

Intersection de coniques et algèbres géométriques

Clément Chomicki¹, Stéphane Breuils², Venceslas Biri¹, and Vincent Nozick¹

¹ LIGM, Univ Gustave Eiffel, CNRS, France

² LAMA, Univ Savoie Mont-Blanc, CNRS, France

Abstract. Les coniques sont couramment rencontrées dans un large éventail de disciplines, y compris l’optique, la physique et divers autres domaines. Par conséquent, la communauté des algèbres géométriques s’engage activement dans le développement d’algèbres qui permettent une représentation et manipulation efficace des coniques.

Les objets d’intersection de conique sont connus et pris en charge par des algèbres spécialisées dans la représentation des coniques, mais il n’existe pas encore de formule élégante pour extraire les points d’intersection de ceux-ci. Cet article propose une méthode d’extraction de points à partir d’une intersection de coniques à travers le concept de faisceaux. Elle sera basée sur QC2GA, la version 2D de QCGA (Quadric Conformal Geometric Algebra), que nous prouvons également être équivalente à GAC (Geometric Algebra for Conics).

Keywords: Coniques, Algèbres Géométriques, Géométrie Projective, Algèbres de Clifford, QCGA (Quadric Conformal Geometric Algebra), GAC (Geometric Algebra for Conics), Faisceaux

1 Introduction

Les algèbres géométriques (GA) sont un moyen pratique de représenter et manipuler des primitives géométriques. Ils sont utilisés en physique depuis des décennies [13,16,17] afin d’unifier et de simplifier certains modèles.

De nos jours, les algèbres géométriques sont très répandues en informatique, et se retrouvent dans divers domaines tels que les réseaux neuronaux [4,22] et l’informatique graphique - où elles peuvent être utilisées pour manipuler des primitives géométriques [7]. Pour une introduction aux algèbres géométriques, veuillez vous référer aux manuels de Perwass et Dorst [20,8].

Faire des plongements polynomiaux [20] en algèbre géométrique étant chose aisée, les algèbres géométriques et les courbes polynomiales forment une union prometteuse. C’est pourquoi plusieurs algèbres ont déjà été proposées pour représenter, transformer et intersecter plusieurs courbes et surfaces polynomiales de divers degrés.

Perwass [20] commença par une simple approche basée sur des lames de $\mathbb{G}_{5,3}$ pour représenter des coniques construite avec 5 points. Cet algèbre supporte les translations et rotations de coniques, mais était présenté comme une preuve

de concept à développer ultérieurement. Plus tard, Goldman et al. ont proposé $\mathbb{R}(4, 4)$ [12], un algèbre composé de deux bases projectives \mathbb{R}^4 et capable de supporter les surfaces quadriques (et donc les coniques en retirant une dimension). Similairement, DCGA - un algèbre introduit par Easter et al. [9] - est composé de deux bases de CGA et permet la représentation de cyclides de Darboux, qui incluent les surfaces quadriques. La faiblesse commune de ces deux algèbres est qu'elles ne permettent pas la construction de leurs objets à partir de points.

Un autre algèbre supportant les quadriques est QCGA (Quadric Conformal Geometric Algebra)[1] de Breuils et al., qui est une extension en 3D de l'algèbre proposé au départ par Perwass explorant les intersections et transformations plus avant. QCGA peut être réduit en dimension (en retirant 7 vecteur de sa base) pour construire QC2GA, un algèbre pour manipuler les coniques. Et finalement, GAC (Geometric Algebra for Conics) [18] est une proposition plus récente de Hrdina et al. , qui contrairement aux autres est spécialement dédiée aux coniques. GAC a cependant été prouvée équivalente à QC2GA [6], ce papier ne fera donc mention que de QC2GA pour la suite. Ces deux algèbres, tout comme celle de Perwass, permettent la construction de leurs objets à partir de points, et ont comme point commun la capacité d'intersecter deux coniques en un objet d'intersections de coniques, renfermant leurs points d'intersections mais sous une forme non-trivialement extractible. En plus de pouvoir extraire ces points d'intersections, il serait intéressant de pouvoir déterminer le type d'une conique, notamment s'il s'agit d'une pair de ligne, et le cas échéant pouvoir extraire ces lignes. Les coniques ont également un centre et plusieurs autres lignes et foyers pertinents qu'il serait intéressant d'extraire et de manipuler. GAC est en fait capable d'extraire des points d'intersection coniques dans certains cas très spécifiques [5], mais il serait préférable d'avoir une méthode générale qui fonctionne sur n'importe quelle intersection.

