the RCP is changed.

Algorithm 3 Swap instructions

Input: A RCP $R = (I, C_-, \mathbf{A}_+, \mathbf{A}_0)$, an index i, a new instruction I'_i, a table T of original instructions

1: Let $I_i \in I$ be $\mathbf{p}_{i+1} = interCC(\mathbf{p}_{i^1}, \mathbf{a}_{i^2}, \mathbf{p}_{i^3}, \mathbf{a}_{i^4})$
2: Let I'_i be $\mathbf{p}_{i+1} = interCC(\mathbf{p}_{i^1}, \mathbf{a}_{i^2}, \mathbf{p}'_{i^3}, \mathbf{a}'_{i^4})$
3: Let c be the constraint $distance(\mathbf{p}_{i+1}, \mathbf{p}_{i^3}) = \mathbf{a}_{i^4}$
4: **if not** $c \in C_-$ **then**   ▷ introduce new instruction
5:      $\mathbf{A}_+ \leftarrow \mathbf{A}_+ \cup \{\mathbf{a}'_{i^4}\}$
6:      $\mathbf{A}_0 \leftarrow \mathbf{A}_0 \cup \{\mathbf{p}'_{i^3}\}$
7:      $I \leftarrow (I_1 \cup \ldots \cup I_{i-1} \cup I'_i \cup I_{i+1} \cup \ldots \cup I_l)$
8:      $C_- \leftarrow C_- \cup \{c\}$
9:      $T[i] \leftarrow I_i$
10: **else**   ▷ restore original instruction
11:     $\mathbf{A}_+ \leftarrow \mathbf{A}_+ \setminus \{\mathbf{a}'_{i^4}\}$
12:     $\mathbf{A}_0 \leftarrow \mathbf{A}_0 \setminus \{\mathbf{p}'_{i^3}\}$
13:     $I \leftarrow (I_1 \cup \ldots \cup I_{i-1} \cup I'_i \cup I_{i+1} \cup \ldots \cup I_l)$
14:     $C_- \leftarrow C_- \setminus \{c\}$
15:     $T[i] \leftarrow \emptyset$
16: **return**

4.3.1   Changing RCP

In Algo 1, when a point (A_+, t) s.t. $^{[b]}\gamma_I(A_+, t) = \alpha$ is reached, the RCP or at least the values of the reference are changed. The basic principle of the mechanism that changes the RCP or the reference is to identify the instruction(s) I_i s.t. $\gamma_i(X) = \alpha$, where $X = {}^{[b]}I(a(t) \uplus A_+)$.

Let I_i be $\mathbf{p}_{i+1} = interCC(\mathbf{p}_{i^1}, \mathbf{a}_{i^2}, \mathbf{p}_{i^3}, \mathbf{a}_{i^4})$ s.t. $\gamma_i(X) = \alpha$. Then either $\mathbf{a}_{i^4} \in \mathbf{A}_+$ and $\mathbf{p}_{i^3} \in \mathbf{A}_0$ is a reference point involved only in I_i, or $\mathbf{a}_{i^4} \in \mathbf{A}$, $\mathbf{p}_{i^3} \notin \mathbf{A}_0$ and we suppose without loss of generality that $a_{i^2} \geq a_{i^4}$ (otherwise arguments of the instruction are swapped).

In the first case, new values for $\mathbf{p}_{i^3}, \mathbf{a}_{i^4}$ are computed s.t. $\gamma_i(X) > \alpha$ thanks to Algo. 4. In the second case, I_i is exchanged with the instruction I'_i defined as $\mathbf{p}_{i+1} = interCC(\mathbf{p}_{i^1}, \mathbf{a}_{i^2}, \mathbf{p}'_{i^3}, \mathbf{a}_{d+1})$ and involving the new driving parameter $\mathbf{a}_{d+1}$ and the new reference point $\mathbf{p}'_{i^3}$. Then values a_{d+1}, p'_{i^3} are computed s.t. $\gamma_i(X) > \alpha$, where γ_i is the distance to a boundary configuration of I'_i. This relaxation is counterbalanced by adding $distance(\mathbf{p}_{i+1}, \mathbf{p}_{i^3}) = \mathbf{a}_{i^4}$ to the set C_- of removed constraints of the new RCP.

A table T, that does not appear in Algo. 1 to ease its description, is used to save the instructions of the original RCP (*i.e.* given as input of Algo. 1). Entries of T are initially empty, and each time an instruction I_i of the original RCP is exchanged with I'_i, I_i is stored in the i-th entry of T. When I_i could be restored while ensuring that the distance to a boundary configuration stays greater than α, I_i is restored and $T[i]$ is re-set to $\emptyset$. Hence an instruction I_i of the current RCP is an instruction of the original RCP if $T[i]$ is empty.

Algo. 2 details the mechanism to change a RCP, and Algo. 3 details the way instructions are swapped. They both modify in place the RCP. Notice that it could exist several indices i such that $\gamma_i(X) = \alpha$, and that Algo. 2 is designed to take it into account.

4.3.2   Stopping Condition (sc1)

The tracking process in Algo. 1 also stops when the condition (sc1) is satisfied for a current point (A_+^{cur}, t^{cur}). We define here this stopping condition.

Algorithm 4 Shift Reference

Input: An instruction $\mathbf{p}_{i+1} = interCC(\mathbf{p}_{i^1}, \mathbf{a}_{i^2}, \mathbf{p}_{i^3}, \mathbf{a}_{i^4})$, values $p_{i+1}, p_{i^1}, a_{i^2}, p_{i^3}, a_{i^4}$ for $\mathbf{p}_{i+1}, \mathbf{p}_{i^1}, \mathbf{a}_{i^2}, \mathbf{p}_{i^3}, \mathbf{a}_{i^4}$.

