HAL open science

Intersection of conic sections using geometric algebra

Clément Chomicki, Stéphane Breuils, Venceslas Biri, Vincent Nozick

To cite this version:

Clément Chomicki, Stéphane Breuils, Venceslas Biri, Vincent Nozick. Intersection of conic sections using geometric algebra. CGI ENGAGE 2023, Aug 2023, Shanghai, China. pp.175-187, 10.1007/978-3-031-50078-7_14 . hal-04210085

HAL Id: hal-04210085

https://hal.science/hal-04210085v1

Submitted on 18 Sep 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Intersection of conic sections using geometric algebra

Clément Chomicki[1], Stéphane Breuils[2], Venceslas Biri[1], and Vincent Nozick[1]

[1] LIGM, Univ Gustave Eiffel, CNRS, France
[2] LAMA, Univ Savoie Mont-Blanc, CNRS, France

Abstract. Conic sections are extensively encountered in a wide range of disciplines, including optics, physics, and various other fields. Consequently, the geometric algebra community is actively engaged in developing frameworks that enable efficient support and manipulation of conic sections. Conic-conic intersection objects are known and supported by algebras specialized in conic sections representation, but there is yet no elegant formula to extract the intersection points from them. This paper proposes a method for point extraction from an conic intersection through the concept of pencils. It will be based on QC2GA, the 2D version of QCGA (Quadric Conformal Geometric Algebra), that we also prove to be equivalent to GAC (Geometric Algebra for Conics).

Keywords: Conics, Geometric Algebra, Projective Geometry, Clifford Algebra, QCGA (Quadric Conformal Geometric Algebra), GAC (Geometric Algebra for Conics), Pencil

1 Introduction

Geometric algebras (GA) are a convenient way to represent and manipulate geometric primitives. They have been used in physics for decades [12,15,16] to unify and simplify some models.

Presently, GA applications in computer sciences are widespread, used in various domains such as neural networking [4,21] and computer graphics - where they can be used to manipulate geometric primitives [6]. For an introduction to geometric algebras, please refer to Perwass' and Dorst's textbooks [19,7].

Because polynomial embedding [19] is done very naturally in geometric algebras, polynomial curves and geometric algebra form a promising marriage. This is why several GA frameworks have already been proposed for both curves and surfaces in order to represent, transform and intersect these objects.

Perwass [19] started with a simple blade-based approach in $\mathbb{G}_{5,3}$ to represent 2D conics constructed from 5 points. This algebra also supports translations and rotations of conics, but is presented as just a proof of concept to be further developed. Later on, Goldman et al. proposed $\mathbb{R}(4,4)$ [11], an algebra composed of two projective $\mathbb{R}^4$ basis capable of supporting quadrics (and so conic when one dimension is removed). Similarly, DCGA introduced by Easter et al. [8] is composed of two CGA basis and can represent general Darboux cyclide, which

embeds quadrics. The common weakness of those two algebras is that their curves and surfaces are not constructed from control points.

Another algebra supporting quadric surfaces is QCGA (Quadric Conformal Geometric Algebra) [1] from Breuils et al, which is a 3D extension of Perwass' conic algebra that explores intersections and transformations. QCGA can be lowered in dimension (by removing 7 vectors of its basis) to get QC2GA, an algebra for handling conics. Finally, GAC (Geometric Algebra for Conics) [17] is a more recent proposition from Hrdina et al., which unlike the other ones, is fully dedicated to conics. It is very similar to QC2GA, the 2D version of QCGA. These two algebras, just like Perwass', support constructing their object from points, and notable distinction between the two lies in the inclusion of a third spatial dimension within QCGA, while GAC (Geometric Algebra of Conics) offers a broader range of geometric transformations. The common point of these algebras is the lack of way to process conic intersection objects into points, and other things that would be extremely useful such as determining the type of a conic. We also want to be able to tell if a conic is a pair of lines, and if so, we should have a way to extract these two lines. Conic section also have a center and several other relevant lines and foci that would be interesting to extract and manipulate. GAC is actually able to extract points from conic intersection on some very specific and easy cases [5], but what we want is a general method that works on any intersection.

This paper presents a method to extract the points from any conic intersection objects. The paper is organized as follows : Section 2 introduces state-of-the-art of conic intersection in both projective geometry and geometric algebra. Section 3 focuses on QC2GA and GAC. Section 4, as a first contribution, demonstrates that they are actually equivalent. Our major contribution is detailed in Section 5 that presents an algorithm to find the intersection points of two conics in QC2GA by using an associated cubic polynomial and the naturally-supported pencils of conics.

2 Conics theoretical background

This section introduces various ways of representing conics and their properties.

2.1 Conics representation

Conics' traces can be found as far as 380 BC by Menaechmus in the ancient Greece. In the last couple of centuries, mathematicians linked these planar curves to the quadratic equations of two variables and projective algebra. This is the formalism that this paper uses.