Ce papier présente une méthode d'extraction de points de n'importe quelle intersection de coniques. Il est organisé ainsi: Section 2 introduit un état de l'art de l'intersection de coniques en formalisme projectif et en algèbre géométrique. Section 3 présente QC2GA et GAC, et Section 4 apporte l'algorithme d'extraction de points des objets d'intersections de coniques en QC2GA, en utilisant un polynôme cubique et les faisceaux de coniques naturellement supportés par l'algèbre.

2 Coniques : contexte théorique

Cette section introduit plusieurs moyens de représenter des coniques et leurs propriétés.

2.1 Représentation des Coniques

On peut retracer les coniques jusqu'à la Grèce antique de Menaechmus en 380 av. J-C. Ces quelques derniers siècles, les mathématiciens ont lié ces courbes

planaires aux polynômes de degré deux à deux variables et à l'algèbre projectif. C'est le formalisme qui est utilisé dans ce papier.

Comme expliqué par Faucette [10], les coniques sont des courbes polynomiales planaires de degré 2 :

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

Avec $(a, b, c, d, e, f) \in \mathbb{K}^6$ et $\mathbb{K} = \mathbb{R}$ or \mathbb{C} . Quand non-dégénérée et non-complexe, une conique peut être vue comme l'intersection d'un double cône infini et d'un plan.

Il est courant de travailler en espace projectif \mathbb{P}^2 pour représenter les points et \mathbb{P}^5 pour les coniques.

\mathbb{P}^n est l'ensemble de toutes les classes d'équivalence $(a_i)_{i \in [1, n+1]} \in \mathbb{K}^{n+1} \setminus \{0_n\}$ sous la relation d'équivalence $(a_1, \dots, a_{n+1}) \sim (b_1, \dots, b_{n+1}) \iff \exists \lambda \in \mathbb{K}, (a_1, \dots, a_{n+1}) = \lambda(b_1, \dots, b_{n+1})$. Un point fini (x, y) de \mathbb{K}^2 est alors plongé en tant que $(x, y, 1)$ (ou plus généralement en tant que (wx, wy, w)) dans \mathbb{P}^2 , et les points infinis de direction (x, y) sont incorporés en tant que $(x, y, 0)$. Afin d'incorporer \mathbb{P}^2 dans \mathbb{P}^5 , nous considérons l'application polynomiale $Q : \mathbb{P}^2 \rightarrow \mathbb{P}^5$ telle que $Q(x, y, w) = (x^2, y^2, xy, xw, yw, w^2)$.

Les coniques sont donc représentées par les vecteurs de \mathbb{P}^5 et l'équation (1) devient alors :

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

Les coniques sont aussi souvent représentées par leur matrices hessiennes, qui permettent de reformuler l'équation (1):

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

Considérons cinq points linéairement indépendants $(p_i)_{i \in [1, 5]}$ de coordonnées $(x_i, y_i, w_i)_{i \in [1, 5]} \in \mathbb{P}^2$ et un point supplémentaire p_0 de coordonnées $(x, y, w) \in \mathbb{P}^2$. Soit $q_i = Q(p_i), \forall i \in [0, 5]$. L'équation de conique (C) est souvent exprimée sous la forme du déterminant nul $\mathcal{P} = [q_{i,j}]$ [14] :

$$C : \det(\mathcal{P}) = \begin{vmatrix} x^2 & y^2 & xy & xw & yw & w^2 \\ x_1^2 & y_1^2 & x_1 y_1 & x_1 w_1 & y_1 w_1 & w_1^2 \\ x_2^2 & y_2^2 & x_2 y_2 & x_2 w_2 & y_2 w_2 & w_2^2 \\ x_3^2 & y_3^2 & x_3 y_3 & x_3 w_3 & y_3 w_3 & w_3^2 \\ x_4^2 & y_4^2 & x_4 y_4 & x_4 w_4 & y_4 w_4 & w_4^2 \\ x_5^2 & y_5^2 & x_5 y_5 & x_5 w_5 & y_5 w_5 & w_5^2 \end{vmatrix} = 0 \quad (4)$$

Puis, si on note $m_i^j(A)$ le mineur respectif de la matrice A , i.e. le déterminant de A avec sa i -ème colone et j -ème ligne retirée. Si on omet l'exposant pour $j = 1$, i.e. $m_i = m_i^1$, on a:

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

On peut tirer une règle plus générale : l'ensemble des coniques passant par n points non-alignés est un sous-espace vectoriel de \mathbb{P}^5 de dimension $(5 - n)$ [11]. Nous pourrions alors argumenter que 4 points de contrôle formeraient au plus un sous-espace vectoriel de dimension 1, 3 points un sous-espace vectoriel de dimension 2, etc. Cela implique que la n -intersection de deux coniques C_a et C_b contient plus que leurs n points communs, car les 3 points d'une 3-intersection génèrent un sous-espace vectoriel de dimension 2 alors que les deux coniques ne créent qu'un sous-espace vectoriel de dimension 1.

2.2 Type d'une conique et cas dégénéré

Quand $\mathbb{K} = \mathbb{R}$, le type d'une conique est édicté par ces discriminants [21]:

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

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

$\Delta_2(C)$	$\Delta_3(C)$	type of C
+	$\neq 0$	Ellipse (peut être imaginaire)
0	$\neq 0$	Parabole
-	$\neq 0$	Hyperbole
+	0	Point
0	0	Deux lignes parallèles
-	0	Deux lignes sécantes

Lorsque $\mathbb{K} = \mathbb{R}$ ou \mathbb{C} , si $\Delta_3(C) = 0$, alors C est appelée **dégénérée** (ou **réductible**), ce qui veut dire qu'elle peut être factorisée en deux lignes complexes.

2.3 Faisceaux

Soient $C_a : g_a(x, y) = 0$ et $C_b : g_b(x, y) = 0$ deux coniques et $\lambda \in \mathbb{R}^*$. $C_\lambda : g_\lambda(x, y) = g_a(x, y) + \lambda g_b(x, y) = 0$ est aussi une conique car g_a sont g_b des formes linéaires de \mathbb{P}^5 et

$$C_a \cap C_b = C_a \cap C_\lambda = C_b \cap C_\lambda \quad (8)$$

Où \cap l'opérateur ensembliste d'intersection.

Cela est dû au fait que $g_\lambda(x, y, w) = g_a(x, y, w) + \lambda g_b(x, y, w)$, et donc si deux des termes sont nuls, alors le troisième l'est aussi, et si un des termes est

non-nul, alors au moins un autre est non-nul aussi. Le 1-espace vectoriel généré par C_a et C_b est appelé leur **faisceau**.

$$\text{Faisceau}(C_a, C_b) = \{K(C_a + \lambda C_b) \mid K, \lambda \in \mathbb{K}\} \quad (9)$$

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

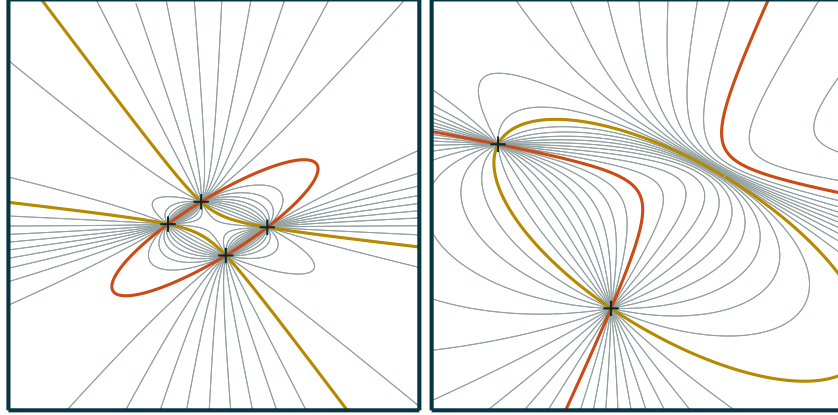


Fig. 1. Le faisceau (en gris) généré par deux coniques (en rouge et jaune) avec 4 points d'intersection (à gauche) et 2 points d'intersections (à droite).

Quatre points non-alignés génèrent aussi un faisceau de conique, mais la réciproque n'est pas vraie: l'intersection de deux coniques peut contenir moins de 4 points (lorsque $\mathbb{K} = \mathbb{R}$) ou peut avoir un ou plusieurs points multiples (voir Fig. 1).

