On the realisability of double-cross matrices by polylines in the plane

Bart Kuijpers[1] and Bart Moelans

Databases and Theoretical Computer Science, Hasselt University, Belgium

Abstract

We study a decision problem that emerges from the area of spatial reasoning, but that is also of interest to the area of computational algebraic geometry. This decision problem concerns the use of constraint calculi in qualitative spatial reasoning. One such qualitative calculus describes polylines in the plane by means of their *double-cross matrix*. In such a matrix, the relative position (or orientation) of each pairs of line segments of a polyline is expresses by means of a 4-tuple, whose entries come from the set $\{-, 0, +\}$. However, not any $N \times N$ matrix of 4-tuples from $\{-, 0, +\}$ is the double-cross matrix of a polyline with N line segments. This gives rise to the following *decision problem*: given an $N \times N$ matrix of 4-tuples from $\{-, 0, +\}$, decide whether it is the double-cross matrix of a polyline with N line segments, and if it is, given an example of a polyline that realises the matrix.

It is known that this problem is decidable, but it is NP-hard and the best, known algorithms have exponential time complexity. In this paper, we give polynomial time algorithms for the case where the attention is restricted to polylines in which consecutive line segments make angles that are multiples of 90° or 45°, respectively. For the more complicated case of 45°-polylines, we also introduce the polar-coordinate representation of double-cross matrices.

Keywords: Spatial reasoning, Double-cross calculus, Qualitative description of polylines, Computational algebraic geometry, Algorithmic complexity

[1]Corresponding author: `bart.kuijpers@uhasselt.be`

1. Introduction and summary of results

Polylines arise in Geographical Information Science (GIS) in a multitude of ways. One recent example comes from the collection of moving object data, where trajectories of moving persons (or animals), that carry GPS-equipped devices, are collected in the form of time-space points that are registered at certain (ir)regular moments in time. The spatial trace of this movement is a collection of points in two-dimensional geographical space, that form a polyline, when in between the measured sample points, for instance, linear interpolation is applied (Güting and Schneider (2005)). Another example comes from shape recognition and retrieval, which arises in domains, such as computer vision and image analysis. Here, closed polylines or polygons, often occur as the boundary of two-dimensional shapes or regions.

In examples, such as the above, there are, roughly speaking, two very distinct approaches to deal with polygonal curves and shapes. On the one hand, there are the *quantitative* approaches and on the other hand, there are the *qualitative* approaches. Initially, most research efforts have dealt with the quantitative methods (Bookstein (1986); Dryden and Mardia (1998); Kent and Mardia (1986); Mokhtarian and Mackworth (1992)). Only afterwards, the qualitative approaches have gained more attention, mainly supported by research in cognitive science that provides evidence that qualitative models of shape representation are much more expressive than their quantitative counterpart and reflect better the way in which humans reason about their environment (Gero (1999)). The principles behind qualitative approaches to deal with polylines are also related to the field of spatial reasoning, which has as one of its main objectives to present geographic information in a qualitative way to be able to reason about it (see, for example, Chapter 12 in (Giannotti and Pedreschi (2008)), also for spatio-temporal reasoning). The reason for using a *qualitative representation* is that the available information is often imprecise, partial and subjective (Freksa (1992)). If we return to the example of trajectory data, we can see that for navigational problems, a person will remember: "I left the station and went straight; passing a church to my right; then taking two left turns; ...", rather than precise metric information about her/his spatial environment and trajectory.

One of the formalisms to qualitatively describe polylines in the plane is given by the *double-cross calculus*. In this method, a *double-cross matrix* captures the relative position (or orientation) of any two line segments in a polyline by describing it with respect to a double cross based on the starting points of these line segments (Freksa (1992); Zimmermann and Freksa (1996)). For an overview of the use of constraint calculi in qualitative spatial reasoning, we refer to (Renz and Nebel (2007)). In the $N \times N$ double-cross matrix of a polyline with N line segments (or $N+1$ vertices), the relative position (or orientation) of two (oriented) line segments is encoded by means of a 4-tuple, whose entries come from the set $\{-, 0, +\}$.

However, not every $N \times N$ matrix of 4-tuples from $\{-, 0, +\}$ is the double-cross matrix of a polyline with $N+1$ vertices. This gives rise to the following decision problem: *Given an $N \times N$ matrix of 4-tuples from $\{-, 0, +\}$, decide whether it is the double-cross matrix of a polyline (with $N+1$ vertices), and if it is, given an example (or many examples) of a polyline that realises the matrix.*

To start with, we give a known collection of polynomial (in)equalities on the coordinates of the vertices of a polyline, that express the information contained in the double-cross matrix of a polyline. Since first-order logic over the reals (or elementary geometry) is decidable (Tarski (1951)), it follows that our decision problem is also decidable. However, we are left with the question of its time complexity.

In computational algebraic geometry, the problem can be viewed as a satisfiability problem of a system of quadratic equations in $2(N+1)$ variables. However, the known best algorithms to solve our problem (including the production of sample points) takes exponential time. Our decision problem has many particularities (the polynomials are homogeneous of degree 2; they use few monomials and each of them uses only six variables), nevertheless the problem is known to be NP-hard. Whether or not this problem is in NP is less obvious, since no apriori polynomial bound on the complexity of sample points (to be guessed) is obvious. We discuss this problem in more detail in Section 3.

In this paper, we focus on subclasses of the above decision problem for which we can give polynomial time decision algorithms. A first subclass is obtained by restricting the attention to polylines in which consecutive line segments make angles that are multiples of 90°. For this sub-problem, we give a $O(N^2)$-time decision procedure. Next, we turn our attention to polylines in which consecutive line segments make angles that are multiples of 45°. To solve the more complicated case of 45°-polylines, we introduce the polar-coordinate representation of double-cross matrices. We give two-way translations between the Cartesian- and the polar-coordinate representations. Using polar coordinates, our decision problem can be reduced to a linear programming problem, with algebraic coefficients, however. Also here, we obtain a polynomial time decision procedure. This result has some implications on the convexity of the solution set consisting of all 45°-polylines that realise a matrix. It is not the intention of this paper to discuss implementations of and experiments with the proposed methods.

Organization. This paper is organized as follows. Section 2 gives the definition of a polyline, the double-cross matrix of a polyline and the known results on the algebraic interpretation of the double-cross matrix. In Section 3, we state our decision problem in a more technical way and discuss some of its general properties. Section 4 gives a $O(N^2)$-time decision procedure for the case of 90°-polylines. In Section 5, we introduce the polar-coordinate representation of double-cross matrices, which prepares the solution for the 45°-case in Section 6. The paper ends with concluding remarks that include variants of our decision problem.

2. Definition and preliminaries

In this section, we give the definitions of a polyline, an α-polyline and of the double-cross matrix of a polyline. We also give an algebraic interpretation of the double-cross matrix.

We start with the following notational conventions. Let $\mathbf{R}$ denote the sets of the real numbers, and let $\mathbf{R}^2$ denote the two-dimensional real plane. To stress that some real values are constants, we use sans serif characters: $\mathsf{x}, \mathsf{y}, \mathsf{x}_0, \mathsf{y}_0, \mathsf{x}_1, \mathsf{y}_1, \ldots$. Real variables are denoted in normal characters. For constant points of $\mathbf{R}^2$, we use the sans serif characters $\mathsf{p}, \mathsf{p}_0, \mathsf{p}_1, \ldots$

2.1. Polylines and α-polylines

The following definition specifies what we mean by *polylines*. We define polylines as a finite sequences of points in $\mathbf{R}^2$ (which is often used as their finite representation). When we add the line segments between consecutive points we obtain what we call the *semantics* of the polyline. We also introduce some terminology about polylines.

Definition 1. A *polyline* (in $\mathbf{R}^2$) is an ordered list $P = \langle(\mathsf{x}_0, \mathsf{y}_0), (\mathsf{x}_1, \mathsf{y}_1), \ldots, (\mathsf{x}_N, \mathsf{y}_N)\rangle$ of points in $\mathbf{R}^2$. We call the points $(\mathsf{x}_i, \mathsf{y}_i)$, $0 \le i \le N$, the *vertices* of the polyline. We assume that no two consecutive vertices are identical, that is: $(\mathsf{x}_i, \mathsf{y}_i) \neq (\mathsf{x}_{i+1}, \mathsf{y}_{i+1})$, for $0 \le i < N$.

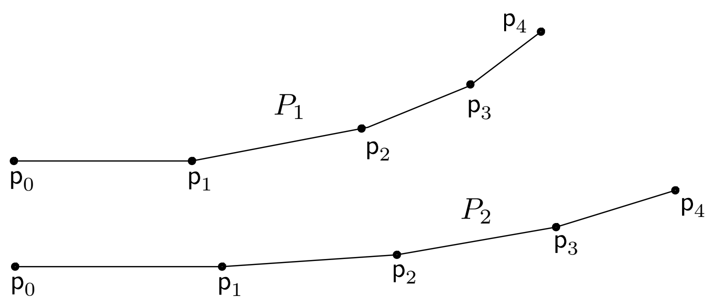

Figure 1: An example of two polylines, P_1 and P_2, of size 4 (the dots) and their semantics (the lines).

The vertices $(\mathsf{x}_0, \mathsf{y}_0)$ and $(\mathsf{x}_N, \mathsf{y}_N)$ are respectively called the *start* and *end vertex* of P. The line segments connecting the points $(\mathsf{x}_i, \mathsf{y}_i)$ and $(\mathsf{x}_{i+1}, \mathsf{y}_{i+1})$, for $0 \leq i < N$, are called the *(line) segments* of the polyline P. The *semantics* of P, denoted $\mathsf{sem}(P)$, is the union of the line segments of P. We call N, the number of line segments, the *size* of the polyline P. □

Figure 1 gives an example of two polylines, P_1 and P_2, of size 4 and their semantics. Further on, we will loosely use the term polyline also to refer to the semantics of a polyline, although, *stricto sensu*, a polyline is a ordered list of points in $\mathbf{R}^2$.

We remark that, by the above definition, two polylines with a different number of vertices, may have the same semantics. We also remark that the line segments, appearing in the semantics, may intersect in points which may or may be not vertices. Finally, we remark that it is reasonable to assume that polylines coming from GIS applications have vertices with rational coordinates (or that are finitely representable in some other way).

We use the following additional notational conventions. As a standard, for vertices of a polyline, we abbreviate $(\mathsf{x}_i, \mathsf{y}_i)$ by p_i. The *(located) vector*[2] from p_i to p_j is denoted by $\overrightarrow{\mathsf{p}_i\mathsf{p}_j}$. The counter-clockwise angle (expressed in degrees) measured from $\overrightarrow{\mathsf{p}_i\mathsf{p}_j}$ to $\overrightarrow{\mathsf{p}_i\mathsf{p}_k}$ is denoted by $\angle(\overrightarrow{\mathsf{p}_i\mathsf{p}_j}, \overrightarrow{\mathsf{p}_i\mathsf{p}_k})$, as illustrated in Figure 2.

[2]By the *located vector* from p to q, we mean an ordered pair (p, q) of points of $\mathbf{R}^2$, which we denote $\overrightarrow{\mathsf{pq}}$. We use this concept to represent the oriented line segment between p and q.

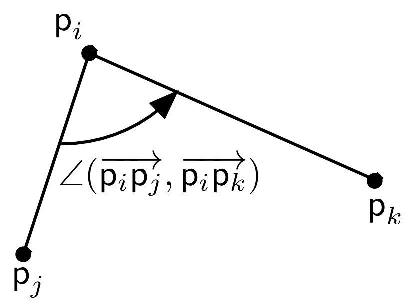

Figure 2: The counter-clockwise angle $\angle(\overrightarrow{\mathsf{p}_i\mathsf{p}_j}, \overrightarrow{\mathsf{p}_i\mathsf{p}_k})$ from $\overrightarrow{\mathsf{p}_i\mathsf{p}_j}$ to $\overrightarrow{\mathsf{p}_i\mathsf{p}_k}$.

In this paper, we use 45°- and 90°-polylines, which are special cases of α-polylines

Definition 2. Let α, $0° < \alpha < 360°$, be an angle such that $\frac{360°}{\alpha} = k_\alpha$ is a natural number. Let $P = \langle \mathsf{p}_0, \mathsf{p}_1, \ldots, \mathsf{p}_N \rangle$ be a polyline. We call P an *α-polyline* if all angles $\angle(\overrightarrow{\mathsf{p}_i\mathsf{p}_{i-1}}, \overrightarrow{\mathsf{p}_i\mathsf{p}_{i+1}})$ are multiples of α, for $0 < i < N$, that is, if $\angle(\overrightarrow{\mathsf{p}_i\mathsf{p}_{i-1}}, \overrightarrow{\mathsf{p}_i\mathsf{p}_{i+1}})$ is of the form $n_i\alpha$, with $n_i \in \{0, 1, ..., k_\alpha\}$. □

Figure 3 shows the 90°-polyline P_1 and the 45°-polylines P_1 and P_2. Indeed, in the polyline P_1, for instance, the consecutive angles are $90°, 90°, 270°$ and $270°$, assuming that the start vertex is at the left bottom.