As stated by Faucette [9], conics are planar polynomial curves of degree 2:

$$C : g(x, y) = ax^2 + by^2 + cxy + dx + ey + f = 0 \tag{1}$$

With $(a, b, c, d, e, f) \in \mathbb{K}^6$ and $\mathbb{K} = \mathbb{R}$ or $\mathbb{C}$. When not degenerate and not complex, a conic can be seen as the intersection between a double cone and a plane.

It is common to work in the projective space $\mathbb{P}^2$ to represent points and in $\mathbb{P}^5$ to represent conics. $\mathbb{P}^n$ is the set of all equivalence classes $(a_i)_{i\in[1,n+1]} \in \mathbb{K}^{n+1} \setminus \{0_n\}$ under the equivalence relation $(a_1, \dots, a_{n+1}) \sim (b_1, \dots, b_{n+1}) \iff \exists \lambda \in \mathbb{K}, (a_1 \dots, a_{n+1}) = \lambda(b_0, \dots, b_{n+1})$. A finite point (x, y) of $\mathbb{K}^2$ is then embedded as $(x, y, 1)$ (or more generally as (wx, wy, w)) in $\mathbb{P}^2$, and infinite points of direction (x, y) is embedded as $(x, y, 0)$. In order to embed $\mathbb{P}^2$ into $\mathbb{P}^5$, we consider the polynomial map $Q : \mathbb{P}^2 \to \mathbb{P}^5$ so that $Q(x, y, w) = (x^2, y^2, xy, xw, yw, w^2)$. Conics are then represented by the vectors of $\mathbb{P}^5$, and the equation (1) becomes

$$C : \mathfrak{C}^\top \cdot Q(x, y, w) = 0 \quad \text{with} \quad \mathfrak{C} = [a, b, c, d, e, f]^\top \in \mathbb{P}^5 \tag{2}$$

Conics are also often represented by their Hessian matrix, which enables to reformulate equation (1):

$$C : p^\top \mathcal{H} p = 0 \qquad \text{with} \quad \mathcal{H} = \begin{bmatrix} a & \frac{c}{2} & \frac{d}{2} \\ \frac{c}{2} & b & \frac{e}{2} \\ \frac{d}{2} & \frac{e}{2} & f \end{bmatrix} \tag{3}$$

Assume five linearly independent points $(p_i)_{i\in[1,5]}$ with coordinates $(x_i, y_i, w_i)_{i\in[1,5]} \in \mathbb{P}^2$ and another point p_0 of coordinate $(x, y, w) \in \mathbb{P}^2$. Let $q_i = Q(p_i), \forall i \in [0, 5]$. The conic equation (C) is often expressed in the form of the vanishing determinant of a matrix $\mathcal{P} = [q_{i,j}]$ [13]:

$$C : \det(\mathcal{P}) = \begin{vmatrix} x^2 & y^2 & xy & xw & yw & w^2 \\ x_1^2 & y_1^2 & x_1y_1 & x_1w_1 & y_1w_1 & w_1^2 \\ x_2^2 & y_2^2 & x_2y_2 & x_2w_2 & y_2w_2 & w_2^2 \\ x_3^2 & y_3^2 & x_3y_3 & x_3w_3 & y_3w_3 & w_3^2 \\ x_4^2 & y_4^2 & x_4y_4 & x_4w_4 & y_4w_4 & w_4^2 \\ x_5^2 & y_5^2 & x_5y_5 & x_5w_5 & y_5w_5 & w_5^2 \end{vmatrix} = 0 \tag{4}$$

Then, if we denote by $m_i^j(A)$ the respective minor of matrix A, i.e. the determinant of A with its i-th column and j-th line removed. If we drop the superscript for $j = 1$, i.e. $m_i = m_i^1$ we have:

$$\begin{aligned} a &= m_1(\mathcal{P}) & b &= -m_2(\mathcal{P}) & c &= m_3(\mathcal{P}) \\ d &= -m_4(\mathcal{P}) & e &= m_5(\mathcal{P}) & f &= -m_6(\mathcal{P}) \end{aligned} \tag{5}$$

There is a more general rule to that: the set of conics passing through n non-aligned points is a dimension $(5 - n)$ vector subspace of $\mathbb{P}^5$ [10]. We could then argue that 4 control points would form at most a 1-vector subspace, 3 points a 2-vector subspace, etc. This implies that the n-intersection of two conics C_a and C_b is more than their n common points, as the 3 points of a 3-intersection generates a 2-vector subspace when the two conics only creates a 1-vector subspace.