Output: New values p'_{i^3}, a'_{i^4} for $\mathbf{p}_{i^3}, \mathbf{a}_{i^4}$

1: Let p be the projection of p_{i+1} on the line (p_{i^1}, p_{i^3})
2: Let $p'_{i^3} = p_{i+1} + a_{i^2} \frac{\overrightarrow{p_{i+1}p}}{p_{i+1}p}$
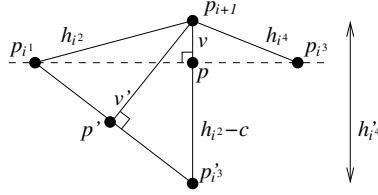3: Let $a'_{i^4} = a_{i^2}$
4: **return** p'_{i^3}, a'_{i^4}

Figure 7: New placement of points given by Algo. 4. p (resp. p') is the projection of p_{i+1} on the line (p_{i^1}, p_{i^3}) (resp. (p_{i^1}, p'_{i^3})) and v (resp. v') is the distance from p_{i+1} to p (resp. p').

Let I_i be an instruction of I s.t. $T[i] \neq \emptyset$. Hence I_i has been introduced by Algo. 2 to replace the original instruction $I'_i = T[i]$. Let I_i be $\mathbf{p}_{i+1} = interCC(\mathbf{p}_{i^1}, \mathbf{a}_{i^2}, \mathbf{p}'_{i^3}, \mathbf{a}'_{i^4})$, and I'_i be $\mathbf{p}_{i+1} = interCC(\mathbf{p}_{i^1}, \mathbf{a}_{i^2}, \mathbf{p}_{i^3}, \mathbf{a}_{i^4})$. Recall that in our homotopy context, values for $\mathbf{a}_{i^2}, \mathbf{a}_{i^4}$ are $a_{i^4}(t)$ and $a_{i^2}(t)$.

(sc1) is satisfied if it exists $2 \leq i \leq l$ s.t. $T[i] \neq \emptyset$ and $a_{i^4}(t^{cur}) > a_{i^2}(t^{cur})$. When (sc1) is satisfied on a point of a path, Algo. 2 is called. Unless the instruction I_i making (sc1) to be satisfied has been restored in steps 15-16, the if condition in step 17 of Algo. 2 is satisfied, and steps 18-19 are performed: the original instruction I'_i is restored, and when entering step 6, its arguments are swapped (*i.e.* $\mathbf{p}_{i+1} = interCC(\mathbf{p}_{i^1}, \mathbf{a}_{i^2}, \mathbf{p}_{i^3}, \mathbf{a}_{i^4})$ is replaced by $\mathbf{p}_{i+1} = interCC(\mathbf{p}_{i^3}, \mathbf{a}_{i^4}, \mathbf{p}_{i^1}, \mathbf{a}_{i^2})$). Then a new driving parameter is introduced.

When returning to Algo. 1 after Algo. 2 have been performed, the stopping condition (sc1) is not satisfied. We will state in Sec. 5.1 that this mechanism ensures that *(h6)* holds.

4.3.3 Shifting Reference

Algo. 4 is called in steps 5, 11 and 21 of Algo. 2. It computes values for driving parameters and reference point of an instruction I_i in order that the constructed figure X satisfies $\gamma_i(X) > \alpha$. The following proposition states that this goal is achieved after applying Algo. 4 if $0 < \alpha < \frac{1}{2}$.

Proposition 2 *Let γ_i be the distance to a boundary configuration associated with the instruction $\mathbf{p}_{i+1} = interCC(\mathbf{p}_{i^1}, \mathbf{a}_{i^2}, \mathbf{p}_{i^3}, \mathbf{a}_{i^4})$ and $(p_{i+1}, p_{i^1}, a_{i^2}, p_{i^3}, a_{i^4})$ be values s.t. $\gamma_i(p_{i+1}, p_{i^1}, p_{i^3}) \leq \alpha$. If $0 < \alpha < \frac{1}{2}$ and p'_{i^3} has been obtained with Algo. 4 then $\gamma_i(p_{i+1}, p_{i^1}, p'_{i^3}) > \alpha$.*

The proof of Prop. 2 is depicted in fig. 7, where p is the projection of p_{i+1} on the line (p_{i^1}, p_{i^3}) and v is the distance from p_{i+1} to p. Let p'_{i^3} be the new placement of $\mathbf{p}_{i^3}$ obtained with Algo. 4. The criterion γ_i is computed by considering the distance v' from p_{i+1} to p' where p' is the projection of p_{i+1} on the line (p_{i^1}, p'_{i^3}). One has $\frac{v'^2}{(a_{i^2})^2} = \frac{1}{2} + \frac{v}{2a_{i^2}}$ and the result follows. $\square$

5   Correctness and Termination

Let $\mathcal{S}$ be the homotopy path of $\mathcal{H}$ to which belongs $(X^{sk}, 0)$. We show here that Algo. 1 terminates, and that all solutions (X, t) of $\mathcal{H}$ with $t = 1$ lying on $\mathcal{S}$ are in $\mathcal{L}_{sol}$ at the end of Algo. 1.

Algo. 1 involves an iterative path tracker that is assumed to track in a finite number of steps a R-reduced path if it is a manifold, and if its points are not closer than α to a boundary configuration. The correctness and the termination of our method is proved when considering such an abstract path tracker. Sec. 7 proposes an implementation of such a path-tracker.

