On Multi-conditioned Conic Fitting in Geometric Algebra for Conics

Pavel Loučka and Petr Vašík

Abstract. We introduce several modifications of conic fitting in Geometric algebra for conics by incorporating additional conditions into the optimisation problem. Each of these extra conditions ensure additional geometric properties of a fitted conic, in particular, centre point position at the origin of coordinate system, axial alignment with coordinate axes, or, eventually, combination of both. All derived algorithms are accompanied by a discussion of the underlying algebra and computational optimisation issues. Finally, we present examples of use on a sample dataset and offer possible applications of the algorithms.

Keywords. Conic fit, Geometric algebra, Clifford algebra, Centre position, Axial alignment.

1. Introduction

This paper is an extension of [12], intended to describe the algebraic background of the derived conic fitting algorithms.

In this paper we provide modifications to the conic fitting algorithm introduced in [10], i.e. an algorithm based on conic representation in Geometric Algebra for Conics (GAC) [9]. Particularly, besides the normalisation condition natural for GAC that was used in the original algorithm, we impose additional geometric conditions on a fitted conic, more specifically, the following: the centre point’s position at the origin of coordinate system, axial alignment with coordinate axes, and, eventually, the combination of both. The only similar geometric modification of the conic fitting algorithms known to the authors may be found in [15], where the conic is fitted in such a way that its centre lies on a prescribed line. In contrast with this approach, we obtain our solution explicitly as a direct result of the optimisation problem, thanks to its geometric formulation in GAC.

This article is part of the Topical Collection on ENGAGE 2021 edited by Andreas Aristedou.

*Corresponding author.
One of the fields, where the mentioned fits can be used, are so called *switched systems*, with conics representing the integral curves (trajectories) of underlying system of ODE’s. At the beginning, the system of ODE’s is numerically solved (e.g. by Runge-Kutta method) with the initial condition at the starting point \( A \). This will give us a set of points representing the initial conic. Subsequently, the GAC conic fitting algorithm is applied and thus we get the conic in IPNS representation, [3].

We recall the original GAC fitting algorithm in Sect. 3, where we also present its modification which decreases computational demands. Consequently, in Sect. 4 we introduce additional geometric conditions in terms of GAC and modify the original algorithm accordingly. In the end of the section, we offer the overview of the described algorithms. In Sect. 5 we provide an analysis and visualisation of the testing examples and finally, we mention a possible use of the algorithms. For MATLAB implementation of chosen conic fitting algorithms from this paper, see [12], or GitHub repository [6], where all the algorithms presented in this paper can be found.

For a detailed survey of standard algorithms not using additional geometric conditions we refer to [4]. Although all classical algorithms use a linear or quadratic constraint, each of them can be linearised in the sense that the best fit may be found using an eigenvector of a matrix operator. Yet this is the only similarity of the classical algorithms to our approach.

2. Geometric Algebra

By a geometric algebra (GA) we mean a Clifford algebra with a specific embedding of an Euclidean space (of arbitrary dimension) in such a way that the predefined geometric primitives as well as their transformations are viewed as the algebra elements, precisely multivectors. This concept has been introduced by Hestenes in [7] and has been used in various mathematical and engineering applications since, see e.g. [1]. For Geometric Algebra for Conics (GAC), proposed by Perwass [13], to generalise the concept of (2-dimensional) conformal geometric algebra \( \mathbb{C}_{3,1} \) also known as a Compass Ruler Algebra (CRA) [8], we are using the notation of [10]. Particularly, we denote the corresponding basis elements as follows

\[
\bar{n}_+, \bar{n}_-, \bar{n}_x, e_1, e_2, n_+, n_-, n_x.
\]

This notation suggests that the basis elements \( e_1, e_2 \) will play the usual role of the standard basis of the plane while the null vectors \( \bar{n}, n \) will represent either the origin or the infinity. Note that there are three orthogonal ‘origins’ \( \bar{n} \) and three corresponding orthogonal ‘infinities’ \( n \). In terms of this basis, a point of the plane \( \mathbf{x} \in \mathbb{R}^2 \) defined by \( \mathbf{x} = xe_1 + ye_2 \) is embedded using the operator \( C : \mathbb{R}^2 \to \text{Cone} \subset \mathbb{R}^{5,3} \), which is defined by

\[
C(x, y) = \bar{n}_+ + xe_1 + ye_2 + \frac{1}{2}(x^2 + y^2)n_+ + \frac{1}{2}(x^2 - y^2)n_- + xyn_x,
\]

where the image \( \text{Cone} \) of the plane in \( \mathbb{R}^{5,3} \) is an analogue of the conformal cone.
Let us also recall that the images of conics in GAC have two possible representations. Particularly, if a conic $Q \subset \mathbb{R}^2$ is represented by a vector $Q_I$ in GAC, then for a point $x \in Q$ it holds that $C(x) \cdot Q_I = 0$ and $Q_I$ is then called the IPNS (Inner Product Null Space) representation of $Q$. On the other hand, given five points $x_1, \ldots, x_5 \in \mathbb{R}^2$ determining a conic $Q$ uniquely, their wedge $C(x_1) \wedge \cdots \wedge C(x_5)$ has the property that $C(x_1) \wedge \cdots \wedge C(x_5) \wedge C(x) = 0$ for any other $x \in Q$ and the resulting five-vector is then called the OPNS (Outer Product Null Space) of $Q$. The connection between these two representation is called the duality, the details may be found in [9].

The type of a given unknown conic can be read off its matrix representation, which in our case for a conic given by vector

$$Q_I = \bar{\nu}^+ \bar{n}_+ + \bar{\nu}^- \bar{n}_- + \bar{\nu}^\times \bar{n}_x + v^1 e_1 + v^2 e_2 + v^+ n_+$$

reads

$$Q = \begin{pmatrix} -\frac{1}{2} (\bar{\nu}^+ + \bar{\nu}^-) & -\frac{1}{2} \bar{\nu}^\times & \frac{1}{2} v^1 \\ -\frac{1}{2} \bar{\nu}^\times & -\frac{1}{2} (\bar{\nu}^+ - \bar{\nu}^-) & \frac{1}{2} v^2 \\ \frac{1}{2} v^1 & \frac{1}{2} v^2 & -v^+ \end{pmatrix}.$$  \hspace{1cm} (2)

The internal parameters of an unknown conic, its position and the orientation in the plane can also be obtained using the matrix (2), [11]. Note that all this can be determined from the GAC vector $Q_I$ by means of the inner product.

For proofs and further details see [10]. To learn more about manipulations with conics such as intersections, see e.g. [2].

Altogether, GAC is constituted as a Clifford algebra $\mathcal{C}l(5,3)$ with an associated bilinear form given by a matrix

$$B = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$  \hspace{1cm} (3)

and an embedded GAC point (1) has the form

$$P = \begin{pmatrix} 0 & 0 & 1 & x & y & \frac{1}{2} (x^2 + y^2) & \frac{1}{2} (x^2 - y^2) & xy \end{pmatrix}^T,$$  \hspace{1cm} (4)

while the IPNS representation of a general conic section reads

$$Q_I = (\bar{\nu}^\times \ \bar{\nu}^- \ \bar{\nu}^+ \ v^1 \ v^2 \ v^+ \ 0 \ 0)^T.$$  \hspace{1cm} (5)

The advantage of this notation is that the $4 \times 4$ matrix in the middle of (3) is the usual matrix for the inner product in the conformal geometric algebra. Consequently, CRA objects in GAC are built of vectors with vanishing first two and last two coordinates and thus they are easier to recognise.
3. Conic Fitting in GAC—Original Algorithm

Having GAC in hand, where conic sections are the intrinsic geometric primitives, we can formulate a fitting algorithm for the conics analogous to the algorithm of fitting circles in CRA.\textsuperscript{1} Namely, for a conic represented by a vector $Q$ of the form (5) and for given points represented by vectors $P_i$ of the form (4), we assume the objective function (cost function) to be given by

$$Q \mapsto \sum_i (P_i \cdot Q)^2,$$  \hspace{1cm} (6)

where we used the inner product between vectors in GAC. Therefore, the used objective function represents the sum of the squared algebraic point-to-conic distances. The best conic fitting the points w.r.t. this function is represented by the $Q$ that minimises this function. The geometrically meaningless minimum $Q = 0$ is not of the interest and thus the authors of [10] consider the natural geometric constraint

$$Q^2 = 1.$$  \hspace{1cm} (7)

Then the objective function is zero if and only if $P_i \cdot Q = 0$ for all $i$, i.e. all points lie on the conic $Q$. Note that the square in (7) is taken w.r.t. the geometric product which is the same as the inner product in this case. Both the objective function and the constraint are defined in a covariant way in terms of GAC. The objective function is obviously invariant under all conformal transformations while the constraint is invariant under rotations and scaling only. As argued in [10], it is not invariant under translations. Hence the solution of the optimisation problem minimising (6) subject to (7) enjoys the same invariant properties by definition.

This formulation of the problem of fitting conics to points in terms of GAC gives not only a geometric insight but also allows an effective computation of the solution as an eigenvector of a $4 \times 4$ matrix which is easy to implement, see Proposition 1 below. We use the matrix notation from now on since it is convenient for the implementation. Using the matrix of bilinear form (3), the objective function (6) reads

$$Q \mapsto \sum_i (P_i BQ)^2 = \sum_i Q^T B P_i P_i^T B Q = Q^T P Q,$$

and thus it is a quadratic form on $\mathbb{R}^{5,3}$ with the matrix

$$P = \sum_i B P_i P_i^T B.$$  \hspace{1cm} (8)

Let us note that matrix $P$ can be equivalently computed as

$$P = B D D^T B^T,$$  \hspace{1cm} (9)

where $D$ is the data matrix whose $i$-th column is the GAC vector $P_i$ representing the $i$-th point from the data set according to (4).

