VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV MATEMATIKY

Ing. Pavel Loučka

ALGORITHMS FOR CONICS IN GEOMETRIC ALGEBRAS
ALGORITMY PRO KUŽELOSEČKY V GEOMETRICKÝCH ALGEBRÁCH

Zkrácená verze disertační práce

OBOR: Aplikovaná matematika

ŠKOLITEL: doc. Mgr. Petr Vašík, Ph.D., ÚM FSI VUT v Brně
OPONENTI:

DATUM OBHAJOBY:

KEYWORDS
conic section, conic fitting, algorithm, geometry, geometric algebra, Clifford algebra, projectivisation, axial alignment, centre, waypoint, wedge, proper point, improper point, ideal point, point at infinity, degenerate conic, line-pair, parabola

KLÍČOVÁ SLOVA
kuželosečka, fitování kuželoseček, algoritmus, geometrie, geometrická algebra, Cliffordova algebra, projektivizace, osové zarovnání, střed, průchozí bod, wedge, vlastní bod, nevlastní bod, bod v nekonečnu, singulární kuželosečka, dvojice přímek, parabola

MÍSTO ULOŽENÍ DISERTAČNÍ PRÁCE:
Ústav matematiky
Fakulta strojního inženýrství
Vysoké učení technické v Brně
Technická 2896/2
616 69 Brno

ISBN
ISSN 1213-4198

Contents

Introduction

Having originated in the ancient times, the conic sections have found many applications since then, for example, in areas such as architecture, mathematics, physics, engineering and many more, [9, 13]. The intense development of computer science in the 20$^{\text{th}}$ and the 21$^{\text{st}}$ centuries has also encouraged the use of conic sections in computer graphics and computer vision, [1].

In the thesis we present several algorithms concerning different kinds of geometric problems related to conic sections, using so called Geometric Algebra for Conics (GAC) which is a geometric algebra originally proposed by Perwass in [28], Hrdina et al. then described the full concept including transformations and its structural position among other geometric algebras in [17]. GAC has already proved a great framework for working with conics in various fields, such as object detection, [2, 6], dynamical systems, [7], electrical power quality analysis, [35], and geometry, [18, 3, 5]. In addition to GAC, there are also references to 2D Conformal Geometric Algebra (2D CGA), [16], and classical projective geometry of conics throughout the work.

In Chapter 1, we offer a brief introduction to GAC and also make a distinction between *proper* points, i.e. points of the real plane $\mathbb{R}^2$, and *improper* points, that is, *points at infinity*, also called *ideal points*, which constitute the infinitely distant points of the real projective plane $\mathbb{RP}^2$. We also present an extended embedding that maps both proper and improper points into GAC in a geometrically reasonable way. Such an extension of the domain of GAC has already been presented by us in [21] and [22]. The notion of proper and improper points is then further used across the whole work, since the use of the elements at infinity is inherent to some types of conics.

Chapter 2 discusses *conic fitting*, i.e. fitting a conic section among the set of discrete points as tightly as possible, which is used in several computer vision tasks. Typical applications of conic fitting include object detection, [19, 4, 29], camera calibration, [25, 39], and miscellaneous problems in fields like reverse engineering, [30], or computer assisted surgery, [8]. Nowadays, even though conic fitting is considered a classical problem of mathematical optimisation, and many conic fitting algorithms have been invented over time, novel algorithms have been being created in the recent years, too, [37, 38, 18]. For a list of some of the standard ones, see for example [10]. Some of the fitting algorithms are also type-specific, i.e. the type of the fitted conic is set in advance, for instance, ellipse-fitting algorithms by Fitzgibbon, [11], and Rosin, [32], or a circle-fitting algorithm by Gander et al., [12]. These can be convenient in practice, since the objects-to-be-detected in computer vision are of circular or elliptical shape very often.

Consequently, in the chapter we discuss and present various conic fitting algorithms based on GAC, namely, they are created by modifying the original GAC-based conic fitting algorithm by Hrdina et al., [18]. Besides possible imposing of the type-specificity on a conic fitting algorithm, there are also other additional geometric constraints which could be a priori prescribed to the fitted conic, such as the tilt of the principal axes, the centre position, or the lengths of the conic's semi-axes. GAC-based algorithms fitting a conic with some of these additional geometric conditions have been derived and presented by us in [23, 24] and are described in the chapter, as well. Additionally, we describe two iterative algorithms that were invented in order to overcome non-invariance w.r.t. translation of the fitted data points, out of which one has appeared in [6]. Besides the geometric constraints mentioned above, it is also shown in the work that constructing a conic through a prescribed point or a group of points called *waypoints*—while fitting a conic among the other data points—is also possible, [21, 22]. Moreover, thanks to the inclusion of improper points in GAC, they can be used as waypoints as well. Let us note that the only other conic-fitting algorithm using some kind of additional geometric condition known to us is [36], where an ellipse is fitted among the data set in such a way that its centre approximately lies on a prescribed line.

Chapter 2 is also equipped with a number of examples using the presented conic fitting algorithms on sample datasets and the influence of noise on functioning of the algorithms is discussed, too. Furthermore, MATLAB implementation of the algorithms and their use on the presented examples are electronically attached to the work.

Chapter 3 discusses the construction of conics using GAC outer product, called *wedge product*, or simply, *wedge*. In compliance with the fact that a conic is generally determined by five points, [27, 31], the beginning of the chapter deals with a construction of a conic passing through a group of five points. The classification of such conics is accompanied by the corresponding examples, including the conics passing through improper points. Next, attention is drawn upon the intersection of two conics, which is represented in GAC as a so called *four-point*. It is shown in the chapter that a conic can be constructed by applying wedge product to the four-point and another point not lying on the four-point. Also, a great portion of the chapter focuses on the concept of a *pencil of conics*, [33, 31, 13], that is a set of all conics passing through the intersection of two generating conics. Consequently, GAC wedge is used to construct two special subsets of conics that are found in a pencil of conics, namely, the *line-pairs* and the *generalised parabolas*. In particular, the alternative ways of constructing such conics using a four-point and another point is described and demonstrated in examples. Furthermore, classification of a number of generalised parabolas to be found in a particular pencil was described w.r.t. the relations between the generating conics of the pencil.

The examples in Chapter 3 were implemented in Maple software and they are available in the electronic appendix of the work.

All the examples, theorems, associated proofs and other material not present in this short version of the thesis can be found in its full version.

1. GAC

1.1. Basics

Geometric Algebra for Conics (GAC), proposed by Perwass in [28], and further elaborated by Hrdina, Návrat and Vašík in [17], generalises the concept of 2-dimensional conformal geometric algebra $\mathbb{G}_{3,1}$ (2D CGA) also known as Compass Ruler Algebra (CRA), [15, 16], to represent conic sections. In particular, GAC constitutes a Clifford algebra $\mathbb{G}_{5,3}$ with an embedding $C : \mathbb{R}^2 \to \mathbb{R}^{5,3}$ of a proper point $p = xe_1 + ye_2$ from the plane $\mathbb{R}^2$ to a 6-dimensional subspace of one-vectors in GAC, in the form

$$C(x, y) = \bar{n}_+ + xe_1 + ye_2 + \frac{1}{2}(x^2 + y^2)n_+ + \frac{1}{2}(x^2 - y^2)n_- + xyn_\times, \qquad (1.1)$$

where

$$\{\bar{n}_\times, \bar{n}_-, \bar{n}_+, e_1, e_2, n_+, n_-, n_\times\} \qquad (1.2)$$

is the 8-dimensional vector basis of $\mathbb{G}_{5,3}$, [18], together with an associated bilinear form of the inner product of vectors in GAC given by the matrix[1]

$$B = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

The meaning of the basis vectors is following: $\bar{n}$'s denote three orthogonal 'origins', e_1 and e_2 stay for the basis vectors of the Euclidean plane and n's represent three orthogonal 'infinities', respectively. Let us note that a pair of an origin and the corresponding infinity is called a Witt pair and that both origins and infinities are null vectors. Let us also recall the types of representations given in GAC:

Definition 1.1. An element $A_I \in \mathbb{G}_{5,3}$ is the *inner product representation* (or *inner product null space representation*, abbr. *IPNS representation*) of a geometric entity $A \subset \mathbb{R}^2$ if and only if

$$A = \left\{p \in \mathbb{R}^2 : C(p) \cdot A_I = 0\right\},$$

where "$\cdot$" denotes the inner product between vectors in GAC, [17].

Consequently, the inner product null space (IPNS) representation of a general conic section Q in GAC is given by 1-vector

$$Q_I = \bar{v}^{\times}\bar{n}_{\times} + \bar{v}^{-}\bar{n}_{-} + \bar{v}^{+}\bar{n}_{+} + v^1 e_1 + v^2 e_2 + v^{+} n_{+}. \tag{1.3}$$

In a similar way, IPNS representation of a proper point $p = xe_1 + ye_2$, $p \in \mathbb{R}^2$, is given by embedding (1.1), therefore

$$P_I = \bar{n}_{+} + xe_1 + ye_2 + \frac{1}{2}(x^2 + y^2)n_{+} + \frac{1}{2}(x^2 - y^2)n_{-} + xyn_{\times}. \tag{1.4}$$

Thus, both a point and a conic are embedded into six-dimensional subspaces of $\mathbb{R}^{5,3}$. Let us also state that, w.r.t. basis (1.2), an IPNS conic section (1.3) and an IPNS point (1.4) can be represented, respectively, by usual vectors

$$Q_I = \begin{pmatrix} \bar{v}^{\times} & \bar{v}^{-} & \bar{v}^{+} & v^1 & v^2 & v^{+} & 0 & 0 \end{pmatrix}^T, \tag{1.5}$$

$$P_I = \begin{pmatrix} 0 & 0 & 1 & x & y & \frac{1}{2}(x^2 + y^2) & \frac{1}{2}(x^2 - y^2) & xy \end{pmatrix}^T. \tag{1.6}$$

Next, using the outer (wedge) product, we can represent objects in GAC as well:

[1]The order of the basis vectors and the elements of matrix B in the text follow notation of [18] and not the one given in [17], where basis vectors $\bar{n}_{\times}$ and $\bar{n}_{+}$ are swapped and the corresponding coefficients in matrix B are interchanged accordingly.