2.2 Type of a conic and degenerate case

When $\mathbb{K} = \mathbb{R}$, the type of the conic is ruled by these two discriminants [20]:

$$\Delta_2(C) = m_3^3(\mathcal{H}) = ab - \frac{1}{4}c^2 \tag{6}$$

$$\Delta_3(C) = \det(\mathcal{H}) = abf + \frac{ced - c^2 f - bd^2 - ae^2}{4} \tag{7}$$

$\Delta_2(C)$	$\Delta_3(C)$	type of C
$+$	$\neq 0$	Ellipse (may be imaginary)
0	$\neq 0$	Parabola
$-$	$\neq 0$	Hyperbola
$+$	0	Point
0	0	Two parallel lines
$-$	0	Two intersecting lines

Whether $\mathbb{K} = \mathbb{R}$ or $\mathbb{C}$, if $\Delta_3(C) = 0$, C is called **degenerate** or **reducible**, meaning that it can be factored into two complex lines.

2.3 Pencils

Let $C_a : g_a(x, y) = 0$ and $C_b : g_b(x, y) = 0$ be two conics and $\lambda \in \mathbb{R}^*$. $C_\lambda : g_\lambda(x, y) = g_a(x, y) + \lambda g_b(x, y) = 0$ is also a conic since g_a and g_b are linear forms of $\mathbb{P}^5$ and

$$C_a \cap C_b = C_a \cap C_\lambda = C_b \cap C_\lambda \tag{8}$$

Where $\cap$ is the set theory intersection operator.

This is because $g_\lambda(x, y, w) = g_a(x, y, w) + \lambda g_b(x, y, w)$ then if two of the terms are 0, the third one is also 0, and if one term is not zero, then at least one other is not zero. The 1-vector space generated C_a and C_b is called their **pencil**.

$$\text{Pencil}(C_a, C_b) = \{K(C_a + \lambda C_b) \qquad K, \lambda \in \mathbb{K}\} \tag{9}$$

$$= \{K(\cos(\theta)C_a + \sin(\theta)C_b) \qquad K, \theta \in \mathbb{K}\} \tag{10}$$

Four non-aligned points also generate a pencil of conics, but the reciprocal is not true: the intersection of two conics might contain less than 4 points (when $\mathbb{K} = \mathbb{R}$) or have one or several multiple points (see figure 1).

2.4 Intersecting conics

Finding the intersection between two conic sections is an old problem. Two different conics can have from 0 to 4 points of intersections (the intersection of a conic with itself is of course the whole conic). Gröbner basis [14] can be used to express the conic intersection problem as a quartic polynomial. Also, Faucette [9] describes a method to solve any quartic polynomial by finding the intersection points of two conics. Therefore, finding the intersection points of two conic sections is equivalent to find the root of a quartic polynomial.

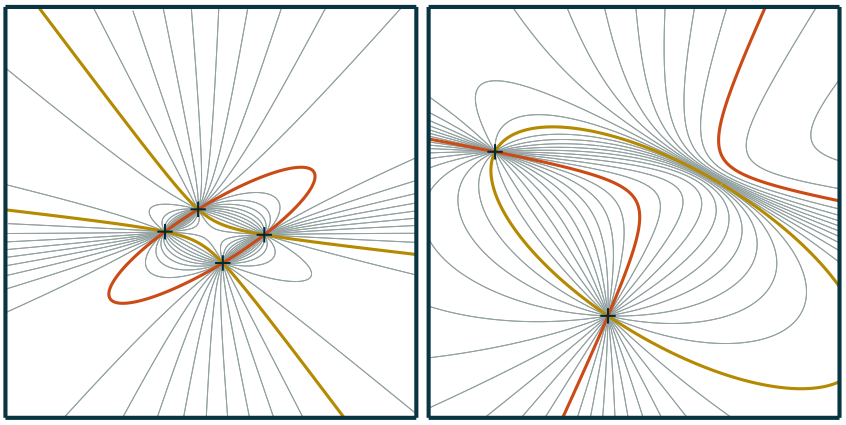

Fig. 1. The pencil (in grey) generated by two conics (in red and yellow) with 4 intersection points (left) and 2 intersection points (right).

The method described by Faucette [9] consists of finding a degenerate conic (i.e. that can be factored into two lines) in the pencil generated by the two intersecting conics. That pair of lines is then separated into two distinct lines, which can be subsequently used to find the intersection points with one of the conics of the pencil. Another method is to directly solve the associated quartic equation of the problem . Richter-Geber [20] reformulates this method in a more complete way and gives a detailed process to find the intersections. Just like [9], they solve the associated cubic equation, find any complex solution using a given formula, split the resulting degenerate complex conic into two complex lines and intersect the two resulting complex lines with one of the intersecting conic. Both of these method heavily relies on complex numbers and lines, but one could choose to only use real lines if they limited themselves to conic-lines intersections.

Now that conic intersections are introduced, the following section will present two algebras that handle conics.