\textsuperscript{1}For the sake of easy referencing, let us denote the original conic fitting algorithm as Algorithm $Q$. 


To formulate the solution of our optimisation problem we need to decompose matrix $P$ into blocks. Note that the points in GAC lie in a six-dimensional subspace and thus the rank of the matrix is six. More precisely, it has the form

$$
P = \begin{pmatrix}
P_0 & P_1 & 0 \\
P_1^T & P_c & 0 \\
0 & 0 & 0
\end{pmatrix},
$$

(10)

where $P_0$ is a $2 \times 2$ matrix, $P_1$ is a $2 \times 4$ matrix and $P_c$ is a $4 \times 4$ matrix. The subscript $c$ denotes that this block corresponds to the CRA part in GAC. Similarly, $B_c$ denotes the middle part of $B$ and it coincides with the matrix of the inner product in CRA, i.e.

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

(11)

Thus, we are ready to formulate the following Proposition, [10].

**Proposition 1.** The solution to the optimisation problem (6), (7) for conic fitting in GAC is given by $Q = (w \quad v \quad 0)^T$, where $v$ is a 4-dimensional eigenvector corresponding to the minimal non-negative eigenvalue of

$$
P_{\text{con}} = B_c(P_c - P_1^TP_0^{-1}P_1)
$$

(12)

and $w$ is a 2-dimensional vector of the form

$$
w = -P_0^{-1}P_1v.
$$

(13)

**Proof.** Note that the inversion in (13) exists up to the case when all points lie on a single line. This singular case must be detected and treated separately. We solve the optimisation problem by the method of Lagrange multipliers. The constraint (7) of the optimisation problem reads $Q^TBQ = 1$ in matrix form. Hence the Lagrange function is

$$
P_\lambda(Q) := Q^TPQ + \lambda(1 - Q^TBQ).
$$

We find its stationary points by differentiation w.r.t. unknown vector $Q$ and we use the block form of $P$ given by (10). Using the block decomposition of $Q$ from the proposition the Lagrange function reads

$$
P_\lambda(Q) = w^TP_0w + 2w^TP_1v + v^TP_cv + \lambda(1 - v^TB_cv),
$$

and its stationary points satisfy the following system of linear equations

$$
0 = \frac{\partial P_\lambda}{\partial w} = 2(P_0w + P_1v),
$$

$$
0 = \frac{\partial P_\lambda}{\partial v} = 2(P_1^T w + P_cv - \lambda B_cv),
$$

$$
0 = \frac{\partial P_\lambda}{\partial \lambda} = 1 - v^TB_cv.
$$
From the first equation we directly get (13) and the substitution of the corresponding term in the second equation yields the equation for generalised eigenvectors

$$(P_c - P_1^T P_0^{-1} P_1)v = \lambda B_c v.$$  

Since $B_c$ squares to one, it can be reduced to the equation for standard eigenvectors by multiplying it by $B_c$ from left. Hence $v$-part of each stationary point of our Lagrange function is an eigenvalue of operator (12). Moreover, it must be normalised according to the third equation of the system above

$$v^T B_c v = 1.$$  \hspace{1cm} (14)

Note that this is not always possible since $B_c$ is not positive definite.

The only thing left is to find the minimum among the stationary points. Therefore, we compute the values of the objective function (6). Let $v_\lambda$ denote the eigenvector corresponding to eigenvalue $\lambda$ and let $Q_\lambda$ be the corresponding conic $Q_\lambda = (-P_0^{-1} P_1 v_\lambda ~ v_\lambda ~ 0)$. By means of the block decomposition of $P$ and the definition of $v_\lambda$ we compute

$$P(Q_\lambda) = Q_\lambda^T P Q_\lambda = v_\lambda^T (P_c - P_1^T P_0^{-1} P_1) v_\lambda \lambda v_\lambda^T B_c v_\lambda = \lambda v_\lambda^T B_c v_\lambda = \lambda > 0,$$

assuming that $v_\lambda$ is normalised as in (14). The eigenvectors corresponding to negative eigenvalues cannot be normalised in the sense of (14) and thus they are not stationary points. So the minimum of the objective function is attained if $v$ is the eigenvector corresponding to the least but non-negative eigenvalue of $P_{con}$. \hspace{1cm} □

Let us also remark that Algorithm Q may be further optimised by modification, namely reduction, of particular vectors and matrices. For example, we can see that the first two rows of the data matrix $D$ are by definition zero, resulting then in a block of zeroes in the matrix $P$, while this block is no further used in the consequent computation. The zero block in the matrix $P$ (though redundant) is not that critical itself, but the computation of the matrix $\tilde{P}$ according to (9) uses multiplication of matrix $D$ with its transpose and thus constitutes both unnecessary element multiplications and storage of useless zero rows of matrix $D$. Therefore, by removing first two rows from the vector (4) we can define the $i$-th reduced GAC point vector $\tilde{P}_i$ of the form

$$\tilde{P}_i = (1, x_i, y_i, \frac{1}{2}(x_i^2 + y_i^2), \frac{1}{2}(x_i^2 - y_i^2), x_i y_i)^T$$

and create the corresponding reduced data matrix $\tilde{D}$ of size $6 \times N$, where the $i$-th column is $\tilde{P}_i$. Additionally, by removing the first two columns and the last two rows from the matrix $B$ of the form (3), we also define a reduced matrix for GAC inner product

$$\tilde{B} = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & -1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}.$$
As a result, the matrix $P$ computed using reduced matrices and vectors would not contain the zero blocks as in (10) and would take the form of a reduced matrix

$$\hat{P} = \begin{pmatrix} P_0 & P_1 \\ P_1^T & P_c \end{pmatrix}$$

of size $6 \times 6$. Let us also note that we can also analogously omit zeroes in GAC conic vector of the form (5), and thus create a reduced conic vector

$$\hat{Q} = (\bar{v}^\times \bar{v} - \bar{v}^+ v^1 v^2 v^+)^T,$$

and then compute the objective function (6) equivalently as

$$\hat{Q} \mapsto \hat{Q}^T \hat{P} \hat{Q}.$$

4. Conic Fitting in GAC—Additional Conditions

One of the main features of the above-presented algorithm 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 tilt of its axes or its centre point position in the plane are not known beforehand, being just a result of the optimisation process.

While this 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 coordinate axes
2. conic having its centre point at the coordinate system origin
3. conic satisfying both 1. and 2.

We note that these additional conditions may also be included in classical conic fitting algorithms by setting some entries of the appropriate optimisation quadratic form to zero, but we stick to the implementation in terms of geometric algebra. We add that we have found no reference to classical conic fitting algorithms with additional conditions, either. We can only assume that this is because motivations, such as switched systems, for including the additional conditions into these algorithms, are rather recent topics.

Before further discussion on the above mentioned three types of conic fitting, let us briefly recall that a conic $Q_I$ in the IPNS representation of the form (5) can be as well described as an implicit two variable function

$$Q(x, y) = Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0$$

(15)

with the corresponding matrix

$$Q = \begin{pmatrix} A & B/2 & D/2 \\ B/2 & C & E/2 \\ D/2 & E/2 & F \end{pmatrix}.$$  (16)

Let us just note that the coefficients of (15) are intentionally written in calligraphic style, in order to avoid confusion in notation.
By comparing the matrices (16) and (2) we can easily describe the relationships between the coefficients of $Q(x, y)$ and the elements of vector $Q_I$, for example

$$F = -v^+.$$ 

Such a direct connection between the IPNS representation and implicit function of a conic will later allow us to easily express the geometric conditions in terms of GAC. As we will see, thanks to this connection and the matrix–vector formulation of the original problem, our conic fitting algorithms with additional conditions can be described analogously to the original one and solved in a way very similar to Proposition 1.

4.1. Axes-Aligned Conic

**Theorem 1.** Conic $Q$ represented in the form of vector (5) is axes-aligned if and only if

$$\bar{v}^\times = 0.$$ 

**Proof.** The necessary and sufficient condition for axial alignment of a conic $Q(x, y)$ in the form (15) is the zero coefficient $B$, [11], i.e.

$$Q(x, y) \text{ is axes-aligned } \iff B = 0.$$ 

By comparing the entries of matrices (16) and (2) we get

$$\bar{v}^\times = -B,$$

therefore

$$Q \text{ is axes-aligned } \iff \bar{v}^\times = 0.$$ 

\[\square\]

For more details on the properties of conics, see [11] or [14].

With the axial alignment expressed in terms of GAC, we can substitute the acquired condition into the vector (5), yielding a vector IPNS representation of an axes-aligned conic $Q_{al}$ of the form

$$Q_{al} = (0 \, \bar{v}^- \, v^1 \, v^2 \, v^+ \, 0 \, 0)^T. \tag{17}$$

Consequently, we may formulate an optimisation problem almost identical to the original problem of minimising objective function (6) subject to (7), however, instead of searching for a general conic $Q$ of the form (5) we will assume the optimal conic to be axes-aligned. Hence, for an axes-aligned conic $Q_{al}$ of the form (17) and for given points represented by vectors $P_i$ of the form (4), the objective function reads

$$Q_{al} \mapsto \sum_i (P_i \cdot Q_{al})^2, \tag{18}$$

where $\cdot$ represents the inner product between GAC vectors. As well as in the original fitting problem, we consider the constraint

$$Q_{al}^2 = 1. \tag{19}$$
Let us stress that the square in (19) is taken w.r.t. the geometric product which is the same as the inner product in this case.