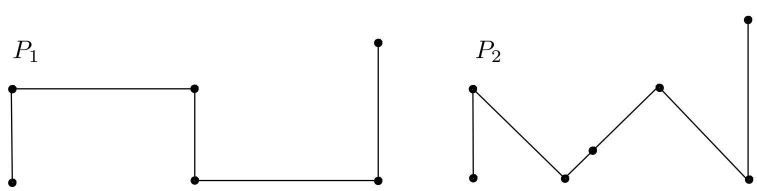

Figure 3: An example of a 90°-polyline (P_1) and two 45°-polylines (P_1 and P_2).

2.2. The double-cross matrix of a polyline

As mentioned in the Introduction, in the double-cross formalism, the relative position (or orientation) of two (located) vectors of a polyline is encoded by means of a 4-tuple, whose entries come from the set $\{-, 0, +\}$ (Freksa (1992); Zimmermann and Freksa (1996)). Such a 4-tuple expresses the relative orientation of two vectors with respect to each other.

In this section, we define the *double-cross matrix* of a polyline. We associate to a polyline $P = \langle(\mathsf{x}_0, \mathsf{y}_0), (\mathsf{x}_1, \mathsf{y}_1), , \ldots, (\mathsf{x}_N, \mathsf{y}_N)\rangle$, with $\mathsf{p}_i = (\mathsf{x}_i, \mathsf{y}_i)$, the (located) vectors $\overrightarrow{\mathsf{p}_0\mathsf{p}_1}, \overrightarrow{\mathsf{p}_1\mathsf{p}_2}, \ldots, \overrightarrow{\mathsf{p}_{N-1}\mathsf{p}_N}$, representing the oriented line segments between the consecutive vertices of P. Because of the assumption in Definition 1, the vectors $\overrightarrow{\mathsf{p}_0\mathsf{p}_1}, \overrightarrow{\mathsf{p}_1\mathsf{p}_2}, \ldots, \overrightarrow{\mathsf{p}_{N-1}\mathsf{p}_N}$ all have a strictly positive length. In the double-cross formalism, the relative orientation between $\overrightarrow{\mathsf{p}_i\mathsf{p}_{i+1}}$ and $\overrightarrow{\mathsf{p}_j\mathsf{p}_{j+1}}$ is given by means of a 4-tuple $(C_1\ C_2\ C_3\ C_4) \in \{-, 0, +\}^4$. We follow the traditional notation of this 4-tuple *without commas*. To determine C_1, C_2, C_3 and C_4, for $\mathsf{p}_i \neq \mathsf{p}_j$, first of all, a *double cross* is defined for the vectors $\overrightarrow{\mathsf{p}_i\mathsf{p}_{i+1}}$ and $\overrightarrow{\mathsf{p}_j\mathsf{p}_{j+1}}$, determined by the following three lines:

- the line L_{ij} through p_i and p_j;
- the line P_{iji} through p_i, perpendicular on L_{ij}; and
- the line P_{ijj} through p_j, perpendicular on L_{ij}.

These three lines are illustrated in Figure 4. These three lines determine a cross at p_i and a cross at p_j. Hence the name "double cross." The entries C_1, C_2, C_3 and C_4 express in which quadrants or on which half lines p_{i+1} and p_{j+1} are located with respect to the double cross.

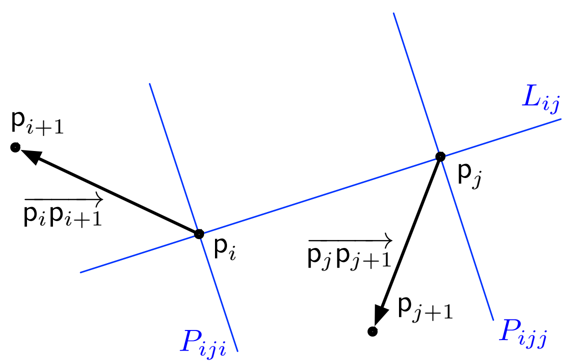

Figure 4: The double cross (in blue): the lines L_{ij}, P_{iji} and P_{ijj}.

We now define this more formally and follow the historical use of the double cross. In this definition, an interval (a, b) of angles, represents the open interval between a and b on the counter-clockwise oriented circle.

Definition 3. Let $P = \langle(\mathsf{x}_0, \mathsf{y}_0), (\mathsf{x}_1, \mathsf{y}_1), , \ldots, (\mathsf{x}_N, \mathsf{y}_N)\rangle$ be a polyline, with $\mathsf{p}_i = (\mathsf{x}_i, \mathsf{y}_i)$, for $0 \leq i \leq N$, and with associated vectors $\overrightarrow{\mathsf{p}_0\mathsf{p}_1}, \overrightarrow{\mathsf{p}_1\mathsf{p}_2}, \ldots, \overrightarrow{\mathsf{p}_{N-1}\mathsf{p}_N}$. For $\overrightarrow{\mathsf{p}_i\mathsf{p}_{i+1}}$ and $\overrightarrow{\mathsf{p}_j\mathsf{p}_{j+1}}$ with $0 \leq i, j < N$, $i \neq j$ and $\mathsf{p}_i \neq \mathsf{p}_j$, we define

$$\mathsf{DC}(\overrightarrow{\mathsf{p}_i\mathsf{p}_{i+1}}, \overrightarrow{\mathsf{p}_j\mathsf{p}_{j+1}}) = (C_1\ C_2\ C_3\ C_4)$$

as follows:

$$C_1 \quad = \quad \begin{cases} - & \text{if} \quad \angle(\overrightarrow{\mathsf{p}_i\mathsf{p}_j}, \overrightarrow{\mathsf{p}_i\mathsf{p}_{i+1}}) \in (-90^\circ, 90^\circ) \\ 0 & \text{if} \quad \angle(\overrightarrow{\mathsf{p}_i\mathsf{p}_j}, \overrightarrow{\mathsf{p}_i\mathsf{p}_{i+1}}) \in \{-90^\circ, 90^\circ\} \\ + & \text{else} \end{cases}$$

$$C_2 \quad = \quad \begin{cases} - & \text{if} \quad \angle(\overrightarrow{\mathsf{p}_j\mathsf{p}_i}, \overrightarrow{\mathsf{p}_j\mathsf{p}_{j+1}}) \in (-90^\circ, 90^\circ) \\ 0 & \text{if} \quad \angle(\overrightarrow{\mathsf{p}_j\mathsf{p}_i}, \overrightarrow{\mathsf{p}_j\mathsf{p}_{j+1}}) \in \{-90^\circ, 90^\circ\} \\ + & \text{else} \end{cases}$$

$$C_3 \quad = \quad \begin{cases} - & \text{if} \quad \angle(\overrightarrow{\mathsf{p}_i\mathsf{p}_j}, \overrightarrow{\mathsf{p}_i\mathsf{p}_{i+1}}) \in (0^\circ, 180^\circ) \\ 0 & \text{if} \quad \angle(\overrightarrow{\mathsf{p}_i\mathsf{p}_j}, \overrightarrow{\mathsf{p}_i\mathsf{p}_{i+1}}) \in \{0^\circ, 180^\circ\} \\ + & \text{else} \end{cases}$$

$$C_4 \quad = \quad \begin{cases} - & \text{if} \quad \angle(\overrightarrow{\mathsf{p}_j\mathsf{p}_i}, \overrightarrow{\mathsf{p}_j\mathsf{p}_{j+1}}) \in (0^\circ, 180^\circ) \\ 0 & \text{if} \quad \angle(\overrightarrow{\mathsf{p}_j\mathsf{p}_i}, \overrightarrow{\mathsf{p}_j\mathsf{p}_{j+1}}) \in \{0^\circ, 180^\circ\} \\ + & \text{else.} \end{cases}$$

For $\overrightarrow{\mathsf{p}_i\mathsf{p}_{i+1}}$ and $\overrightarrow{\mathsf{p}_j\mathsf{p}_{j+1}}$, with $\mathsf{p}_i = \mathsf{p}_j$, we define, for reasons of continuity,[3]

$$\mathsf{DC}(\overrightarrow{\mathsf{p}_i\mathsf{p}_{i+1}}, \overrightarrow{\mathsf{p}_j\mathsf{p}_{j+1}}) = (0\ 0\ 0\ 0).$$

The *double-cross matrix* of P, denoted $\mathsf{DCM}(P)$, is the $N \times N$ matrix with the entries $\mathsf{DCM}(P)[i, j] = \mathsf{DC}(\overrightarrow{p_ip_{i+1}}, \overrightarrow{p_jp_{j+1}})$, for $0 \leq i, j < N$. □

So, in particular, when $i = j$, we have $\mathsf{DC}(\overrightarrow{\mathsf{p}_i\mathsf{p}_{i+1}}, \overrightarrow{\mathsf{p}_j\mathsf{p}_{j+1}}) = (0\ 0\ 0\ 0)$.

We remark that the values C_1 and C_3 describe the location of the point p_{i+1} or, equivalently, the orientation of the vector $\overrightarrow{\mathsf{p}_i\mathsf{p}_{i+1}}$ with respect to the cross at p_i (formed by the lines L_{ij} and P_{iji}). We see that each of the four quadrants and four half lines determined by the cross at p_i are determined by a unique combination of C_1 and C_3 values. Similarly, the values C_2 and

[3]This argumentation is given in (Forbus (1990)).

C_4 describe the location of the point p_{j+1} or, equivalently, the orientation of the vector $\overrightarrow{\mathsf{p}_j\mathsf{p}_{j+1}}$ with respect to the cross at p_j (formed by the lines L_{ij} and P_{ijj}). The quadrants and half lines where C_1, C_2, C_3 and C_4 take different values are graphically illustrated in Figure 5. For example, the 4-tuple $\mathsf{DC}(\overrightarrow{\mathsf{p}_i\mathsf{p}_{i+1}}, \overrightarrow{\mathsf{p}_j\mathsf{p}_{j+1}})$ for the vectors $\overrightarrow{\mathsf{p}_i\mathsf{p}_{i+1}}$ and $\overrightarrow{\mathsf{p}_j\mathsf{p}_{j+1}}$, shown in Figure 4, is $(+\ -\ -\ -)$.

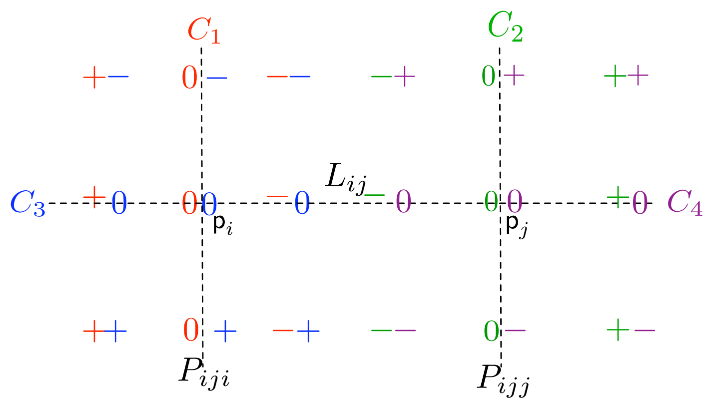

Figure 5: The quadrants and half lines where C_1, C_2, C_3 and C_4 take different values.

For example, the entries of the double-cross matrix of the polylines P_1 and P_2 of Figure 1 are given in Table 1. Polylines, such as P_1 and P_2 of Figure 1, that have the same double-cross matrix, are called *double-cross similar*.

This first example can be used to illustrate some properties of this matrix (Kuijpers et al. (2006)). First, we observe that on the diagonal always $(0\ \ 0\ \ 0\ \ 0)$ appears. We also see that there is a certain degree of *symmetry along the diagonal*. If $\mathsf{DC}(\overrightarrow{\mathsf{p}_i\mathsf{p}_{i+1}}, \overrightarrow{\mathsf{p}_j\mathsf{p}_{j+1}}) = (C_1\ C_2\ C_3\ C_4)$, then we have $\mathsf{DC}(\overrightarrow{\mathsf{p}_j\mathsf{p}_{j+1}}, \overrightarrow{\mathsf{p}_i\mathsf{p}_{i+1}}) = (C_2\ C_1\ C_4\ C_3)$. These two observations imply that it suffices to know a double-cross matrix *above its diagonal*.

Input matrices, that do not posses these symmetry properties, are therefor, apriori, *not* realisable.