3 A look into QC2GA and GAC

3.1 Conics and Geometric Algebra

Let's look back at equations (4) and (5), assuming that $\mathbb{K} = \mathbb{R}$. Computing the determinant of the matrix $\mathcal{P}$ can be done by working in $\mathbb{G}_6 = \bigwedge \mathbb{P}^5$ [19,18], where we find that the parameters of the conic (a, b, c, d, e, f) are the coordinates of the blade of grade 5, $\bigwedge_{i\in[1,5]} q_i$. The equation of the conic (4) then becomes.

$$C : q_1 \wedge q_2 \wedge q_3 \wedge q_4 \wedge q_5 \wedge q = 0 \tag{11}$$

Hence five points determines a conic gives us a natural outer product representation of the implicit equation of the conic. This leads directly to Perwass'

proposition [19], and then to QC2GA [1] and GAC [17]. In these algebras, points and conics are represented this way with their respective $\mathbb{G}_{5,3}$ bases:

$$p = x^2 e_1 + y^2 e_2 + xy e_3 + x e_4 + y e_5 + e_6 \tag{12}$$

$$C = p_1 \wedge p_2 \wedge p_3 \wedge p_4 \wedge p_5 \wedge e_7 \wedge e_8 \tag{13}$$

$$= a e_1^c + b e_2^c + c e_3^c + d e_4^c + e e_5^c + f e_6^c \tag{14}$$

In (12), $e_1^c, \ldots, e_6^c$ denotes the (right) complement-dual of the multivectors $e_1, \ldots, e_6$, defined through $m \wedge m^c = I$ for all multivector m [3].

3.2 Two-dimensional Quadric Conformal Algebra (QC2GA)

QC2GA is the 2D-version of the QCGA algebra [1] by Breuils et al. Its signature is $\mathbb{R}^{5,3}$, but we use this more convenient non-diagonal basis:

	e_1	e_2	e_{o_1}	e_{∞_1}	e_{o_2}	e_{∞_2}	e_{o_3}	e_{∞_3}
e_1	1	.	.	.	.	.	.	.
e_2	.	1	.	.	.	.	.	.
e_{o_1}	.	.	0	-1	.	.	.	.
e_{∞_1}	.	.	-1	0	.	.	.	.
e_{o_2}	.	.	.	.	0	-1	.	.
e_{∞_2}	.	.	.	.	-1	0	.	.
e_{o_3}	.	.	.	.	.	.	0	-1
e_{∞_3}	.	.	.	.	.	.	-1	0

QC2GA formalism relies on the following blades:

$$e_o = e_{o_1} + e_{o_2} \tag{15}$$

$$e_\infty = \frac{e_{\infty_1} + e_{\infty_2}}{2} \tag{16}$$

$$I_o^{\triangleright} = (e_{o_1} - e_{o_2}) \wedge e_{o_2} \tag{17}$$

$$I_\infty^{\triangleright} = (e_{\infty_1} - e_{\infty_2}) \wedge e_{\infty_2} \tag{18}$$

$$I_o = e_{o_1} \wedge e_{o_2} \wedge e_{o_2} \tag{19}$$

$$I_\infty = e_{\infty_1} \wedge e_{\infty_2} \wedge e_{\infty_3} \tag{20}$$

$$I_\epsilon = e_1 \wedge e_2 \wedge e_3 \tag{21}$$

$$I = I_\epsilon \wedge I_\infty \wedge I_o \tag{22}$$

QC2GA points $\overset{\text{QC2GA}}{p}$ and conics $\overset{\text{QC2GA}}{C}$ are defined as follows:

$$\overset{\text{QC2GA}}{p} = e_o + x e_1 + y e_2 + x^2 \frac{e_{\infty_1}}{2} + y^2 \frac{e_{\infty_2}}{2} + xy e_{\infty_3} \tag{23}$$

$$\overset{\text{QC2GA}}{C} = \overset{\text{QC2GA}}{p_1} \wedge \overset{\text{QC2GA}}{p_2} \wedge \overset{\text{QC2GA}}{p_3} \wedge \overset{\text{QC2GA}}{p_4} \wedge \overset{\text{QC2GA}}{p_5} \wedge I_o^{\triangleright} \tag{24}$$

$$\overset{\text{QC2GA}}{C^*} = a(\frac{e_{\infty_1}}{2})^{-1} + b(\frac{e_{\infty_2}}{2})^{-1} + c e_{\infty_3}^{-1} + d e_1^{-1} + e e_2^{-1} + f e_o^{-1} \tag{25}$$

$$= -2a e_{o_1} - 2b e_{o_2} - c e_{o_3} + d e_1 + e e_2 - f e_\infty \tag{26}$$

With $(a, b, c, d, e, f) \in \mathbb{R}^6$ the parameters of the conic C represented by $\overset{\text{QC2GA}}{C}$.

QC2GA supports intersections of conics, and evaluating if a point lies on one:

$$\overset{\text{QC2GA}}{Inter} = \overset{\text{QC2GA}}{C_1} \vee \overset{\text{QC2GA}}{C_2} = (\overset{\text{QC2GA}}{C_1^*} \wedge \overset{\text{QC2GA}}{C_2^*})^* \tag{27}$$

$$\overset{\text{QC2GA}}{p} \in \overset{\text{QC2GA}}{C} \iff \overset{\text{QC2GA}}{p} \wedge \overset{\text{QC2GA}}{C} = 0 \iff \overset{\text{QC2GA}}{p} \cdot (\overset{\text{QC2GA}}{C^*}) = 0 \tag{28}$$

3.3 Geometric Algebra for Conics (GAC)

GAC is another geometric algebra for conics from [17] by J. Hrdina et al. Its basis $(e_1, e_2, \bar{n}_+, n_+, \bar{n}_-, n_-, \bar{n}_\times, n_\times)$ has the same signature as QC2GA.