Definition 1.2. An element $A_O \in \mathbb{G}_{5,3}$ is the *outer product representation* (or *outer product null space representation*, abbr. *OPNS representation*) of a geometric entity $A \subset \mathbb{R}^2$ if and only if

$$A = \left\{p \in \mathbb{R}^2 : C(p) \wedge A_O \wedge \bar{n}_- \wedge \bar{n}_\times = 0\right\}.$$

Pseudoscalar is in GAC given by

$$I = \bar{n}_\times \bar{n}_- \bar{n}_+ e_1 e_2 n_+ n_- n_\times$$

and duality between IPNS and OPNS representation then reads

$$\begin{aligned} A_O &= (A_I \wedge n_- \wedge n_\times)^*, \\ A_I &= (A_O \wedge \bar{n}_- \wedge \bar{n}_\times)^*. \end{aligned}$$

Furthermore, the conversion between OPNS and IPNS can be carried out easily using two "pseudoscalars"

$$\begin{aligned} I_{OI} &= \bar{n}_+ \bar{n}_- \bar{n}_\times e_1 e_2 n_+, \\ I_{IO} &= \bar{n}_+ e_1 e_2 n_+ n_- n_\times, \end{aligned}$$

and the inner product as

$$\begin{aligned} A_O &= A_I \cdot I_{IO}, \\ A_I &= A_O \cdot I_{OI}. \end{aligned} \tag{1.7}$$

Additionally, an IPNS representation of the intersection of two conics is computed as wedge of two IPNS conics, [17], i.e.

$$\left(Q^1 \cap Q^2\right)_I = Q^1_I \wedge Q^2_I. \tag{1.8}$$

Such a representation of the intersection of two conics is called a *four-point*.

Let us also note that a type and features of conic Q can be read off its matrix representation, [20], which is obtained using (1.3), as the symmetric matrix

$$M = \begin{pmatrix} -\frac{1}{2}(\bar{v}^+ + \bar{v}^-) & -\frac{1}{2}\bar{v}^\times & \frac{1}{2}v^1 \\ -\frac{1}{2}\bar{v}^\times & -\frac{1}{2}(\bar{v}^+ - \bar{v}^-) & \frac{1}{2}v^2 \\ \frac{1}{2}v^1 & \frac{1}{2}v^2 & -v^+ \end{pmatrix} = \begin{pmatrix} q_{11} & q_{12} & q_{13} \\ q_{12} & q_{22} & q_{23} \\ q_{13} & q_{23} & q_{33} \end{pmatrix}. \tag{1.9}$$

Additionally, its principal 2×2 submatrix is also used in classification:

$$M = \begin{pmatrix} -\frac{1}{2}(\bar{v}^+ + \bar{v}^-) & -\frac{1}{2}\bar{v}^\times \\ -\frac{1}{2}\bar{v}^\times & -\frac{1}{2}(\bar{v}^+ - \bar{v}^-) \end{pmatrix} = \begin{pmatrix} q_{11} & q_{12} \\ q_{12} & q_{22} \end{pmatrix}. \tag{1.10}$$

The coefficients of matrix (1.9) also define usual equations of a conic in $\mathbb{R}^2$ and $\mathbb{RP}^2$, respectively, [20, 13]:

$$Q_{\mathbb{R}^2} : q_{11}x^2 + 2q_{12}xy + q_{22}y^2 + 2q_{13}x + 2q_{23}y + q_{33} = 0, \tag{1.11}$$

$$Q_{\mathbb{RP}^2} : q_{11}x^2 + 2q_{12}xy + q_{22}y^2 + 2q_{13}xz + 2q_{23}yz + q_{33}z^2 = 0. \tag{1.12}$$

1.2. Projectivisation of GAC

Although the improper points are geometrically essential for conic sections, GAC originally covered the representation of proper points only, [17]. Fortunately, as shown in [21, 22], GAC can be used for representation of both proper and improper points. Namely, this GAC representation can be reached by extending the domain of the point embedding $C : \mathbb{R}^2 \to \mathbb{R}^{5,3}$ to $\mathbb{RP}^2$ according to the following definition.

Definition 1.3. Using embedding $C : \mathbb{R}^2 \to \mathbb{R}^{5,3}$ of the form (1.1), we define projective embedding $C\mathbb{P} : \mathbb{RP}^2 \to \mathbb{R}^{5,3}$ of a point $p = (a, b, c)$, $(a, b, c) \neq (0, 0, 0)$, of the real projective plane $\mathbb{RP}^2$ as

$$C\mathbb{P}(a, b, c) = c^2\bar{n}_+ + ace_1 + bce_2 + \frac{1}{2}(a^2 + b^2)n_+ + \frac{1}{2}(a^2 - b^2)n_- + abn_\times.$$

Corollary 1.1. *Since a proper point $p = (x, y)$ has homogeneous coordinates $(x, y, 1)$, both projective embedding $C\mathbb{P}$ and embedding C maps a proper point into GAC without difference:*

$$C\mathbb{P}(x, y, 1) \equiv C(x, y) = \bar{n}_+ + xe_1 + ye_2 + \frac{1}{2}(x^2 + y^2)n_+ + \frac{1}{2}(x^2 - y^2)n_- + xyn_\times.$$

Furthermore, an improper point $p_\infty = (s, t)$ with homogeneous coordinates $(s, t, 0)$ is embedded into GAC in a similar way:

$$C\mathbb{P}(s, t, 0) = \frac{1}{2}(s^2 + t^2)n_+ + \frac{1}{2}(s^2 - t^2)n_- + stn_\times.$$

Subsequently, creating an analogy with the IPNS vector form (1.6) of a proper point, we can define an IPNS vector form of an improper point as

$$P_{\infty I} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & \frac{1}{2}(s^2 + t^2) & \frac{1}{2}(s^2 - t^2) & st \end{pmatrix}^T.$$

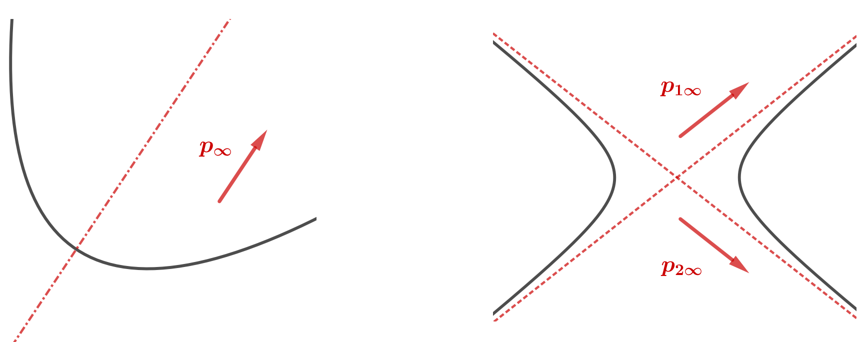


Figure 1.1: Improper points of a parabola and a hyperbola

2. Conic fitting

2.1. Overview of algorithms

2.1.1. Original GAC fitting algorithm

The first implementation of a GAC-based conic fitting algorithm was proposed by Hrdina, Návrat and Vašík in [18], and the corresponding conic fitting problem is formulated as follows: For N_D data points embedded into GAC as vectors P_i of the form (1.6) and for a conic represented by a vector Q of the form (1.5), let us minimise the objective function

$$Q \mapsto \sum_i (P_i \cdot Q)^2,$$

i.e. a sum of squared point-to-conic algebraic distances. The authors of [18] also assume the normalisation constraint

$$Q^2 = 1,$$

where the square is meant w.r.t. inner product in GAC. The algorithms presented further are based upon this very algorithm, to which we refer throughout the work as to algorithm **Q**.

2.1.2. Algorithms with additional geometric conditions

One of the main features of algorithm **Q** is that the sought conic is not limited in advance by any prescribed geometric condition. In other words, parameters of the fitted conic such as its axial tilt or the centre point position in the plane are not known beforehand, being just a result of the optimisation process.

While this general type of fit may be useful in some cases, there are problems demanding a fit with one or more additional geometric conditions constraining the output conic. In this section, we focus on forming three types of fitting algorithms, each resulting in either of following conics:

1. *conic having its axes aligned with (i.e. parallel to) the coordinate axes*
2. *conic having its centre point at the origin of the coordinate system*
3. *conic satisfying both* 1. *and* 2.

After inspecting the meaning of the coefficients of IPNS conic Q of the form (1.5) and using a part of the conic section theory, it can be shown that either of conics of types 1.–3. can be obtained from the vector representation (1.5) simply by setting some of the coefficients to zero, namely, according to Theorem 2.1, [21, 22].

Theorem 2.1. *Conic Q represented in the form of vector* (1.5) *is*

1. ***axes-aligned*** $\Leftrightarrow \bar{v}^\times = 0$,
2. ***origin-centred*** $\Leftrightarrow (v^1 = 0)$ *and* $(v^2 = 0)$,
3. ***axes-aligned, origin-centred*** $\Leftrightarrow (\bar{v}^\times = 0)$ *and* $(v^1 = 0)$ *and* $(v^2 = 0)$.

In other words, after denoting the mentioned conics respectively as Q^{al}, Q^0 and $Q^{al,0}$, their general forms are:

$$\begin{aligned} Q^{al} &= \begin{pmatrix} 0 & \bar{v}^- & \bar{v}^+ & v^1 & v^2 & v^+ & 0 & 0 \end{pmatrix}^T, \\ Q^0 &= \begin{pmatrix} \bar{v}^\times & \bar{v}^- & \bar{v}^+ & 0 & 0 & v^+ & 0 & 0 \end{pmatrix}^T, \\ Q^{al,0} &= \begin{pmatrix} 0 & \bar{v}^- & \bar{v}^+ & 0 & 0 & v^+ & 0 & 0 \end{pmatrix}^T. \end{aligned}$$

Following the example of the previous algorithms, let us call the algorithms with additional geometric conditions presented above after their respective resulting conics, i.e. algorithms **QAL**, **Q0** and **QAL0**.

