AIMS Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 9(2): 4294–4308.
DOI: 10.3934/math.2024212
Received: 06 November 2023
Revised: 05 January 2024
Accepted: 09 January 2024
Published: 16 January 2024

Research article

Detecting isometries and symmetries of implicit algebraic surfaces

Uğur Gözütok[1,*] **and Hüsnü Anıl Çoban**[2]

[1] Department of Natural Sciences, National Defence University, İstanbul, Türkiye
[2] Department of Mathematics, Karadeniz Technical University, Trabzon, Türkiye

* **Correspondence:** Email: ugozutok@hho.msu.edu.tr.

Abstract: We presented a new and complete algorithm for detecting isometries and symmetries of implicit algebraic surfaces. First, our method reduced the problem to the case of isometries fixing the origin. Second, using tools from elimination theory and polynomial factoring, we determined the desired isometries between the surfaces. We have implemented the algorithm in Maple to provide evidences of the efficiency of the method.

Keywords: implicit surfaces; isometries; symmetries; equivalences; computer algebra
Mathematics Subject Classification: 14Q10, 14Q20

1. Introduction

Computing equivalences and symmetries of geometric objects in a specific geometric setup, e.g., projective, affine, or Euclidean, plays a significant role in computer vision, computer graphics, pattern recognition, and computer-aided geometric design. Specifically, the studies addressing the problems regarding algebraic curves and surfaces have recently gained particular interest. Some of these studies are [1–3, 5, 7, 9–12, 15, 17] for both rational and implicit curves, and [4, 6, 8, 9, 18, 19] for both rational and implicit surfaces.

In this paper, similar to the above studies, we aim to address the problem of computing isometries between two implicit surfaces. An isometry is a transformation that preserves the metric properties of the underlying space. We refer the reader to [13] for an extensive account regarding isometries. We call two surfaces isometric or congruent if one of the surfaces is the image of the other under an isometry. Also, we call a surface symmetric if there exists an isometry, other than the identity, which leaves the surface invariant.

For rational surfaces, there are certain approaches [4, 8, 18] addressing the detection of isometries. However, to the best of our knowledge, there is only one algorithm [9] regarding the isometries of implicit surfaces. In [9], the authors present a comprehensive approach to compute projective and affine equivalences, starting from projective equivalences between finite sets of points, possibly given as roots of univariate polynomials. However, the approach in [9] leads to solving multivariate polynomial equations, something that we avoid here. Also, in a very recent paper [6], a novel approach concerning symmetries of implicit surfaces is provided. In [6], the authors present a complete algorithm to compute four types of notable symmetries, namely, rotational, axial, reflectional, and central symmetries. Nonetheless, the method given in [6] is not global, i.e., it requires a separate computation for each type of symmetry. On the contrary, when applied to just one surface, in which case we determine its symmetries, our method can determine all symmetries of a surface in one go.

The two main features of the method presented in this paper are the reduction to isometries fixing the origin, which allows us to transfer the problem to polynomials of smaller degree, namely some homogeneous forms of the resulting polynomials, and the use of polynomial factoring and gcd computation as an alternative to polynomial system solving. The reduction step employs a previous idea already provided in [6, 7], where the problem is reduced to computing transformations fixing the origin. In some nongeneric bad cases, it is not possible to perform the reduction step; for those cases we provide a backup method inspired by [9]. The main algorithm is implemented in Maple and we provide extensive tests, all of which can be found publicly in the first author's website [16].

The paper is organized as follows. In Section 2, we recall some basic notions regarding isometries of surfaces, and present the statement of the problem. In Section 3, which comprises three subsections, we first explain the general strategy of the method, show how to reduce the problem to computing isometries fixing the origin, and provide a full algorithm. In Section 4, we provide results concerning extensive experimentation performed on the algorithm. If the reduction method fails, we provide a backup method in Section 5. Finally, we conclude the paper in Section 6.