2.4 Intersecter des coniques

La recherche des points d'intersection de deux conique est un problème ancien. Deux coniques différentes peuvent avoir de 0 à 4 points d'intersection (l'intersection d'une conique avec elle-même est bien sûr la conique entière). Les bases de Gröbner [15] peuvent être utilisées pour exprimer le problème d'intersection conique sous la forme d'un polynôme quartique. De plus, Faucette [10] décrit une méthode pour résoudre n'importe quel polynôme quartique en trouvant les points d'intersection de deux coniques. Par conséquent, trouver les points d'intersection de deux sections coniques revient à trouver la racine d'un polynôme quartique.

La méthode décrite par Faucette [10] consiste à trouver une conique dégénérée (c'est-à-dire qui peut être factorisée en deux droites) dans le faisceau généré par les deux coniques que l'on intersecte. Cette paire de lignes est ensuite séparée en deux droites distinctes, qui peuvent être ensuite utilisées pour trouver

les points d'intersection avec l'une des coniques du faisceau. Une autre méthode consiste à résoudre directement l'équation quartique associée au problème. Richter-Geber [21] reformule cette méthode de manière plus complète et donne un processus détaillé pour trouver les intersections. Tout comme [10], ils résolvent l'équation cubique associée, trouvent une solution complexe quelconque en utilisant une formule donnée, séparent la conique complexe dégénérée obtenue en deux droites complexes et les intersecte l'une des coniques. Ces deux méthodes reposent fortement sur les nombres complexes, mais on pourrait choisir de n'utiliser que des droites réelles si on se limitait aux intersections lignes-coniques.

Maintenant que les intersections de coniques sont introduits, la sections suivante présente deux algèbres supportant les coniques.

3 Un aperçu de QC2GA et GAC

3.1 Coniques et algèbres géométriques

Revenons aux équations (4) et (5), en considérant $\mathbb{K} = \mathbb{R}$. Calculer le déterminant de la matrice \mathcal{P} peut être fait en travaillant dans $\mathbb{G}_6 = \bigwedge \mathbb{P}^5$ [20,19], où on trouve que les paramètres de la conique (a, b, c, d, e, f) sont les coordonnées de la lame de grade 5 $\bigwedge_{i \in [1,5]} q_i$. L'équation de conique (4) devient alors :

$$C : q_1 \wedge q_2 \wedge q_3 \wedge q_4 \wedge q_5 \wedge q = 0 \quad (11)$$

Ainsi le produit extérieur de cinq points nous donne une représentation naturelle de l'équation implicite des coniques. Cela mène directement à la proposition de Perwass [20] , et donc à QC2GA [1] et GAC [18]. Dans ces algèbres, les points et coniques sont représentés ainsi dans leurs bases $\mathbb{G}_{5,3}$ respectives :

$$p = x^2 e_1 + y^2 e_2 + xye_3 + xe_4 + ye_5 + e_6 \quad (12)$$

$$C = p_1 \wedge p_2 \wedge p_3 \wedge p_4 \wedge p_5 \wedge e_7 \wedge e_8 \quad (13)$$

$$= ae_1^c + be_2^c + ce_3^c + de_4^c + ee_5^c + fe_6^c \quad (14)$$

Dans (12), les e_1^c, \dots, e_6^c représentent les compléments à droite des multivecteurs e_1, \dots, e_6 , opérateur défini par $m \wedge m^c = I$ pour tout multivecteur m [3].

3.2 Two-dimensional Quadric Conformal Algebra (QC2GA)

QC2GA est la version 2D de QCGA [1] par Breuils et al. Sa signature est $\mathbb{R}^{5,3}$, mais on utilise la base non-diagonale suivante qui est bien plus pratique :

	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

Le formalisme de QC2GA repose sur les lames suivantes:

$$e_o = e_{o_1} + e_{o_2} \quad (15) \quad e_\infty = \frac{e_{\infty_1} + e_{\infty_2}}{2} \quad (16)$$

$$I_o^\triangleright = (e_{o_1} - e_{o_2}) \wedge e_{o_2} \quad (17) \quad I_\infty^\triangleright = (e_{\infty_1} - e_{\infty_2}) \wedge e_{\infty_2} \quad (18)$$

$$I_o = e_{o_1} \wedge e_{o_2} \wedge e_{o_2} \quad (19) \quad I_\infty = e_{\infty_1} \wedge e_{\infty_2} \wedge e_{\infty_3} \quad (20)$$

$$I_\epsilon = e_1 \wedge e_2 \wedge e_3 \quad (21) \quad I = I_\epsilon \wedge I_\infty \wedge I_o \quad (22)$$