Moreover, as shown in [24], there are more ways to fit an origin-centred conic Q^0 or an axes-aligned, origin-centred conic $Q^{al,0}$ to a point set. The first alternative way is reached after inspection of the used matrices and vectors, using their elements and thus yielding the direct computation of the fitted conics, without need to obtain the solution with an eigenproblem. Let us denote these algorithms as **Q0-dir** and **QAL0-dir** and remark that, as proved in [24], they produce conics equivalent to those of algorithms **Q0** and **QAL0**, respectively.

The second alternative way of fitting Q^0 and $Q^{al,0}$ is *symmetrisation* of the point set as a preprocessing step and then fitting the symmetrised point set using algorithm **Q**. Let us denote these algorithms as **Q0-sym** and **QAL0-sym** and stress that, again, they produce conics equivalent to those of algorithms **Q0** (and **Q0-dir**) and **QAL0** (and **QAL0-dir**), respectively, [24].

Application of the presented algorithms (including the original algorithm **Q**) on a sample dataset of 10 points can be seen in Figure 2.1.

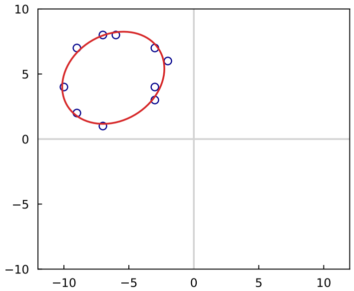
2.1.3. Iterative algorithms

As shown in [18], the original GAC-based algorithm possesses invariance w.r.t. rotations and isotropic scaling, but it is not invariant w.r.t. translations, which is rather an inconvenient property of the fit, since a point set is generally fitted with a different conic than the same point set translated. Fortunately, we can largely remedy this flaw by using the iterative algorithm described in [6], which is "almost invariant" (i.e. invariant up to the prescribed precision) w.r.t. translations, rotations and scaling as well.

From now on, let us refer to the presented iterative algorithm as "**Q-iter**". Detailed steps of the algorithm can be found in [6].

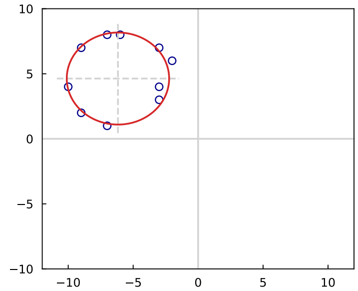
Comparison of algorithms **Q** and **Q-iter** regarding the translational (non-)invariance is shown by example in Figure 2.2, where an identical point set is translated to different locations and then fitted by both algorithms. The prescribed precision for **Q-iter** was $\varepsilon = 10^{-6}$. Let us note that besides succeeding in terms of "imperfect invariance" w.r.t. translations, iterative fitting also reduced overall point-to-conic distances. Moreover, all the ellipses obtained by **Q-iter** seem almost identical, only translated. It is also apparent that the point set closest to the origin was fitted very closely even by algorithm **Q**—indeed, empirical experience indicates that the point sets closer to the origin are fitted more tightly by **Q** than those more distant.

Let us note that a similar iterative adjustment can also be made with algorithm **QAL**, since it also entails non-invariance w.r.t. translation. By adding the same iterative fitting



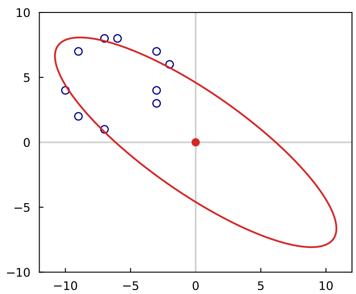
Q

(a) general conic Q

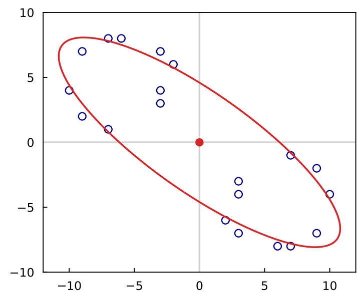


QAL

(b) axes-aligned conic Q^{al}

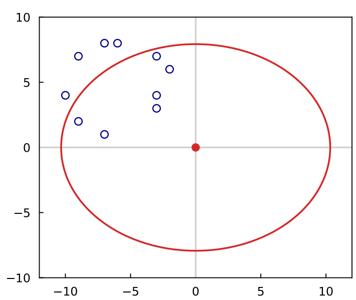


Q0, **Q0-dir**

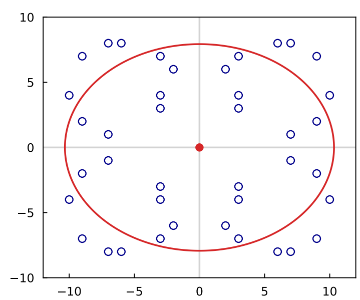


Q0-sym

(c) origin-centred conic Q^0



QAL0, **QAL0-dir**



QAL0-sym

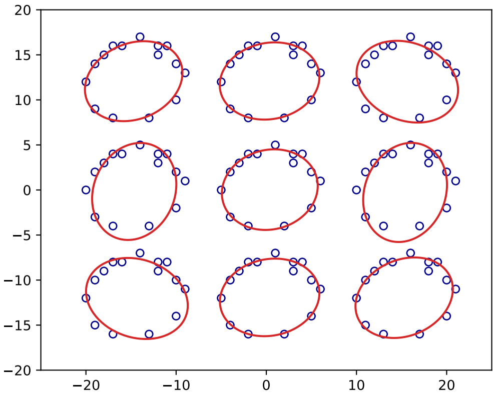
(d) axes-aligned, origin-centred conic $Q^{al,0}$

Figure 2.1: Conics with different geometrical conditions fitted by the respective algorithms

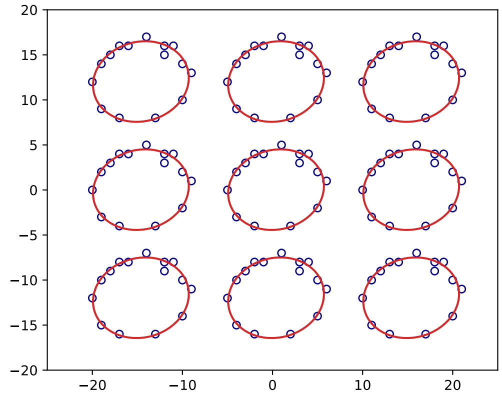
and final inverse translation of the conic into this algorithm, we can define an iterative GAC-based algorithm capable of fitting axes-aligned conics, algorithm **QAL-iter**.

2.1.4. Fitting through waypoints

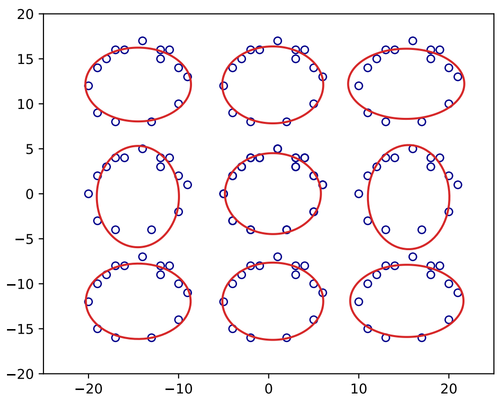
In addition to fitting a conic among the data points as closely as possible, we can also guarantee that the sought conic passes exactly through one or more points called *waypoints*. To achieve this type of fitting we can, again, make use of the vector-matrix formulation of conic fitting in terms of GAC by solving a certain eigenproblem, [21, 22].



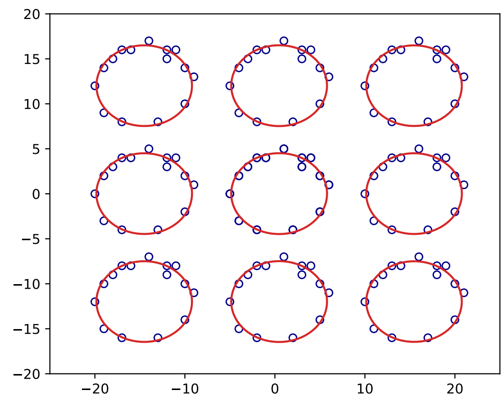
(a) Direct fitting (algorithm **Q**)



(b) Iterative fitting (algorithm **Q-iter**)



(c) Direct fitting (algorithm **QAL**)



(d) Iterative fitting (algorithm **QAL-iter**)

Figure 2.2: Comparison of the conic fitting algorithms applied on a differently translated sample point set. Precision prescribed for the iterative algorithms was $\varepsilon = 10^{-6}$, in all cases 4 or 5 iterations sufficed to achieve the precision.

Next, let us list two conic fitting algorithms solving the given optimisation problem.

The first algorithm is algorithm **QW-pseudoinv**, which has already been introduced in [21] as algorithm **QW**[2].

To also cover the cases in which algorithm **QW-pseudoinv** fails to fit the conic through all the waypoints (as will be further elaborated later) we introduce a new conic fitting algorithm, algorithm **QW-null**, [22]. As indicated by its name, the algorithm computes the fitted conic passing through given waypoint(s) using a null space of a certain matrix.

In Figure 2.3, we can see four cases of fitting the elliptical dataset, each fit being the result of fitting a conic with one more proper waypoint than in the preceding subfigure. In case 1A, both algorithms fit an ellipse passing through the waypoint, as expected. Although both algorithms work in case 2A as well, the result is still somehow unex-

[2]Renaming of the algorithm was done in order to distinguish it from a newer algorithm which fits a conic through given waypoint(s) as well. Moreover, the new name of the algorithm stresses that it reaches the solution using the Moore-Penrose inverse, also called *pseudoinverse*, [26]

pected, since both fitted conics are not only similar, they are actually (both algorithms are equivalent when exactly two points are employed, as discussed in [22]).