2.3. An algebraic interpretation of the double-cross matrix

In this section, we give an algebraic interpretation of the double-cross matrix. In the following theorem, taken from (Kuijpers et al. (2006)), we use

	$\overrightarrow{\mathsf{p}_0\mathsf{p}_1}$	$\overrightarrow{\mathsf{p}_1\mathsf{p}_2}$	$\overrightarrow{\mathsf{p}_2\mathsf{p}_3}$	$\overrightarrow{\mathsf{p}_3\mathsf{p}_4}$
$\overrightarrow{\mathsf{p}_0\mathsf{p}_1}$	$(0\ 0\ 0\ 0)$	$(-\ +\ 0\ +)$	$(-\ +\ +\ +)$	$(-\ +\ +\ +)$
$\overrightarrow{\mathsf{p}_1\mathsf{p}_2}$	$(+\ -\ +\ 0)$	$(0\ 0\ 0\ 0)$	$(-\ +\ 0\ +)$	$(-\ +\ +\ +)$
$\overrightarrow{\mathsf{p}_2\mathsf{p}_3}$	$(+\ -\ +\ +)$	$(+\ -\ +\ 0)$	$(0\ 0\ 0\ 0)$	$(-\ +\ 0\ +)$
$\overrightarrow{\mathsf{p}_3\mathsf{p}_4}$	$(+\ -\ +\ +)$	$(+\ -\ +\ +)$	$(+\ -\ +\ 0)$	$(0\ 0\ 0\ 0)$

Table 1: The entries of the double-cross matrix of the polylines P_1 and P_2 of Figure 1.

the function

$$\text{sign} : \mathbf{R} \to \{-, 0, +\} : x \mapsto \text{sign}(x) = \begin{cases} - & \text{if} \quad x < 0; \\ 0 & \text{if} \quad x = 0; \text{ and} \\ + & \text{if} \quad x > 0. \end{cases}$$

We also work with the following convention concerning signs: $--$ is $+$; -0 is 0; and $-+$ is $-$.

Theorem 1. *Let $P = \langle (\mathsf{x}_0, \mathsf{y}_0), (\mathsf{x}_1, \mathsf{y}_1), , \ldots, (\mathsf{x}_N, \mathsf{y}_N) \rangle$ be a polyline and let $\mathsf{p}_i = (\mathsf{x}_i, \mathsf{y}_i)$, for $0 \leq i \leq N$. Then, $\mathsf{DC}(\overrightarrow{\mathsf{p}_i\mathsf{p}_{i+1}}, \overrightarrow{\mathsf{p}_j\mathsf{p}_{j+1}}) = (C_1\ C_2\ C_3\ C_4)$ with*

$$\begin{array}{lcll} C_1 & = & - & \text{sign}((\mathsf{x}_j - \mathsf{x}_i) \cdot (\mathsf{x}_{i+1} - \mathsf{x}_i) + (\mathsf{y}_j - \mathsf{y}_i) \cdot (\mathsf{y}_{i+1} - \mathsf{y}_i)); \\ C_2 & = & & \text{sign}((\mathsf{x}_j - \mathsf{x}_i) \cdot (\mathsf{x}_{j+1} - \mathsf{x}_j) + (\mathsf{y}_j - \mathsf{y}_i) \cdot (\mathsf{y}_{j+1} - \mathsf{y}_j)); \\ C_3 & = & - & \text{sign}((\mathsf{x}_j - \mathsf{x}_i) \cdot (\mathsf{y}_{i+1} - \mathsf{y}_i) - (\mathsf{y}_j - \mathsf{y}_i) \cdot (\mathsf{x}_{i+1} - \mathsf{x}_i)); \textit{ and} \\ C_4 & = & & \text{sign}((\mathsf{x}_j - \mathsf{x}_i) \cdot (\mathsf{y}_{j+1} - \mathsf{y}_j) - (\mathsf{y}_j - \mathsf{y}_i) \cdot (\mathsf{x}_{j+1} - \mathsf{x}_j)). \end{array}$$

□

3. Problem statement and discussion

In this section, we state the decision problem, already given in the Introduction, more formally and we devote some theoretical discussion to it.

3.1. Problem statement

In this papers, we address the following decision problem (relative to some class $\mathcal{P}$ of polylines in the plane $\mathbf{R}^2$).

Problem 1 (Realisability). Given is an $N \times N$ matrix M of 4-tuples

$$(C_1\ C_2\ C_3\ C_4) \in \{-, 0, +\}^4.$$

(a) Decide whether M is the double-cross matrix of some polyline (from a class $\mathcal{P}$) in $\mathbf{R}^2$ of size N; and

(b) If the answer to question (a) is *yes*, then produce an example (or many examples) of a polyline P with $\mathsf{DCM}(P) = M$. □

Initially, we take the class of polylines $\mathcal{P}$ as broad as possible. For instance, when we look at polylines that have, as first line segment, the unit interval on the x-axis of $\mathbf{R}^2$, it is sufficient to look at all polylines whose vertices have algebraic coordinates.

3.2. Discussion

By Theorem 1, the entries of an input matrix M to Problem 1 can be translated into sign conditions on quadratic polynomial equalities and inequalities. Therefor, Problem 1 is equivalent to deciding the first-order sentence

$$\exists x_0 \exists y_0 \exists x_1 \exists y_1 \cdots \exists x_N \exists y_N \bigwedge_{0 \le i < j \le N} \left\{ \begin{array}{lll} (x_j - x_i) \cdot (x_{i+1} - x_i) + (y_j - y_i) \cdot (y_{i+1} - y_i) & \alpha_{ij} & 0 \\ (x_j - x_i) \cdot (x_{j+1} - x_j) + (y_j - y_i) \cdot (y_{j+1} - y_j) & \beta_{ij} & 0 \\ (x_j - x_i) \cdot (y_{i+1} - y_i) - (y_j - y_i) \cdot (x_{i+1} - x_i) & \gamma_{ij} & 0 \\ (x_j - x_i) \cdot (y_{j+1} - y_j) - (y_j - y_i) \cdot (x_{j+1} - x_j) & \delta_{ij} & 0 \end{array} \right.$$

where $\alpha_{ij}, \beta_{ij}, \gamma_{ij}, \delta_{ij} \in \{=, <, >\}$, for $0 \le i < j \le N$ are determined by the input matrix M, over the reals. The minus signs before the equations for C_1 and C_3 are assumed to be incorporated in the α_{ij} and γ_{ij}. We remark that the above sentence expresses the entries of the input matrix strictly above its diagonal (as we can apriori discard non-symmetric input matrices).

The $4\frac{N(N-1)}{2}$ equalities and inequalities describe a semi-algebraic subsets of $\mathbf{R}^{2(N+1)}$ (Bochnak et al. (1998)). We make the following observations about this system:

- there are $2N(N-1)$ (in)equalities in $2(N+1)$ variables $x_0, y_0, ..., x_N, y_N$;
- each polynomial uses 6 variables from $x_0, y_0, x_1, y_1, ..., x_N, y_N$ and has at most 8 monomial terms;
- each of the polynomials is homogeneous of degree 2;
- all the coefficients of the polynomials are 0, 1 or -1.

The first-order theory of the real ordered field is decidable (Tarski (1951)) and various implementations of decision procedures, that are based on Cylindrical Algebraic Decomposition (Collins (1975)) or other techniques, for this theory exist. We refer to QEPCAD (Hong (2000)), REDLOG (Dolzmann and Sturm (1997)) and and MATHEMATICA (Wolfram Research (2015)) as a few examples. This type of software could be used, in theory, to answer Problem 1 (a) in practice. If there is a solution, these implementations also provide, as a byproduct of the above decision problem, sample points, thus, also, effectively answering question Problem 1 (b). But it is also known that the above mentioned implementations are slow and fail in practice to produce answers as soon as the number of variables increases. This is due to the intrinsic high time complexity of quantifier elimination in the ordered field of the reals (Heintz et al. (2013)). The theory of computational algebraic geometry gives an upper complexity bound. In particular, Theorem 13.13 in (Basu et al. (2006)) gives an upper bound on determining realisable sign conditions of a collection of polynomials. When applied to our decision problem, we obtain that there exists an algorithm to compute the set of *all* realisable sign conditions of the above system of polynomial (in)equalities with complexity $(2N(N-1))^{2N+3} \cdot 2^{O(N)}$. The complexity of deciding the satisfiability of the system is the same, as well as that of generating a sample point in case of non-emptiness. The use of alternative data structures to codify the polynomials can improve the time complexity, but not below exponential time (Giusti and Heintz (2001)). For a more recent discussion on lower bounds of the complexity, we refer to (Heintz et al. (2013)).

However, we have the following, negative result: Problem 1 (a) is NP-hard (Scivos and Nebel (2001); Renz and Nebel (2007)). Whether or not this problem is in NP is less obvious. It is known that if there is a solution to the above system of polynomial (in)equalities, there is also an solution with algebraic coordinates (Basu et al. (2006)). We could, for instance, try to guess the coordinates of the vertices of a polyline and then verify whether it satisfies the above system. Guessing algebraic coordinates could be implemented by guessing a polynomial and a root of this polynomial. However, an apriori polynomial bound on the complexity of sample points (to be guessed) is not obvious (Basu et al. (2006)). Above, we have observed that each polynomial uses at most 6 variables from $x_0, y_0, x_1, y_1, ..., x_N, y_N$ and has at most 8 monomial terms. This implies our problem is part of the field of "fewnomials" (Khovanskii (1991)), where problems are notoriously difficult. And our problem and the production of sample points, is not covered by the known solutions there.

On the positive side, we can remark that, from the definition of the double-cross matrix in Section 2.2, it is clear that translations, rotations and scalings of a polyline do not change its double-cross matrix. Double-cross matrices are, in fact, invariant under *similarities* of $\mathbf{R}^2$. Thus, we can conclude, that if Problem 1 (a) has a positive answer, we can always find a polyline, to witness this fact, that starts of with the vertices $(\mathsf{x}_0, \mathsf{y}_0) = (0, 0)$ and $(\mathsf{x}_1, \mathsf{y}_1) = (1, 0)$ and in which the other vertices have coordinates that are algebraic numbers.

4. A realisability test for 90°-polylines

In this section, we give an efficient solution for a special case of Problem 1, for $\mathcal{P} = \mathcal{P}_{90^\circ}$, the class of 90°-polylines (again with vertices with algebraic coordinates). As we have remarked, for the problem of realisability, we may assume, without loss of any generality, that the polyline that realises a matrix M, if it exists, starts with the unit interval on the x-axis, that is, $\mathsf{p}_0 = (\mathsf{x}_0, \mathsf{y}_0) = (0, 0)$ and $\mathsf{p}_1 = (\mathsf{x}_1, \mathsf{y}_1) = (1, 0)$.

The following, straightforward, property gives a first necessary condition for the input to our decision problem, the matrix M.

Property 1. *Let $P = \langle \mathsf{p}_0, \mathsf{p}_1, \mathsf{p}_2, ..., \mathsf{p}_N \rangle$ be a polyline. A necessary and sufficient condition for P to be a 90°-polyline is that for all i, $0 \leq i < N - 1$,* $\mathsf{DC}(\overrightarrow{\mathsf{p}_i \mathsf{p}_{i+1}}, \overrightarrow{\mathsf{p}_{i+1} \mathsf{p}_{i+2}}) =$

- $(-\ -\ 0\ 0)$ *(reverse turn);*
- $(-\ 0\ 0\ -)$ *(right turn);*
- $(-\ +\ 0\ 0)$ *(straight); or*
- $(-\ 0\ 0\ +)$ *(left turn).* □

Since we take $\mathsf{p}_0 = (\mathsf{x}_0, \mathsf{y}_0) = (0, 0)$ and $\mathsf{p}_1 = (\mathsf{x}_1, \mathsf{y}_1) = (1, 0)$, all line segments of the polyline, realising M, should be or horizontal or vertical in $\mathbf{R}^2$ with respect to the standard coordinate axes. In fact, we have for each i, $0 \leq i < N$ that $\mathsf{x}_i = \mathsf{x}_{i+1} \wedge (\mathsf{y}_i < \mathsf{y}_{i+1} \vee \mathsf{y}_i > \mathsf{y}_{i+1})$ or $\mathsf{y}_i = \mathsf{y}_{i+1} \wedge (\mathsf{x}_i < \mathsf{x}_{i+1} \vee \mathsf{x}_i > \mathsf{x}_{i+1})$. Here, for $0 \leq i < N$, we are in exactly one of the following four situations (always with $\ell_{i+1} > 0$):

$$\begin{cases} x_{i+1} = x_i + \ell_{i+1} \\ y_{i+1} = y_i; \end{cases} \quad \begin{cases} x_{i+1} = x_i - \ell_{i+1} \\ y_{i+1} = y_i; \end{cases} \quad \begin{cases} x_{i+1} = x_i \\ y_{i+1} = y_i + \ell_{i+1}; \end{cases} \quad \begin{cases} x_{i+1} = x_i \\ y_{i+1} = y_i - \ell_{i+1}. \end{cases}$$

Before we give an efficient solution to Problem 1 for for $\mathcal{P} = \mathcal{P}_{90^\circ}$, we prove a lemma that explains in which quadrant, determined by a horizontal or vertical line segment of a polyline, a vertex of a polyline is located. For clarity, we state and prove the lemma for horizontal and vertical segments of a polyline that coincide with the unit interval on the x- and y-axis (or their negatives), but the lemma can be easily extended and applied to any horizontal and vertical polyline segments after applying a scaling and translation of $\mathbf{R}^2$.