Points $\overset{\text{GAC}}{p}$ and conics $\overset{\text{GAC}}{C}$ are constructed this way in GAC:

$$\overset{\text{GAC}}{p} = \bar{n}_+ + xe_1 + ye_2 + \frac{x^2+y^2}{2}n_+ + \frac{x^2-y^2}{2}n_- + xyn_\times \tag{29}$$

$$\overset{\text{GAC}}{C} = \overset{\text{GAC}}{p_1} \wedge \overset{\text{GAC}}{p_2} \wedge \overset{\text{GAC}}{p_3} \wedge \overset{\text{GAC}}{p_4} \wedge \overset{\text{GAC}}{p_5} \wedge e_{o_2} \wedge e_{o_3} \tag{30}$$

$$\overset{\text{GAC}}{C^*} = -(a+b)\bar{n}_+ - (a-b)\bar{n}_- - c\bar{n}_\times + de_1 + ee_2 - fn_+ \tag{31}$$

The usage we have of GAC is the same than QC2GA.

$$\overset{\text{GAC}}{Inter} = \overset{\text{GAC}}{C_1} \vee \overset{\text{GAC}}{C_2} = (\overset{\text{GAC}}{C_1^*} \wedge \overset{\text{GAC}}{C_2^*})^* \tag{32}$$

$$\overset{\text{GAC}}{p} \in \overset{\text{GAC}}{C} \iff \overset{\text{GAC}}{p} \wedge \overset{\text{GAC}}{C} = 0 \iff \overset{\text{GAC}}{p} \cdot (\overset{\text{GAC}}{C^*}) = 0 \tag{33}$$

4 GAC and QC2GA are equivalent

The metric of GAC and QC2GA are identical, and at first sight the objects of these algebras looks very similar. It would make sense to find out that these two algebras are the same thing, which would unify all the work done on these two algebras. We actually can express GAC basis with QC2GA's, which shows a direct equivalence between QCGA and GAC objects:

$$\bar{n}_+ = e_{o_1} + e_{o_2} \quad (34) \qquad \bar{n}_- = e_{o_1} - e_{o_2} \quad (35) \qquad \bar{n}_\times = e_{o_3} \quad (36)$$

$$n_+ = \frac{e_{\infty_1} + e_{\infty_2}}{2} \quad (37) \qquad n_- = \frac{e_{\infty_1} - e_{\infty_2}}{2} \quad (38) \qquad n_\times = e_{\infty_3} \quad (39)$$

$$\overset{\text{GAC}}{p} = e_o + xe_1 + ye_2 + \frac{x^2 e_{\infty_1} + y^2 e_{\infty_2}}{2} + xye_{\infty_3} = \overset{\text{QC2GA}}{p} \tag{40}$$

$$\overset{\text{GAC}}{C} = -2ae_{o_1} - 2be_{o_2} - ce_{o_3} + de_1 + ee_2 - fe_\infty = \overset{\text{QC2GA}}{C} \tag{41}$$

Thanks to this reformulation, we establish the equivalence of GAC and QC2GA. GAC possess versors for rotation, translations, dilation and even "general reflection" (which looks similar as a CGA's spherical inversion but with conics [19]). QC2GA also have rotations and translations (inherited directly from Perwass' conformal conic algebra), and its translator is easier to express than GAC's. Due to the equivalence of these two algebras, tools from GAC and QC2GA can be utilized interchangeably. For the rest of this article, we will then omit the "QC2GA" or "GAC" on top of the geometric objects and follow the formalism of QC2GA, and every property relative to QC2GA will also hold for GAC.

5 Extracting points from QC2GA conics intersections

An intersection might contain from 0 to 4 points. However, the grade of that object is always 6. Therefore it is neither trivial to distinguish the type of intersection nor to extract the points, hence our method. By selecting a real degenerate conic from the intersection object's pencil, the conic is factorized into two lines. These lines are then intersected with another conic from the same pencil, resulting in the desired intersection points. Notably, this method exclusively operates on real objects, unlike from Faucette's approach that uses complex lines.