A striking difference between the two algorithms can be seen in cases 3A and 4A—while **QW-pseudoinv** fails to fit a conic through the given waypoints, **QW-null** succeeds in that. In particular, when we employ more than two waypoints in our conic fitting problem, an overdetermined system of linear equations occurs in the process for **QW-null** (for details see [22]). On the other hand, conics fitted by **QW-null** pass (and must pass) through the given waypoints because of the definition of the solution given as a part of the null space of a certain matrix.

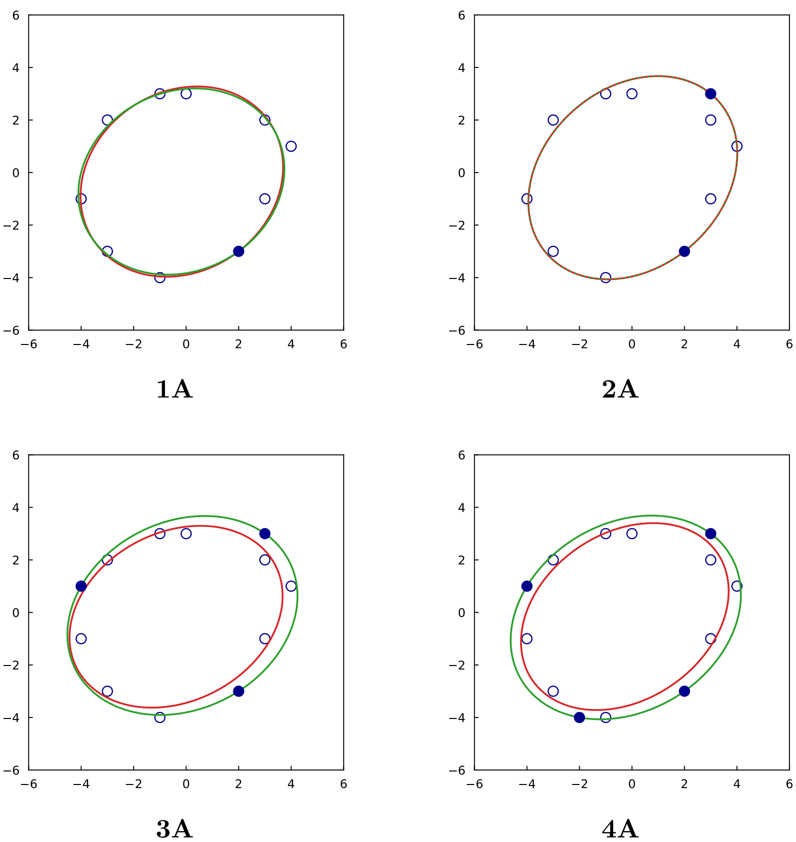


Figure 2.3: Conic fits through 1–4 proper waypoints (**QW-pseudoinv** red, **QW-null** green)

2.2. Influence of noise

Generally, noise is a natural part of empirical data, and, as such, its influence on practical functioning of the fitting algorithms is worthy of attention. In addition to the previous subsections—where the algorithms were demonstrated on small datasets—let us further evaluate stability and meaningfulness of the presented algorithms by using them on bigger point sets corrupted by noise.

Let us show functioning of the algorithms **Q**, **QAL**, **Q0** and **QAL0** under the influence of noise. For a more detailed analysis of influence of noise on the algorithms, see the full version of the thesis.

Example 2.1. *Let us assume a group of reference ellipses, each characterised by a quintuplet of parameters of the form*

$$\{x_c, y_c, a, b, \theta\},$$

i.e. the coordinates of the centre, the lengths of semi-major and semi-minor axes and the angle of tilt between the positive x-axis and the major principal axis of the ellipse. In particular, the first reference ellipse reads

$$E_{ref}^{al,0} = \{0, 0, 6, 3, 0\},$$

so it is origin-centred, it has the lengths of semi-major and semi-minor axis 6 and 3, respectively, and it is axes-aligned. By successive rotation and/or translation, we create the other reference ellipses:

$$\begin{aligned} E_{ref} &= \left\{-8, 5, 6, 3, \frac{\pi}{6}\right\}, \\ E_{ref}^{al} &= \{-8, 5, 6, 3, 0\}, \\ E_{ref}^{0} &= \left\{0, 0, 6, 3, \frac{\pi}{6}\right\}. \end{aligned}$$

Therefore, $E_{ref}, E_{ref}^{al}, E_{ref}^{0}$ *and* $E_{ref}^{al,0}$ *comprise an ellipse in general position, an axes-aligned ellipse, an origin-centred ellipse and an axes-aligned, origin-centred ellipse, respectively.*

Now, let us create four reference point sets by dividing the circumference of the reference ellipses into 50 equidistant points, and consequently corrupt these point sets with additive zero-mean Gaussian noise with standard deviation σ*. Namely, let us alter the reference point sets using the noise with* σ *varying within the values* $\{2^{-5}, 2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}\}$*, hence, we obtain 5 noisy reference point sets for each of the original reference point sets.*

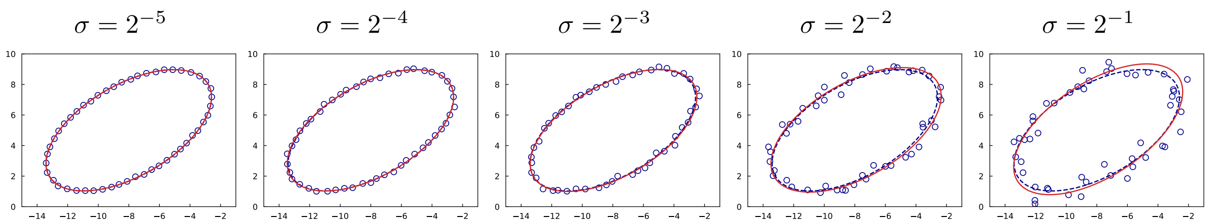
Next, let us use the algorithms ***Q****,* ***QAL****,* ***Q0*** *and* ***QAL0*** *on the noisy reference point sets with the corresponding geometric characterisation—for instance, algorithm* ***Q*** *can be used for fitting the general conics, so it will be used on the noisy reference points created from* E_{ref}*, algorithm* ***QAL*** *fits the axes-aligned conics, so it will be tested on the points from* E_{ref}^{al}*, and so on.*

Visual results of such an experiment can be found in Figure 2.4: each algorithm fitted the noisy reference point sets with an ellipse (red), which should—as much as possible—resemble the original reference ellipse (dark blue, dashed). As apparent from the figure, lower levels of the noise affect the resulting fitted ellipses very little, indeed, it is rather difficult to distinguish the fitted ellipse and the reference one. Although these differences are more evident for $\sigma = 2^{-2}, 2^{-1}$*, the fitted ellipses still resemble the reference ellipses to a great extent.*

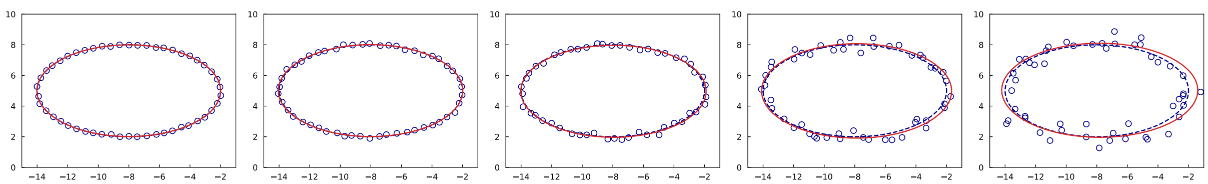
For the statistical evaluation of the repeated version of the experiment and other examples see the full version of the thesis.

3. Construction of conics using the wedge product

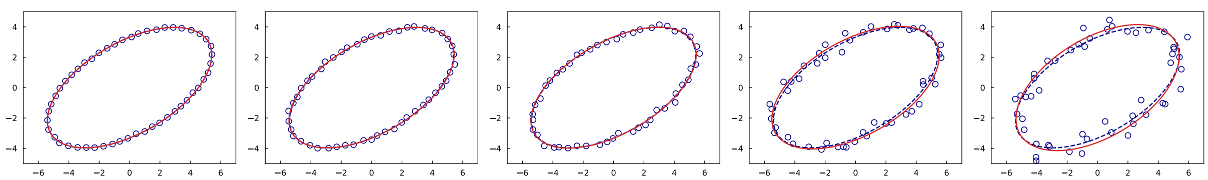
In contrast with the previous chapter where the conic fitting was discussed, the problems investigated here substantially differ in their geometrical nature. While the main goal of



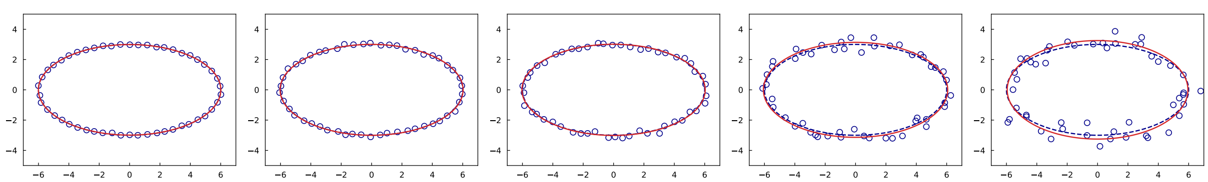
(a) $\mathbf{Q}$, general ellipses



(b) $\mathbf{QAL}$, axes-aligned ellipses



(c) $\mathbf{Q0}$, origin-centred ellipses



(d) $\mathbf{QAL0}$, axes-aligned, origin-centred ellipses

Figure 2.4: One realisation of the experiment from Example 2.1: Noisy data fitted with the respective algorithms and comparison of the fitted ellipses (red) with the reference ellipses (dark blue, dashed).

conic fitting is to find a conic reasonably *approximating* given data, the wedge product construction of conics creates a curve that passes *exactly* through a small set of points. In light of this, conic fitting through given waypoints can be viewed as a kind of intermediate problem between ordinary conic fitting and the construction of conics using the wedge product described in this chapter.