Lemma 1. *Let $P = \langle \mathsf{p}_0, \mathsf{p}_1, , \ldots, \mathsf{p}_N \rangle$ be a polyline and assume that $\mathsf{p}_i = (0,0)$ and $\mathsf{p}_{i+1} = (\pm 1, 0)$ or $\mathsf{p}_i = (0,0)$ and $\mathsf{p}_{i+1} = (0, \pm 1)$. Let $\mathsf{p}_j = (\mathsf{x}_j, \mathsf{y}_j)$, for $0 \leq i \leq N$ and $i+1 < j$. From the first and third component of $\mathsf{DC}(\overrightarrow{\mathsf{p}_i \mathsf{p}_{i+1}}, \overrightarrow{\mathsf{p}_j \mathsf{p}_{j+1}})$, we can determine $\mathrm{sign}(\mathsf{x}_j)$ and $\mathrm{sign}(\mathsf{y}_j)$.*

Proof. First, let $\mathsf{p}_i = (0,0)$ and $\mathsf{p}_{i+1} = (\pm 1, 0)$. From Theorem 1 it is clear that $C_1 = -\mathrm{sign}((\mathsf{x}_j - 0) \cdot \pm 1 + \mathsf{y}_j \cdot 0) = -\mathrm{sign}(\pm \mathsf{x}_j)$ and that $C_3 = -\mathrm{sign}(\mathsf{x}_j \cdot 0 - \mathsf{y}_j \cdot \pm 1) = \mathrm{sign}(\pm \mathsf{y}_j)$.

Secondly, let $\mathsf{p}_i = (0,0)$ and $\mathsf{p}_{i+1} = (0, \pm 1)$. Similarly, Theorem 1 implies $C_1 = -\mathrm{sign}((\mathsf{x}_j - 0) \cdot 0 + \mathsf{y}_j \cdot \pm 1) = -\mathrm{sign}(\pm \mathsf{y}_j)$ and that $C_3 = -\mathrm{sign}(\mathsf{x}_j \cdot \pm 1 - \mathsf{y}_j \cdot 0) = -\mathrm{sign}(\pm \mathsf{x}_j)$. $\square$

Theorem 2. *It can be decided in time $O(N^2)$ whether a $N \times N$ matrix M of 4-tuples $(C_1\ C_2\ C_3\ C_4) \in \{-, 0, +\}^4$ is the double-cross matrix of some 90°-polyline in $\mathbf{R}^2$ of size N. If M this is the case, also witnesses to this can be produced in time $O(N^2)$.*

Proof. We now describe a decision procedure for Problem 1: in a first step, we determine the relationship $(<, =, >)$ between coordinates of consecutive vertices. In a second step, we do it for all remaining vertices

Let M be a $N \times N$ matrix of 4-tuples $(C_1\ C_2\ C_3\ C_4) \in \{-, 0, +\}^4$. In a first step, we determine the relationship $(<, =, >)$ between coordinates of consecutive vertices. In a second step, we do it for all remaining vertices. As an apriori step, we check whether M doesn't have $(0\ 0\ 0\ 0)$ entries on its diagonal or doesn't have the "symmetry" properties, discussed in Section 2.2. If M fails this symmetry-test, we can already answer *no*, else we proceed.

Step 1. First, we inspect all entries $M[i, i+1]$, $0 \leq i < N$ of M. They should all be of the form

- $(- \ - \ 0 \ 0)$ (reverse turn);
- $(- \ 0 \ 0 \ -)$ (right turn);
- $(- \ + \ 0 \ 0)$ (straight); or
- $(- \ 0 \ 0 \ +)$ (left turn).

If this is not the case, we can already answer *no*. In the other case, we deduce the arrangement[4] of x_i and x_{i+1} of the coordinates of candidate vertices of a polyline. We do the same for y_i and y_{i+1} and determine whether $\mathsf{y}_i < \mathsf{y}_{i+1}$, $\mathsf{y}_i = \mathsf{y}_{i+1}$ or $\mathsf{y}_i > \mathsf{y}_{i+1}$. Then we proceed to Step 2.

Step 2. Now, we inspect all entries $M[i, j]$, $1 \leq i+1 < j < N$ of M. Now, per entry, two cases have to be considered.
Case 1 ($\mathsf{x}_i = \mathsf{x}_{i+1}$): Taking into account, $\mathsf{y}_i < \mathsf{y}_{i+1}$ or $\mathsf{y}_i > \mathsf{y}_{i+1}$, and $\mathsf{y}_j < \mathsf{y}_{j+1}$ or $\mathsf{y}_j > \mathsf{y}_{j+1}$, we can use the vertical version of Lemma 1, to determine the quadrant in which $(\mathsf{x}_j, \mathsf{y}_j)$ is located compared to the vertical line segment that connects p_i and p_{i+1}. This gives us the the arrangement of x_j and x_i on the one hand and y_j and y_i on the other hand.
Case 2 ($\mathsf{y}_i = \mathsf{y}_{i+1}$): This case is analogous to the previous one. We can get the arrangement of x_j and x_i on the one hand and y_j and y_i on the other hand, but now using the horizontal version of version of Lemma 1.

Now, we have now complete information on how the x-coordinate values $\mathsf{x}_0, \mathsf{x}_1, ..., \mathsf{x}_N$ are arranged (or ordered) and how the y-coordinate values $\mathsf{y}_0, \mathsf{y}_1, ..., \mathsf{y}_N$ are arranged (since, form the definition of the double-cross matrix, the length of the last line segment is irrelevant, one of x_N and y_N may be undetermined, but we know the direction of the final line segment). We can store this arrangement information in two matrices (similarly to the double-cross matrix). The first matrix can be used to verify whether an ordering of $\mathsf{x}_0, \mathsf{x}_1, ..., \mathsf{x}_N$ is possible. To this purpose, we scan the first matrix column per column. The first column will allow us to place x_0 and x_1 on the real line

[4]By *arrangement*, we mean which of the cases $\mathsf{x}_i < \mathsf{x}_{i+1}$, $\mathsf{x}_i = \mathsf{x}_{i+1}$ and $\mathsf{x}_i > \mathsf{x}_{i+1}$ holds.

(according to their arrangement). This results in at most five locations to place x_2 (before; between; after; or on x_0 and x_1). The second column of the matrix tells us where. We repeat this process until all the candidate values $\mathsf{x}_0, \mathsf{x}_1, ..., \mathsf{x}_N$ are placed on the real line. Then, we use the second matrix to place the y-coordinate values $\mathsf{y}_0, \mathsf{y}_1, ..., \mathsf{y}_N$ on the y-axis. If, in this process, we find it impossible to find a location to place one of the x_i or y_i (due to a contradiction), we answer *no*. If we have never found a contradiction ad all x- and y-values can be ordered, we are ready to answer *yes*. This ordering process takes $O(N^2)$ time.

If we have found k_x different values $\mathsf{x}_0, \mathsf{x}_1, ..., \mathsf{x}_N$ and k_y different values $\mathsf{y}_0, \mathsf{y}_1, ..., \mathsf{y}_N$, we can draw an example of a polyline that realises M on the grid $\{0, 1, ..., k_x - 1\} \times \mathbf{R} \cup \mathbf{R} \times \{0, 1, ..., k_x - 1\}$, with vertices belonging to $\{0, 1, ..., k_x - 1\} \times \{0, 1, ..., k_x - 1\}$. This drawing serves aa a sample point and answers Problem 1 (b).

It is clear that the above inspection of the matrix M takes $O(N^2)$ time. The reconstruction of a polyline can be done in the same amount of time. This completes the proof. $\square$

5. The polar coordinate representation of a polyline

In this section, we define the polar coordinate representation of a polyline and we describe how to go from the Cartesian coordinate representation to the polar coordinate representation and vice versa.

Definition 4. Let $P = \langle(\mathsf{x}_0, \mathsf{y}_0), (\mathsf{x}_1, \mathsf{y}_1), \ldots, (\mathsf{x}_N, \mathsf{y}_N)\rangle$ be a polyline (in Cartesian coordinate representation) and let $\mathsf{p}_i = (\mathsf{x}_i, \mathsf{y}_i)$, $0 \leq i \leq N$. The *polar coordinate representation* of the polyline P is the list

$$\langle \ell_1, \theta_1, \ell_2, \theta_2, ..., \ell_{N-1}, \theta_{N-1}, \ell_N \rangle,$$

where ℓ_i is the length of the line segment $\mathsf{p}_{i-1}\mathsf{p}_i$ and θ_i is the counter-clockwise angel at p_i between the line connecting p_i and p_{i-1} and the line connecting p_i and p_{i+1}. $\square$

If at p_i, the polyline turns to the left or goes straight, $\theta_i = 180° - \angle(\overrightarrow{\mathsf{p}_i\mathsf{p}_{i-1}}, \overrightarrow{\mathsf{p}_i\mathsf{p}_{i+1}})$ and if at p_i, the polyline turns to the right or returns, $\theta_i = 180° + \angle(\overrightarrow{\mathsf{p}_i\mathsf{p}_{i-1}}, \overrightarrow{\mathsf{p}_i\mathsf{p}_{i+1}})$.

So, θ_i captures the (counter-clockwise) change in direction when going from the line segment $\mathsf{p}_{i-1}\mathsf{p}_i$ to the line segment $\mathsf{p}_i\mathsf{p}_{i+1}$. This is illustrated in Figure 6.

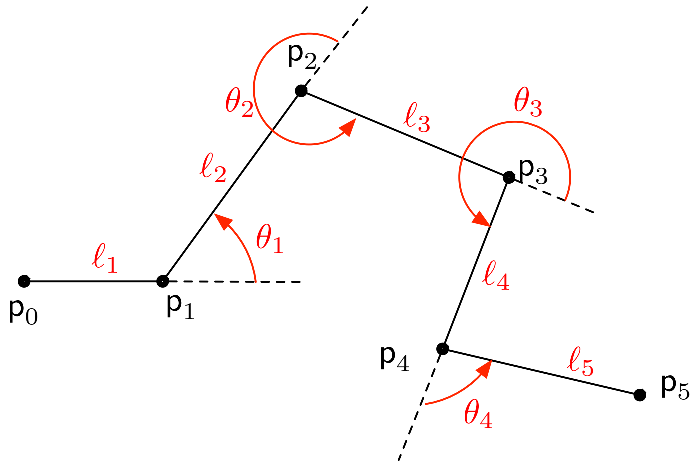

Figure 6: The polar coordinates $\langle \ell_1, \theta_1, \ell_2, \theta_2, \ell_3, \theta_3, \ell_4, \theta_4, \ell_5 \rangle$ (in red) of the polyline $\langle \mathsf{p}_0, \mathsf{p}_1, \mathsf{p}_2, \mathsf{p}_3, \mathsf{p}_4, \mathsf{p}_5 \rangle$ (in black).

5.1. From the Cartesian coordinate to the polar coordinate representation

To convert a polyline $P = \langle (\mathsf{x}_0, \mathsf{y}_0), (\mathsf{x}_1, \mathsf{y}_1), , \dots, (\mathsf{x}_N, \mathsf{y}_N) \rangle$ given by the Cartesian coordinates of its vertices to polar coordinate representation is easy. For ℓ_i, we take the length of the line segment $\mathsf{p}_{i-1}\mathsf{p}_i$. By definition $\theta_i = 180° - \angle(\overrightarrow{\mathsf{p}_i\mathsf{p}_{i-1}}, \overrightarrow{\mathsf{p}_i\mathsf{p}_{i+1}})$ if the polyline turns to the left or goes straight and $\theta_i = 180° + \angle(\overrightarrow{\mathsf{p}_i\mathsf{p}_{i-1}}, \overrightarrow{\mathsf{p}_i\mathsf{p}_{i+1}})$ if the polyline turns to the right or returns. Therefore, the angle θ_i is given by the formula

$$\pi - \arccos\left(\frac{(\mathsf{x}_{i-1} - \mathsf{x}_i, \mathsf{y}_{i-1} - \mathsf{y}_i) \cdot (\mathsf{x}_{i+1} - \mathsf{x}_i, \mathsf{y}_{i+1} - \mathsf{y}_i)}{|(\mathsf{x}_{i-1} - \mathsf{x}_i, \mathsf{y}_{i-1} - \mathsf{y}_i)| \cdot |(\mathsf{x}_{i+1} - \mathsf{x}_i, \mathsf{y}_{i+1} - \mathsf{y}_i)|}\right)$$

if the polyline turns to the left or goes straight, and by

$$\pi + \arccos\left(\frac{(\mathsf{x}_{i-1} - \mathsf{x}_i, \mathsf{y}_{i-1} - \mathsf{y}_i) \cdot (\mathsf{x}_{i+1} - \mathsf{x}_i, \mathsf{y}_{i+1} - \mathsf{y}_i)}{|(\mathsf{x}_{i-1} - \mathsf{x}_i, \mathsf{y}_{i-1} - \mathsf{y}_i)| \cdot |(\mathsf{x}_{i+1} - \mathsf{x}_i, \mathsf{y}_{i+1} - \mathsf{y}_i)|}\right)$$

if the polyline turns to the right or returns.[5]

[5]Here, the $\cdot$ in the numerator denotes the *inner product* of two vectors and the $\cdot$ in the denominator is the product of norms.