Let $R = (I, C_-, \mathbf{A}_+, \mathbf{A}_0)$ with $I = (I_i)_{1 \le i \le l}$ be the RCP given as input of Algo. 1. The latter procedure computes a sequence $(R^j)_{j \in N}$ of RCP where $N \subseteq \mathbb{N}_*$, $R^1 = R$ and $R^j = (I^j, C_-^j, \mathbf{A}_+^j, \mathbf{A}_0^j)$ with $I^j = (I_i^j)_{1 \le i \le l}$. A sequence $(\mathcal{S}^j)_{j \in N}$ of connected pieces of R^j-reduced paths is followed, and we note bj the branch of I^j such that $^{[bj]}R^j(\mathcal{S}^j) = 0$. We will note γ_{I^j} the distance to a boundary configuration associated with I^j and $^{[bj]}\gamma_{I^j}(\mathbf{A}_+^j, \mathbf{t})$ for $\gamma_{I^j}(^{[bj]}I^j(a(\mathbf{t}) \uplus \mathbf{A}_+^j))$. We will note $^{[bj]}\varphi_j$ and φ_j' the mappings defined in Eqs. 3 and 4 specialized to the RCP R^j. Let finally $\pi_{\mathbf{t}}$ be the projection with respect to the $\mathbf{t}$-coordinate.

The main points of the proof are:

(i) $^{[bj]}\gamma_{I^j}$ is well defined on $\mathcal{S}^j$,

(ii) if $(A_+^j, t) \in \mathcal{S}^j$ and $\alpha < \frac{1}{2}$ then $^{[bj]}\gamma_{I^j}(A_+^j, t) \ge \alpha$,

(iii) $(\mathcal{S}^j)_{j \in N}$ are 1-dimensional manifolds,

(iv) $\bigcup_{j \in N} {}^{[bj]}\varphi_j(\mathcal{S}^j) \subseteq \mathcal{S}$,

(v) N is a finite subset of $\mathbb{N}$ and $\bigcup_{j \in N} {}^{[bj]}\varphi_j(\mathcal{S}^j) = \mathcal{S}$.

Remark that the stopping condition (sc1) is not satisfied on a point $(A_+^j, t) \in \mathcal{S}^j$, for $j \in N$. We will prove (i) in Sec. 5.1 by showing that *(h4)* and *(h5)* hold for R^j, and *(h6)* holds for R^j on $\pi_{\mathbf{t}}(\mathcal{S}^j)$. Then (ii) holds thanks to Prop. 2.

(iii) and (iv) are consequences of Prop. 1: $^{[bj]}\varphi_j$ is a diffeomorphism from $\mathcal{S}^j$ to a connected subset of $\mathcal{S}$. Notice that (ii) and (iii) are the two conditions under which the abstract path-tracker used in Algo. 1 computes $\mathcal{S}^j$.

(v) is proved in Sec. 5.2. It has as a direct consequence that Algo. 1 terminates and all solutions (X, t) of $\mathcal{H}$ with $t = 1$ lying on $\mathcal{S}$ are found.

5.1   Proof of Point (i)

As stated in Sec. 4.2, the distance to a boundary configuration associated with a RCP is well defined when *(h4)*, *(h5)* and *(h6)* hold. It is established in the following proposition, and point (ii) follows as a corollary.

Proposition 3 *If assumptions (h4), (h5) and (h6) hold for R, then $\forall j \in N$, (h4) and (h5) hold for R^j, and (h6) holds for R^j at least on $\pi_{\mathbf{t}}(\mathcal{S}^j)$.*

Proof of Prop. 3: Let $j \in N$. The first instruction of the RCP is never changed in Algo. 2. $I_1^j = I_1$ follows and assumptions *(h4)* holds for R^j.

Let $2 \le i \le l$ and I_i^j be the instruction $\mathbf{p}_{i+1} = interCC(\mathbf{p}_{i^1}, \mathbf{a}_{i^2}, \mathbf{p}_{i^3}, \mathbf{a}_{i^4})$. If $\mathbf{a}_{i^4} \notin \mathbf{A}_+^j$ then $I_i^j = I_i$. Since *(h5)* holds for R, *(h5)* holds for R_j.

Suppose now $\mathbf{a}_{i^4} \in \mathbf{A}_+^j$. If $\mathbf{a}_{i^4} \in \mathbf{A}_+$, *i.e.* $\mathbf{a}_{i^4}$ is a driving parameter of R, then $I_i^j = I_i$ and $a_{i^2}(t) > 0$ since *(h6)* holds for R when $t \in supp_+(a)$.

Otherwise, I_i^j is not an original instruction. Let $\mathbf{p}_{i+1} = interCC(\mathbf{p}_{i^1}, \mathbf{a}_{i^2}, \mathbf{p}_{i^3}', \mathbf{a}_{i^4}')$ be the original instruction I_i and a_{i^4} be the interpolation function for values of $\mathbf{a}_{i^4}'$. $a_{i^2}(t)$ and $a_{i^4}(t)$ does not both vanish according to *(h5)*. Since condition (sc1) is not satisfied for $(A_+^j, t) \in \mathcal{S}^j$, $a_{i^2}(t) \geq a_{i^4}(t)$ holds and $a_{i^2}(t)$ does not vanish on $\pi_{\mathbf{t}}(\mathcal{S}^j)$. Thus *(h6)* holds for R_j on $\pi_{\mathbf{t}}(\mathcal{S}^j)$. □

Corollary 2 *If assumptions (h4), (h5) and (h6) hold for R, then $\forall j \in N$, $^{[bj]}\gamma_{I^j}$ is well defined on $\mathcal{S}^j$.*

5.2 Proof of Point (v)

We consider first the case where $N = \{1, \ldots, n\}$. The termination condition of step 6 of Algo. 1 is reached, hence the set $\bigcup_{j \in N} {}^{[bj]}\varphi_j(\mathcal{S}^j)$ is diffeomorphic to a circle. Since $\bigcup_{j \in N} {}^{[bj]}\varphi_j(\mathcal{S}^j) \subseteq \mathcal{S}$, $\bigcup_{j \in N} {}^{[bj]}\varphi_j(\mathcal{S}^j) = \mathcal{S}$ follows.