3.1. Five distinct points

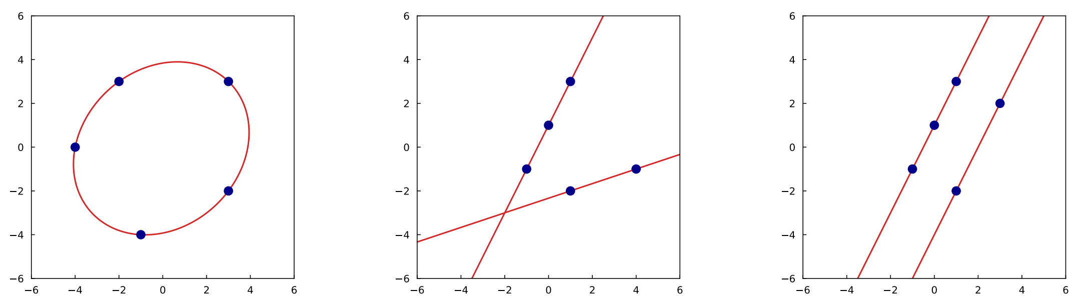
As shown in [17], an outer product null space (OPNS) representation of conic Q can be generally constructed from five distinct GAC points $P_1, \ldots, P_5$ of the form (1.4) using the wedge product:

$$Q_O = P_1 \wedge P_2 \wedge P_3 \wedge P_4 \wedge P_5.$$

Consequently, the resulting conic either passes through all the points $P_1, \ldots, P_5$ or cannot be determined uniquely, that is, it does not exist, and in such a case it holds that

$$P_1 \wedge P_2 \wedge P_3 \wedge P_4 \wedge P_5 = 0.$$

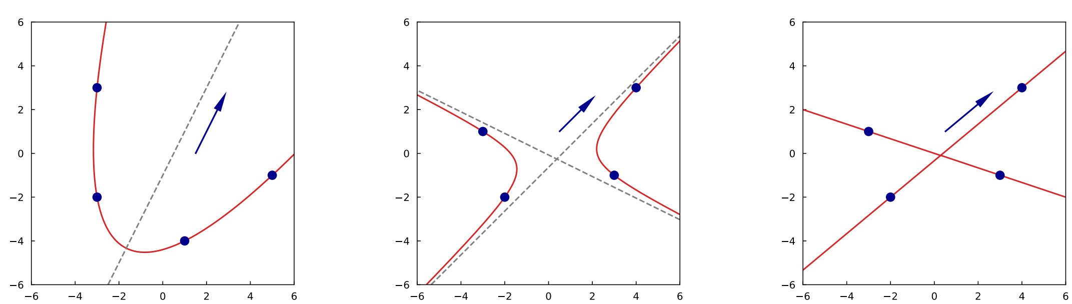
Now, let us further inspect examples of the construction of conics using GAC wedge product: An example of a non-degenerate conic spanned by 5 points in a general linear position can be seen in Figure 3.1 (a). The case when 3 out of 5 points in the set are collinear is depicted in the same figure, subfigures (b), (c)—in both (b), (c), the point set spans a unique but degenerate conic, in particular, a pair of lines (3 collinear points form the first line and the remaining 2 points form the second one).



(a) 5 points in general linear position **(b)** 3 collinear points **(c)** 3 collinear points

Figure 3.1: Conics spanned by a set of 5 proper points

An example of a parabola spanned by 4 proper points and 1 improper point can be seen in Figure 3.2 (a). For the sake of comparison, let us also examine subfigures (b), (c), where a set of 4 proper points and 1 improper point spans different conics: In (b), the improper point made a direction of an asymptote of the spanned hyperbola, while the other asymptote was a mere by-product of the construction. In (c), the situation is similar to (b), but the improper point was chosen in such a direction that it coincides with the direction connecting two of the proper points, thus resulting in a pair of lines out of which one connects the two aforementioned proper points and the other one necessarily connects the remaining two proper points.



(a) parabola **(b)** hyperbola **(c)** intersecting lines

Figure 3.2: Conics spanned by a set of 4 proper points and 1 improper point

3.2. Four-point and another point

Let us reiterate that by means of GAC, we can represent the intersection of two conics as a *four-point* (1.8). In the current subsection, we examine the construction of a conic as wedge product of a four-point and another point. Such a construction is possible, even without the necessity of computing the individual points of the four-point used, as shown by the following theorem:

Theorem 3.1. *Let Q_I^1, Q_I^2 be IPNS representations of two distinct conics $Q^1, Q^2 \subset \mathbb{RP}^2$ and P_I be an IPNS representation of point $p \in \mathbb{RP}^2$ which does not lie at the intersection of the conics Q_I^1, Q_I^2. Then Q_O computed as*

$$Q_O = \left(Q_I^1 \wedge Q_I^2\right)^* \wedge P_I \tag{3.1}$$

*where * signifies duality between IPNS and OPNS given by transition equations* (1.7) *and Q_O of the form* (3.1) *is an OPNS representation of conic $Q \subset \mathbb{RP}^2$.*

Now, before applying the wedge product construction of a conics described above to some special problems, let us recall a term closely related to the four-point and the intersection of conics—a *pencil of conics* (also called *bundle of conics*), [31, 13].

Definition 3.1. *Pencil of conics* generated by conics Q^1 and Q^2 is a family of all conics passing through the points of intersection of Q^1 and Q^2. Moreover, if Q^1 and Q^2 are represented by equations $E_1 = 0$ and $E_2 = 0$, respectively, such a pencil can also be characterised as a set of all conics represented by a non-trivial linear combination of the equations $E_1 = 0$ and $E_2 = 0$, i.e. as

$$\{Q : \lambda E_1 + \mu E_2 = 0, \quad \lambda, \mu \in \mathbb{R}, (\lambda, \mu) \neq (0, 0)\}.$$

$E_1 = 0$ and $E_2 = 0$ are equations of the form (1.11) or (1.12), depending on whether the conics are considered in $\mathbb{R}^2$ or $\mathbb{RP}^2$, [27, 31, 5].

The points of intersection of the two generating conics in the pencil are then called the *base points*.

With the knowledge of the concepts of a four-point and a pencil of conics, construction of a conic by wedging a four-point and another point can be perceived as choosing one conic out of the whole pencil of conics. In the following text, we will use this finding as a means to constructing some of the geometrically significant conics that are contained in pencils of conics.

3.2.1. Line-pairs in a pencil of conics

One of the important sets of conics found in a pencil of conics is the set of *degenerate conics*, consisting of at most three *line-pairs* passing through all the base points of the pencil. Furthermore, a point where two lines of an intersecting line-pair intersect is called a *double point* (or a *diagonal point*), therefore, every pencil also contains at most three double points. An example of a pencil of conics and the corresponding line-pairs can be seen in Figure 3.3. Moreover, let us note that *not* all the base points, double points and line-pairs are necessarily real, [31, 34, 13].

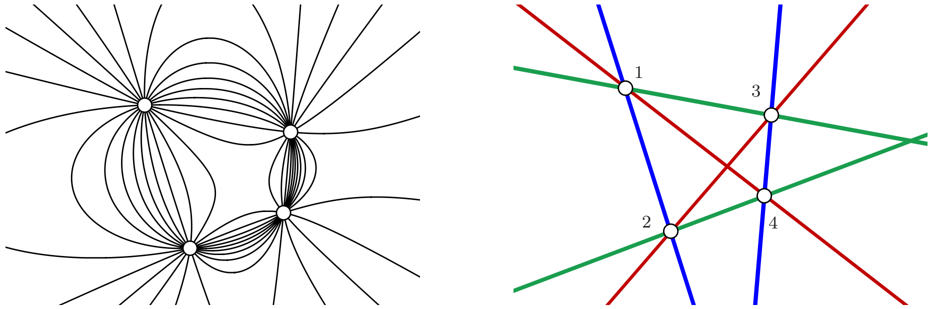


Figure 3.3: Pencil of conics through four points and three degenerate conics, i.e. line-pairs ([31])

The most usual way of finding the degenerate members of a pencil is by exploiting the algebraic characterisation of a degenerate conic, i.e. that a degenerate conic is represented by a singular matrix. As stated earlier, any conic Q in a pencil generated by conics Q^1 and Q^2 can be expressed as a linear combination of these two conics and the same holds for the matrices representing the conics. Thus, if conics Q^1 and Q^2 are represented by matrices M_1 and M_2 of the form (1.9) then Q is represented by $M = \lambda M_1 + \mu M_2$. Consequently, for Q to be degenerate, M must satisfy

$$\det(M) \equiv \det(\lambda M_1 + \mu M_2) = 0 \tag{3.2}$$

which comprises a cubic equation in variables λ and μ. Algorithm for solving the equation (3.2) is in detail described in [31]. Let us note that since equation (3.2) is cubic, it can have either three real solutions corresponding to three real line-pairs or one real and two complex solutions leading to one real line-pair and two complex line-pairs. Moreover, the solutions are three including possible multiplicities, so the line-pairs are not necessarily distinct.

Now, let us describe an alternative way of computing the line-pairs of a pencil in terms of GAC setting: Since each line-pair passes through all the base points of a pencil (which are represented by a four-point) *and* through its double point, we can construct each of the line-pairs exactly as wedge product of the four-point and the double point corresponding to the particular line-pair.

While the idea of the construction is quite straightforward, finding the three double points without the explicit knowledge of the individual points of intersection is not so evident. Fortunately, double points of a pencil of conics can be computed using the concept of *polarity*, [14].

Definition 3.2. Let Q be a conic represented by matrix M and $p = (x_p, y_p, z_p)^T$ be a homogeneous point in $\mathbb{RP}^2$. Furthermore, by abuse of notation, let us identify a homogeneous line $l : ax + by + cz = 0$ with the vector of its coefficients, i.e. $l = (a, b, c)^T$. Then a homogeneous line $l = Mp$ is called a *polar line* (or simply a *polar*) of point p with respect to conic Q. Point p is then called a *pole*, [31].

Definition 3.3. Let Q be a conic and p_1, p_2, p_3 three homogeneous points in $\mathbb{RP}^2$. If polar of any of these points with respect to conic Q passes through the remaining two points, then the triangle $\triangle p_1p_2p_3$ is called a *self-polar triangle* of conic Q, [14].