5.2. From the polar coordinate to the Cartesian coordinate representation

Now, we turn to transforming the polar coordinate representation into the classical Cartesian coordinate representation, which is more laborious. Here, we can use some techniques that are also known in the description of robot arms with multiple joints (see, for instance, Chapter 6 of (Cox et al. (1997))).

Hereto, we first need some technical results. Let $P = \langle(\mathsf{x}_0, \mathsf{y}_0), (\mathsf{x}_1, \mathsf{y}_1), \ldots, (\mathsf{x}_N, \mathsf{y}_N)\rangle$ be a polyline and let $\mathsf{p}_i = (\mathsf{x}_i, \mathsf{y}_i)$, $0 \leq i \leq N$. In each vertex $(\mathsf{x}_i, \mathsf{y}_i)$, we create a local coordinate system (X_i, Y_i). The origin of this coordinate system is $(\mathsf{x}_i, \mathsf{y}_i)$ and the positive X_i-axis is points from $(\mathsf{x}_i, \mathsf{y}_i)$ to $(\mathsf{x}_{i+1}, \mathsf{y}_{i+1})$. The Y_i-axis is perpendicular to the X_i-axis in $(\mathsf{x}_i, \mathsf{y}_i)$ in the usual way. This is illustrated in Figure 7.

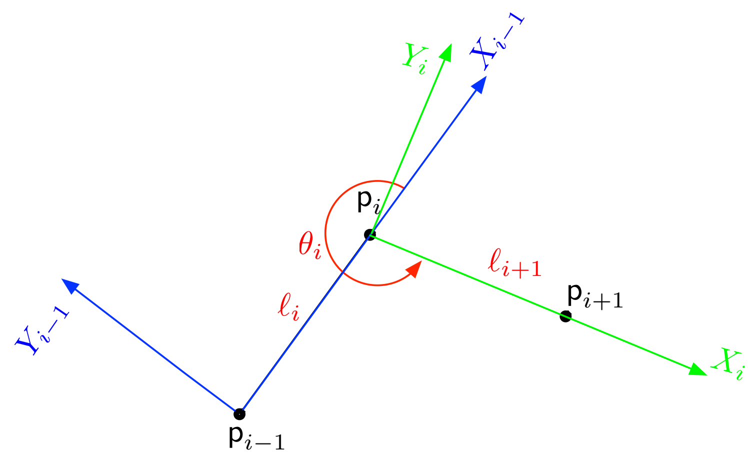

Figure 7: The local coordinate systems (X_{i-1}, Y_{i-1}) (in blue) and (X_i, Y_i) (in green) on the vertices p_{i-1} and p_i of a polyline.

The following property is well known from linear algebra and also from the field of multiple joint robot arms (see, Chapter 6, page 262, in Cox et al. (1997)).

Property 2. *Let p_{i-1}, p_i and p_{i+1} be three consecutive vertices on a polyline $P = \langle(\mathsf{x}_0, \mathsf{y}_0), (\mathsf{x}_1, \mathsf{y}_1), \ldots, (\mathsf{x}_N, \mathsf{y}_N)\rangle$ with $\mathsf{p}_i = (\mathsf{x}_i, \mathsf{y}_i)$, $0 \leq i \leq N$. If a point q in $\mathbf{R}^2$ has coordinates (a_{i-1}, b_{i-1}) and (a_i, b_i), respectively, in the local coordinate systems (X_{i-1}, Y_{i-1}) and (X_i, Y_i), respectively, then*

$$\begin{pmatrix} a_{i-1} \\ b_{i-1} \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\theta_i & -\sin\theta_i & \ell_i \\ \sin\theta_i & \cos\theta_i & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} a_i \\ b_i \\ 1 \end{pmatrix}.$$

$\square$

For a polyline P, given by its polar coordinate representation $\langle \ell_1, \theta_1, \ell_2, \theta_2, ..., \ell_{N-1}, \theta_{N-1}, \ell_N \rangle$, we set

$$P_i = \begin{pmatrix} \cos\theta_i & -\sin\theta_i & \ell_i \\ \sin\theta_i & \cos\theta_i & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

From now on, we only consider polylines with $(\mathsf{x}_0, \mathsf{y}_0) = (0,0)$ and $(\mathsf{x}_1, \mathsf{y}_1) = (1,0)$, such that (X_0, Y_0) is the standard coordinate system.

The following property, based on the previous property, has a straightforward induction proof.

Property 3. *Let* $P = \langle (\mathsf{x}_0, \mathsf{y}_0), (\mathsf{x}_1, \mathsf{y}_1), , \ldots, (\mathsf{x}_N, \mathsf{y}_N) \rangle$ *be a polyline. If a point* q *in* $\mathbf{R}^2$ *has coordinates* (a_i, b_i) *in the local coordinate system* (X_i, Y_i)*, then it has absolute Cartesian coordinates* (a_0, b_0) *in* (X_0, Y_0)*, with*

$$\begin{pmatrix} a_0 \\ b_0 \\ 1 \end{pmatrix} = P_1 \cdot P_2 \cdots P_i \cdot \begin{pmatrix} a_i \\ b_i \\ 1 \end{pmatrix}.$$

$\square$

The following property tells us what the matrix product $P_1 \cdot P_2 \cdots P_i$ looks like.

Property 4. *For* $1 \leq i < N$*, we have*

$$P_1 \cdot P_2 \cdots P_i = \begin{pmatrix} \cos\Theta_1^i & -\sin\Theta_1^i & \sum_{j=1}^{i} \ell_j \cos\Theta_1^{j-1} \\ \sin\Theta_1^i & \cos\Theta_1^i & \sum_{j=1}^{i} \ell_j \sin\Theta_1^{j-1} \\ 0 & 0 & 1 \end{pmatrix},$$

where Θ_i^j *abbreviates* $\theta_i + \theta_{i+1} + \cdots + \theta_j$*, for* $i \leq j$*.*

Proof. We proceed by induction on i. For $i = 1$, we have $\ell_1 \cos 0 = \ell_1$ and $\ell_1 \sin 0 = 0$, which clearly gives P_1.

Now, we proceed from i to $i+1$. By the induction hypothesis, $P_1 \cdot P_2 \cdots P_i \cdot P_{i+1}$ equals

$$\begin{pmatrix} \cos\Theta_1^i & -\sin\Theta_1^i & \sum_{j=1}^{i} \ell_j \cos\Theta_1^{j-1} \\ \sin\Theta_1^i & \cos\Theta_1^i & \sum_{j=1}^{i} \ell_j \sin\Theta_1^{j-1} \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \cos\theta_{i+1} & -\sin\theta_{i+1} & \ell_{i+1} \\ \sin\theta_{i+1} & \cos\theta_{i+1} & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

which is

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

with

- $a_{11} = \cos\Theta_1^i \cdot \cos\theta_{i+1} - \sin\Theta_1^i \cdot \sin\theta_{i+1} = \cos(\Theta_1^i + \theta_{i+1}) = \cos(\Theta_1^{i+1})$;
- $a_{12} = -\cos\Theta_1^i \cdot \sin\theta_{i+1} - \sin\Theta_1^i \cdot \cos\theta_{i+1} = -\sin(\Theta_1^i + \theta_{i+1}) = -\sin(\Theta_1^{i+1})$;
- $a_{13} = \ell_{i+1} \cos\Theta_1^i + \sum_{j=1}^{i} \ell_j \cos\Theta_1^{j-1} = \sum_{j=1}^{i+1} \ell_j \cos\Theta_1^{j-1}$;
- $a_{21} = \sin\Theta_1^i \cdot \cos\theta_{i+1} + \cos\Theta_1^i \cdot \sin\theta_{i+1} = \sin(\Theta_1^i + \theta_{i+1}) = \sin(\Theta_1^{i+1})$;
- $a_{22} = -\sin\Theta_1^i \cdot \sin\theta_{i+1} + \cos\Theta_1^i \cdot \cos\theta_{i+1} = \cos(\Theta_1^i + \theta_{i+1}) = \cos(\Theta_1^{i+1})$;
- $a_{23} = \ell_{i+1} \sin\Theta_1^i + \sum_{j=1}^{i} \ell_j \sin\Theta_1^{j-1} = \sum_{j=1}^{i+1} \ell_j \sin\Theta_1^{j-1}$;
- $a_{31} = 0 + 0 + 0 = 0$;
- $a_{32} = 0 + 0 + 0 = 0$; and
- $a_{33} = 0 + 0 + 1 = 1$;

where we have used the well-known formulas for cosinus and sinus of the sum of angles. This gives the desired matrix and concludes the proof. $\square$

The following theorem tells us how to translate from polar coordinates to Cartesian coordinates.

Theorem 3. *Let* $P = \langle (\mathsf{x}_0, \mathsf{y}_0), (\mathsf{x}_1, \mathsf{y}_1), , \ldots, (\mathsf{x}_N, \mathsf{y}_N) \rangle$ *be a polyline that is given by its polar coordinate representation* $\langle \ell_1, \theta_1, \ell_2, \theta_2, ..., \ell_{N-1}, \theta_{N-1}, \ell_N \rangle$*. If we assume that* $(\mathsf{x}_0, \mathsf{y}_0) = (0, 0)$ *and* $(\mathsf{x}_1, \mathsf{y}_1) = (1, 0)$*, then*

$$\left\{ \begin{array}{lcl} x_i & = & \sum_{j=1}^{i} \ell_j \cos(\theta_1 + \cdots + \theta_{j-1}) \\ y_i & = & \sum_{j=1}^{i} \ell_j \sin(\theta_1 + \cdots + \theta_{j-1}) \end{array} \right.$$

for $2 \leq i \leq N$*.*

We remark that we could also have written

$$\left\{ \begin{array}{l} x_i = 1 + \sum_{j=2}^{i} \ell_j \cos(\theta_1 + \cdots + \theta_{j-1}) \\ y_i = \sum_{j=2}^{i} \ell_j \sin(\theta_1 + \cdots + \theta_{j-1}) \end{array} \right.$$

in the statement of this theorem, since $\ell_1 = 1$, $\cos 0 = 1$ and $\sin 0 = 0$. For esthetic reasons, we will stick to the earlier expressions.

Proof. In the local coordinate system (X_{i-1}, Y_{i-1}), the coordinates op $\mathsf{p}_i = (\mathsf{x}_i, \mathsf{y}_i)$ are $(\ell_i, 0)$. By Property 3, the coordinates of p_i in the standard coordinate system (X_0, Y_0) are given by

$$\begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = P_1 \cdot P_2 \cdots P_{i-1} \cdot \begin{pmatrix} \ell_i \\ 0 \\ 1 \end{pmatrix}.$$

By Property 4, this means

$$\begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\Theta_1^{i-1} & -\sin\Theta_1^{i-1} & \sum_{j=1}^{i-1} \ell_j \cos\Theta_1^{j-1} \\ \sin\Theta_1^{i-1} & \cos\Theta_1^{i-1} & \sum_{j=1}^{i-1} \ell_j \sin\Theta_1^{j-1} \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \ell_i \\ 0 \\ 1 \end{pmatrix}$$

or

$$\left\{ \begin{array}{lcl} x_i & = & \ell_i \cos\Theta_1^{i-1} + \sum_{j=1}^{i-1} \ell_j \cos\Theta_1^{j-1} = \sum_{j=1}^{i} \ell_j \cos\Theta_1^{j-1} \\ y_i & = & \ell_i \sin\Theta_1^{i-1} + \sum_{j=1}^{i-1} \ell_j \sin\Theta_1^{j-1} = \sum_{j=1}^{i} \ell_j \sin\Theta_1^{j-1} \end{array} \right.$$

which concludes the proof. $\square$

5.3. The double-cross conditions for polar coordinates

For a polyline $P = \langle(\mathsf{x}_0, \mathsf{y}_0), (\mathsf{x}_1, \mathsf{y}_1), , \ldots, (\mathsf{x}_N, \mathsf{y}_N)\rangle$ with $\mathsf{p}_i = (\mathsf{x}_i, \mathsf{y}_i)$, $0 \leq i \leq N$, Theorem 1, gives us sign conditions on polynomials for C_1, C_2, C_3 and C_4 in $\mathsf{DC}(\overrightarrow{\mathsf{p}_i\mathsf{p}_{i+1}}, \overrightarrow{\mathsf{p}_j\mathsf{p}_{j+1}}) = (C_1\ C_2\ C_3\ C_4)$. Now, if the polyline $P = \langle(\mathsf{x}_0, \mathsf{y}_0), (\mathsf{x}_1, \mathsf{y}_1), , \ldots, (\mathsf{x}_N, \mathsf{y}_N)\rangle$ is given by its polar coordinate representation $\langle \ell_1, \theta_1, \ell_2, \theta_2, ..., \ell_{N-1}, \theta_{N-1}, \ell_N\rangle$, Theorem 1 allows us to translate these conditions into polar coordinates.