We consider now the case $N = \mathbb{N}_*$ and we show that it never happens. Algo. 1 constructs a sequence $((A_+^j, t^j))_{j \in N}$ of points s.t. either $^{[bj]}\gamma_{I^j}(A_+^j, t^j) = \alpha$ or (sc1) is satisfied, and $^{[bj]}\gamma_{I^j}(A_+^{j-1}, t^{j-1}) > \alpha$ and (sc1) is not satisfied. Consider the sequence $((X^j, t^j))_{j \in N}$ where $(X^j, t^j) = {}^{[bj]}\varphi_j(A_+^j, t^j)$. From point (iv), $((X^j, t^j))_{j \in N}$ is a sequence of points of $\mathcal{S}$.

From Cor. 1, $\mathcal{S}$ is diffeomorphic to a circle, hence $\mathcal{S} \setminus \{(X^{sk}, 0)\}$ is diffeomorphic to a bounded open interval, and it exists a diffeomorphism $S :]0, 1[\to \mathcal{S} \setminus \{(X^{sk}, 0)\}$ that maps to $s \in]0, 1[$ a point of $\mathcal{S} \setminus \{(X^{sk}, 0)\}$. Reciprocally, S^{-1} maps to a point (X^j, t^j) a real number $s^j \in]0, 1[$.

We show that the sequence $(s^j)_{j \in N}$ satisfies $\forall j \geq 2, s^j > s^{j-1}$ and does not have any accumulation point. As a consequence, it cannot be infinite.

Suppose it exists $j \geq 2$ s.t. $s^j = s^{j-1}$, hence $(X^j, t^j) = (X^{j-1}, t^{j-1})$. From Cor. 2, γ_{I^j} is well defined on $X^j = X^{j-1}$ and $\gamma_{I^j}(X^{j-1}) > \alpha$ hence (sc1) is not satisfied for (X^{j-1}, t^{j-1}), and either $\gamma_{I^j}(X^j) = \alpha$ or (sc1) is satisfied for (X^j, t^j) hence a contradiction follows. In Algo. 1, paths $\mathcal{S}^j$ are followed with an orientation that ensures a progression along $\mathcal{S}$, hence we have $s^j > s^{j-1}$.

Suppose now that the sequence $(s^j)_{j \in N}$ has an accumulation point s^*, hence $((X^j, t^j))_{j \in \mathbb{N}}$ has an accumulation point (X^*, t^*), and it exists a subsequence $((X^j, t^j))_{j \in N^*}$, with $N^* \subseteq N$, converging to (X^*, t^*). From assumption *(h3)*, there is an index j^1 s.t. $\forall j > j^1, \forall 1 \leq i^1, i^2 \leq m$ the sign of $a_{i^1}(t^j) - a_{i^2}(t^j)$ does not change. Hence for $j > j^1$, for each instruction $p_{i+1} = interCC(\mathbf{p}_{i^1}, \mathbf{a}_{i^2}, \mathbf{p}_{i^3}, \mathbf{a}_{i^4})$ of I^j with $i \geq 2$, $a_{i^2}^j > a_{i^4}^j$ and since assumptions *(h5)* and *(h6)* hold, it exists $r > 0$ s.t. $\forall j > j^1, \forall i \geq 2, a_{i^2}^j > r$. Now, for each $\epsilon > 0$, it exists an index $j^2 > j^1$ s.t. $\forall j > j^2$, points and distances between points vary no more than ϵ between X^j and X^{j-1}. Remark that since signs of $a_{i^1}(t^j) - a_{i^2}(t^j)$ does not change when $j \geq j^2$ grows, (sc1) is satisfied neither for X^j nor for X^{j-1} and it follows that $\gamma_{I^j}(X^{j-1}) > \frac{1}{2}$ (from Prop. 2) and $\gamma_{I^j}(X^j) = \alpha < \frac{1}{2}$ when $j \geq 2$. Taking ϵ sufficiently small (strictly less than $(\frac{1}{2} - \alpha)r$) leads to a contradiction. □

6 Generalization to 3D PDSP

3D PDSP fit well to the method depicted in this paper: results of [9] as well as reparameterization approach stay valid. Given a PDSP G in a 3D geometric universe, solutions of G up to rigid motions are found by fixing a reference consisting in a point $\mathbf{p}_1$, a line $\mathbf{l}_1$ and a plane $\mathbf{pl}_1$. Values p_1, l_1, pl_1 are fixed s.t. $p_1 \in l_1$ and $l_1 \in pl_1$. A CP of G has the structure:

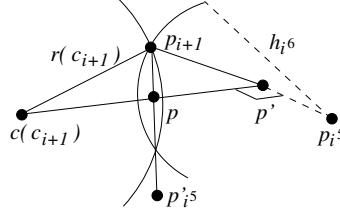(I_1) $\mathbf{p}_2 = interSL(\mathbf{p}_1, \mathbf{a}_1, \mathbf{l}_1)$

Figure 8: Placement of point $\mathbf{p}'_{i^5}$ when swapping a $interCS$ instruction. Dashed lines does not belong to the plane of circle c_{i+1}.

(I_2) $\mathbf{p}_3 = interSSP(\mathbf{p}_1, \mathbf{a}_2, \mathbf{p}_2, \mathbf{a}_3, \mathbf{pl}_1)$

(I_3) $\mathbf{p}_4 = interSSS(\mathbf{p}_1, \mathbf{a}_4, \mathbf{p}_2, \mathbf{a}_5, \mathbf{p}_3, \mathbf{a}_6)$

   $\ldots$

(I_i) $\mathbf{p}_{i+1} = interSSS(\mathbf{p}_{i^1}, \mathbf{a}_{i^2}, \mathbf{p}_{i^3}, \mathbf{a}_{i^4}, \mathbf{p}_{i^5}, \mathbf{a}_{i^6})$

   $\ldots$

where $interSL$ is a sphere-line intersection, $interSSP$ is the intersection of two spheres and one plane, and $interSSS$ is a three spheres intersection.