Definition 3.4. Let Q^1 and Q^2 be two conics and p_1, p_2, p_3 three homogeneous points in $\mathbb{RP}^2$. If p_1, p_2, p_3 form a self-polar triangle of both conics simultaneously, the triangle is then called a *common self-polar triangle* of conics Q^1 and Q^2, [14].

Theorem 3.2. *The double points of a pencil of conics form the vertices of a common self-polar triangle of the generating conics Q^1 and Q^2, [33].*

Using the self-polarity of the double points of a pencil, their computation can be derived and summed up in the following theorem:

Theorem 3.3. *Let Q^1 and Q^2 be generating conics of a pencil of conics and M_1 and M_2 their matrices. Then, if a common self-polar triangle of the conics Q^1 and Q^2 exists, its vertices can be found as eigenvectors p of generalised eigenproblem*

$$M_1 p = \lambda M_2 p. \tag{3.3}$$

Now, to construct the line-pairs in a pencil generated by conics Q^1 and Q^2 we can first find the double points p_i according to equation (3.3) and wedge their GAC representatives with the four-point $Q^1 \wedge Q^2$ as in (3.1). In other words, we can construct an OPNS representation of the i-th line-pair LP^i of the pencil as

$$LP^i_O = \left(Q^1_I \wedge Q^2_I\right)^* \wedge C\mathbb{P}(p_i).$$

Example 3.1. *Let us consider two ellipses E^1, E^2 with equations*

$$\begin{aligned} E^1 &: 9x^2 + 25y^2 - 225 = 0, \\ E^2 &: 4x^2 + y^2 - 16x - 2y + 1 = 0, \end{aligned}$$

IPNS representations

$$\begin{aligned} E^1_I &= 16\bar{n}_- - 34\bar{n}_+ + 225 n_+, \\ E^2_I &= -3\bar{n}_- - 5\bar{n}_+ - 16e_1 - 2e_2 - n_+, \end{aligned}$$

and the associated matrices

$$M_1 = \begin{pmatrix} 9 & 0 & 0 \\ 0 & 25 & 0 \\ 0 & 0 & -225 \end{pmatrix}, \quad M_2 = \begin{pmatrix} 4 & 0 & -8 \\ 0 & 1 & -1 \\ -8 & -1 & 1 \end{pmatrix}.$$

Consequently, the solution to generalised eigenproblem (3.3) *is given by*

$$\begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{pmatrix} \approx \begin{pmatrix} 13.0501 \\ 21.6502 \\ 2.7997 \end{pmatrix}, \quad \begin{pmatrix} p_1 & p_2 & p_3 \end{pmatrix} \approx \begin{pmatrix} 2.4167 & 2.2320 & 10.1865 \\ -1.0921 & -6.4631 & -0.1261 \\ 1 & 1 & 1 \end{pmatrix},$$

and the IPNS representation of the points of intersection P_1, P_2, P_3 of the sought line-pairs are computed as an embedding of the obtained eigenvectors to GAC, i.e.

$$P_i = C\mathbb{P}(p_i), \qquad i = 1, 2, 3.$$

Finally, each of the line-pairs is computed using wedge product according to (3.1)*:*

$$LP_O^i = \left(E_I^1 \wedge E_I^2\right)^* \wedge P_i, \qquad i = 1, 2, 3.$$

After conversion to IPNS, the representations of the acquired line-pairs read

$$LP_I^1 \approx -6181.4079\bar{n}_- + 5067.3281\bar{n}_+ - 11348.6259e_1 - 1418.5782e_2 - 57712.2031n_+,$$
$$LP_I^2 \approx 602.7278\bar{n}_- + 341.5317\bar{n}_+ + 2281.9561e_1 + 285.2445e_2 + 2601.6050n_+,$$
$$LP_I^3 \approx -3231.0342\bar{n}_- - 2963.6289\bar{n}_+ - 13826.2175e_1 - 1728.2772e_2 - 9844.7100n_+.$$

Both conics, the intersection four-point and the constructed line-pairs with their points of intersection can be seen in Figure 3.4.

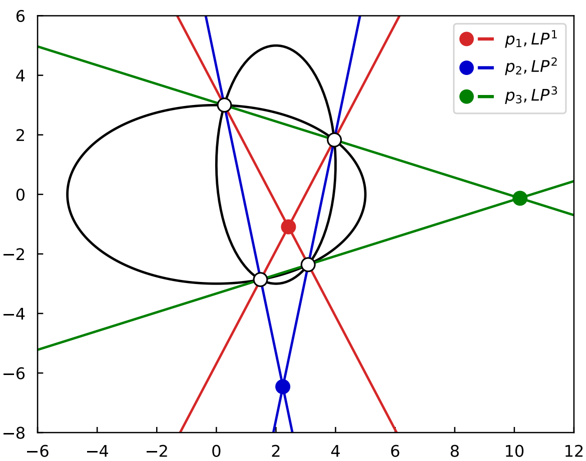


Figure 3.4: Four-point obtained as an intersection of two conics from Example 3.1. Each of the three line-pairs was constructed by wedging the four-point and the corresponding point of intersection.

Furthermore, a construction of the line-pairs of a pencil can be particularly easy in certain settings, e.g. the cases when two intersecting conics are concentric, as seen in the following example.

Example 3.2. *Let us consider two concentric ellipses* E^1, E^2 *depicted in Figure 3.5(a). Since both are concentric, it is obvious that their intersection points are symmetric with respect to their common centre, hence, one of the line-pairs passing through the intersection four-point must also pass through the centre, which is the point* $(0, 0, 1)$ *in this case. Additionally, because of concentricity of both conics, the other two line-pairs must be pairs of parallel lines intersecting in improper points. Also, both conics are axes-aligned, therefore, the parallels pass through the improper points corresponding to the directions of the x-axis (point* $(1, 0, 0)$*) and the y-axis (point* $(0, 1, 0)$*), respectively. Thus, thanks to the trivial setting of the intersecting conics and the possibility of expressing improper points in terms of GAC, the construction of the line-pairs may proceed in a more straightforward* *way than in Example 3.1, without a need to compute the eigenvectors. Wedging in this case simply proceeds in following way:*

$$LP_O^1 = \left(E_I^1 \wedge E_I^2\right)^* \wedge C\mathbb{P}(0,0,1),$$
$$LP_O^2 = \left(E_I^1 \wedge E_I^2\right)^* \wedge C\mathbb{P}(0,1,0),$$
$$LP_O^3 = \left(E_I^1 \wedge E_I^2\right)^* \wedge C\mathbb{P}(1,0,0).$$

Ellipses E^1, E^2 in Figure 3.5 (b) are also concentric with the common centre $(0,0,1)$, but one of the ellipses is rotated by 30°. The situation is almost the same as in case (a): there will be a line-pair passing through the common centre and two remaining line-pairs will be two pairs of parallels. A crucial difference between cases (a) and (b) is that the directions of the parallels do not coincide with the directions of the coordinate axes and must be computed using eigenproblem (3.3)*. The eigenvectors for these two conics yield*

$$\begin{pmatrix} p_1 & p_2 & p_3 \end{pmatrix} \approx \begin{pmatrix} 0 & -0.2525 & 2.2281 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix},$$

and the construction of the line-pairs is given as

$$LP_O^1 = \left(E_I^1 \wedge E_I^2\right)^* \wedge C\mathbb{P}(p_1),$$
$$LP_O^2 = \left(E_I^1 \wedge E_I^2\right)^* \wedge C\mathbb{P}(p_2),$$
$$LP_O^3 = \left(E_I^1 \wedge E_I^2\right)^* \wedge C\mathbb{P}(p_3).$$

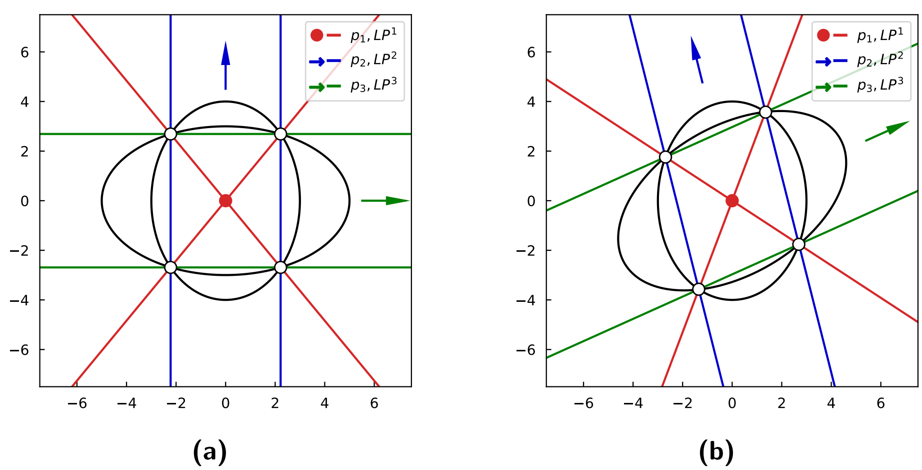


Figure 3.5: Four-point as an intersection of two concentric conics. One out of three line-pairs passing through the four-point is constructed by wedging the four-point with the common centre; the remaining two line-pairs were obtained by wedging the four-point with an improper point.

3.2.2. Generalised parabolas in a pencil of conics

Another special subset of conics found in a pencil of conics consists of so called *generalised parabolas*. As will be shown further, there can be either none, one, two or—under special circumstances—an infinite number of generalised parabolas in a pencil of conics.

Let us recall that a parabola is a non-degenerate, non-central conic, therefore, if it is characterised by matrix M and principal submatrix $\bar{M}$, it holds that

$$\det(M) \neq 0,$$
$$\det(\bar{M}) = 0.$$

Nevertheless, as a hyperbola can degenerate into a pair of intersecting lines, so a parabola can degenerate, in particular, into a pair of parallel lines or into a double line. Moreover, the double line can also be a double line at infinity. Consequently, when looking for parabolas in a pencil of conics, let us further assume parabolas in a more general sense, i.e. all non-zero, non-central conics. Therefore, such *generalised parabolas* should satisfy relations

$$M \neq 0, \tag{3.4}$$
$$\det(\bar{M}) = 0. \tag{3.5}$$

Furthermore, in addition to the mentioned generalised parabolas, let us also add that a union of an ordinary real line and the line at infinity also constitutes a generalised parabola, since it satisfies both (3.4) and (3.5).