Where needed, we use the abbreviations

$$\begin{cases} \mathsf{c}_i = \cos\Theta_1^i = \cos(\theta_1 + \cdots + \theta_i); \\ \mathsf{s}_i = \sin\Theta_1^i = \sin(\theta_1 + \cdots + \theta_i). \end{cases}$$

to control the length of the expressions.

The following theorem gives the double-cross conditions in polar form.

Theorem 4. *If now the polyline $P = \langle \mathsf{p}_0, \mathsf{p}_1, \ldots, \mathsf{p}_N\rangle$ is given by its polar coordinate representation $\langle \ell_1, \theta_1, \ell_2, \theta_2, ..., \ell_{N-1}, \theta_{N-1}, \ell_N\rangle$, and if we assume that $\mathsf{p}_0 = (0, 0)$ and $\mathsf{p}_1 = (1, 0)$, then $\mathsf{DC}(\overrightarrow{\mathsf{p}_i\mathsf{p}_{i+1}}, \overrightarrow{\mathsf{p}_j\mathsf{p}_{j+1}}) = (C_1\ C_2\ C_3\ C_4)$, for $0 \leq i < j < N$, are expressed in polar coordinates as follows:*

$$\begin{array}{lcll} C_1 & = & - & \mathrm{sign}(\sum_{k=i+1}^{j} \ell_k \cos(\theta_{i+1} + \cdots + \theta_{k-1})); \\ C_2 & = & & \mathrm{sign}(\sum_{k=i+1}^{j} \ell_k \cos(\theta_k + \cdots + \theta_j)); \\ C_3 & = & - & \mathrm{sign}(\sum_{k=i+1}^{j} \ell_k \sin(\theta_{i+1} + \cdots + \theta_{k-1}));\ \textit{and} \\ C_4 & = & & \mathrm{sign}(\sum_{k=i+1}^{j} \ell_k \sin(\theta_k + \cdots + \theta_j)), \end{array}$$

where we agree that the empty sum of angles equals 0.

Proof. Let P be as in the statement of the theorem. From Theorem 3, we get

$$\begin{cases} x_i = \sum_{k=1}^{i} \ell_k \cos(\theta_1 + \cdots + \theta_{k-1}) \\ y_i = \sum_{k=1}^{i} \ell_k \sin(\theta_1 + \cdots + \theta_{k-1}), \end{cases}$$

for $0 \le i \le N$. So, we obtain, for $0 \le i < j < N$,

$$\begin{cases} x_j - x_i & = & \sum_{k=1}^{j} \ell_k \cos(\theta_1 + \cdots + \theta_{k-1}) - \sum_{k=1}^{i} \ell_k \cos(\theta_1 + \cdots + \theta_{k-1}) \\ & = & \sum_{k=i+1}^{j} \ell_k \cos(\theta_1 + \cdots + \theta_{k-1}) \\ y_j - y_i & = & \sum_{k=1}^{j} \ell_k \sin(\theta_1 + \cdots + \theta_{k-1}) - \sum_{k=1}^{i} \ell_k \sin(\theta_1 + \cdots + \theta_{k-1}) \\ & = & \sum_{k=i+1}^{j} \ell_k \sin(\theta_1 + \cdots + \theta_{k-1}) \\ x_{i+1} - x_i & = & \sum_{k=1}^{i+1} \ell_k \cos(\theta_1 + \cdots + \theta_{k-1}) - \sum_{k=1}^{i} \ell_k \cos(\theta_1 + \cdots + \theta_{k-1}) \\ & = & \ell_{i+1} \cos(\theta_1 + \cdots + \theta_i) \\ y_{i+1} - y_i & = & \sum_{k=1}^{i+1} \ell_k \sin(\theta_1 + \cdots + \theta_{k-1}) - \sum_{k=1}^{i} \ell_k \sin(\theta_1 + \cdots + \theta_{k-1}) \\ & = & \ell_{i+1} \sin(\theta_1 + \cdots + \theta_i) \\ x_{j+1} - x_j & = & \sum_{k=1}^{j+1} \ell_k \cos(\theta_1 + \cdots + \theta_{k-1}) - \sum_{k=1}^{j} \ell_k \cos(\theta_1 + \cdots + \theta_{k-1}) \\ & = & \ell_{j+1} \cos(\theta_1 + \cdots + \theta_j) \\ y_{j+1} - y_j & = & \sum_{k=1}^{j+1} \ell_k \sin(\theta_1 + \cdots + \theta_{k-1}) - \sum_{k=1}^{j} \ell_k \sin(\theta_1 + \cdots + \theta_{k-1}) \\ & = & \ell_{j+1} \sin(\theta_1 + \cdots + \theta_j). \end{cases}$$

If we plug these identities in the equations of Theorem 1, we get

$$\begin{array}{rcll} C_1 & = & - & \mathrm{sign}(\sum_{k=i+1}^{j} \ell_k(\mathsf{c}_i\mathsf{c}_{k-1} + \mathsf{s}_i\mathsf{s}_{k-1}) \\ & = & - & \mathrm{sign}(\sum_{k=i+1}^{j} \ell_k \cos(\theta_{i+1} + \cdots + \theta_{k-1})); \\ C_2 & = & & \mathrm{sign}(\sum_{k=i+1}^{j} \ell_k(\mathsf{c}_j\mathsf{c}_{k-1} + \mathsf{s}_j\mathsf{s}_{k-1}) \\ & = & & \mathrm{sign}(\sum_{k=i+1}^{j} \ell_k \cos(\theta_k + \cdots + \theta_j)); \\ C_3 & = & - & \mathrm{sign}(\sum_{k=i+1}^{j} \ell_k(\mathsf{s}_i\mathsf{c}_{k-1} - \mathsf{c}_i\mathsf{s}_{k-1}) \\ & = & - & \mathrm{sign}(\sum_{k=i+1}^{j} \ell_k \sin(\theta_{i+1} + \cdots + \theta_{k-1})); \text{ and} \\ C_4 & = & & \mathrm{sign}(\sum_{k=i+1}^{j} \ell_k(\mathsf{s}_j\mathsf{c}_{k-1} - \mathsf{c}_j\mathsf{s}_{k-1}) \\ & = & & \mathrm{sign}(\sum_{k=i+1}^{j} \ell_k \sin(\theta_k + \cdots + \theta_j)). \end{array}$$

In the last equalities we used the well-known formulas $\sin(\alpha \pm \beta) = \sin\alpha\cos\beta \pm \cos\alpha\sin\beta$ and $\cos(\alpha \pm \beta) = \cos\alpha\cos\beta \mp \sin\alpha\sin\beta$. This concludes the proof. $\square$

We remark that all the double-cross conditions in the above theorem are *linear* expressions in the lengths $\ell_1, ..., \ell_{N-1}$. We also remark that an alternative way to write these conditions is

$$\begin{array}{rcll} C_1 & = & - & \mathrm{sign}(\ell_{i+1} + \sum_{k=i+2}^{j} \ell_k \cos(\theta_{i+1} + \cdots + \theta_{k-1})); \\ C_2 & = & & \mathrm{sign}(\sum_{k=i+1}^{j} \ell_k \cos(\theta_k + \cdots + \theta_j)); \\ C_3 & = & - & \mathrm{sign}(\sum_{k=i+2}^{j} \ell_k \sin(\theta_{i+1} + \cdots + \theta_{k-1})); \text{ and} \\ C_4 & = & & \mathrm{sign}(\sum_{k=i+1}^{j} \ell_k \sin(\theta_k + \cdots + \theta_j)). \end{array}$$

We end this section with a remark about the double cross entries for consecutive line segments.

Because of the special location of $(\mathsf{x}_0, \mathsf{y}_0) = (0, 0)$ and $(\mathsf{x}_1, \mathsf{y}_1) = (1, 0)$, we look at a special case of this theorem, namely $i = 0$ and $j = 1$. Here, we have $\mathsf{DC}(\overrightarrow{\mathsf{p}_0\mathsf{p}_1}, \overrightarrow{\mathsf{p}_1\mathsf{p}_2}) = (C_1\ C_2\ C_3\ C_4)$, with

$$\begin{array}{rcll}
C_1 & = & - & \text{sign}(1) = -; \\
C_2 & = & & \text{sign}(\ell_1 + \ell_2\mathsf{c}_1 - 1) = \text{sign}(\ell_2\mathsf{c}_1); \\
C_3 & = & - & \text{sign}(0) = 0; \quad \text{and} \\
C_4 & = & & \text{sign}(\ell_2\mathsf{s}_1);
\end{array}$$

Because, by the assumption in Definition 1, two consecutive vertices in a polyline are never identical, we have $\ell_2 > 0$, we can simplfy conditions C_2 and C_4 and we get

$$\begin{array}{rcll}
C_1 & = & - & \\
C_2 & = & \text{sign}(\cos\theta_1) & \\
C_3 & = & 0 & \text{and} \\
C_4 & = & \text{sign}(\sin\theta_1). &
\end{array}$$

More generally, we look at the following special case of consecutive line segments of a polyline.

Corollary 1. *If now the polyline $P = \langle \mathsf{p}_0, \mathsf{p}_1, \ldots, \mathsf{p}_N \rangle$ is given by its polar coordinate representation $\langle \ell_1, \theta_1, \ell_2, \theta_2, ..., \ell_{N-1}, \theta_{N-1}, \ell_N \rangle$, and if we assume that $\mathsf{p}_0 = (0, 0)$ and $\mathsf{p}_1 = (1, 0)$, then $\mathsf{DC}(\overrightarrow{\mathsf{p}_i\mathsf{p}_{i+1}}, \overrightarrow{\mathsf{p}_{i+1}\mathsf{p}_{i+2}}) = (C_1\ C_2\ C_3\ C_4)$, for $0 \leq i < N - 1$, are expressed in polar coordinates as follows:*

$$\begin{array}{rcl}
C_1 & = & -; \\
C_2 & = & \text{sign}(\cos\theta_{i+1}); \\
C_3 & = & 0; \\
C_4 & = & \text{sign}(\sin\theta_{i+1}).
\end{array}$$

$\square$

6. A realisability test for 45°-polylines and some remarks on convexity

In this section, we describe how it can be decided whether a given $N \times N$ matrix is realisable in the plane by a 45°-polyline. So, we look at Problem 1 for $\mathcal{P} = \mathcal{P}_{45^\circ}$, the class of 45°-polylines (again with vertices with algebraic coordinates). At the end of this section, we discuss some implications of our result on the convexity of the solution set, determined by a matrix that is realisable in the plane by a 45°-polyline.

6.1. A realisability test for 45°-polylines

For the problem of realisability, here again, we may assume, without loss of any generality, that the polyline that realises a matrix M, if it exists, starts with the unit interval on the x-axis, that is, $\mathsf{p}_0 = (\mathsf{x}_0, \mathsf{y}_0) = (0, 0)$ and $\mathsf{p}_1 = (\mathsf{x}_1, \mathsf{y}_1) = (1, 0)$. This also permits us, to use the results on the polar representation from the previous section.

Theorem 5. *It can be decided in polynomial time whether an $N \times N$ matrix M of 4-tuples $(C_1\ C_2\ C_3\ C_4) \in \{-, 0, +\}^4$ is the double-cross matrix of some 45°-polyline of size N in $\mathbf{R}^2$. If this is the case, also witnesses to this can be produced in polynomial time.*

Proof. We now describe a decision procedure that solves Problem 1 for $\mathcal{P} = \mathcal{P}_{45^\circ}$. Let M be a $N \times N$ input matrix of 4-tuples $(C_1\ C_2\ C_3\ C_4)$ $\in \{-, 0, +\}^4$. In a first step, we determine the polar angles of the polyline, we attempt to construct. In a second step, we see if appropriate lengths of line segments can be found. As an apriori step, we check whether M doesn't have $(0\ 0\ 0\ 0)$ entries on its diagonal or doesn't have the "symmetry" properties, discussed in Section 2.2. If M fails this symmetry-test, we can already answer *no*, else we proceed.

Step 1 (Determining the angles $\theta_1, \theta_2, ..., \theta_{N-1}$). First, we inspect the entries $M[i, i+1]$, $0 \leq i < N$ of M. Hereto, we use Corollary 1. So, C_1 should be $-$ and C_3 should be 0. If this is not the case, we can already answer *no.* From C_2 and C_4 in all entries $M[i, i+1]$, we can determine the angles θ_i as is shown in the following table.