5.1 Pencil of conics in QC2GA

Let $Inter$ be an intersection of two conics and p a point not in $Inter$, we define $C = Inter \wedge p$ as the conic passing through every point of $Inter$ and through p. Summation of two conics is allowed by QC2GA, hence we can write $\forall \lambda \in \mathbb{R}, C_\lambda = C_a + \lambda C_b$. Summations of points is also supported, which gives the following:

$$Inter = C_a \vee C_b = (C_a^* \wedge C_b^*)^* \tag{42}$$

$$C_a = Inter \wedge p_a \qquad p_a \in C_a \setminus C_b \tag{43}$$

$$C_b = Inter \wedge p_b \qquad p_b \in C_b \setminus C_a \tag{44}$$

$$C_\lambda = Inter \wedge (p_a + \lambda p_b) \tag{45}$$

The pencil of C_a and C_b is then generated by their intersection $Inter$.

5.2 Extraction method

The next step consists in the extraction of the points contained in an intersection.

The two conics will be used to find a new pair of conics with one degenerate, by finding the root of an associated cubic polynomial. The degenerate conic will then be factored into two lines, that will be intersected with the other conic to get the intersection points using a simpler algorithm.

Building two conics from Inter In order to build two conics from $Inter$, one could just pick two random points and use the formulas of (42)-(45), but this would lead to possibilities of p_a and p_b to generate the same conic. In order to avoid that, it is possible to only pick one point p_a randomly, to generate a conic C_a from it and $Inter$ and to extract C_b as the normed element in the 2D-vector space $Inter$ perpendicular to C_a.

$$C_a = Inter \wedge p_a \tag{46} \qquad C_b = Inter \wedge {C_a}^c \tag{47}$$

Finding a degenerate conic (Algorithm 1) Now that we have two relevant conics to work with, we would like to find a degenerate conic, because a degenerate conic is either a point (which makes the problem trivial as we just take that point as the eventual solution), or a pair of lines that we will treat as two

Algorithm 1: Find a degenerate conic and a different one in the pencil of $Inter$

Function `gen_degen_and_other`

Input: $Inter$

Output: $C_{deg}, C_\perp$

do $p_a \leftarrow$ random_point() **while** $p_a \wedge Inter = 0$

$C_a \leftarrow Inter \wedge p_a \qquad C_b \leftarrow Inter \wedge (C_a^c)$

$a \leftarrow \Delta_3(C_b) \qquad b \leftarrow \dfrac{\Delta_3(C_a + C_b) + \Delta_3(C_a - C_b)}{2} - \Delta_3(C_a)$

$d \leftarrow \Delta_3(C_a) \qquad c \leftarrow \dfrac{\Delta_3(C_a + C_b) - \Delta_3(C_a - C_b)}{2} - \Delta_3(C_b)$

$\Delta_0 \leftarrow b^2 - 3ac \qquad \Delta_1 \leftarrow 2b^3 - 9abc + 27a^2 d$

$\Omega_- \leftarrow \sqrt[3]{\dfrac{\Delta_1 - \sqrt{\Delta_1^2 - 4\Delta_0^3}}{2}} \qquad \Omega_+ \leftarrow \sqrt[3]{\dfrac{\Delta_1 + \sqrt{\Delta_1^2 - 4\Delta_0^3}}{2}}$

$\lambda \leftarrow -3a \qquad \mu \leftarrow b + \Omega_- + \Omega_+$

$C_{deg} \leftarrow \lambda C_a + \mu C_b \qquad C_\perp \leftarrow -\mu C_a + \lambda C_b$

distinct lines, which reduces the conic-conic intersection to the simpler task of determining conic-line intersections. Finding a degenerate conic in the pencil of C_1 and C_2 can be done by solving the equation $\Delta_3(C_1 + \lambda C_2) = 0$, which can be expanded into equation (48).

$$\begin{aligned} &\Delta_3(C_2)\lambda^3 + \left(\frac{\Delta_3(C_1 + C_2) + \Delta_3(C_1 - C_2)}{2} - \Delta_3(C_1)\right)\lambda^2 \\ &+\left(\frac{\Delta_3(C_1 + C_2) - \Delta_3(C_1 - C_2)}{2} - \Delta_3(C_2)\right)\lambda + \Delta_3(C_1) = 0 \end{aligned} \tag{48}$$

To solve this cubic equation, we use a formula similar than Cardano's, but which allows to do less divisions and which allows us to easily obtain a real root when $k = 0$:

$$\Delta_0 = b^2 - 3ac \qquad \Delta_1 = 2b^3 - 9abc + 27a^2 d \tag{49}$$

$$\Omega_\pm = \sqrt[3]{\frac{\Delta_1 \pm \sqrt{\Delta_1^2 - 4\Delta_0^3}}{2}} \qquad x_k = -\frac{b + e^{i\frac{2\pi k}{3}}\Omega_+ + e^{i\frac{-2\pi k}{3}}\Omega_-}{3a} \tag{50}$$

Using this approach, we may write the Algorithm 1 that samples a degenerate conic and another conic significantly different (noticing that $\lambda C_a + \mu C_b$ is a rotation of angle $\arctan(\mu/\lambda)$ in the pencil) from the degenerate one from the pencil of conic of $Inter$. The variables a, b, c, d are not conic parameters, but rather the four parameters of the cubic polynomial.