As proved in the full version of the thesis, existence and the number of generalised parabolas in a pencil of conics is summarised by following theorem.

Theorem 3.4. *Let us assume two conics Q^1 and Q^2 represented by matrices M_1 and M_2 of the form* (1.9) *and principal submatrices $\bar{M}_1$ and $\bar{M}_2$ of the form* (1.10)*. Moreover, let us suppose matrix*

$$N = \begin{pmatrix} \delta_1 & \gamma_{12} \\ \gamma_{12} & \delta_2 \end{pmatrix},$$

where

$$\delta_1 = \det(\bar{M}_1),$$
$$\gamma_{12} = \frac{1}{2} \operatorname{Tr}\left(\operatorname{adj}\left(\bar{M}_1\right) \bar{M}_2\right),$$
$$\delta_2 = \det(\bar{M}_2).$$

Then the number of generalised parabolas in the generated pencil can be classified w.r.t. M_1, M_2 and N according to the overview given by Table 3.1.

Every generalised parabola (except those containing the line at infinity) has exactly one improper point, i.e. a point at infinity associated with the direction of its axis of symmetry, therefore, it is natural to wedge a four-point with this particular improper point. As will be shown further, the way of computing the improper points of the conjugate parabolas structurally resembles the computation of the vertices of the common self-polar triangle of two conics, although its geometric meaning is slightly different.

The concept of so called *conjugate directions* will enable us to compute the improper points of the generalised parabolas in a pencil, as we will see in the following theorem. For the details on the term, see the full version of the thesis.

Table 3.1: Number of generalised parabolas found in a pencil generated by conics Q^1 and Q^2 with matrices M_1 and M_2 w.r.t. their relationship and to the matrix N. Cases denoted with bold style.

	$M_2 \neq kM_1$, $k \neq 0$	$M_2 = kM_1$, $k \neq 0$
$\det(N) < 0$	**A1** 2 real parabolas	**B1** –
$\det(N) = 0$, $N \neq 0$	**A2** 1 real parabola	**B2** no parabola
$\det(N) > 0$	**A3** no real parabola (2 imaginary parabolas)	**B3** –
$N = 0$	**A4** infinite number of distinct real parabolas	**B4** coincident real parabolas Q^1 and Q^2

Theorem 3.5. *Let Q^1 and Q^2 be generating conics of a pencil of conics and $\bar{M}_1$ and $\bar{M}_2$ their principal submatrices. Then, if common conjugate directions of conics Q^1 and Q^2 exist, they can be found as eigenvectors $\bar{p}_\infty$ of generalised eigenproblem*

$$\bar{M}_1 \bar{p}_\infty = \lambda \bar{M}_2 \bar{p}_\infty. \tag{3.6}$$

Now, to construct the generalised parabolas in a pencil generated by Q^1 and Q^2, we will first find the directions $\bar{p}_{\infty j}$ according to equation (3.6), construct the associated improper points $p_{\infty j}$ by adding zero as the third coordinate and then wedge their GAC representatives with the four-point $Q^1 \wedge Q^2$ as in (3.1). More mathematically, we can construct an OPNS representation of the j-th generalised parabola P^j of the pencil as

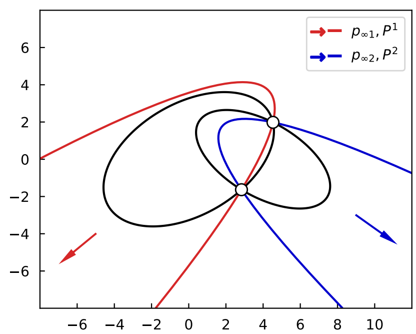
$$P^j_O = \left(Q^1_I \wedge Q^2_I\right)^* \wedge C\mathbb{P}(p_{\infty j}).$$

Now, since construction of general parabolas using the wedge product proves especially useful for cases A1 (as can be seen in the full version of the thesis), let us offer some examples of them.

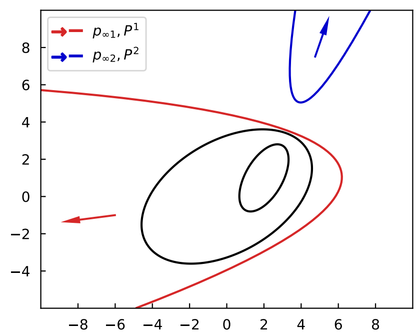
Example 3.3. *(1) Let us consider two settings from where both pairs of conics have four distinct intersection points. In the first case, the conics intersect in two real and two imaginary points of intersection, nevertheless, two axial directions were computed and two associated real parabolas of the pencil were successfully constructed using the wedge product. The second case may come as a bigger surprise, since the conics have no real intersection points, yet two real distinct parabolas of the pencil were found as well. These settings are depicted in Figure 3.6 (a,b).*

(2) Now, let us assume a few pairs of conics with other relative positions. In each case, both axial directions are computed and the associated pair of generalised parabolas is constructed. Respective situations can be seen in Figure 3.6 (c)–(f). It is apparent that in addition to parabolas in usual sense, other generalised parabolas, like parallel lines or a double line, may truly appear in a pencil.

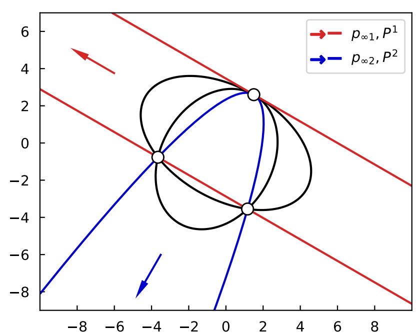
Finally, inspection of matrix N in the mentioned cases would confirm that, indeed, all the cases fall under the category A1, where two real distinct parabolas are found, regardless the relative position of the conics.



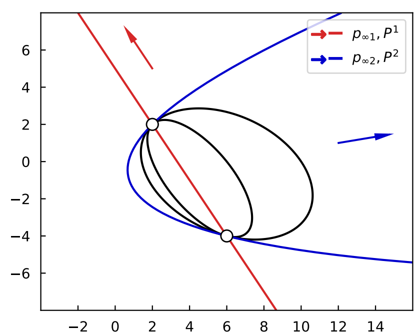
(a) two real and two imaginary points of intersection



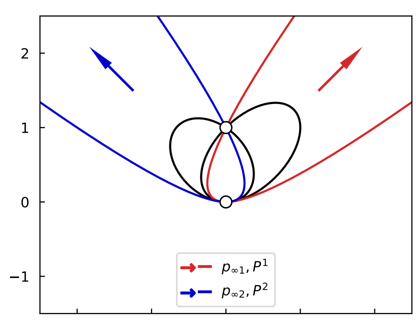
(b) four imaginary points of intersection



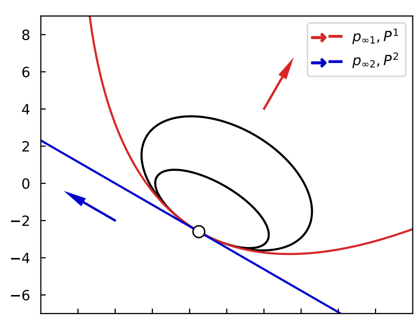
(c) 1 point of twofold contact



(d) 2 points of twofold contact



(e) 1 point of threefold contact



(f) 1 point of fourfold contact

Figure 3.6: Two conics in different relative positions from Example 3.3 generating a pencil and the parabolas of the pencil constructed with wedge product

References

[1] Brown, K., Bourbakis, N. G. (2021). Curve and surface fitting techniques in computer vision. *International Journal of Image and Graphics*, 21(04). https://doi.org/10.1142/s0219467821500418

[2] Byrtus, R., Derevianko, A., Vašík, P. (2020). Outline of tube elbow detection based on GAC. In *Lecture Notes in Computer Science* (pp. 482–491). https://doi.org/10.1007/978-3-030-61864-3_41

[3] Byrtus, R., Derevianko, A., Vašík, P., Hildenbrand, D., Steinmetz, C. (2021). On specific Conic intersections in GAC and symbolic calculations in GAALOPWeb. *Advances in Applied Clifford Algebras*, 32(1). https://doi.org/10.1007/s00006-021-01182-z

[4] Chen, B. X., Tsotsos, J. K. (2019). Fast Visual Object Tracking with Rotated Bounding Boxes. *arXiv (Cornell University)*. https://arxiv.org/pdf/1907.03892.pdf

[5] Chomicki, C., Breuils, S., Biri, V., Nozick, V. (2024). Intersection of Conic Sections Using Geometric Algebra. In: Sheng, B., Bi, L., Kim, J., Magnenat-Thalmann, N., Thalmann, D. (eds) *Advances in Computer Graphics. CGI 2023. Lecture Notes in Computer Science*, vol 14498. Springer, Cham. https://doi.org/10.1007/978-3-031-50078-7_14

[6] Derevianko, A., Loučka, P. (2022). Search for similarity transformation between image point clouds using geometric algebra for conics. In *Lecture Notes in Computer Science* (pp. 215–226). https://doi.org/10.1007/978-3-030-98260-7_13

[7] Derevianko, A., Vašík, P. (2022). Solver-free optimal control for linear dynamical switched system by means of geometric algebra. *Mathematical Methods in the Applied Sciences*. https://doi.org/10.1002/mma.8752

[8] Doignon, C., De Mathelin, M. (2007). A Degenerate Conic-Based Method for a Direct Fitting and 3-D Pose of Cylinders with a Single Perspective View. *Proceedings*. https://doi.org/10.1109/robot.2007.364128

[9] Downs, J. W. (2003). *Practical conic sections: the geometric properties of ellipses, parabolas and hyperbolas.*

[10] Fitzgibbon, A.W., Fisher, R.B. (1995). A Buyer's Guide to Conic Fitting. *British Machine Vision Conference.*