C_2	C_4	θ_i
0	0	answer *no*
0	+	270°
0	−	90°
+	0	180°
+	+	225°
+	−	135°
−	0	0°
−	+	315°
−	−	45°

Obviously, if both $C_2 = \text{sign}(\cos\theta_i)$ and $C_4 = \text{sign}(\sin\theta_i)$ (see Corollary 1) are 0, we have an impossible situation. So, at this point, or we have answered *no*, or we know all the angles $\theta_1, \theta_2, ..., \theta_{N-1}$ of a possible realisation of M. In the latter case, we proceed to Step 2.

Step 2 (Determining $\ell_1, \ell_2, ..., \ell_N$). Once, we have determined the angles $\theta_1, \theta_2, ..., \theta_{N-1}$, we can compute all the values $\cos(\theta_{i+1} + \cdots + \theta_{k-1})$, $\cos(\theta_k + \cdots + \theta_j)$, $\sin(\theta_{i+1} + \cdots + \theta_{k-1})$ and $\sin(\theta_k + \cdots + \theta_j)$ that appear in the expressions given in Theorem 4. Since all these sums of angles are multiples of 45°, these cosines and sines will take values as shown in the following table.

α	$\cos\alpha$	$\sin\alpha$
0°	1	0
45°	$\frac{\sqrt{2}}{2}$	$\frac{\sqrt{2}}{2}$
90°	0	1
135°	$-\frac{\sqrt{2}}{2}$	$\frac{\sqrt{2}}{2}$
180°	-1	0
225°	$-\frac{\sqrt{2}}{2}$	$-\frac{\sqrt{2}}{2}$
270°	0	-1
315°	$\frac{\sqrt{2}}{2}$	$-\frac{\sqrt{2}}{2}$

This means that the double-cross conditions given by Theorem 4, together with the constraints that the ℓ_i are strictly positive lengths, can be seen as linear constraint conditions in $\ell_1, \ell_2, ..., \ell_N$ of the form

$$\left\{\begin{array}{llll} - & \sum_{k=i+1}^{j} \ell_k \cos(\theta_{i+1} + \cdots + \theta_{k-1}) & \alpha_{ij} & 0 \quad (0 \leq i < j < N) \\ & \sum_{k=i+1}^{j} \ell_k \cos(\theta_k + \cdots + \theta_j) & \beta_{ij} & 0 \quad (0 \leq i < j < N) \\ - & \sum_{k=i+1}^{j} \ell_k \sin(\theta_{i+1} + \cdots + \theta_{k-1}) & \gamma_{ij} & 0 \quad (0 \leq i < j < N) \\ & \sum_{k=i+1}^{j} \ell_k \sin(\theta_k + \cdots + \theta_j) & \delta_{ij} & 0 \quad (0 \leq i < j < N) \\ & \ell_i & > & 0 \quad (0 < i \leq N) \end{array}\right. \qquad (*)$$

with $\alpha_{ij}, \beta_{ij}, \gamma_{ij}, \delta_{ij} \in \{=, <, >\}$ determined by the entries of the matrix M. Since all cosines and sines take values in the set $\{0, 1, -1, \frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}\}$, all these conditions are linear in $\ell_1, \ell_2, ..., \ell_N$. Therefore, $(*)$ can be seen as a *linear programming problem*, or at least almost. Normally in a linear programming problem, linear polynomial conditions of the form

$$a_1\ell_1 + a_2\ell_2 + \cdots + a_N\ell_N \geq 0,$$

with the coefficients a_i rational numbers are expected to appear together with the additional conditions

$$\ell_i \geq 0 \quad (0 \leq i \leq N).$$

So, we are left with three problems:

(1) we have $\ell_i > 0$ for $0 < i \leq N$ and not the traditional $\ell_i \geq 0$;

(2) we have $\alpha_{ij}, \beta_{ij}, \gamma_{ij}, \delta_{ij} \in \{=, <, >\}$ and not the traditional $\geq$; and

(3) we have irrational coefficients $\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}$.

The linear polynomial condition

$$a_1\ell_1 + a_2\ell_2 + \cdots + a_N\ell_N = 0$$

is obviously equivalent to

$$a_1\ell_1 + a_2\ell_2 + \cdots + a_N\ell_N \geq 0 \text{ and } a_1\ell_1 + a_2\ell_2 + \cdots + a_N\ell_N \leq 0.$$

This solves the case of equality. Obviously,

$$a_1\ell_1 + a_2\ell_2 + \cdots + a_N\ell_N < 0$$

is equivalent to

$$-a_1\ell_1 - a_2\ell_2 - \cdots - a_N\ell_N > 0.$$

So, we are left with $a_1\ell_1 + a_2\ell_2 + \cdots + a_N\ell_N > 0$. To solve the problem of the strict inequalities in (1) and (2), there is a known trick from the linear programming literature that we can use (see page 22 of Matousek (2007)). We introduce a new variable δ, which stands for the "gap" between the left and the right side of each inequality and we try to make this gap as large as possible. Then $a_1\ell_1 + a_2\ell_2 + \cdots + a_N\ell_N > 0$ is equivalent to

$$\begin{array}{rl} \text{maximize} & \delta \\ \text{subject to} & a_1\ell_1 + a_2\ell_2 + \cdots + a_N\ell_N - \delta \geq 0 \\ \text{and} & \delta \geq 0. \end{array}$$

And this single δ can be used to deal with several strict inequalities all at once. Indeed, the linear program has now an extra variable δ and the optimal δ is strictly positive exactly when the original system with strict inequalities has a solution.

Let us write the first $2N(N-1)$ linear polynomials appearing in $(*)$ as

$$P_{ij}^{\sigma_{ij}}(\ell_1, \ell_2, ..., \ell_N)\ \sigma_{ij}\ 0,$$

with $0 \leq i < j < N$ and $\sigma_{ij} \in \{\alpha_{ij}, \beta_{ij}, \gamma_{ij}, \delta_{ij}\}$.

We define the sets

$$\begin{array}{rcl} S_= & := & \{(i, j, \sigma_{ij} \mid 0 \leq i < j < N, \sigma_{ij} \in \{\alpha_{ij}, \beta_{ij}, \gamma_{ij}, \delta_{ij}\} \text{ and } \sigma_{ij} ==\}; \\ S_> & := & \{(i, j, \sigma_{ij} \mid 0 \leq i < j < N, \sigma_{ij} \in \{\alpha_{ij}, \beta_{ij}, \gamma_{ij}, \delta_{ij}\} \text{ and } \sigma_{ij} =>\}; \text{ and} \\ S_< & := & \{(i, j, \sigma_{ij} \mid 0 \leq i < j < N, \sigma_{ij} \in \{\alpha_{ij}, \beta_{ij}, \gamma_{ij}, \delta_{ij}\} \text{ and } \sigma_{ij} =<\}. \end{array}$$

Now, we can see that $(*)$ can be converted to the following linear programming problem:

$$\begin{array}{lrl} \text{maximize } \delta & & \\ \text{subject to} & P_{ij}^{\sigma_{ij}}(\ell_1, \ell_2, ..., \ell_N) \geq 0 & \text{for } (i, j, \sigma_{ij}) \in S_= \\ & -P_{ij}^{\sigma_{ij}}(\ell_1, \ell_2, ..., \ell_N) \geq 0 & \text{for } (i, j, \sigma_{ij}) \in S_= \\ & P_{ij}^{\sigma_{ij}}(\ell_1, \ell_2, ..., \ell_N) - \delta \geq 0 & \text{for } (i, j, \sigma_{ij}) \in S_> \\ & -P_{ij}^{\sigma_{ij}}(\ell_1, \ell_2, ..., \ell_N) - \delta \geq 0 & \text{for } (i, j, \sigma_{ij}) \in S_< \\ & \ell_i - \delta \geq 0 & \text{for } 0 < i \leq N \\ \text{and} & \delta \geq 0. & \end{array}$$

What remains is Problem (3), namely that we may have the irrational coefficients $\frac{\sqrt{2}}{2}$ and $-\frac{\sqrt{2}}{2}$ in our linear programming problem. However, a result by Adler and Beling (Adler and Beling (1994)) shows that linear programming with algebraic coefficients also has a time complexity that is a polynomial of

(i) the "rank" of the linear system of inequalities, which applied to our example is $O(N^2)$;

(ii) the degree of the extension of the rationals in which we work, which is in our case 2, since $(\mathbf{Q}(\sqrt{2}) : \mathbf{Q}) = 2$; and

(iii) a quantity related to the conjugate norm of the linear system, which in our case is $O(N^2 \log N)$.

So, we can conclude that our linear programming problem can be solved in polynomial time in N. This completes the proof. □

6.2. Convexity properties of 45°-polylines

From Step 1 of the proof of Theorem 5, it follows that a matrix M that is realisable by a 45°-polyline determines the angles θ_i uniquely for $0 < i < N$. This proves the following corollary.

Corollary 2. *If an $N \times N$ matrix M of 4-tuples $(C_1\ C_2\ C_3\ C_4) \in \{-, 0, +\}^4$ is realisable by two 45°-polylines P_1 and P_2, that start with the line segment connecting $(0, 0)$ and $(1, 0)$ and have polar-coordinate representations $\langle \ell_1, \theta_1, \ell_2, \theta_2, ..., \ell_{N-1}, \theta_{N-1}, \ell_N \rangle$ and $\langle \ell'_1, \theta'_1, \ell'_2, \theta'_2, ..., \ell'_{N-1}, \theta'_{N-1}, \ell'_N \rangle$, respectively, then $\theta_i = \theta'_i$ for $0 < i < N$.* □

Also from the proof of Theorem 5, the following property follows.

Corollary 3. *If an $N \times N$ matrix M of 4-tuples $(C_1\ C_2\ C_3\ C_4) \in \{-, 0, +\}^4$ is realisable by two 45°-polylines P_1 and P_2, that start with the line segment connecting $(0, 0)$ and $(1, 0)$ and have polar-coordinate representations $\langle \ell_1, \theta_1, \ell_2, \theta_2, ..., \ell_{N-1}, \theta_{N-1}, \ell_N \rangle$ and $\langle \ell'_1, \theta'_1, \ell'_2, \theta'_2, ..., \ell'_{N-1}, \theta'_{N-1}, \ell'_N \rangle$, respectively, then for any real numbers $\alpha_1, \alpha_2 > 0$, the 45°-polyline given by the polar coordinate representation $\langle \alpha_1 \cdot \ell_1 + \alpha_2 \cdot \ell'_1, \theta_1, \alpha_1 \cdot \ell_2 + \alpha_2 \cdot \ell'_2, \theta_2, ..., \alpha_1 \cdot \ell_{N-1} + \alpha_2 \cdot \ell'_{N-1}, \theta_{N-1}, \alpha_1 \cdot \ell_N + \alpha_2 \cdot \ell'_N \rangle$ also realises M.*

Proof. Corollary 2 takes care of the angles. From Step 2 of the proof of the previous theorem it follows that if P_1 and P_2 are realisations of a matrix M their lengths satisfy the same set of linear conditions of the form $a_1\ell_1+a_2\ell_2+\cdots+a_N\ell_N \;\alpha\; 0$, with $\alpha \in \{<,=,>\}$. Suppose that we have

$$\begin{cases} a_1\ell_1 + a_2\ell_2 + \cdots + a_N\ell_N & > & 0 \text{ and} \\ a_1\ell'_1 + a_2\ell'_2 + \cdots + a_N\ell'_N & > & 0 \end{cases}$$

for P_1 and P_2, for any of these linear conditions. Since both $\alpha_1 > 0$ and $\alpha_2 > 0$, we also have

$$\begin{cases} \alpha_1 \cdot (a_1\ell_1 + a_2\ell_2 + \cdots + a_N\ell_N) & > & 0 \text{ and} \\ \alpha_2 \cdot (a_1\ell'_1 + a_2\ell'_2 + \cdots + a_N\ell'_N) & > & 0. \end{cases}$$

So, also the sum of the two left hand sides,

$$\alpha_1 \cdot (a_1\ell_1 + a_2\ell_2 + \cdots + a_N\ell_N) + \alpha_2 \cdot (a_1\ell'_1 + a_2\ell'_2 + \cdots + a_N\ell'_N)$$

will be strictly larger than 0. The same argument hold when α is $=$ or $<$. This completes the proof. $\square$

We end this chapter with the following *convexity property* for 45°-polylines.