To solve the upcoming problem we will use again the method of Lagrange multipliers and the block decomposition of the matrix $P$ of the form (8). However, because of the slight change of the resulting conic vector, it is necessary to partition matrix $P$ in a manner different to (10). Firstly, let us denote the (generally) non-zero entries of symmetric matrix $P$ as $p_{ij}$, i.e.

$$
P = \begin{pmatrix}
p_{11} & p_{21} & p_{31} & p_{41} & p_{51} & p_{61} & 0 & 0 \\
p_{21} & p_{22} & p_{32} & p_{42} & p_{52} & p_{62} & 0 & 0 \\
p_{31} & p_{32} & p_{33} & p_{43} & p_{53} & p_{63} & 0 & 0 \\
p_{41} & p_{42} & p_{43} & p_{44} & p_{54} & p_{64} & 0 & 0 \\
p_{51} & p_{52} & p_{53} & p_{54} & p_{55} & p_{65} & 0 & 0 \\
p_{61} & p_{62} & p_{63} & p_{64} & p_{65} & p_{66} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 
\end{pmatrix}.
$$

The objective function (18) in the matrix notation reads

$$Q^{al} \mapsto Q^{alT} P Q^{al}$$

and the matrix $P$ is thus a matrix of a quadratic form w.r.t. the vector $Q^{al}$. Since the first entry in the vector $Q^{al}$ is zero, we are not interested in the first row or the first column of the matrix $P$. Therefore, the desired decomposition has the form

$$P = \begin{pmatrix}
p_{11} & p_{21} & p_{31} & p_{41} & p_{51} & p_{61} & 0 & 0 \\
p_{21} & P^{al}_0 & 0 & 0 \\
p_{31} & P^{al}_0 & P^{al}_1 \\
p_{41} & P^{al}_0 & P^{al}_1 & P_c \\
p_{51} & P^{al}_0 & P^{al}_1 & P_c & 0 \\
p_{61} & P^{al}_0 & P^{al}_1 & P_c & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 
\end{pmatrix},
$$

where $P^{al}_0$ is a real number, $P^{al}_1$ is a 4-dimensional vector and $P_c$ is a $4 \times 4$ matrix corresponding to the CRA part in GAC. Let us note that the matrices $P^{al}_0$, $P^{al}_1$ and $P^{al}_1T$ are created as subparts of matrices $P_0$, $P_1$ and $P_1^T$ from partition (10) by omitting the first row and the first column in the matrix $P$. Submatrix $P_c$ remains unaffected by this change, so we keep its notation.

**Proposition 2.** The solution to the optimisation problem (18), (19) for conic fitting in GAC is given by $Q^{al} = (0 \ w^{al} \ v \ 0 \ 0)^T$, where $v$ is a 4-dimensional eigenvector corresponding to the minimal non-negative eigenvalue of

$$P^{al}_{con} = B_c(P_c - P^{al}_1 T P^{al}_0 P^{-1} P^{al}_1)$$
and \( w^{al} \equiv \bar{v}^- \) is a real number acquired as
\[
w^{al} = -P_{0}^{al-1}P_{1}^{al}v.
\] (22)

**Proof.** The inversion in (22) exists up to the case when all points lie on a double-line \( x^2 - y^2 = 0 \), so this case must be detected and treated separately. Let us also note that relabelling of the variable \( \bar{v}^- \) to \( w^{al} \) is not necessary at all and has been made only to emphasise the correspondence of this variable to the matrices \( P_{0}^{al}, P_{1}^{al} \) and \( P_{1}^{al^T} \), respectively.

As already said, this approach to the solution is basically the same as in the proof of Proposition 1. Using the method of the Lagrange multipliers, we obtain the Lagrange function in the form
\[
P_{\lambda}(Q^{al}) := Q^{al^T}PQ^{al} + \lambda(1 - Q^{al^T}BQ^{al}).
\]
and after applying the block decomposition (21), it can be rewritten as
\[
P_{\lambda}(Q^{al}) = w^{al^2}P_{0}^{al} + 2w^{al}P_{1}^{al^T}v + v^{T}P_{c}v + \lambda(1 - v^{T}B_{c}v).
\]
We find its stationary points by differentiation w.r.t. the unknown vector \( Q^{al} \) and by solving the following system of linear equations
\[
0 = \frac{\partial P_{\lambda}}{\partial w^{al}} = 2(w^{al}P_{0}^{al} + P_{1}^{al}v),
\]
\[
0 = \frac{\partial P_{\lambda}}{\partial v} = 2(w^{al}P_{1}^{al^T} + P_{c}v - \lambda B_{c}v),
\]
\[
0 = \frac{\partial P_{\lambda}}{\partial \lambda} = 1 - v^{T}B_{c}v.
\]
The rest of the proof proceeds analogously to the proof of Proposition 1. \( \square \)

The algorithm for fitting an axes-aligned conic \( Q^{al} \), further denoted as Algorithm QAL, has the form very similar to Algorithm Q, containing only a few adjustments of the involved matrices. For the sake of computational optimisation we follow the example of Algorithm Q and offer the use of reduced forms of vectors and matrices within the algorithm. Thus, by removing the first two zeros and the mixed term \( xy \) from the vector (4) we define the \( i \)-th reduced GAC point vector \( \hat{P}_{i}^{al} \) of the form
\[
\hat{P}_{i}^{al} = (1 \ x_{i} \ y_{i} \ \frac{1}{2}(x_{i}^{2} + y_{i}^{2}) \ \frac{1}{2}(x_{i}^{2} - y_{i}^{2}))^{T}
\]
and create the corresponding reduced data matrix \( \hat{D}^{al} \) of the size 5 \( \times \) \( N \), where the \( i \)-th column is \( \hat{P}_{i}^{al} \). Additionally, by removing the rows 1, 7 and 8 and columns 1,2 and 8 from the matrix \( B \) of the form (3), we also define a reduced matrix for GAC inner product
\[
\hat{B}^{al} = \begin{pmatrix}
0 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & -1 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
-1 & 0 & 0 & 0 & 0
\end{pmatrix}
\].
while the submatrix $B_c$ remains the same as in (11). Computation of the matrix $P$ then yields a $5 \times 5$ reduced matrix of the form

$$\hat{P}^{al} = \begin{pmatrix} P_0^{al} & P_1^{al} \\ P_0^{alT} & P_c \end{pmatrix}.$$ 

Furthermore, by omitting zeroes in GAC vector of an axes-aligned conic of the form (17) we define a reduced conic vector

$$\hat{Q}^{al} = (\bar{v}^- \bar{v}^+ v^1 v^2 v^+)^T,$$

and formulate an objective function (18) equivalently as

$$\hat{Q}^{al} \mapsto \hat{Q}^{alT} \hat{P}^{al} \hat{Q}^{al}.$$ 

### 4.2. Origin-Centred Conic

**Theorem 2.** A central conic $Q$ represented by a vector (5) has its centre at the origin of the coordinate system if and only if $(v^1 = 0) \land (v^2 = 0)$.

**Proof.** Firstly, by comparing the entries of matrices (2) and (16) we conclude that

$$v^1 = D,$$

$$v^2 = E,$$

so the proof can be equivalently written in terms of the function $Q(x, y)$ coefficients in the form (15). We have to show that the central conic $Q(x, y)$ has its centre at the origin of the coordinate system if and only if $(D = 0) \land (E = 0)$.

The coordinates $(x_c, y_c)$ of the central conic’s centre-point may be computed using the coefficients (15) as follows

$$x_c = \frac{B E - 2 C D}{4 A C - B^2}, \quad (23)$$

$$y_c = \frac{B D - 2 A E}{4 A C - B^2}. \quad (24)$$

Let us use note that the denominator $4 A C - B^2 \neq 0$ for all central conics. Using the above formulae we can prove the theorem in two steps:

I. $(D = 0) \land (E = 0) \implies Q(x, y)$ has the centre at the origin:

After direct substitution into formulae (23), (24) we obtain immediately that

$$(x_c, y_c) = (0, 0).$$

II. $Q(x, y)$ has the centre at the origin $\implies (D = 0) \land (E = 0)$:

If a central conic $Q(x, y)$ has the centre at the origin, the formulae (23), (24) must be equal to zero, thus after multiplying both equations by their non-zero denominator, we get

$$B E - 2 C D = 0, \quad (25)$$
\[ BD - 2AE = 0. \]  

(26)

To prove that the coefficients \( D \) and \( E \) must be zero for a central conic, we will inspect behaviour of \( D \) and \( E \) depending on whether the remaining coefficients \( A, B, C \) are or are not equal to zero, respectively. Therefore, 8 possibilities summarised by the following table may occur (‘0’ means equal to zero and ‘1’ means non-zero):

|   | 1. | 2. | 3. | 4. | 5. | 6. | 7. | 8. |
|---|----|----|----|----|----|----|----|----|
| \( A \) | 0  | 1  | 0  | 0  | 1  | 0  | 1  | 1  |
| \( B \) | 0  | 0  | 1  | 0  | 1  | 1  | 0  | 1  |
| \( C \) | 0  | 0  | 0  | 1  | 0  | 1  | 1  | 1  |