2. Preliminaries

Let $\mathcal{S}_f, \mathcal{S}_g \subset \mathbb{R}^3$ be two algebraic surfaces, implicitly defined by $f(x,y,z) = 0$, $g(x,y,z) = 0$. Writing $\boldsymbol{x} = (x,y,z)^\intercal$, an isometry of the space is a mapping $\mathcal{T} : \mathbb{R}^3 \to \mathbb{R}^3$,

$$\mathcal{T}(\boldsymbol{x}) = A\boldsymbol{x} + \boldsymbol{b}, \tag{2.1}$$

where A is a 3×3 orthogonal matrix, i.e., $AA^\intercal = I$, and $\boldsymbol{b} \in \mathbb{R}^3$. We will use the notation

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}, \boldsymbol{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}, \tag{2.2}$$

so

$$\begin{aligned} \mathcal{T}(x,y,z) &= (u(x,y,z), v(x,y,z), w(x,y,z)) \\ &= (a_{11}x + a_{12}y + a_{13}z + b_1, a_{21}x + a_{22}y + a_{23}z + b_2, a_{31}x + a_{32}y + a_{33}z + b_3). \end{aligned}$$

We say that the surfaces $\mathcal{S}_f, \mathcal{S}_g$ are *congruent*, if there exists an isometry $\mathcal{T}$ such that $\mathcal{T}(\mathcal{S}_f) = \mathcal{S}_g$. Under the hypothesis that the surfaces $\mathcal{S}_f, \mathcal{S}_g$ are irreducible, this definition can be transferred to the polynomials f, g defining the surfaces, so that $\mathcal{S}_f, \mathcal{S}_g$ are congruent if, and only if,

$$f(u(x,y,z), v(x,y,z), w(x,y,z)) = \lambda g(x,y,z), \tag{2.3}$$

where $\lambda \in \mathbb{R} - \{0\}$.

As a special case, we say that the surface $\mathcal{S}_f$ is *symmetric* if there is an isometry $\mathcal{T}$ other than the identity, leaving $\mathcal{S}_f$ invariant, i.e., $\mathcal{T}(\mathcal{S}_f) = \mathcal{S}_f$. This can be stated in terms of the polynomial f as $f(\mathcal{T}(\boldsymbol{x})) = \lambda f(\boldsymbol{x})$. In this case, it can be shown that $\lambda = \pm 1$ [6].

In the rest of the paper, we will assume that none of the surfaces we are dealing with have multiple components and that they are neither cylindrical nor surfaces of revolution. Cylindrical surfaces and surfaces of revolution are the only surfaces admitting infinitely many symmetries [8], so under this last hypothesis, we can be sure that the number of symmetries between $\mathcal{S}_f, \mathcal{S}_g$ is finite: Indeed, if $\mathcal{T}_1$ and $\mathcal{T}_2$ are two isometries between $\mathcal{S}_f, \mathcal{S}_g$, then $\mathcal{T}_1 \circ \mathcal{T}_2^{-1}$ is a symmetry of $\mathcal{S}_g$ and $\mathcal{T}_2^{-1} \circ \mathcal{T}_1$ is a symmetry of $\mathcal{S}_f$. Thus, the existence of infinitely many isometries between $\mathcal{S}_f, \mathcal{S}_g$ implies that $\mathcal{S}_f, \mathcal{S}_g$ have infinitely many symmetries themselves.

Since isometries preserve the degree, $\mathcal{S}_f$ and $\mathcal{S}_g$ must have the same degree. In our case, we will assume that $\deg(\mathcal{S}_f) = \deg(\mathcal{S}_g) = n \geq 3$. Notice that for $n = 1$ where the surfaces are planes and for $n = 2$ where the surfaces are quadrics, the problem can be solved easily using linear algebra. Thus, we can summarize the problem in the following way.

Problem: *Given two implicit algebraic surfaces $\mathcal{S}_f, \mathcal{S}_g$, implicitly defined by polynomials $f(x, y, z), g(x, y, z)$ of the same degree $n \geq 3$ without multiple components, neither of them are cylindrical or a surface of revolution. Find an algorithm to check whether or not $\mathcal{S}_f, \mathcal{S}_g$ are congruent and to compute the isometries, if any, between them in the affirmative case.*

3. Detecting isometries between implicit surfaces

3.1. General strategy

Our approach consists of the following main steps.

(1) Reducing the problem to isometries fixing the origin. Isometries fixing the origin have the form $\mathcal{T}(\boldsymbol{x}) = A\boldsymbol{x}$, so compared to Eq (2.1), they have three less parameters. In order to reduce our problem to this case, we need to locate two points $\boldsymbol{P}$ and $\boldsymbol{Q}$ such that $\mathcal{T}(\boldsymbol{P}) = \boldsymbol{Q}$. By performing translations in each surface so that both $\boldsymbol{P}, \boldsymbol{Q}$ are taken to the origin, we move to $\mathcal{T}(\boldsymbol{x}) = A\boldsymbol{x}$.