Les points $\overset{\text{QC2GA}}{p}$ et coniques $\overset{\text{QC2GA}}{C}$ de QC2GA sont défini ainsi :

$$\overset{\text{QC2GA}}{p} = e_o + xe_1 + ye_2 + x^2 \frac{e_{\infty_1}}{2} + y^2 \frac{e_{\infty_2}}{2} + xye_{\infty_3} \quad (23)$$

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

$$\overset{\text{QC2GA}}{C^*} = a\left(\frac{e_{\infty_1}}{2}\right)^{-1} + b\left(\frac{e_{\infty_2}}{2}\right)^{-1} + ce_{\infty_3}^{-1} + de_1^{-1} + ee_2^{-1} + fe_o^{-1} \quad (25)$$

$$= -2ae_{o_1} - 2be_{o_2} - ce_{o_3} + de_1 + ee_2 - fe_\infty \quad (26)$$

Avec $(a, b, c, d, e, f) \in \mathbb{R}^6$ les paramètres de la conique C représentée par $\overset{\text{QC2GA}}{C}$.

QC2GA permet l'intersection de conique et l'évaluation de si un point repose sur une conique :

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

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

3.3 Geometric Algebra for Conics (GAC)

GAC est un autre algèbre géométrique fait pour les coniques [18] par J. Hrdina et al. Sa base $(e_1, e_2, \bar{n}_+, n_+, \bar{n}_-, n_-, \bar{n}_\times, n_\times)$ a la même signature que QC2GA.

Les points $\overset{\text{GAC}}{p}$ et coniques $\overset{\text{GAC}}{C}$ sont construits ainsi dans GAC:

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

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

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

L'usage de GAC est le même que celui de QC2GA.

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

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

4 Extraire des points d'intersections de coniques de QC2GA

Une intersection peut contenir de 0 à 4 points. Cependant, le degré de cet objet est toujours de 6. Par conséquent, il n'est ni trivial de distinguer le type d'intersection ni d'extraire les points, d'où notre méthode. En sélectionnant une conique dégénérée réelle à partir du faisceau de l'objet d'intersection, la conique est factorisée en deux droites. Ces droites sont ensuite intersectées avec une autre conique du faisceau de départ, ce qui donne les points d'intersection désirés. Cette méthode opère exclusivement sur des objets réels, contrairement à l'approche de Faucette qui utilise des droites complexes.

4.1 Faisceaux de QC2GA

Soit Inter une intersection de deux coniques et p un point hors de Inter , on défini $C = \text{Inter} \wedge p$ comme la conique passant par tous les points de Inter et par p . L'addition de coniques est autorisé dans QC2GA, donc on peut écrire $\forall \lambda \in \mathbb{R}, C_\lambda = C_a + \lambda C_b$. Les points peuvent aussi être additionnés, ce qui donne :

$$\text{Inter} = C_a \vee C_b = (C_a^* \wedge C_b^*)^* \quad (34)$$

$$C_a = \text{Inter} \wedge p_a \quad p_a \in C_a \setminus C_b \quad (35)$$

$$C_b = \text{Inter} \wedge p_b \quad p_b \in C_b \setminus C_a \quad (36)$$

$$C_\lambda = \text{Inter} \wedge (p_a + \lambda p_b) \quad (37)$$

Le faisceau de C_a et C_b est donc généré par leur intersection Inter .

4.2 Méthode d'extraction

L'étape suivante consiste en l'extraction des points contenus dans l'intersection. Les deux coniques seront utilisées pour trouver une nouvelle paire de coniques, dont l'une est dégénérée, en trouvant la racine d'un polynôme cubique associé. La conique dégénérée sera ensuite factorisée en deux droites, qui seront intersectées avec l'autre conique pour obtenir les points d'intersection en utilisant un algorithme plus simple.