Factorize lines pairs (Algorithm 2) A line pair is of the form:

$$l_1 = \cos(\alpha_1)x + \sin(\alpha_1)y + w_1 \tag{51}$$

$$l_2 = \cos(\alpha_2)x + \sin(\alpha_2)y + w_2 \tag{52}$$

$$C : Kl_1l_2 = ax^2 + by^2 + cxy + dx + ey + f = 0 \tag{53}$$

where $K \in \mathbb{R}^*$ is a multiplicative constant coming from the projective aspect of the conic (it can be any constant). These two lines can be extracted using the method presented by Richter-Gebert [20]. This method is also used by Byrtus et al for their specific GAC intersection extraction [5] . Algorithm 2 implements that.

Algorithm 2: Line pair factorization algorithm

Function `factor_line_pair`

Input: C_{deg}

Output: $\alpha_1, \alpha_2, w_1, w_2$ `// lines angles and offsets`

$H \leftarrow \text{hessian_matrix}(C_{deg}) \quad A \leftarrow \text{adjoint}(H) \quad i \leftarrow \arg\min_k \{A_{k,k}\}$

$$D \leftarrow \begin{bmatrix} 0 & -A_{i,2} & +A_{i,1} \\ +A_{i,2} & 0 & -A_{i,0} \\ -A_{i,1} & +A_{i,0} & 0 \end{bmatrix} / \sqrt{-A_{i,i}} \quad N \leftarrow H + D$$

$u_1, v_1, w_1 \leftarrow N\Big[*, \arg\max_j \{N[1,j]^2 + N[2,j]^2]\}\Big]$

$u_2, v_2, w_2 \leftarrow N\Big[\arg\max_j \{N[j,1]^2 + N[j,2]^2]\}, *\Big]$

$\alpha_1 \leftarrow \text{arctan2}(v_1, u_1) \quad \alpha_2 \leftarrow \text{arctan2}(v_2, u_2)$

return $\{\alpha_1, \alpha_2, w_1/\sqrt{u_1^2 + v_1^2}, w_2/\sqrt{u_2^2 + v_2^2}\}$

Conic-line intersection (Algorithm 3) The idea is to rotate the problem to have the line vertical (which means a known x), and we are then left with a trivial quadratic polynomial in y.

Algorithm 3: Conic-Line intersection algorithm

Function `conic_line_inter`

Input: C, θ, w

Output: points

$R \leftarrow \text{qc2ga_rotor}(-\theta) \quad [a, b, c, d, e, f] \leftarrow RC\tilde{R}$

$x \leftarrow -w \qquad \delta \leftarrow (cx + e)^2 - 4b(ax^2 + dx + f) \qquad \gamma \leftarrow \dfrac{cx + e}{2b}$

$$p_0 \leftarrow \begin{pmatrix} \cos(\theta)x - \gamma\sin(\theta) \\ \sin(\theta)x + \gamma\cos(\theta) \end{pmatrix} \qquad \boldsymbol{u} \leftarrow \frac{1}{2b}\begin{pmatrix} -\sin(\theta) \\ \cos(\theta) \end{pmatrix}$$

switch *sgn(δ)* **do**

  case $+$ **do return** $\{p_0 \pm \sqrt{\delta}\boldsymbol{u}\}$

  case 0 **do return** $\{p_0\}$

  case $-$ **do return** $\{\}$

end

5.3 Full Algorithm

Using the algorithms we just presented, we can write the Algorithm 4 that extracts the points in an intersection object. This algorithm have been implemented in C++ with Garamon [2] (see `https://github.com/technolapin/qc2ga-intersection`). Figure 2 gives outputs of our code.

Algorithm 4: Point extraction algorithm

Function `extract_pts_from_inter`
- **Input:** *Inter*
- **Output:** a set of points

$C_{deg}, C_{\perp} \leftarrow$ gen_degen_and_other($Inter$)
$\alpha_1, \alpha_2, w_1, w_2 \leftarrow$ factor_line_pair(C_{deg})
return conic_line_inter($C_{\perp}, \alpha_1, w_1$) $\cup$ conic_line_inter($C_{\perp}, \alpha_2, w_2$)