   If I_i is an $interSSS$ instructions, it is decomposed into the two instructions

$(I1_i)$ $\mathbf{c}_{i+1} = interSS(\mathbf{p}_{i^1}, \mathbf{a}_{i^2}, \mathbf{p}_{i^3}, \mathbf{a}_{i^4})$

$(I2_i)$ $\mathbf{p}_{i+1} = interCS(c(\mathbf{c}_{i+1}), r(\mathbf{c}_{i+1}), \mathbf{p}_{i^5}, \mathbf{a}_{i^6})$

where $\mathbf{c}_{i+1}$ is a circle, $interSS$ is a sphere-sphere intersection, $c(\mathbf{c}_{i+1})$ and $r(\mathbf{c}_{i+1})$ are respectively the center and the radius of $\mathbf{c}_{i+1}$, and $interCS$ is a sphere-circle intersection.

   $interSSP$ and $interSS$ can be seen as a 3D extension of an $interCC$ instruction in the 2D case. Hence when assumptions *(h1)* to *(h6)* hold, boundary configurations encountered on a homotopy path $\mathcal{S}$ of $\mathcal{H}$ for these instructions are the same than boundary configurations in the 2D case. The function that measures the distance to a boundary configuration of such instructions is the natural extension of the one associated with an $interCC$ instruction. When such instructions are swapped, reference points are fixed as in the 2D case.

   Consider now an $interCS$ instruction $I2_i$. A boundary configuration is reached when the circle and the sphere are tangent. The radius of the center is null only for boundary configurations of $I1_i$. We associate with $(I2_i)$ the function γ_i defined as $\gamma_i(\mathbf{X}) = \dfrac{\dot{\mathbf{p}}_{i+1}\mathbf{p}}{max(\mathbf{p}_{i+1}c(\mathbf{c}_{i+1}), \mathbf{p}_{i+1}\mathbf{p}_{i^5})}$ where $\mathbf{p}'$ is the projection of $\mathbf{p}_{i^5}$ on the plane to which belongs $\mathbf{c}_{i+1}$, and $\mathbf{p}$ is the projection of $\mathbf{p}_{i+1}$ on the line passing by $c(\mathbf{c}_{i+1})$ and $\mathbf{p}'$. γ_i is defined on figures that are not a boundary configuration of $I1_i$. If $\mathbf{a}_{i^6}$ is not already a driving parameter, $\mathbf{p}_{i+1} = interCS(c(\mathbf{c}_{i+1}), r(\mathbf{c}_{i+1}), \mathbf{p}_{i^5}, \mathbf{a}_{i^6})$ can be swapped with $\mathbf{p}_{i+1} = interCS(c(\mathbf{c}_{i+1}), r(\mathbf{c}_{i+1}), \mathbf{p}'_{i^5}, \mathbf{a}'_{i^6})$, and the value p'_{i^5} for the new reference point $\mathbf{p}'_{i^5}$ is set as $p'_{i^5} = p_{i+1} + m\dfrac{\overrightarrow{p_{i+1}p}}{p_{i+1}p}$ where m is the greatest value between $r(c_{i+1})$ and a_{i^6}, and a'_{i^6} is set as m, as illustrated in Fig 8. It is then easy to state a proposition equivalent to Prop. 2 for this placement of point.

7 Implementation and results

Our method has been implemented in `C++`, giving rise to a program that accepts a PDSP G, a RCP, a sketch and provides solutions of G. The path-tracking is achieved by a prediction-correction method with an adaptive prediction step that is described in Sec. 7.1. It requires to compute partial derivatives of the function $^{[b]}H_R$, which appears to be one of the most time consuming step of our method. We propose in Sec. 7.2 to exploit the acyclic nature of a CP to optimize this operation. Numerical results related to the solving of four PDSP in 2D and 3D are given in Subsec. 7.3. It confirms the efficiency of the approach of [9] to provide several solutions (sometimes all of them) of problems that resist to divide and conquer methods and are too large to be solved by classical numeric solvers providing all the solutions. Using a RCP as proposed here brings an important speed-up of this approach.

7.1 Path tracking

Homotopy and R-reduced paths are followed thanks to a classical prediction-correction method: prediction is performed along the tangent of the path by an Euler predictor with step $\delta \in [\delta_{min}, \delta_{max}]$ and correction by Newton-Raphson iterations. The step δ is doubled (respectively halved) if $2\delta \leq \delta_{max}$ (resp. $\delta \geq 2\delta_{min}$) and if the previous correction step did succeed (resp. fail). The Jacobian matrices that are required both in prediction and correction steps are numerically computed with finite differences.

In Algo. 1, when entering for the first time in the main while loop, an orientation (*i.e.* one of the two unit vectors of the tangent) to follow the first R-reduced path is arbitrarily chosen. When entering in the while loop after the RCP has been changed for the $j + 1$-th time, the orientation has to ensure the progression along $\mathcal{S}$. To determine the appropriated orientation, the last unit vector of the tangent used to track $\mathcal{S}^j$ is "translated" in the new space where $\mathcal{S}^{j+1}$ is tracked with the application $\varphi'_{j+1} \circ {}^{[bj]}\varphi_j$, with notations of Sec. 5.

Notice that this simple path tracking algorithm does not avoid jumps between paths. Approaches using interval arithmetic (see [10, 5, 13]) could be used to certify the path tracking. Here we suppose that δ_{max} is small enough to follow considered curves while avoiding such jumps.

7.2 Differentiation of the CP

When tracking a R-reduced path, most of the computation time is spent in the evaluation of the underlying RCP. Most evaluations intervene in the computation of Jacobian matrices by finite differences that needs about d evaluations, where d is the number of driving parameters. Such matrices are computed at each prediction step and at each iteration of the Newton-Raphson method in a correction step. Here we exploit the acyclic computation scheme of a RCP to improve its evaluation.