Possibility 1. violates the definition of a conic itself, because at least one of the coefficients \( A, B, C \) must be non-zero. Possibilities 2. and 4. cannot occur for central conics, since the denominator \( 4AC - B^2 \) must be non-zero. Possibility 3. yields
\[
(BE = 0) \land (B \neq 0) \implies E = 0,
(BD = 0) \land (B \neq 0) \implies D = 0,
\]
so
\[
(D = 0) \land (E = 0).
\]

After inspecting 5., we get
\[
(BE = 0) \land (B \neq 0) \implies E = 0,
(BD - 2AE = 0) \land (B \neq 0) \land (E = 0) \implies D = 0,
\]
so
\[
(D = 0) \land (E = 0).
\]

The approach in 6. and 7. would be similar. Finally, in 8., we multiply the Eq. (25) by a non-zero coefficient \( B \) and the Eq. (26) by a non-zero \( 2C \) to get the system
\[
B^2E - 2BCD = 0,
2BCD - 4AC\varepsilon = 0.
\]

By summing up the resulting equations we get
\[
B^2E - 4AC\varepsilon = 0,
\]
or
\[
\varepsilon(B^2 - 4AC) = 0.
\]
Since \( B^2 - 4AC \neq 0 \) for central conics, it follows that \( \mathcal{E} = 0 \).
Because \( \mathcal{E} = 0 \) and \( B \neq 0 \), we conclude from the Eq. (26) that \( \mathcal{D} = 0 \) as well, so
\[
(\mathcal{D} = 0) \land (\mathcal{E} = 0).
\]

\( \square \)

Description of the origin-centred conic fitting will now proceed analogously to the problem with axes-aligned conic fitting. After substituting the acquired condition into the vector (5), we obtain the IPNS representation of an origin-centred conic \( Q^0 \) in the vector form
\[
Q^0 = (\bar{v} \times \bar{v} - \bar{v} + 0 0 v^+ 0 0)^T.
\]
Consequently, for an origin-centred conic \( Q^0 \) of the form (27) and for given points represented by vectors \( P_i \) of the form (4), we get the objective function
\[
Q^0 \mapsto \sum_i (P_i \cdot Q^0)^2,
\]
where \( \cdot \) represents the inner product between GAC vectors. Similarly to the previous problem, we minimise the objective function (28) subject to normalisation constraint
\[
Q^{02} = 1,
\]
where the square in (29) is w.r.t. the geometric product which is the same as the inner product in this case.

Because the objective function (28) in the matrix notation reads
\[
Q^0 \mapsto Q^0^T P Q^0,
\]
the matrix \( P \) in the form (20) allows a partition according to the vector (34) of the origin-centred conic \( Q^0 \). Since both \( v^1 \) and \( v^2 \) are zero, it follows that the rows 4 and 5 together with the columns 4 and 5 of the matrix \( P \) are of no importance to our problem. Therefore, we consider the following block decomposition:
\[
P = \begin{pmatrix}
P_0 & \sim & p_{41} & p_{51} & \sim & 0 & 0 \\
\sim & \sim & p_{42} & p_{52} & \sim & 0 & 0 \\
\cdot & \cdot & p_{43} & p_{53} & \cdot & 0 & 0 \\
p_{41} & p_{42} & p_{43} & p_{44} & p_{45} & p_{54} & p_{55} & p_{64} & p_{65} & 0 & 0 \\
p_{51} & p_{52} & p_{53} & p_{54} & p_{55} & p_{64} & p_{65} & 0 & 0 & 0 & 0 \\
\cdot & \cdot & p_{64} & p_{65} & \cdot & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix},
\]
where \( P_0 \) is a \( 2 \times 2 \) matrix. The cells inscribed with \( \sim \) indicate the elements of \( 2 \times 2 \) matrix \( P_1^0 \) consisting of the two columns from the matrix \( P \) that are
mutually isolated, i.e.

$$P^0_1 = \begin{pmatrix} p_{31} & p_{61} \\ p_{32} & p_{62} \end{pmatrix}.$$  

Due to the symmetry of matrix $P$, the cells with $\wr$ stands for the matrix $P^0_1^T$. Similarly to $P^0_1$, the isolated elements designated with $\bullet$ comprises $2 \times 2$ matrix $P^0_c$ of the form

$$P^0_c = \begin{pmatrix} p_{33} & p_{63} \\ p_{63} & p_{66} \end{pmatrix}.$$  

Let us remark that the matrices $P^0_1$, $P^0_1^T$ and $P^0_c$ are created from the matrices $P_1$, $P_1^T$ and $P_c$, respectively, used in the partition (10) by omitting the corresponding rows and/or columns in the matrix $P$. Submatrix $P_0$ remains the same as in the original decomposition.

Next, let us recall that the zeroness of variables $v^1$ and $v^2$ in the vector (27) of conic $Q^0$ also affects the final form of the normalisation constraint (29). By default, the matrix formulation of this constraint is

$$Q^0^T B Q^0 = 1,$$

but thanks to sparseness of the matrix $B$, it can be simplified. After defining a vector

$$v^0 = (\bar{v}^+ \ v^+)^T,$$

and a $2 \times 2$ matrix

$$B^0_c = \begin{pmatrix} 0 & -1 \\ -1 & 0 \end{pmatrix},$$  \hspace{1cm} (31)

the normalisation constraint can be reformulated as

$$v^0^T B^0_c v^0 = 1.$$  \hspace{1cm} (32)

Analogously to the matrices $P^0_1$, $P^0_1^T$ and $P^0_c$, the vector $v^0$ and the matrix $B^0_c$ were created from the vector $v$ and matrix $B_c$ by omitting those rows and/or columns corresponding to the zero variables $v^1$ and $v^2$.

**Proposition 3.** The solution to the optimisation problem (28), (29) for conic fitting in GAC is given by $Q^0 = (w \ \bar{v}^+ \ 0 \ 0 \ v^+ \ 0 \ 0)^T$, where $(\bar{v}^+ \ v^+)^T = v^0$ constitutes a 2-dimensional eigenvector corresponding to the minimal non-negative eigenvalue of

$$P^0_{\text{con}} = B^0_c (P^0_c - P^0_1 P^{-1}_0 P^0_1^T)$$  \hspace{1cm} (33)

and $w = (\bar{v}^x \ \bar{v})^T$ is a 2-dimensional vector computed as

$$w = -P^{-1}_0 P^0_1 v^0.$$  \hspace{1cm} (34)

**Proof.** As in Proposition 1, the inversion in (34) exists up to the case when all points lie on a single line.
Again, using the method of Lagrange multipliers, we obtain the Lagrange function in the form
\[
P_\lambda(Q^0) := Q^0^T P Q^0 + \lambda(1 - Q^0^T B Q^0).
\]
and after applying the block decomposition (30), it can be rewritten as
\[
P_\lambda(Q^0) = w^T P_0 w + 2w^T P_1^0 v^0 + v^0^T P_1^0 v^0 + \lambda(1 - v^0^T B^0_c v^0).
\]
We find its stationary points by differentiation w.r.t. the unknown vector \(Q^0\) and solving the corresponding system of linear equations. Rest of the proof is analogous to proof of Proposition 1. □

Not only that Proposition 3 formulates the solution to our fitting problem in a quite compact way, but it can also be used as a foundation for even more direct way of computation of the origin-centred conic \(Q^0\). With the use of matrix \(P^0_{con}\) of the form (33) and its elements denoted as
\[
P^0_{con} = \begin{pmatrix} a & r \\ s & a \end{pmatrix}, \tag{35}
\]
we can indeed derive another method of computation summarised in the following proposition. Moreover, although the proposition is based on the eigenvalue problem formulation, it reaches the solution without computing a single eigenvalue.

**Proposition 4.** The solution to the optimisation problem (28), (29) for conic fitting in GAC is given by \(Q^0 = (w \quad \bar{v}^+ \quad 0 \quad 0 \quad v^+ \quad 0)^T\), where \((\bar{v}^+ \quad v^+)^T = v^0\) is a vector obtained as
\[
v^0 = \begin{pmatrix} \bar{v}^+ \\ v^+ \end{pmatrix} = \begin{pmatrix} \pm \sqrt{1 + 4s} \\ \mp \frac{4 + 4s}{\pm 2\sqrt{1 + 4s}} \end{pmatrix}
\]
and \(w = (\bar{v}^x \quad \bar{v}^-)^T\) is a vector of the form
\[
w = -P_0^{-1} P_1^0 v^0. \tag{36}
\]

**Proof.** The eigenvalue problem derived in Proposition 3 reads
\[
P^0_{con} v^0 = \lambda v^0,
\]
or in the equivalent form
\[
(P^0_{con} - \lambda E) v^0 = 0,
\]
which can be expressed as a system of two equations:
\[
(a - \lambda)\bar{v}^+ + rv^+ = 0, \tag{37}
\]
\[
s\bar{v}^+ + (a - \lambda)v^+ = 0. \tag{38}
\]
After multiplying the Eq. (37) by \(-v^+\) and the Eq. (38) by \(\bar{v}^+\) we obtain
\[
-(a - \lambda)\bar{v}^+ v^+ - rv^+^2 = 0,
\]
\[
sv^+^2 + (a - \lambda)\bar{v}^+ v^+ = 0,
\]
and summing the two equations up yields
\[ s\bar{v}^+ + 2rv^+ = 0. \] (39)