[11] Fitzgibbon, A., Pilu, M., Fisher, R. B. (1999). Direct least square fitting of ellipses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5), 476–480. https://doi.org/10.1109/34.765658

[12] Gander, W., Golub, G. H., Strebel, R. (1994). Least-squares fitting of circles and ellipses. *Bit Numerical Mathematics*, 34(4), 558–578. https://doi.org/10.1007/bf01934268

[13] Glaeser, G., Stachel, H., Odehnal, B. (2018). *The Universe of Conics: From the ancient Greeks to 21st century developments.*

[14] Guo, Y. (2020). Homography Estimation from Ellipse Correspondences Based on the Common Self-polar Triangle. *Journal of Mathematical Imaging and Vision*, 62(2), 169-188. ISSN 0924-9907. https://doi.org/10.1007/s10851-019-00928-6

[15] Hildenbrand, D. (2013). Foundations of Geometric Algebra Computing. In *Geometry and computing*. https://doi.org/10.1007/978-3-642-31794-1

[16] Hildenbrand, D. (2020). *Introduction to Geometric Algebra Computing.*

[17] Hrdina, J., Návrat, A., Vašík, P. (2018). Geometric algebra for Conics. *Advances in Applied Clifford Algebras*, 28(3). https://doi.org/10.1007/s00006-018-0879-2

[18] Hrdina, J., Návrat, A., Vašík, P. (2019). Conic fitting in geometric algebra setting. *Advances in Applied Clifford Algebras*, 29(4). https://doi.org/10.1007/s00006-019-0989-5

[19] Kanatani, K., Sugaya, Y., Kanazawa, Y. (2016). Ellipse Fitting for Computer Vision: Implementation and Applications. *Synthesis Lectures on Computer Vision*, 6(1), 1–141. https://doi.org/10.2200/s00713ed1v01y201603cov008

[20] Korn, G. A. (1961). *Mathematical handbook for scientists and engineers.*

[21] Loučka, P. (2023). On Proper and Improper Points in Geometric Algebra for Conics and Conic Fitting Through Given Waypoints. *Lecture Notes in Computer Science*, 67–79. https://doi.org/10.1007/978-3-031-30923-6_6

[22] Loučka, P., Vašík, P. Algorithms for Conic Fitting Through Given Proper and Improper Waypoints in Geometric Algebra for Conics. *Advances in Applied Clifford Algebras* 34, 6 (2024). https://doi.org/10.1007/s00006-023-01308-5

[23] Loučka, P., Vašík, P. (2021). Algorithms for multi-conditioned conic fitting in geometric algebra for Conics. In *Lecture Notes in Computer Science* (pp. 645–657). https://doi.org/10.1007/978-3-030-89029-2_48

[24] Loučka, P., Vašík, P. (2023) On multi-conditioned conic fitting in Geometric algebra for conics. *Advances in Applied Clifford Algebras*, 33(31). https://doi.org/10.1007/s00006-023-01277-9

[25] Meng, X., Hu, Z. (2003). A new easy camera calibration technique based on circular points. *Pattern Recognition*, 36(5), 1155–1164. https://doi.org/10.1016/s0031-3203(02)00225-x

[26] Moore, E. H. (1920). On the reciprocal of the general algebraic matrix. *Bulletin of the American Mathematical Society*. 26 (9): 394–95. doi:10.1090/S0002-9904-1920-03322-7

[27] Pamfilos, P. (2014). A Gallery of Conics by Five Elements. *Forum Geometricorum*, 14, 295-348.

[28] Perwass, C. (2009). Geometric Algebra with Applications in Engineering. In *Springer eBooks*. https://doi.org/10.1007/978-3-540-89068-3

[29] Qian, Z., Tang, F., Liu, B., Fu, Y., Wu, S., Jia, X., Wu, Y. (2022). Automatic detection and fitting of Ellipse markers using EllipseNet. *2022 26th International Conference on Pattern Recognition (ICPR).* https://doi.org/10.1109/icpr56361.2022.9956313

[30] Rahayem, M., Werghi, N., Kjellander, J. (2012). Best ellipse and cylinder parameters estimation from laser profile scan sections. *Optics and Lasers in Engineering*. 50(9), 1242 – 1259

[31] Richter-Gebert, J. (2011). *Perspectives on Projective Geometry: A guided tour through real and complex geometry.*

[32] Rosin, P. L. (1993). A note on the least squares fitting of ellipses. *Pattern Recognition Letters*, 14(10), 799–808. https://doi.org/10.1016/0167-8655(93)90062-i

[33] Semple, J., Kneebone, G. (1952) *Algebraic Projective Geometry*. Oxford University Press, Oxford.

[34] Thomas, A. (2020) Geometric Characterizations of the Cross Ratio in a Pencil of Conics. https://arxiv.org/abs/2011.03811

[35] Ventura, J., Hrdina, J., Eid, A. H. A., Stodola, M., Arrabal-Campos, F. M., Alcayde, A., Montoya, F. G. (2024). On the Fitting of Three-Phase Voltage Curves with Ellipses Using Geometric Algebra for Conics. *22nd International Conference on Renewable Energies and Power Quality (ICREPQ'24)*, Bilbao, Spain.

[36] Waibel, P., Matthes, J., Gröll, L. (2015). Constrained Ellipse Fitting with Center on a Line. *Journal of Mathematical Imaging and Vision*, 53(3), 364–382. https://doi.org/10.1007/s10851-015-0584-x

[37] Wu, Y., Wang, H., Tang, F. (2017). Conic Fitting: New Easy Geometric Method and Revisiting Sampson Distance. *2017 4th IAPR Asian Conference on Pattern Recognition (ACPR)*, Nanjing, China, pp. 37-42.

[38] Wu, Y., Wang, H., Tang, F., Wang, Z. (2019). Efficient conic fitting with an analytical Polar-N-Direction geometric distance. *Pattern Recognition*, 90, 415–423. https://doi.org/10.1016/j.patcog.2019.01.023

[39] Ying, X., Hu, Z. (2004). Catadioptric camera calibration using geometric invariants. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10), 1260–1271. https://doi.org/10.1109/tpami.2004.79

Curriculum vitae

Ing. Pavel Loučka

Personal info

Date and place of birth:	September 3rd 1994, Brno
Citizenship:	Czech Republic
Nationality:	Czech
E-mail:	Pavel.Loucka@vutbr.cz

Education

2014–2017: Faculty of Mechanical Engineering BUT, study branch: Mathematical Engineering, degree Bc.

2017–2019: Faculty of Mechanical Engineering BUT, study branch: Mathematical Engineering, degree Ing.

Career overview

Feb 2019–Dec 2019: Technical worker at Institute of Machine and Industrial Design, Faculty of Mechanical Engineering BUT

Feb 2020–Jan 2021: Lecturer of Mathematics at Faculty of Economics and Administration, Masaryk University

Sep 2020–Dec 2021: Researcher at University of Defence, Brno

Sep 2022–until now: Junior Lecturer at Institute of Mathematics, Faculty of Mechanical Engineering BUT

Pedagogic activities

- Teaching at FME BUT:
  - seminars: Constructive Geometry, Numerical Methods I

Academic internships abroad

- six-weeks stay at University of Palermo (ERASMUS mobility for employees, 2023)
- one-week stay at University of Palermo (ERASMUS mobility for employees, 2024)

Abstract

We present several algorithms for conic fitting and conic construction based on Geometric Algebra for Conics (GAC). First, conic fitting with additional geometric constraints is presented, i.e. fitting a conic among the dataset while also prescribing some geometric properties of the conic in advance. In particular, constraints successfully incorporated into the offered algorithms are: principal axes parallel to the coordinate axes, position of the centre at the origin of the coordinate system, and, finally, combining both constraints at the same time. Next, the iterative versions of two GAC-based algorithms were described, aiming to overcome the non-invariance w.r.t. translations of the original GAC fitting methods. Additionally, two algorithms fitting a conic among the data points in such a way that it also passes through up to four prescribed points, called waypoints, are presented as well. The second group of algorithms uses GAC outer product, wedge product, for the construction of conics, including the construction of conics from five points. Also, we propose the way of using wedge product for the construction of special conics of the pencils of conics, namely, the line-pairs and the generalised parabolas. Moreover, thanks to the projective extension of GAC presented here, both ordinary (proper) points of the Euclidean plane and the points at infinity (improper points) are used. MATLAB implementation of the conic fitting algorithms and Maple scripts for the wedge constructions presented in the thesis are a part of the electronic appendix of the work.

Abstrakt

Představujeme několik algoritmů na fitování kuželoseček a konstrukci kuželoseček založených na Geometrické algebře pro kuželosečky (zkr. GAC). Nejdříve je představeno fitování kuželoseček s dodatečnými geometrickými podmínkami, tj. fitování kuželoseček, které mají apriori předepsané vybrané geometrické vlastnosti. Jmenovitě se do fitovacích algoritmů podařilo začlenit tyto geometrické podmínky: osy jsou rovnoběžné s osami souřadné soustavy, střed leží v počátku a nakonec obě předešlé podmínky současně. Dále byly popsány iterativní verze dvou fitovacích algoritmů založených na GAC, které si kladou za cíl vypořádat se s neinvariancí vůči posunutí dat u nemodifikovaných algoritmů. Mimoto jsou představeny dva algoritmy, které kromě fitování kuželosečky mezi body dokáží nalézt kuželosečku tak, aby zároveň přesně procházela až čtyřmi předepsanými body. Druhá skupina algoritmů využívá vnější součin definovaný v GAC, zvaný wedge, ke konstrukci kuželoseček, a to včetně konstrukce kuželosečky procházející pěti body. Také navrhujeme použití wedge ke konstrukci speciálních kuželoseček ze svazků kuželoseček, konkrétně dvojic přímek a zobecněných parabol. Díky projektivnímu rozšíření GAC, které je zde představeno, jsou navíc v práci používány jak vlastní body euklidovské roviny, tak nevlastní body, tj. body v nekonečnu. Implementace fitovacích algoritmů v softwaru MATLAB a skripty určené ke konstrukci kuželoseček v softwaru Maple jsou součástí elektronické přílohy práce.