Suppose driving parameters $\mathbf{A}_+ = (\mathbf{k}_1, \ldots, \mathbf{k}_d)$ of R appear in instructions $I_{i^1}, \ldots, I_{i^d}$ with $i^d \geq \ldots \geq i^1$. When computing with finite differences the derivative with respect to $\mathbf{k}_j$ of the numerical function associated to R, the geometric objects of the figures resulting of the two evaluations differ only if they are produced by instructions I_i with $i \geq i^j$ since I_{i^j} is the first step involving $\mathbf{k}_j$.

Hence a manner of optimizing the differentiation of a RCP is to evaluate it entirely a first time and then to compute the partial derivatives with respect to $\mathbf{k}_j$ by evaluating the RCP from the step i^j, for j from d to 1. Our implementation incorporates this optimization.

Figure 9: Octahedron (left) and disulfide (right) problems. Edges are distance constraints.

7.3   Results

We give here numerical results concerning the solving of four PDSP, one in 2D and three in 3D. The method depicted here consists in computing the path $\mathcal{S}$ of $\mathcal{H}$ to which belongs the sketch by using a RCP to track $\mathcal{S}$ in the space of driving parameters instead of tracking it in the space of all coordinates. These two approaches (with and without RCP) yield the same number of solutions. However, using a RCP brings an important gain in term of running times as it appears in our experiments. For each problem we also give the running time and the number of solutions obtained when solving the system $\mathcal{F}_{so}$ with a classical homotopy method implemented by the free software HOM4PS-2.0 (see [18]). We did chose homotopy solving as a witness method because as far as we know, it is the sole approach allowing to find all the solutions of large undecomposable problems. We did choose HOM4PS-2.0 to implement it because among other free softwares implementing homotopy, it seems to be faster to solve sparse systems of polynomials.

Be given a real solution of a PDSP, the elements of its orbit by the action of the group of reflections through x and y axis (resp. (x,y), (y,z) and (x,z) planes in the 3D case) are also solutions of the PDSP. In our experiments, the solutions found on $\mathcal{S}$ belong to different orbits.

7.3.1   Problems and parameters settings

The goal of the octahedron problem is to construct a solid with 6 vertices, 12 edges and 8 triangular faces knowing the lengths of its 12 edges (see Fig. 9). This problem is used in [3] and is related to the parallel robot called Gough-Stewart platform. It results in a system $\mathcal{F}_{so}$ of 12 equations with 12 unknowns.

The second problem comes from molecular chemistry and is picked up from [17]. Coordinates of 8 points in the 3D space have to be found knowing 18 distances. It corresponds to a disulfide molecule (see right part of Fig. 9). A valuation of parameters is exhibited in [17] that leads to 18 solutions up to reflexions all found by a bisection method in more than 10 minutes in [17].

Dodecagon and Icosahedron problems are illustrated on Fig. 10. The former gives rise to a system $\mathcal{F}_{so}$ with 21 equations. The system $\mathcal{F}_{so}$ associated to the icosahedron problem involves 30 equations.

Values X^{sk} of the sketches and A^{so} of parameters are given in appendix A.

Interpolation functions have been chosen such that $a_m(t) = -2t^2 + (a_m^{so} - a_m^{sk} + 2)t + a_m^{sk}$ and $a_i(t) = (1-t)a_i^{sk} + ta_i^{so}$, for $1 \leq i < m$. The value for α has been set for each problem to 0.1, what seems to fit well to our algorithm. The prediction steps vary in the interval $[1^{-10}, \delta_{max}]$. As stated above, δ_{max} has to be chosen small enough to avoid jumping between different paths. The values $\delta_{max} = 0.1$ when tracking $\mathcal{S}$ without RCP and $\delta_{max} = 0.05$ when using a RCP have been chosen after several trials. Further details and discussions concerning the prediction steps are given in 7.3.3.
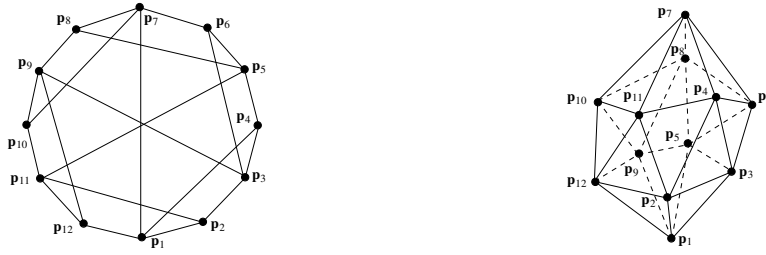
Figure 10: Dodecagon (left) and icosahedron (right) problems. Edges are distance constraints.

Table 1: Sequential running times on an Intel(R) Core(TM) i7-5600U CPU @ 2.60GHz.

	Octahedron	Disulfide	Dodecagon	Icosahedron
m	12	18	21	30
Number of solutions				
complex	72	256	12580	-
real (up to reflections)	4	18	2	-
on $\mathcal{S}$	4	8	2	32
Running times				
HOM4PS-2.0	19.8s	3776s	10h	-
tracking $\mathcal{S}$	0.06s	0.8s	0.07s	8.6s
Algo. 1	0.02s	0.09s	0.01s	1.5s

7.3.2   Data of Table 1

Table 1 gives for each problem the number m of equations of the system $\mathcal{F}_{so}$. In group of lines "Number of solutions", it first gives the total number of complex solutions of $\mathcal{F}_{so}$, all found by HOM4PS-2.0. (line "complex"). The line "real (up to reflections)" gives the number of real solutions up to reflections. The line "on $\mathcal{S}$" gives the number of real solutions lying on the path $\mathcal{S}$ to which belongs the sketch, that is obtained with our method. As remarked above, these solutions are different up to reflections.

The group of lines "Running times" refers to times required to solve each problem with each approach. When using HOM4PS-2.0., most efforts are spent to follow paths leading to complex solutions, what explains the large running times in the line "HOM4PS-2.0.". The line "tracking $\mathcal{S}$" refers to the time required to track $\mathcal{S}$ in the space $\mathbb{R}^m \times \mathbb{R}$, without using a RCP. The line "Algo. 1" refers to the time required to compute solutions on $\mathcal{S}$ with on-the-fly change of RCP; it allows an important gain in term of computation cost.