Since \( v^0 = (\bar{v}^+ \quad v^+)^T \) must comply with the normalisation constraint (32), which can be rewritten simply as
\[ -2\bar{v}^+ v^+ = 1, \]
either of the variables \( \bar{v}^+ \) and \( v^+ \) can be directly expressed in terms of the other one, for example
\[ v^+ = -\frac{1}{2\bar{v}^+}. \] (40)

Substitution of (40) into (39) gives
\[ s\bar{v}^+ + 2 - \frac{r}{4\bar{v}^+} = 0, \]
and after multiplying by \( 4\bar{v}^+ \) we have
\[ 4s\bar{v}^+ + 4 - r = 0, \]
implying that
\[ \bar{v}^+ = \pm \frac{4}{\sqrt{\frac{r}{4s}}}, \]
which together with (40) yields
\[ v^+ = \mp \frac{1}{2\sqrt{\frac{r}{4s}}}. \]

The computation of the vector \( w \) given by (36) is the same as in the Proposition 3, where it was derived and proven valid. \( \square \)

Let us remark that the notation of entries of matrix \( P_{con}^0 \) in (35) is intentional—since we denote the entries of matrix \( P_{con} \) of the form (12) in the following way:
\[
P_{con} = \begin{pmatrix}
a & k & l & r \\
m & t & b & -k \\
n & b & u & -l \\
s & -m & -n & a
\end{pmatrix},
\] (41)
it is obvious that the matrix \( P_{con}^0 \) consists of four corner elements of the matrix \( P_{con} \). This is a consequence of the block decomposition (30) of matrix \( P \) where the two rows and columns corresponding to the zero variables \( v^1 \) and \( v^2 \) are omitted.

Apart from two equivalent solutions formulated in Propositions 3 and 4 we can reach the same solution by yet another way. In contrast with mentioned Propositions, in which we just used subparts of matrix \( P \) to reformulate the original fitting problem, we can use the original fitting approach and extend the point dataset so that we necessarily obtain an origin-centred conic as a result.
Proposition 5. The solution to the optimisation problem (28), (29) for conic fitting in GAC is given by Proposition 1 provided that the dataset consisting of points \( p_i = (x_i, y_i) \), \( i = 1, \ldots, N \), is extended by points \( p_i^{-} = (-x_i, -y_i) \), \( i = 1, \ldots, N \), which constitute centrally symmetric images of points \( p_i \) w.r.t. the origin of the coordinate system.

Proof. Firstly, let us recall that every point \( p_i = (x_i, y_i) \) is in GAC represented by the vector

\[
P_i = \begin{pmatrix} 0 & 0 & 1 & x_i & y_i & \frac{1}{2}(x_i^2 + y_i^2) & \frac{1}{2}(x_i^2 - y_i^2) & x_i y_i \end{pmatrix}^T
\]

therefore, every point \( p_i^{-} = (-x_i, -y_i) \) is represented by a vector \( P_i^{-} \) of the form

\[
P_i^{-} = \begin{pmatrix} 0 & 0 & 1 & -x_i & -y_i & \frac{1}{2}(x_i^2 + y_i^2) & \frac{1}{2}(x_i^2 - y_i^2) & x_i y_i \end{pmatrix}^T.
\]

Consequently, the corresponding matrix \( P_{con} \) is of the form

\[
P_{con}^{0, sym} = \begin{pmatrix} a^* & 0 & 0 & r^* \\ 0 & t^* & b^* & 0 \\ 0 & b^* & u^* & 0 \\ s^* & 0 & 0 & a^* \end{pmatrix},
\]

where superscript asterisks are used to stress that the elements of matrix \( P_{con}^{0, sym} \) are generally different than those of the matrix \( P_{con} \) of the form (41) constructed from the points \( p_i \) only. Fortunately, after further inspection, one can conclude that some elements of the matrix \( P_{con}^{0, sym} \) can be expressed in terms of the entries of matrix \( P_{con} \). In particular, we get

\[
P_{con}^{0, sym} = \begin{pmatrix} a^* & 0 & 0 & r^* \\ 0 & t^* & b^* & 0 \\ 0 & b^* & u^* & 0 \\ s^* & 0 & 0 & a^* \end{pmatrix} = \begin{pmatrix} 2a & 0 & 0 & 2r \\ 0 & t^* & b^* & 0 \\ 0 & b^* & u^* & 0 \\ 2s & 0 & 0 & 2a \end{pmatrix}.
\]

Due to the special form of the matrix \( P_{con}^{0, sym} \), the set \( V^* \) of its eigenvectors

\[
v_1^* = \begin{pmatrix} v_{11}^* \\ v_{12}^* \\ v_{13}^* \\ v_{14}^* \end{pmatrix}, \ldots, v_4^* = \begin{pmatrix} v_{41}^* \\ v_{42}^* \\ v_{43}^* \\ v_{44}^* \end{pmatrix},
\]

can be described as

\[
V^* = \begin{pmatrix} v_1^* & v_2^* & v_3^* & v_4^* \end{pmatrix} = \begin{pmatrix} v_{11}^* & 0 & 0 & v_{41}^* \\ 0 & v_{22}^* & v_{32}^* & 0 \\ 0 & v_{23}^* & v_{33}^* & 0 \\ v_{14}^* & 0 & 0 & v_{44}^* \end{pmatrix},
\]

with \( \lambda_1^*, \ldots, \lambda_4^* \) as their corresponding eigenvalues. From the geometric nature of the problem, the dataset centrally symmetric w.r.t. the coordinate origin must necessarily be fitted with origin-centred conic, therefore, neither of the eigenvectors \( v_2^* \) and \( v_3^* \) can be the solution, since they do not correspond to an origin-centred conic (also, neither of the eigenvalues \( \lambda_2^* \) and \( \lambda_3^* \) can
represent the optimal solution). Regarding the eigenvalues \( \lambda_1^* \) and \( \lambda_4^* \), they are computed as

\[
\begin{pmatrix}
\lambda_1^* \\
\lambda_4^*
\end{pmatrix} = \begin{pmatrix}
2(a + \sqrt{rs}) \\
2(a - \sqrt{rs})
\end{pmatrix}
\]

and their corresponding eigenvectors as

\[
\begin{pmatrix} v_1^* & v_4^* \end{pmatrix} = \begin{pmatrix}
\sqrt{\frac{r}{s}} & -\sqrt{\frac{r}{s}} \\
0 & 0 \\
0 & 0 \\
1 & 1
\end{pmatrix},
\]

while the eigenvalues \( \lambda_1^0 \) and \( \lambda_2^0 \) of the matrix \( P_{con}^0 \) can be expressed as

\[
\begin{pmatrix}
\lambda_1^0 \\
\lambda_2^0
\end{pmatrix} = \begin{pmatrix}
a + \sqrt{rs} \\
a - \sqrt{rs}
\end{pmatrix}
\]

with the corresponding eigenvectors \( v_1^0 \) and \( v_2^0 \) of the form

\[
\begin{pmatrix} v_1^0 & v_2^0 \end{pmatrix} = \begin{pmatrix}
\sqrt{\frac{r}{s}} & -\sqrt{\frac{r}{s}} \\
1 & 1
\end{pmatrix}.
\]

We can see that

\[
\begin{pmatrix}
\lambda_1^* \\
\lambda_4^*
\end{pmatrix} = 2 \begin{pmatrix}
\lambda_1^0 \\
\lambda_2^0
\end{pmatrix}
\]

and that

\[
\begin{pmatrix} v_1^* & v_4^* \end{pmatrix} \approx \begin{pmatrix} v_1^0 & v_2^0 \end{pmatrix},
\]

where by \( \approx \) we mean that after omitting zero elements in the eigenvectors \( v_1^* \) and \( v_4^* \), the eigenvectors of both matrices are the same (note that in the case of eigenvectors \( v_1^0 \) and \( v_2^0 \), two zeroes omitted in the eigenvectors \( v_1^* \) and \( v_4^* \) would eventually be added anyway, in order to form a complete conic vector). Consequently, the optimal eigenvalue of the matrix \( P_{con}^{0, sym} \) corresponds to the same solution as the one obtained by using the matrix \( P_{con}^0 \) in Proposition 3, which is also equivalent to the solution acquired in Proposition 4. \( \Box \)

Despite the tempting simplicity of this approach, we must keep in mind that the solution in Proposition 5 is reached by doubling the number of points in the dataset. Such an enlargement of the input data necessarily results into the increase of computational demands, which is generally preferred to be avoided, especially when having a large number of points at the beginning already.