Extraire deux coniques de *Inter* Pour construire deux coniques à partir de *Inter*, on pourrait simplement choisir deux points au hasard et utiliser les formules de (34)-(37), mais cela pourrait conduire à des cas où p_a et p_b génèrent la même conique. Pour éviter cela, il est possible de ne choisir qu'un seul point p_a au hasard, de générer une conique C_a à partir de lui et de *Inter*, et d'extraire C_b comme l'élément normé dans le 2-sous-espace vectoriel *Inter* perpendiculaire à C_a .

$$C_a = Inter \wedge p_a \quad (38) \quad C_b = Inter \wedge C_a^c \quad (39)$$

Trouver une conique dégénérée (Algorithme 1) Maintenant que nous avons deux coniques pertinentes avec lesquelles travailler, nous aimerions trouver une conique dégénérée, car une conique dégénérée est soit un point (ce qui rend le problème trivial car nous prenons simplement ce point comme solution éventuelle), soit une paire de droites que nous traiterons comme deux droites distinctes, ce qui réduit l'intersection conique-conique à la tâche plus simple de déterminer des intersections ligne-conique. La recherche d'une conique dégénérée dans le faisceau de C_1 et C_2 peut être réalisée en résolvant l'équation $\Delta_3(C_1 + \lambda C_2) = 0$, qui peut être développée en l'équation (40)

$$\begin{aligned} & \Delta_3(C_2)\lambda^3 + \left(\frac{\Delta_3(C_1 + C_2) + \Delta_3(C_1 - C_2)}{2} - \Delta_3(C_1) \right) \lambda^2 \\ & + \left(\frac{\Delta_3(C_1 + C_2) - \Delta_3(C_1 - C_2)}{2} - \Delta_3(C_2) \right) \lambda + \Delta_3(C_1) = 0 \end{aligned} \quad (40)$$

Pour résoudre cette équation, on utilise une formule similaire à celle de Cardano, mais qui évite certaines divisions et qui permet d'obtenir très facilement la racine réelle garantie en choisissant $k = 0$.

$$\Delta_0 = b^2 - 3ac \quad \Delta_1 = 2b^3 - 9abc + 27a^2d \quad (41)$$

$$\Omega_{\pm} = \sqrt[3]{\frac{\Delta_1 \pm \sqrt{\Delta_1^2 - 4\Delta_0^3}}{2}} \quad x_k = -\frac{b + e^{i\frac{2\pi k}{3}}\Omega_+ + e^{i\frac{-2\pi k}{3}}\Omega_-}{3a} \quad (42)$$

En utilisant cette approche, on peut écrire l'Algorithme 1 qui extraient du faisceau de *Inter* une conique dégénérée et une autre conique significativement différente de la dégénérée. Les variables a, b, c, d ne sont pas les paramètres d'une conique, mais les paramètres du polynôme cubique.

Factoriser une paire de ligne (Algorithme 2) Une paire de lignes et de la forme :

$$l_1 = \cos(\alpha_1)x + \sin(\alpha_1)y + w_1 \quad (43)$$

$$l_2 = \cos(\alpha_2)x + \sin(\alpha_2)y + w_2 \quad (44)$$

$$C : Kl_1l_2 = ax^2 + by^2 + cxy + dx + ey + f = 0 \quad (45)$$

Algorithm 1: Trouve une conique dégénérée et une autre dans le faisceau de $Inter$

Function `gen_degen_and_other`
Input: $Inter$
Output: C_{deg}, C_{\perp}

do $p_a \leftarrow \text{random_point}()$ **while** $p_a \wedge Inter = 0$
 $C_a \leftarrow Inter \wedge p_a$ $C_b \leftarrow Inter \wedge (C_a^c)$
 $a \leftarrow \Delta_3(C_b)$ $b \leftarrow \frac{\Delta_3(C_a + C_b) + \Delta_3(C_a - C_b)}{2} - \Delta_3(C_a)$
 $d \leftarrow \Delta_3(C_a)$ $c \leftarrow \frac{\Delta_3(C_a + C_b) - \Delta_3(C_a - C_b)}{2} - \Delta_3(C_b)$
 $\Delta_0 \leftarrow b^2 - 3ac$ $\Delta_1 \leftarrow 2b^3 - 9abc + 27a^2d$
 $\Omega_- \leftarrow \sqrt[3]{\frac{\Delta_1 - \sqrt{\Delta_1^2 - 4\Delta_0^3}}{2}}$ $\Omega_+ \leftarrow \sqrt[3]{\frac{\Delta_1 + \sqrt{\Delta_1^2 - 4\Delta_0^3}}{2}}$
 $\lambda \leftarrow -3a$ $\mu \leftarrow b + \Omega_- + \Omega_+$
 $C_{deg} \leftarrow \lambda C_a + \mu C_b$ $C_{\perp} \leftarrow -\mu C_a + \lambda C_b$

où $K \in \mathbb{R}^*$ est une constante quelconque multiplicative venant de l'aspect projectif de la conique. Ces deux lignes peuvent être extraites via la méthode présentée par Richter-Gerbert [21]. Cette méthode est aussi utilisée par Byrtus et al. pour leur travail sur leurs cas spécifiques d'intersections [5]. L'Algorithme 2 implémente cela.