For the icosahedron problem, we do not give total number of solutions and running times for HOM4PS-2.0 since the solving process did not finish.

7.3.3   Details on execution

Table 2 gives for each problem details about paths tracking without RCP (in the first columns) and with RCP (in the second columns). The row "time t in s" recalls the execution time in seconds.

The row "smallest δ" gives the smallest prediction step used during the tracking process. It shows that tracking a path with a RCP requires to take smaller prediction steps than without a RCP. Together with the fact that δ_{max} has to be smaller when using a RCP, it suggests that

Table 2: Details about path-tracking. For each problem, the first (resp. second) columns refers to the path tracking without (resp. with) RCP.

| | Octahedron | | Disulfide | | Dodecagon | | Icosahedron | |
m	12		18		21		30	
time t in s	0.06	0.02	0.8	0.09	0.07	0.01	8.6	1.5
smallest δ	0.1	0.03	0.1	0.01	0.05	0.02	0.05	0.001
nb. i of iterations	157	173	1097	694	42	46	4074	3377
t/i in ms	0.4	0.1	0.78	0.13	1.78	0.3	2.1	0.43
nb. of RCP changing	-	13	-	28	-	2	-	250
average nb. of DP	-	1.3	-	1.98	-	4	-	3.8
max. nb. of DP	-	2	-	3	-	4	-	6

R-reduced paths have higher curvature than the corresponding paths in the space of figures, and are more difficult to track.

The lines "nb. i of iterations" and "t/i in ms" give the number of iterations of prediction-correction and the average time needed for each iteration. The latter information underlines the main advantage of using a RCP: each iteration of prediction-correction involves less computation than without RCP since the path is tracked in a space of smaller dimension. The line "time t in s" shows that this gain counterbalances the drawback mentioned above.

The line "nb. of RCP changing" gives the number of times the RCP has been changed during the tracking process. This number can be large (see the case of the icosahedron problem): as mentioned in the penultimate paragraph of Sec. 3.4, figures with a boundary configuration are the solutions of systems of $m + 1$ equations in $m + 1$ unknowns and can be in an exponential number. Hence the approach proposed in this paper could require, in the worst case, to change exponentially many times the RCP. The rows "average nb. of DP" and "max. nb. of DP" give respectively the average and the maximum number of driving parameters involved in the RCP and show that the number of driving parameters involved in successive RCP stays much lower than m what keeps the method efficient.

8   Conclusion

Well-constrained point distance solving problems often have many solutions. The existing solvers that offer all the solutions are of limited practical interest because either the class of problems they solve is reduced or their complexity is exponential. But even if not all solutions are needed, several ones similar in shape to the sketch must be provided.

An approach to fulfill this requirement is to use the sketch to define a real homotopy such that the homotopy path to which belongs the sketch is diffeomorphic to a circle and contains several solutions, that are similar to the sketch in the sense that they belongs to the same homotopy path.

In this article we made this approach more efficient by reducing the dimension of the space where the homotopy path is tracked by using a symbolic geometric constructions program. The latter is modified on-the-fly in order to stay robust to critical geometric configurations it could induce.

This original idea has been implemented to prove its soundness. In the examples discussed solutions are produced quicker when a construction program is used. Moreover the presented experiments show that our approach can provide several solutions to problems that are too large

to be solved with numerical solvers searching all the solutions such as homotopy.

Notice finally that our method could be extended to more general geometric constraints such as angles, collinearities, coplanarities, and so on. When considering these constraints, homotopy paths are not necessarily diffeomorphic to circles but can converge to special geometric configurations that can be detected when using a construction program to stop the path tracking process.

References

[1] E.L. Allgower and K. Georg. Numerical path following. *Handbook of Numerical Analysis*, 5(3):207, 1997.

[2] Hichem Barki, Lincong Fang, Dominique Michelucci, and Sebti Foufou. Re-parameterization reduces irreducible geometric constraint systems. *Computer-Aided Design*, 70:182–192, 2016.

[3] C. Durand and C.M. Hoffmann. A systematic framework for solving geometric constraints analytically. *Journal of Symbolic Computation*, 30(5):493–519, 2000.

[4] Arnaud Fabre and Pascal Schreck. Combining symbolic and numerical solvers to simplify indecomposable systems solving. In *Proceedings of ACM Symposium on Applied Computing SAC 2008)*, pages 1838–1842, New York, NY, USA, 2008. ACM.

[5] D. Faudot and D. Michelucci. A new robust algorithm to trace curves. *Reliable computing*, 13(4):309–324, 2007.

[6] S. Foufou and D. Michelucci. The Bernstein basis and its applications in solving geometric constraint systems. *Reliable Computing*, 17(2):192–208, 2012.

[7] Xiao-Shan Gao, Christoph M. Hoffmann, and Wei-Qiang Yang. Solving spatial basic geometric constraint configurations with locus intersection. In *Proceedings of the seventh ACM symposium on Solid modeling and applications*, SMA '02, pages 95–104, New York, NY, USA, 2002. ACM.

[8] R. Imbach, P. Mathis, and P. Schreck. Tracking method for reparametrized geometrical constraint systems. In *2011 13th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, pages 31–38. IEEE, 2011.

[9] Rémi Imbach, Pascal Schreck, and Pascal Mathis. Leading a continuation method by geometry for solving geometric constraints. *Computer-Aided Design*, 46:138–147, 2014.

[10] R.B. Kearfott and Z. Xing. An interval step control for continuation methods. *SIAM Journal on Numerical Analysis*, 31(3):892–914, 1994.

[11] Hervé Lamure and Dominique Michelucci. Solving geometric constraints by homotopy. In *Proceedings of the third ACM symposium on Solid modeling and applications*, SMA '95, pages 263–269, New York, NY, USA, 1995. ACM.