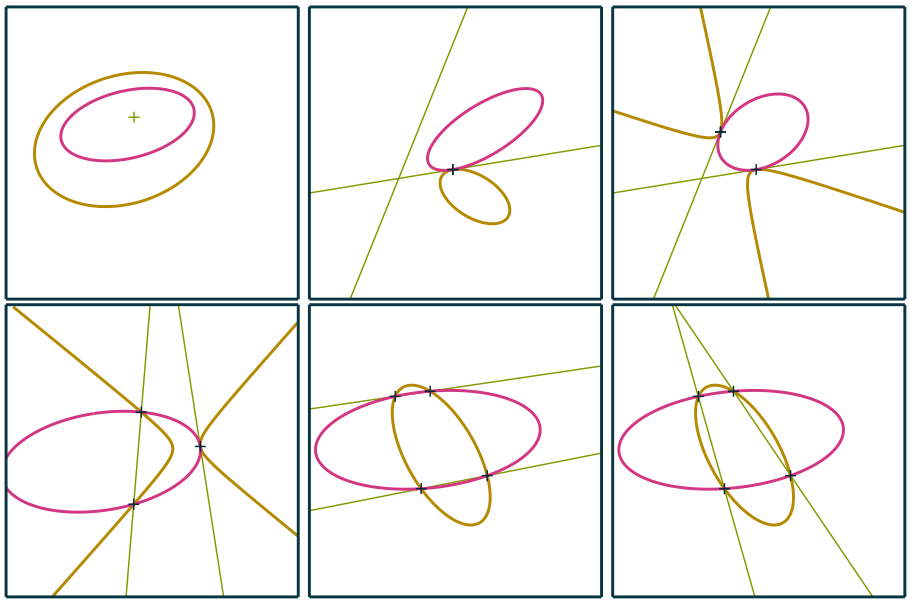

Fig. 2. Extraction of 0,1,2,3 and 4-intersections where the green curve is the degenerate conic, which can be a line pair or a point. The 4-intersection is depicted twice with different line pairs.

6 Conclusion

This paper established the equivalence of QC2GA and GAC, and presented a geometric-algebra driven method for the decomposition of a conic-conic inter-

section object into points. Further work is to be done to increase the part of geometric algebra in the process.

References

1. Breuils, S., Fuchs, L., Hitzer, E., Nozick, V., Sugimoto, A.: Three-dimensional quadrics in extended conformal geometric algebras of higher dimensions from control points, implicit equations and axis alignment. Advances in Applied Clifford Algebras **29**, 1–22 (2019)
2. Breuils, S., Nozick, V., Fuchs, L.: Garamon: A Geometric Algebra Library Generator. Advances in Applied Clifford Algebras **29**(4), 69 (2019)
3. Browne, J.: Grassmann Algebra Volume 1: Foundations: Exploring Extended Vector Algebra with Mathematica, vol. 1. John M Browne (2012)
4. Buchholz, S., Tachibana, K., Hitzer, E.M.: Optimal learning rates for Clifford neurons. In: International conference on artificial neural networks, pp. 864–873. Springer (2007)
5. Byrtus, R., Derevianko, A., Vašík, P., Hildenbrand, D., Steinmetz, C.: On specific conic intersections in gac and symbolic calculations in gaalopweb. Advances in Applied Clifford Algebras **32**(1), 2 (2021)
6. De Keninck, S.: Non-parametric realtime rendering of subspace objects in arbitrary geometric algebras. In: M. Gavrilova, J. Chang, N.M. Thalmann, E. Hitzer, H. Ishikawa (eds.) Advances in Computer Graphics, pp. 549–555. Springer International Publishing, Cham (2019)
7. Dorst, L.: The inner products of geometric algebra. In: Applications of Geometric Algebra in Computer Science and Engineering, pp. 35–46. Springer (2002)
8. Easter, Robert Benjamin and Hitzer, Eckhard: Double conformal geometric algebra. Advances in Applied Clifford Algebras **27**(3), 2175–2199 (2017)
9. Faucette, W.M.: A geometric interpretation of the solution of the general quartic polynomial. The American Mathematical Monthly **103**(1), 51–57 (1996)
10. Fulton, W.: Algebraic curves, vol. 54. Addison-Wesley (2008)
11. Goldman, R., Mann, S.: R(4, 4) As a Computational Framework for 3-Dimensional Computer Graphics. Advances in Applied Clifford Algebras **25**(1), 113–149 (2015)
12. Gregory, A.L., Lasenby, J., Agarwal, A.: The elastic theory of shells using geometric algebra. Royal Society open science **4**(3), 170,065 (2017)
13. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge books online. Cambridge University Press (2003)
14. Heck, A.: A Bird's-Eye View of Gröbner Bases, pp. 697–746. Springer New York, New York, NY (2003)
15. Hestenes, D.: The zitterbewegung interpretation of quantum mechanics. Foundations of Physics **20**(10), 1213–1232 (1990)
16. Hestenes, D.: New foundations for classical mechanics, vol. 15. Springer Science & Business Media (2012)
17. Hrdina, J., Návrat, A., Vašík, P.: Geometric algebra for conics. Advances in Applied Clifford Algebras **28**, 1–21 (2018)
18. Kirillov, A.A.: Elements of the Theory of Representations. Springer Berlin Heidelberg (1976)
19. Perwass, C.: Geometric algebra with applications in engineering, *Geometry and Computing*, vol. 4. Springer (2009)
20. Richter-Gebert, J.: Perspectives on projective geometry (2011)
21. Ruhe, D., Gupta, J.K., Keninck, S.D., Welling, M., Brandstetter, J.: Geometric clifford algebra networks. CoRR **abs/2302.06594** (2023)