Corollary 4. *If an $N \times N$ matrix M of 4-tuples $(C_1\ C_2\ C_3\ C_4) \in \{-,0,+\}^4$ is realisable by two 45°-polylines P_1 and P_2, that start with the line segment connecting $(0,0)$ and $(1,0)$ and have polar-coordinate representations $\langle \ell_1, \theta_1, \ell_2, \theta_2, ..., \ell_{N-1}, \theta_{N-1}, \ell_N\rangle$ and $\langle \ell'_1, \theta'_1, \ell'_2, \theta'_2, ..., \ell'_{N-1}, \theta'_{N-1}, \ell'_N\rangle$, respectively, then for any λ, with $0 \le \lambda \le 1$, the 45°-polyline given by the polar coordinate representation*

$$\begin{aligned} &\langle \lambda \cdot \ell_1 + (1-\lambda)\cdot \ell'_1, \lambda\cdot\theta_1 + (1-\lambda)\cdot\theta'_1, \\ &\quad \lambda\cdot\ell_2 + (1-\lambda)\cdot\ell'_2, \lambda\cdot\theta_2 + (1-\lambda)\cdot\theta'_2, ..., \lambda\cdot\ell_{N-1} + (1-\lambda)\cdot\ell'_{N-1}, \\ &\qquad \lambda\cdot\theta_{N-1} + (1-\lambda)\cdot\theta'_{N-1}, \lambda\cdot\ell_N + (1-\lambda)\cdot\ell'_N\rangle \end{aligned}$$

also realises M.

Proof. From Corollary 2, it is clear that $\lambda \cdot \theta_i + (1 - \lambda) \cdot \theta'_i = \theta_i = \theta'_i$ for $0 < i < N$.

For λ with $0 \leq \lambda \leq 1$, we observe that if we take $\lambda = 0$, we get P_2 and if we take $\lambda = 1$, we get P_1. This leaves us with the case $0 < \lambda < 1$. But here, both λ and $1 - \lambda$ are strictly larger than 0 and Corollary 3 applies with $\alpha_1 = \lambda$ and $\alpha_2 = 1 - \lambda$. This completes the proof. $\square$

7. Conclusion and discussion

We have studied the decision problem that asks whether a $N \times N$ matrix of 4-tuples from $\{-, 0, +\}$ is the double-cross matrix of a polyline with N line segments. This problem is, in general, NP-hard. In this paper, we have given a conceptually easy $O(N^2)$-time algorithms for the case where the attention is restricted to polylines in which consecutive line segments make angles that are multiples of 90°. Next, we have given a more complicated algorithm that solves the question for 45°-polylines. For this more complicated case of 45°-polylines, we have introduce the polar-coordinate representation of double-cross matrices.

It is not easy to see how the techniques, developed for the 45°-case can be generalized to, for instance the reliability question for of 30°-polylines. In Step 1 of the proof of Theorem 5, we would have to consider two possibilities per quadrant of the double cross (essentially corresponding to 30° and 60°). It is not clear how an exponential blow-up in N can be avoided.

It was not the intention of this paper to implement the proposed methods for 90°-polylines and 45°-polylines and experiment with them. But, we have some experiments in a related setting. The question of a $N \times N$ matrix of 4-tuples from $\{-, 0, +\}$ being realisable in restricted classes of polylines also has other variants. Instead of taking the angles from a fixed set of angles, we can also take the lengths of the line segments in the polyline to come from a set of fixed lengths. One particular case of this is where we ask if a matrix is realisable by polylines in which all line segments have a fixed length $L > 0$. It turns out that in this setting the reconstruction task is also considerably simplified as is demonstrated by the following experiment. For a given polyline P of size N, we compute the double-cross matrix M of size N by N; we set the desired lengths of the line segments $L > 0$; and the granularity (the number of candidates to be generated) s. The reconstruction algorithm works as follows. In the first step, the algorithm creates a line of length L with as start vertex $(0,0)$ and end vertex $(L,0)$. In the second step, the algorithm creates a set of candidate second line segments. By looking at $M[1,2]$, we know what the minimum and maximum angle is between the first and the second line segment. There are two possibilities: the minimum angle is equal to maximum angle; or it is not. In the first case, there is a unique solution to position the second line segment. In the other case, we create s candidate solutions, equally distributed between the minimum and maximum angle.

In the i-th step of the algorithm ($i \leq N$), we take one by one the polylines created in the $(i-1)$-th step, and create new polylines by adding s possible last line setments (as we did in the second step). The only difference now is that when we have a candidate polyline P_{new}, we first check whether $\mathsf{DCM}(P_{new})$ corresponds to the given matrix M. If this is not the case, this candidate is pruned. Only when this is the case, P_{new} is further considered as possible realisation of M.

Given the double-cross matrix in Table 2, which is the double-cross matrix of the polyline in Figure 8, the output of the algorithm with $L = 3$ and $s = 10$, in the second step looks like the polyline in Figure 9.

Table 2: The double-cross matrix (only the part strictly above the diagonal is shown) of the polyline in Figure 8.

$-+0+$	$-+++$	$-++-$	$--++$	$--++$	$--+$-	$--++$	$++++$
	$-+0+$	$-++-$	$--++$	$--++$	$--++$	$-+++$	$+++-$
		$--0-$	$---+$	$-+-+$	$--++$	$-+++$	$-++-$
			$--0+$	$--++$	$+-++$	$++--$	$++--$
				$-+0+$	$--++$	$-++-$	$-++-$
					$--0+$	$-++-$	$-++-$
						$--0-$	$----$
							$-+0-$

Given the double-cross matrix in Table 2, which is the double-cross matrix of the polyline in Figure 8, the output of the algorithm with $L = 3$ and $s = 10$, in the second step are the polylines given in Figure 9.

Figure 10 gives all output polylines that have the double-cross matrix of Table 2 (and thus are double-cross similar to the polyline of Figure 8). In this experiment, the output consists of 632 polylines, satisfying the double-cross matrix of Table 2. They were created in approximately 3 seconds on a Apple

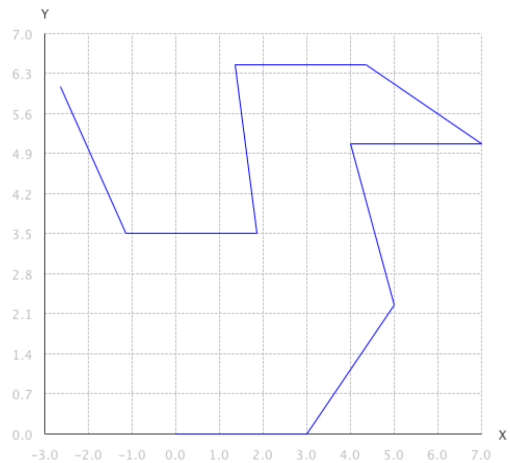

Figure 8: The original polyline

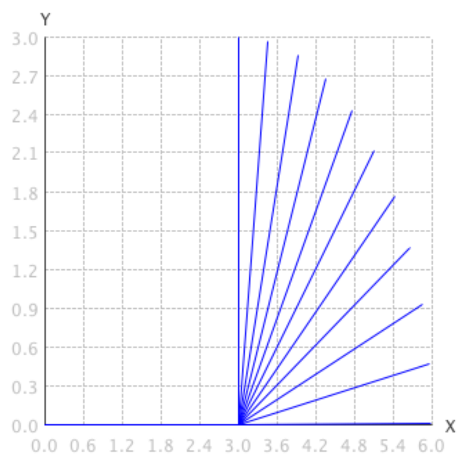

Figure 9: The output after the second step of the algorithm using the double-cross matrix in Table 2.

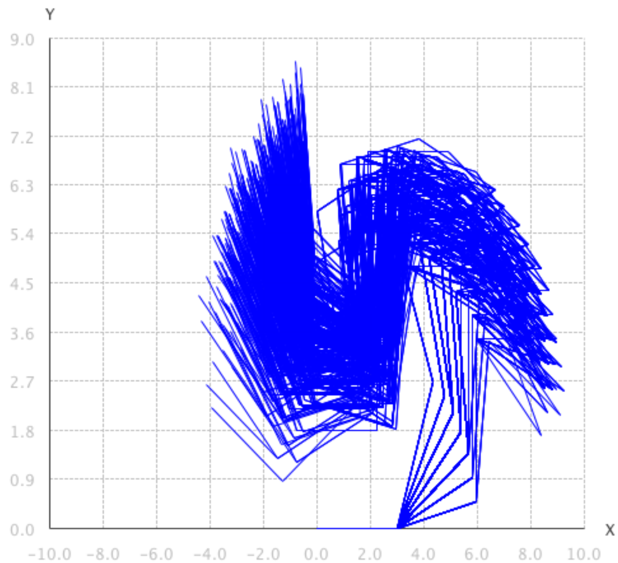

Figure 10: A complete set of output polylines.

Macbook with 2.16 GHz Intel Core 2 Duo processor and 1 GB RAM.

It should be clear that the described algorithm is not guaranteed to find a sample polyline, even when one exists. By creating s candidate solutions, equally distributed between the minimum and maximum angle, we might find no solution, but miss an existing solution.

Adler, I., Beling, P. A., 1994. Polynomial algorithms for linear programming over the algebraic numbers. Algorithmic 12 (6), 436–457.

Basu, S., Pollack, R., Roy, M.-F., 2006. Algorithms in Real Algebraic Geometry (Algorithms and Computation in Mathematics). Springer-Verlag New York, Inc.

Bochnak, J., Coste, M., Roy, M.-F., 1998. Real Algebraic Geometry. Vol. 36 of Ergebenisse der Mathematik und ihrer Grenzgebiete. Folge 3. Springer-Verlag.

Bookstein, F. L., 1986. Size and shape spaces for landmark data in two dimensions. Statistical Science 1, 181–242.

Collins, G., 1975. Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In: Automata Theory and Formal Languages. Vol. 33 of Lecture Notes in Computer Science.

Cox, D. A., Little, J., O'Shea, D., 1997. Ideals, varieties, and algorithms - an introduction to computational algebraic geometry and commutative algebra (2. ed.). Undergraduate texts in mathematics. Springer.

Dolzmann, A., Sturm, T., 1997. Redlog: Computer algebra meets computer logic. ACM SIGSAM Bulletin 32 (2), 2–9, (see also redlog.dolzmann.de/).

Dryden, I., Mardia, K. V., 1998. Statistical Shape Analysis. Wiley.

Forbus, K. D., 1990. Qualitative physics: past present and future. In: Readings in qualitative reasoning about physical systems. Morgan Kaufmann, pp. 11–39.

Freksa, C., 1992. Using orientation information for qualitative spatial reasoning. In: Spatio-Temporal Reasoning (GIS'92). pp. 162–178.

Gero, J. S., 1999. Representation and reasoning about shapes: cognitive and computational studies in visual reasoning in design. In: et al., K. (Ed.), Proceedings of the International Conference on Spatial Information Theory (COSIT'99). Vol. Lecture Notes in Computer Science, 2825. Springer, pp. 315–330.

Giannotti, F., Pedreschi, D., 2008. Mobility, Data Mining and Privacy. Springer.

Giusti, M., Heintz, J., 2001. Kronecker's smart, little black boxes. In: DeVore, R., A., I., Suli, E. (Eds.), Foundations of Computational Mathematics. Cambridge University Press, Cambridge, pp. 69–104.

Güting, R. H., Schneider, M., 2005. Moving Objects Databases. Morgan Kaufmann.

Heintz, J., Kuijpers, B., Paredes, A. R., 2013. Software engineering and complexity in effective algebraic geometry. Journal of Complexity 29 (1), 92–138.

Hong, H., 2000. QEPCAD. `www.usna.edu/CS/qepcadweb/B/QEPCAD.html`.

Kent, J. T., Mardia, K. V., 1986. Shape, procrustes tangent projections and bilateral symmetry. Biometrika 88, 469–485.

Khovanskii, A., 1991. Fewnomials. Translations of mathematical monographs. American Mathematical Society.

Kuijpers, B., Moelans, B., Van de Weghe, N., 2006. Qualitative polyline similarity testing with applications to query-by-sketch, indexing and classification. In: Proceedings of the 14th annual ACM International Symposium on Advances in Geographic Information Systems. ACM, pp. 11–18.

Matousek, J., G. B., 2007. Understanding and Using Linear Programming. No. v. 36 in Universitext. Springer.

Mokhtarian, F., Mackworth, A. K., 1992. A theory of multiscale, curvature-based shape representation for planar curves. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) 14, 789–805.

Renz, J., Nebel, B., 2007. Qualitative spatial reasoning using constraint calculi. In: Aiello, M., Pratt-Hartmann, I., van Benthem, J. (Eds.), Handbook of Spatial Logics. Springer.

Scivos, A., Nebel, B., 2001. Double-crossing: Decidability and computational complexity of a qualitative calculus for navigation. In: Spatial Information Theory: Foundations of Geographic Information Science, International Conference, COSIT 2001. pp. 431–446.

Tarski, A., 1951. A Decision Method for Elementary Algebra and Geometry. University of California Press.

Wolfram Research, 2015. Mathematica 9. `www.wolfram.com`.

Zimmermann, K., Freksa, C., 1996. Qualitative spatial reasoning using orientation, distance, and path knowledge. Applied Intelligence 6 (1), 49–58.