[12] TY Li and Xiao Shen Wang. Solving real polynomial systems with real homotopies. *mathematics of computation*, 60(202):669–680, 1993.

[13] Benjamin Martin, Alexandre Goldsztejn, Laurent Granvilliers, and Christophe Jermann. Certified parallelotope continuation for one-manifolds. *SIAM Journal on Numerical Analysis*, 51(6):3373–3401, 2013.

[14] Pascal Mathis, Pascal Schreck, and Rémi Imbach. Decomposition of geometrical constraint systems with reparameterization. In Sascha Ossowski and Paola Lecca, editors, *SAC*, pages 102–108. ACM, 2012.

[15] B. Mourrain and J. P. Pavone. Subdivision methods for solving polynomial equations. *J. Symb. Comput.*, 44(3):292–306, March 2009.

[16] M.D. Plummer and L. Lovász. *Matching Theory*. North-Holland Mathematics Studies. Elsevier Science, 1986.

[17] Josep M Porta, Lluís Ros, Federico Thomas, Francesc Corcho, Josep Cantó, and Juan Jesús Pérez. Complete maps of molecular-loop conformational spaces. *Journal of computational chemistry*, 28(13):2170–2189, 2007.

[18] T. Y. Liet T. L. Lee and C. H. Tsai. Hom4ps-2.0: a software package for solving polynomial systems by the polyhedral homotopy continuation method. *COMPUTING*, 83:109–133, 2008.

A   Numerical values of A^{so} and X^{sk}

Octahedron   Values A^{so} for parameters are picked up from [3].

$$p_1 = (0,0,0), \quad p_2 = (1,0,0), \quad p_3 = (0.5, 1.3, 0)$$
$$p_4 = (1.2, 0.5, 1.5), \quad p_5 = (0, 1.8, 0.9), \quad p_6 = (-0.2, -0.7, 1.3)$$

Disulfide molecule   The reader is referred to [17] to get values A^{so} of parameters.

$$p_1 = (0,0,0), \quad p_2 = (2.5, 5.3, 0), \quad p_3 = (3.2, 5.8, 5.9), \quad p_4 = (1, 8.7, 8.3),$$
$$p_5 = (4.2, 5.6, 7.9), \quad p_6 = (-2.2, 1.3, 6.3), \quad p_7 = (5.5, 2.5, 6.5), \quad p_8 = (5, 0, 0).$$

Dodecagone   In the tables below, $a_{i,j}$ denotes the value of the parameter of $distance(\mathbf{p}_i, \mathbf{p}_j)$.

$$a_{1,2} = 3, \quad a_{2,3} = 1.75, \quad a_{3,4} = 1.7, \quad a_{4,5} = 2.05 \quad a_{5,6} = 1.5, \quad a_{6,7} = 1.85,$$
$$a_{7,8} = 1.45, \quad a_{8,9} = 1.35 \quad a_{9,10} = 1, \quad a_{10,11} = 1.4, \quad a_{11,12} = 1, \quad a_{12,1} = 0.6,$$
$$a_{1,4} = 4.4, \quad a_{3,6} = 5.1, \quad a_{5,8} = 3.9, \quad a_{7,10} = 3.05, \quad a_{9,12} = 3.35, \quad a_{11,2} = 4.4,$$
$$a_{1,7} = 4.45, \quad a_{3,9} = 6.65, \quad a_{5,11} = 4.65.$$

$$p_1 = (0,0), \quad p_2 = (2.9, 0), \quad p_3 = (3.4, 1.8), \quad p_4 = (2.8, 3.4),$$
$$p_5 = (1.3, 4.6), \quad p_6 = (0.1, 5.7), \quad p_7 = (-1.0, 4.4), \quad p_8 = (-2.2, 3.6),$$
$$p_9 = (-3.0, 2.3), \quad p_{10} = (-2.4, 1.5), \quad p_{11} = (-1.2, 0.8), \quad p_{12} = (-0.6, 0.4)$$

Icosahedron

$$a_{1,2} = 2.0, \quad a_{2,4} = 4, \quad a_{4,3} = 4.5, \quad a_{3,6} = 3.9, \quad a_{6,5} = 4.45, \quad a_{5,8} = 3.8,$$
$$a_{8,9} = 4.4, \quad a_{9,10} = 3.7, \quad a_{10,12} = 4.35, \quad a_{12,11} = 3.65, \quad a_{11,2} = 4.3, \quad a_{11,7} = 2.5,$$
$$a_{4,7} = 2.6, \quad a_{6,7} = 2.7, \quad a_{8,7} = 2.8, \quad a_{10,7} = 2.9, \quad a_{11,4} = 3.0, \quad a_{11,8} = 2.95,$$
$$a_{10,8} = 3.05, \quad a_{8,6} = 2.9, \quad a_{6,4} = 3.1, \quad a_{2,3} = 3.0, \quad a_{3,5} = 2.9, \quad a_{5,9} = 3.1,$$
$$a_{9,12} = 2.8, \quad a_{12,2} = 3.2, \quad a_{1,9} = 2.1, \quad a_{1,5} = 2.2, \quad a_{1,3} = 2.3, \quad a_{1,12} = 2.4.$$

$$p_1 = (0,0,0), \quad p_2 = (1,0,0), \quad p_3 = (0.5, 1, 0), \quad p_4 = (11, 1.2, 1.3),$$
$$p_5 = (-0.5, 0.5, 1), \quad p_6 = (1.7, 2.3, 1.1), \quad p_7 = (3, 3, 3), \quad p_8 = (0.8, 1.6, 2.2),$$
$$p_9 = (-0, 0, 2), \quad p_{10} = (1.1, 1.3, 3.2), \quad p_{11} = (1.8, 1.2, 2.1), \quad p_{12} = (0.6, 0.2, 1.3)$$