Algorithm 2: Algorithme de factorisation de paire de lignes

Function `factor_line_pair`
Input: C_{deg}
Output: $\alpha_1, \alpha_2, w_1, w_2$ // lines angles and offsets

$H \leftarrow \text{hessian_matrix}(C_{deg})$ $A \leftarrow \text{adjoint}(H)$ $i \leftarrow \arg \min_k \{A_{k,k}\}$
 $D \leftarrow \begin{bmatrix} 0 & -A_{i,2} + A_{i,1} \\ +A_{i,2} & 0 & -A_{i,0} \\ -A_{i,1} + A_{i,0} & 0 & 0 \end{bmatrix} / \sqrt{-A_{i,i}}$ $N \leftarrow H + D$
 $u_1, v_1, w_1 \leftarrow N \left[*, \arg \max_j \{N[1,j]^2 + N[2,j]^2\} \right]$
 $u_2, v_2, w_2 \leftarrow N \left[\arg \max_j \{N[j,1]^2 + N[j,2]^2\}, * \right]$
 $\alpha_1 \leftarrow \arctan2(v_1, u_1)$ $\alpha_2 \leftarrow \arctan2(v_2, u_2)$
return $\{\alpha_1, \alpha_2, w_1/\sqrt{u_1^2 + v_1^2}, w_2/\sqrt{u_2^2 + v_2^2}\}$

Ligne-conique intersection (Algorithme 3) L'idée est d'appliquer une rotation au problème pour se retrouver avec une ligne verticale (ce qui signifie un x connu), et on se retrouve avec un polynôme quadratique en y trivial.

Algorithm 3: Algorithme d'intersection ligne-conique

```

Function conic_line_inter
  Input:  $C, \theta, w$ 
  Output: points
   $R \leftarrow \text{qc2ga\_rotor}(-\theta) \quad [a, b, c, d, e, f] \leftarrow RC\tilde{R}$ 
   $x \leftarrow -w \quad \delta \leftarrow (cx + e)^2 - 4b(ax^2 + dx + f) \quad \gamma \leftarrow \frac{cx + e}{2b}$ 
   $p_0 \leftarrow \begin{pmatrix} \cos(\theta)x - \gamma \sin(\theta) \\ \sin(\theta)x + \gamma \cos(\theta) \end{pmatrix} \quad \mathbf{u} \leftarrow \frac{1}{2b} \begin{pmatrix} -\sin(\theta) \\ \cos(\theta) \end{pmatrix}$ 
  switch  $\text{sgn}(\delta)$  do
    case  $+$  do return  $\{p_0 \pm \sqrt{\delta}\mathbf{u}\}$ 
    case  $0$  do return  $\{p_0\}$ 
    case  $-$  do return  $\{\}$ 
  end

```

4.3 Algorithme Complet

En utilisant les algorithmes présentés dans les sections précédentes, on peut écrire l'Algorithme 4 qui extrait les points d'intersection d'un objet d'intersection de coniques. Cet algorithme a été implémenté en C++ avec Garamon [2] (voir <https://github.com/technolapin/qc2ga-intersection>). Figure 2 donne des résultats de cette implémentation.

Algorithm 4: Algorithme d'extraction de points

```

Function extract_pts_from_inter
  Input:  $Inter$ 
  Output: a set of points
   $C_{deg}, C_{\perp} \leftarrow \text{gen\_degen\_and\_other}(Inter)$ 
   $\alpha_1, \alpha_2, w_1, w_2 \leftarrow \text{factor\_line\_pair}(C_{deg})$ 
  return  $\text{conic\_line\_inter}(C_{\perp}, \alpha_1, w_1) \cup \text{conic\_line\_inter}(C_{\perp}, \alpha_2, w_2)$ 

```

5 Conclusion

Ce papier a présenté une méthode basée sur les algèbres géométriques pour la décomposition d'intersection conique-conique en points. Un travail plus avant est à fournir pour augmenter la part d'algèbre géométrique dans le processus.