Since we have formulated three ways of fitting an origin-centred conic \( Q_0 \), we denote three corresponding algorithms as:

- **Q0**—via eigenvalues and eigenvectors from Proposition 3
- **Q0-dir**—‘direct’ computation using Proposition 4 without computing eigenvalues
- **Q0-sym**—central symmetrisation of the dataset according to Proposition 5
As in the case of Algorithm \textbf{QAL}, we can use in Algorithms \textbf{Q0} and \textbf{Q0-dir} the reduced forms of vectors and matrices. Therefore, we define the $i$-th reduced GAC point
\[
\hat{P}_i^0 = \begin{pmatrix} 1 & \frac{1}{2}(x_i^2 + y_i^2) & \frac{1}{2}(x_i^2 - y_i^2) \\ x_i & y_i \end{pmatrix}^T
\]
and create the corresponding reduced data matrix $\hat{D}^0$ of the size $4 \times N$, where the $i$-th column is $\hat{P}_i^0$. Additionally, by removing the rows 4, 5, 7 and 8 and columns 1, 2, 4 and 5 from the matrix $B$ of the form (3), we also define a reduced matrix for GAC inner product
\[
\hat{B}^0 = \begin{pmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{pmatrix}.
\]
A reduction of the matrix $B_c$ used in Algorithms \textbf{Q0} and \textbf{Q0-dir} is the matrix $B_c^0$ of the form (31). Computation of the matrix $P$ then yields a $4 \times 4$ reduced matrix of the form
\[
\hat{P}^0 = \begin{pmatrix} P_0^0 \\ P_1^0 \\ P_2^0 \\ P_c^0 \end{pmatrix}.
\]
Furthermore, after defining the reduced GAC vector of the origin-centred conic
\[
\hat{Q}^0 = \begin{pmatrix} \bar{v}^x \\ \bar{v}^- \\ \bar{v}^+ \\ v^+ \end{pmatrix}^T,
\]
the objective function (28) can be equivalently computed as
\[
\hat{Q}^0 \mapsto \hat{Q}^0 \hat{P}^0 \hat{Q}^0.
\]
Due to Proposition 4 we can reuse some steps of Algorithm \textbf{Q0} in Algorithm \textbf{Q0-dir}, while the other steps of the algorithm will be altered.

4.3. Axes-Aligned Origin-Centred Conic

\textbf{Theorem 3.} A central conic $Q$ represented in the form of a vector (5) has a centre at the origin of coordinate system and is axes-aligned if and only if
\[
(\bar{v}^x = 0) \land (v^1 = 0) \land (v^2 = 0).
\]

\textit{Proof.} This Theorem is a direct consequence of Theorems 1 and 2. \hfill \Box

It is evident that to describe the fitting of an axes-aligned origin-centred conic it is efficient to merge the approaches from two previous methods. After substituting the geometric conditions into the vector (5), we obtain the IPNS representation of an axes-aligned origin-centred conic $Q^{al,0}$
\[
Q^{al,0} = \begin{pmatrix} 0 & \bar{v}^- & \bar{v}^+ & 0 & 0 & v^+ \end{pmatrix}^T.
\]
Hence, for an axes-aligned origin-centred conic $Q^{al,0}$ of the form (42) and for given points represented by vectors $P_i$ of the form (4), we get the objective function
\[
Q^{al,0} \mapsto \sum_i (P_i \cdot Q^{al,0})^2
\]
with \( \cdot \) representing the inner product between GAC vectors. The normalisation constraint reads

\[
Q^{al,0^2} = 1, \tag{44}
\]

where the square in (44) is w.r.t. the geometric product which is the same as the inner product in this case.

Since the elements \( \bar{v}^\times, v^1 \) and \( v^2 \) in the vector \( Q^{al,0} \) of an axes-aligned origin-centred conic are all zero, in the partition of the matrix \( P \) we omit the rows and columns 1, 4 and 5. Therefore, the resulting block decomposition of the matrix \( P \) is of the form

\[
P = \begin{pmatrix}
\begin{array}{cccccc}
p_{11} & p_{21} & p_{31} & p_{41} & p_{51} & p_{61}
p_{21} & P_{al}^0 & \sim & p_{42} & p_{52} & \sim & 0 & 0 
p_{34} & \cdot & p_{43} & p_{53} & \cdot & 0 & 0 
p_{41} & p_{42} & p_{43} & p_{44} & p_{54} & p_{64} & 0 & 0 
p_{51} & p_{52} & p_{53} & p_{54} & p_{55} & p_{65} & 0 & 0 
p_{61} & \cdot & p_{64} & p_{65} & \cdot & 0 & 0 
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 
\end{array}
\end{pmatrix}, \tag{45}
\]

where \( P_{al}^0 \) is the same real number as in the decomposition (21), the cells denoted \( \cdot \) constitutes \( 2 \times 2 \) matrix \( P_{c}^0 \) from decomposition (30) and the cells inscribed with \( \sim \) indicate the elements of vector \( P_{1}^{al,0} \), i.e.

\[
P_{1}^{al,0} = \begin{pmatrix} p_{32} 
\end{pmatrix}.
\]

Symmetrically, cells with \( \approx \) stand for the matrix \( P_{1}^{al,0^T} \).

The default matrix form of the normalisation constraint (44) is

\[
Q^{al,o^T}BQ^{al,o} = 1
\]

and due to the zero variables in vector (42) of the conic it can be rewritten as

\[
v^0^T B_{c}^0 v^0 = 1, \tag{46}
\]

which coincides with the normalisation constraint (29) used in the origin-centred conic fitting problem.

**Proposition 6.** The solution to the optimisation problem (43), (44) for conic fitting in GAC is given by \( Q^{al,0} = (0 \ w^{al} \ \bar{v}^+ \ 0 \ 0 \ v^+ \ 0 \ 0)^T \), where \( (\bar{v}^+ \ v^+)^T = v^0 \) constitutes a 2-dimensional eigenvector corresponding to the minimal non-negative eigenvalue of

\[
P_{con}^{al,0} = B_{c}^0 (P_{c}^0 - P_{1}^{al,0^T} P_{0}^{al,-1} P_{1}^{al,0}) \tag{47}
\]

and \( w^{al} \equiv \bar{v}^- \) is a real number computed as

\[
w^{al} = -P_{0}^{al,-1} P_{1}^{al,0} v^0. \tag{48}
\]
Proof. Let us recall that, as well as in Proposition 2, the inversion in (48) exists up to the case when all points lie on double-line $x^2 - y^2 = 0$.

Using the method of Lagrange multipliers, we acquire the Lagrange function in the form

$$P_\lambda(Q^{al,0}) := Q^{al,0T}PQ^{al,0} + \lambda(1 - Q^{al,0T}BQ^{al,0})$$

and after applying the block decomposition (45), it can be rewritten as

$$P_\lambda(Q^{al,0}) = w^{alT}P_0^{al}w^{al} + 2w^{alT}P_1^{al,0}v^0 + v^0TP_0^0v^0 + \lambda(1 - v^0T B_0^0v^0).$$

After differentiation w.r.t. the unknown vector $Q^{al,0}$ and solving the resulting system of linear equations, the rest of the proof proceeds similarly to proof of Proposition 1.

As in the case of the origin-centred conic fitting, we apply simple normalisation constraint (46) and with the use of the matrix $P^{al,0}_{con}$ in the form (47) reach the solution without computing a single eigenvalue. Since the matrix $P^{al,0}_{con}$ is generally different from $P^{0}_{con}$ and does not consist of the entries of matrix $P_{con}$, we must distinguish the notation of its elements as well, for instance by adding apostrophes:

$$P^{al,0}_{con} = \begin{pmatrix} a' & r' & \bar{s}' \\ s' & a' \end{pmatrix}. \quad (49)$$

**Proposition 7.** The solution to the optimisation problem (43), (44) for conic fitting in GAC is given by $Q^{al,0} = (0, w^{al}, \bar{v}^+, 0, 0, v^+, 0, 0)^T$, where $(\bar{v}^+ \ v^+)^T = v^0$ is a vector obtained as

$$v^0 = \begin{pmatrix} \bar{v}^+ \\ v^+ \end{pmatrix} = \begin{pmatrix} \pm \sqrt{r'/4s'} \\ \mp \frac{1}{2\sqrt{r'/4s'}} \end{pmatrix}$$

and $w^{al} \equiv \bar{v}^-$ is a real number computed as

$$w^{al} = -P^{al,-1}_{0} P^{al,0}_{1} v^0. \quad (50)$$

Proof. The eigenvalue problem derived in Proposition 6 reads

$$P^{al,0}_{con}v^0 = \lambda v^0$$

and is almost the same as in the proof of Proposition 4, only the matrix $P^{0}_{con}$ is replaced with $P^{al,0}_{con}$. Therefore, the rest of the proof for the vector $v^0$ computation proceeds in a way analogous to the Proposition 4, only using the elements of $P^{al,0}_{con}$.

Additionally, computation of the vector $w^{al}$ given by (50) is the same as in Proposition 6.

Similarly to fitting an origin-centred conic in Sect. 4.2, it is possible to obtain the solution using Algorithm $Q$ on an appropriately extended dataset.
Proposition 8. The solution to the optimisation problem (43), (44) for conic fitting in GAC is given by Proposition 1 provided that the dataset consisting of the points \( p_i = (x_i, y_i) \), \( i = 1, \ldots, N \), is extended by points \( p_i^{-} = (-x_i, -y_i) \), \( i = 1, \ldots, N \), which constitute centrally symmetric images of points \( p_i \) w.r.t. the origin of the coordinate system and by points \( p_i^{++} = (x_i, y_i) \), \( i = 1, \ldots, N \), and points \( p_i^{-+} = (-x_i, y_i) \), \( i = 1, \ldots, N \), which are axially symmetric images of points \( p_i \) w.r.t. y-axis and x-axis, respectively.