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)

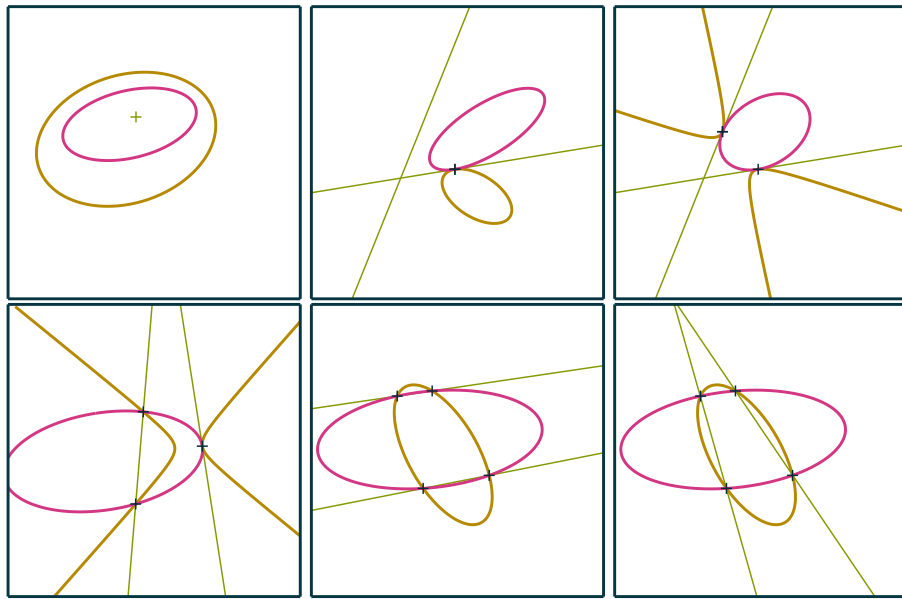


Fig. 2. Extraction de 0,1,2,3 et 4-intersections où la courbe verte est la conique dégénérée, qui peut être une paire de ligne ou un point. La 4-intersection est présentée deux fois avec deux paires de lignes.

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. Chomicki, C., Breuils, S., Biri, V., Nozick, V.: Intersection of Conic Sections Using Geometric Algebra. In: B. Sheng, L. Bi, J. Kim, N. Magnenat-Thalmann, D. Thalmann (eds.) *Advances in Computer Graphics, Lecture Notes in Computer Science*, pp. 175–187. Springer Nature Switzerland, Cham (2023). https://doi.org/10.1007/978-3-031-50078-7_14
7. 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)
8. Dorst, L.: The inner products of geometric algebra. In: *Applications of Geometric Algebra in Computer Science and Engineering*, pp. 35–46. Springer (2002)
9. Easter, Robert Benjamin and Hitzer, Eckhard: Double conformal geometric algebra. *Advances in Applied Clifford Algebras* **27**(3), 2175–2199 (2017)
10. Faucette, W.M.: A geometric interpretation of the solution of the general quartic polynomial. *The American Mathematical Monthly* **103**(1), 51–57 (1996)
11. Fulton, W.: *Algebraic curves*, vol. 54. Addison-Wesley (2008)
12. 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)
13. Gregory, A.L., Lasenby, J., Agarwal, A.: The elastic theory of shells using geometric algebra. *Royal Society open science* **4**(3), 170,065 (2017)
14. Hartley, R., Zisserman, A.: *Multiple View Geometry in Computer Vision*. Cambridge books online. Cambridge University Press (2003)
15. Heck, A.: A Bird’s-Eye View of Gröbner Bases, pp. 697–746. Springer New York, New York, NY (2003)
16. Hestenes, D.: The zitterbewegung interpretation of quantum mechanics. *Foundations of Physics* **20**(10), 1213–1232 (1990)
17. Hestenes, D.: *New foundations for classical mechanics*, vol. 15. Springer Science & Business Media (2012)
18. Hrdina, J., Návrát, A., Vašík, P.: Geometric algebra for conics. *Advances in Applied Clifford Algebras* **28**, 1–21 (2018)
19. Kirillov, A.A.: *Elements of the Theory of Representations*. Springer Berlin Heidelberg (1976)
20. Perwass, C.: Geometric algebra with applications in engineering, *Geometry and Computing*, vol. 4. Springer (2009)
21. Richter-Gebert, J.: *Perspectives on projective geometry* (2011)
22. Ruhe, D., Gupta, J.K., Keninck, S.D., Welling, M., Brandstetter, J.: Geometric clifford algebra networks. *CoRR* **abs/2302.06594** (2023)