Proof. Proof is conducted in a way similar to the proof of Proposition 5. Let us note that the points \( p_i, p_i^{--}, p_i^{-+}, p_i^{++} \) are in GAC represented by respective vectors

\[
P_i = \begin{pmatrix} 0 & 0 & 1 \\ x_i & y_i & \frac{1}{2}(x_i^2 + y_i^2) \\ \frac{1}{2}(x_i^2 - y_i^2) & x_i y_i \end{pmatrix}^T,
\]

\[
P_i^{--} = \begin{pmatrix} 0 & 0 & 1 \\ -x_i & -y_i & \frac{1}{2}(x_i^2 + y_i^2) \\ \frac{1}{2}(x_i^2 - y_i^2) & x_i y_i \end{pmatrix}^T,
\]

\[
P_i^{-+} = \begin{pmatrix} 0 & 0 & 1 \\ x_i & -y_i & \frac{1}{2}(x_i^2 + y_i^2) \\ \frac{1}{2}(x_i^2 - y_i^2) & x_i y_i \end{pmatrix}^T,
\]

\[
P_i^{++} = \begin{pmatrix} 0 & 0 & 1 \\ -x_i & y_i & \frac{1}{2}(x_i^2 + y_i^2) \\ \frac{1}{2}(x_i^2 - y_i^2) & x_i y_i \end{pmatrix}^T.
\]

Consequently, the matrix \( P_{\text{con}} \) in our case is of the form

\[
P_{\text{con}}^{\text{al,0,sym}} = \begin{pmatrix} a^{**} & 0 & 0 & r^{**} \\ 0 & t^{**} & b^{**} & 0 \\ 0 & b^{**} & u^{**} & 0 \\ s^{**} & 0 & 0 & a^{**} \end{pmatrix}.
\]

Fortunately, the corner elements of \( P_{\text{con}}^{\text{al,0,sym}} \) can be expressed in terms of the entries of \( P_{\text{con}}^{\text{al,0}} \) of the form (49). Specifically, we get

\[
P_{\text{con}}^{\text{al,0,sym}} = \begin{pmatrix} a^{**} & 0 & 0 & r^{**} \\ 0 & t^{**} & b^{**} & 0 \\ 0 & b^{**} & u^{**} & 0 \\ s^{**} & 0 & 0 & a^{**} \end{pmatrix} = \begin{pmatrix} 4a' & 0 & 0 & 4r' \\ 0 & t^{**} & b^{**} & 0 \\ 0 & b^{**} & u^{**} & 0 \\ 4s' & 0 & 0 & 4a' \end{pmatrix}.
\]

Therefore, the set \( V^{**} \) of its eigenvectors

\[
v_1^{**} = \begin{pmatrix} v_{11}^{**} \\ v_{12}^{**} \\ v_{13}^{**} \\ v_{14}^{**} \end{pmatrix}, \ldots, v_4^{**} = \begin{pmatrix} v_{41}^{**} \\ v_{42}^{**} \\ v_{43}^{**} \\ v_{44}^{**} \end{pmatrix},
\]

yields

\[
V^{**} = \begin{pmatrix} v_1^{**} & v_2^{**} & v_3^{**} & v_4^{**} \end{pmatrix} = \begin{pmatrix} v_{11}^{**} & 0 & 0 & v_{14}^{**} \\ 0 & v_{32}^{**} & 0 & 0 \\ v_{23}^{**} & 0 & 0 \\ v_{14}^{**} & 0 & 0 & v_{44}^{**} \end{pmatrix},
\]

with \( \lambda_1^{**}, \ldots, \lambda_4^{**} \) as their corresponding eigenvalues. From the geometric nature of the problem, the dataset centrally symmetric w.r.t. the coordinate origin and axially symmetric w.r.t. both x- and y-axis must necessarily be fitted with axes-aligned, origin-centred conic, therefore, neither of the eigenvectors \( v_2^{**} \) and \( v_3^{**} \) can be the solution, since they do not correspond to an
origin-centred conic (also, neither of the eigenvalues $\lambda_2^{**}$ and $\lambda_3^{**}$ can represent the optimal solution). Further analysis would yield

$$
\begin{pmatrix}
\lambda_1^{**} \\
\lambda_4^{**}
\end{pmatrix} = 4
\begin{pmatrix}
\lambda_{1al}^{0} \\
\lambda_{2al}^{0}
\end{pmatrix}
$$

and

$$
\begin{pmatrix}
v_1^{**} \\
v_4^{**}
\end{pmatrix} \approx
\begin{pmatrix}
v_1^{al,0} \\
v_2^{al,0}
\end{pmatrix}.
$$

Therefore, similarly to situation in proof of Proposition 5, the optimal eigenvalue of the matrix $P_{con}^{al,0,sym}$ corresponds to the same solution as the one obtained by using the matrix $P_{con}^{al,0}$ in Proposition 6, which is also equivalent to the solution acquired in Proposition 7.

Let us further denote three algorithms for fitting an axes-aligned origin-centred conic $Q^{al,0}$, each of them corresponding to one of the presented Propositions, as:

- **QAL0**—via eigenvalues and eigenvectors in Proposition 6
- **QAL0-dir**—‘direct’ computation using Proposition 7 without computing eigenvalues
- **QAL0-sym**—central and axial symmetrisation of the dataset according to Proposition 8

As in the case of previous algorithms, we can use in Algorithms QAL0 and QAL0-dir the reduced forms of vectors and matrices. Therefore, we can define the $i$-th reduced GAC point

$$
\hat{P}_{i}^{al,0} = \begin{pmatrix}
1 & \frac{1}{2}(x_i^2 + y_i^2) & \frac{1}{2}(x_i^2 - y_i^2)
\end{pmatrix}^T
$$

and create the corresponding reduced data matrix $\hat{D}_{i}^{al,0}$ of the size $3 \times N$, where the $i$-th column is $\hat{P}_{i}^{al,0}$. Additionally, by removing the rows 1, 4, 5, 7 and 8 and columns 1, 2, 4, 5 and 8 from the matrix $B$ of the form (3), we also define a reduced matrix for GAC inner product

$$
\hat{B}_{c}^{al,0} =
\begin{pmatrix}
0 & 0 & -1 \\
0 & -1 & 0 \\
-1 & 0 & 0
\end{pmatrix}.
$$

Reduction of the matrix $B_c$ used in Algorithms QAL0 and QAL0-dir is the matrix $B_c^0$ of the form (31), while computation of the matrix $P$ yields a $3 \times 3$ reduced matrix of the form

$$
\hat{P}_{al,0} =
\begin{pmatrix}
P_{0}^{al} & P_{1}^{al,0} \\
P_{1}^{al,0}^T & P_{c}^{0}
\end{pmatrix}.
$$

Therefore, after defining a reduced GAC vector of origin-centred conic

$$
\hat{Q}_{al,0} = (\bar{v}^- \bar{v}^+ v^+)^T,
$$

the objective function (43) can be equivalently formulated as

$$
\hat{Q}_{al,0} \mapsto \hat{Q}_{al,0}^T \hat{P}_{al,0} \hat{Q}_{al,0}.
$$
Also, we can reuse some steps of Algorithm QAL0 in Algorithm QAL0-dir, while the other steps would be altered. Let us also note that Algorithm QAL0-sym uses four times bigger dataset than the previous two algorithms, so it is not ideal for larger amount of data, at least with regard to computational complexity. Nevertheless, we included it because of its geometrical significance.

4.4. Overview of Fitting Methods

Together with the original Algorithm Q, we derived eight conic fitting algorithms in total, each of them fitting one of four types of conics: a general conic $Q$, an axes-aligned conic $Q_{al}$, an origin-centred conic $Q^0$ or an axes-aligned origin-centred conic $Q_{al,0}$. Overview of the fitting methods by a conic type with the corresponding algorithms, types of computational methods and the objective functions is given by Table 1.

Let us point out that in the algorithms using the eigenvalues and eigenvectors of a certain matrix operator, the calculation of the corresponding operator has essentially the same structure for all the algorithms. As the default equation we take (12), where the matrix $P_{con}$—the operator used in Algorithm Q—is computed using the central subblock $B_c$ of the matrix $B$ and the subblocks $P_0$, $P_1$ and $P_c$ of matrix $P$ given by the decomposition (10). Consequently, the matrix operators used in Algorithms QAL, Q0 and QAL0 are, indeed, calculated with equations almost identical to Eq. (12), only the subblock of the matrix $B$ and the subblocks of matrix $P$ are generally substituted by different subblocks, according to the corresponding decomposition of matrices $B$ and $P$. The overview of the matrix operators computation by a conic type is given by Table 2.

5. Experimental Results and Applications

Finally, we apply each of the presented algorithms (including the original Algorithm Q) on a sample dataset of 10 points described by Table 3. After doing so, we compare the resulting conics both visually and by values of the corresponding objective functions. Resulting conics sorted by type and used algorithm can be seen in Figs. 1, 2, 3 and 4.

Even though all three algorithms for fitting an origin-centred conic $Q^0$ produce the same result, an additional subfigure (b) was made in Fig. 3 to emphasise the fact that Algorithm Q0-sym uses symmetrically extended dataset, unlike the cases of Algorithms Q0 and Q0-dir. Two subfigures in Fig. 4 have the same purpose in relation to axes-aligned origin-centred conic $Q_{al,0}$. It is obvious that in our case the new conic fitting methods with additional geometric constraints produce conics with greater total distance from the data points than general conic $Q$ acquired by Algorithm Q. Even though the difference is not that evident when comparing conics $Q$ and $Q_{al}$, one cannot overlook the increase of total distance from the data points to conic $Q^0$, let alone the case of conic $Q_{al,0}$. Consequently, we can conclude that the ability

\[ \text{Let us note that the comparison of the original Algorithm Q with other conic fitting algorithms without additional geometric conditions can be found in [10].} \]
| Conic                              | Algorithm | Computational method                        | Objective function |
|------------------------------------|-----------|--------------------------------------------|--------------------|
| General conic $Q$                  | $Q$       | Eigenvalues and eigenvectors               | $Q^T PQ$           |
| Axes-aligned conic $Q^{al}$        | $Q_{AL}$  | Eigenvalues and eigenvectors               | $Q^{alT} PQ^{al}$  |
| Origin-centred conic $Q^0$         | $Q_0$     | Eigenvalues and eigenvectors               | $Q^{0T} PQ^0$      |
|                                     | $Q_0$-dir | Direct use of matrix entries               |                    |
|                                     | $Q_0$-sym | Central symmetrisation of point dataset    |                    |
| Axes-aligned origin-centred conic $Q^{al,0}$ | $Q_{AL0}$ | Eigenvalues and eigenvectors               | $Q^{al,0T} PQ^{al,0}$ |
|                                     | $Q_{AL0}$-dir | Direct use of matrix entries               |                    |
|                                     | $Q_{AL0}$-sym | Central and axial symmetrisation of point dataset |                    |
Table 2. Construction of matrix operator for eigenvalue problem by conic type

| Conic | Subblock of $B$ | Subblocks of $P$ | Matrix operator |
|-------|-----------------|------------------|-----------------|
| $Q$   | $B_c$           | $P_0$ $P_1$ $P_c$ | $P_{con}^* = B_c^* (P_c^* - P_1^T P_0^{-1} P_1^*)$ |
| $Q^{al}$ | $B_c$ | $P_0^{al}$ $P_1^{al}$ $P_c$ | |
| $Q^0$  | $B_0^c$        | $P_0$ $P_0^0$ $P_0^0$ | |
| $Q^{al,0}$ | $B_0^c$ | $P_0^{al}$ $P_0^{al,0}$ $P_0^0$ | |

Table 3. Fitted points

| $x_i$ | $-3$ | $-3$ | $-2$ | $-3$ | $-6$ | $-7$ | $-9$ | $-10$ | $-9$ | $-7$ |
|-------|------|------|------|------|------|------|------|-------|------|------|
| $y_i$ | 3    | 4    | 6    | 7    | 8    | 8    | 7    | 4     | 2    | 1    |

Figure 1. General conic $Q$

to achieve a tight fit is decreasing with the fitted conic’s degree of freedom reduction (at least for our dataset), in other words, value of objective function is inversely proportional to the degrees of freedom (d.f.). Indeed, we can see this trend in Figs. 1, 2, 3 and 4: a general conic $Q$ has five d.f. in total; an axes-aligned conic $Q^{al}$ has one less d.f. due to the fixed tilt of axes; an origin-centred conic $Q^0$ loses two d.f. because of the fixed position of its centre-point; finally, a conic $Q^{al,0}$ has three d.f. less than a general conic $Q$, since it has both the tilt of axes and the centre-point position fixed. The fits depicted in Figs. 1, 2, 3 and 4 are characterised by the corresponding conic type, value of the objective function and degrees of freedom in Table 4. Due to the layout of the points in the used dataset, conics $Q^0$ and $Q^{al,0}$ are not exactly tightly fitted, nonetheless, such fits can be conveniently used
for data consisting of points that approximately lie on an origin-centred, or axes-aligned origin-centred conic, respectively.

To address the above mentioned application in switched systems, [3], we refer to the corresponding algorithms as follows. In the cases, when we know beforehand that the resulting integral curves of ODE’s should be origin-centred conics, it is appropriate to fit the points with one of Algorithms \( Q_0 \), \( Q_0\text{-dir} \) and \( Q_0\text{-sym} \) instead of Algorithm \( Q \), which cannot ensure additional geometric constraints. Moreover, other ODE’s lead to trajectories simultaneously axes-aligned and origin-centred, so any of Algorithms \( \text{QAL0} \), \( \text{QAL0-dir} \) and \( \text{QAL0-sym} \) can be applied in that case. Such an approach makes the initial trajectories and the following GAC calculations very precise.
Table 4. Objective function and geometric freedom

| Conic       | Value of objective function | Degrees of freedom |
|-------------|-----------------------------|--------------------|
| $Q$         | 0.0245                      | 5                  |
| $Q^{al}$    | 0.0871                      | 4                  |
| $Q^0$       | 1.0369                      | 3                  |
| $Q^{al,0}$  | 4.3361                      | 2                  |

6. Conclusion

At the beginning of the text, we briefly summarised the concept of GAC and recalled the original conic fitting algorithm using GAC and specific normalisation constraint. Consequently, by using basic knowledge of conic sections theory and exploiting the matrix formulation of the original fitting problem, [10], we derived methods of fitting conics with one of three additional geometric properties, in particular: having the axes aligned with the coordinate axes, having the centre located at the origin of coordinates, and, eventually, two previous properties combined.

All in all, we presented eight conic fitting algorithms in total, each resulting in one of four geometrically (un)constrained types of conics. Moreover, redundant elements used in the derivations of the fitting methods were mostly omitted in the described algorithms (including the original algorithm), thus the computational demands on both time and memory have been reduced.

All the methods were tested on a sample dataset and the resulting conics were analysed visually and by the values of the corresponding objective functions. In particular, our empirical results showed a relationship between a fitted conic’s degrees of freedom and tightness of the fit to the data points: the more degrees of freedom, the tighter the fit. We point out that all GAC fitting
algorithms are not invariant under translation, which is a natural property due to the GAC normalisation constraint.

Also, we mentioned possible use of the algorithms in the switched systems problem.

In the future, we can also consider inclusion of other normalisation constraints, such as Fitzgibbon’s, [5], to our optimisation process, but this needs deeper analysis of conditions on the existence of the optimal solution.

Acknowledgements
The research was supported by a grant no. FSI-S-23-8161.

Data availability Inclusion of a data availability statement is not applicable.

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

References
[1] Byrtus, R., Derevianko, A., Vašík, P.: Outline of tube elbow detection based on GAC. In: Magnenat-Thalmann N. et al. (eds.) Advances in Computer Graphics. CGI 2020. Lecture Notes in Computer Science, vol. 12221. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-61864-3_41
[2] Byrtus, R., Derevianko, A., Vašík, P., Hildenbrand, D., Steinmetz, C.: On specific conic intersections in GAC and symbolic calculations in GAALOP-Web. Adv. Appl. Clifford Algebras 32, 2 (2022). https://doi.org/10.1007/s00006-021-01182-z
[3] Derevianko, A., Vašík, P.: Solver-free Optimal Control for Linear Dynamical Switched System by means of Geometric Algebra. Mathematical Methods in the Applied Sciences (2022). https://doi.org/10.1002/mma.8752
[4] Fitzgibbon, A.W., Fisher, R.B.: A Buyer’s guide to conic fitting. In: Proceedings of the 6th British Conference on Machine Vision, vol. 2, pp. 513–522 (1995)
[5] Fitzgibbon, A., Pilu, M., Fisher, R.B.: Direct least square fitting of ellipses. IEEE Trans. Pattern Anal. Mach. Intell. 21(5), 476–480. ISSN 01628828
[6] GitHub Repository. Conic fitting algorithms using Geometric Algebra for Conics (GAC). https://github.com/Palouk94/Conic-fitting-algorithms-using-Geometric-Algebra-for-Conics-GAC
[7] Hestenes, D.: Space-Time Algebra. Gordon and Breach, New York (1966)
[8] Hildenbrand, D.: Introduction to Geometric Algebra Computing. CRC Press, Taylor & Francis Group, Boca Raton (2019)
[9] Hrdina, J., Návrat, A., Vašík, P.: Geometric algebra for conics. Adv. Appl. Clifford Algebras 28, 66 (2018). https://doi.org/10.1007/s00006-018-0879-2

[10] Hrdina, J., Návrat, A., Vašík, P.: Conic fitting in geometric algebra setting. Adv. Appl. Clifford Algebras 29, 72 (2019). https://doi.org/10.1007/s00006-019-0989-5

[11] Korn, G.A., Korn, T.M.: Mathematical Handbook for Scientists and Engineers. McGraw-Hill Book Company, New York (1961)

[12] Loučka, P., Vašík, P.: (2021) Algorithms for multi-conditioned conic fitting in geometric algebra for conics. In: Magnenat-Thalmann, N. et al. (eds.) Advances in Computer Graphics. CGI (2021)

[13] Perwass, Ch.: Geometric Algebra with Applications in Engineering. Springer, Berlin (2009)

[14] Richter-Gebert, J.: Perspectives on Projective Geometry: A Guided Tour Through Real and Complex Geometry. Springer, Berlin (2016)

[15] Waibel, P., Matthes, J., Gröll, L.: Constrained ellipse fitting with center on a line. J. Math. Imaging Vis. 53, 364–382 (2015)

Pavel Loučka and Petr Vašík
Technická 2896/2
616 69 Brno
Czech Republic
e-mail: Petr.Vasik@vutbr.cz

Pavel Loučka
e-mail: Pavel.Loucka@vutbr.cz

Received: November 16, 2021.
Accepted: April 19, 2023.