(2) Eliminating λ. If $f(x, y, z)$ implicitly defines $\mathcal{S}_f$, where the degree of $f(x, y, z)$ is n, then we can write

$$f(x, y, z) = f_n(x, y, z) + f_{n-1}(x, y, z) + \cdots + f_0(x, y, z), \tag{3.1}$$

where $f_d(x, y, z)$ are the homogeneous forms of f of degree d, with $d = 0, 1, \ldots, n$; similarly for $g(x, y, z)$ and $\mathcal{S}_g$. If $\mathcal{T}(\boldsymbol{x}) = A\boldsymbol{x}$ is a congruence between $\mathcal{S}_f, \mathcal{S}_g$, we have that $\mathcal{T}(\boldsymbol{x}) = A\boldsymbol{x}$ is also a congruence between the surfaces defined by f_d, g_d, for all d, so

$$f_d(\mathcal{T}(\boldsymbol{x})) = \lambda g_d(\boldsymbol{x}) \tag{3.2}$$

for all d. λ can be written in terms of $\boldsymbol{x}$ and the entries of the matrix A and, therefore, it is eliminated.

(3) Elimination and factorization. We first determine a polynomial system in the original variables x, y, z, and the components u, v, w of the congruence $\mathcal{T}$ using (2.3). After computing a Gröbner basis to eliminate, say, u, v, we can recover the functions $w = w(x, y, z)$ as the linear factors in w of the resulting polynomial. For u, v, we proceed in a similar way.

In the first and third steps, we need to make use of two notions from Vector calculus, namely, the Laplacian and Gradient, which behave nicely under isometries. Recall that the laplacian operator $\Delta := \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$ satisfies [2]

$$\Delta(f(\mathcal{T}(\boldsymbol{x}))) = \Delta f(\mathcal{T}(\boldsymbol{x})), \tag{3.3}$$

and the gradient operator $\nabla := (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})$ satisfies [6]

$$\|\nabla f(\mathcal{T}(\boldsymbol{x}))\|^2 = \|(\nabla f)(\mathcal{T}(\boldsymbol{x}))\|^2, \tag{3.4}$$

where $\|\cdot\|$ denotes the Euclidean norm of a vector.

3.2. Reduction of the problem to isometries fixing the origin

We start by reducing the problem to finding isometries fixing the origin. In order to do this, we form two other surfaces from the original surface, namely, the surface itself and the surfaces obtained by taking the Laplacian Δf and the square of the norm $\|\nabla f\|^2$ of the gradient of the original surface. Let us denote the surfaces defined by Δf, $\|\nabla f\|^2$ and by $\mathcal{L}_f$, $\mathcal{N}_f$. Note that since $f(x, y, z)$ is, by hypothesis, not a plane, $\|\nabla f\|^2$ is not a constant. However, Δf might be a constant, in which case we replace Δf by the Hessian of f, Hf. An account regarding the invariance of the Hessian under isometries can be found in [7].

Now, using Eqs (3.3) and (3.4), we get that if two surfaces, $\mathcal{S}_f$ and $\mathcal{S}_g$, are congruent by an isometry $\mathcal{T}(\boldsymbol{x}) = A\boldsymbol{x} + \boldsymbol{b}$, then the surfaces $\mathcal{L}_f$,$\mathcal{N}_f$ and $\mathcal{L}_g$, $\mathcal{N}_g$ are also mutually congruent by the same isometry $\mathcal{T}$. Using this fact, we conclude that the intersection points of the systems $\{\mathcal{S}_f, \mathcal{L}_f, \mathcal{N}_f\}$ and $\{\mathcal{S}_g, \mathcal{L}_g, \mathcal{N}_g\}$ are congruent by $\mathcal{T}$. Assume that the set of intersection points of the system $\{\mathcal{S}_f, \mathcal{L}_f, \mathcal{N}_f\}$ is $\mathcal{P} := \{P_i\}_{i=1}^m$ and that the set of intersection points of the system $\{\mathcal{S}_g, \mathcal{L}_g, \mathcal{N}_g\}$ is $\mathcal{Q} := \{Q_i\}_{i=1}^m$. Then, since $\mathcal{T}$ is an isometry and $\mathcal{T}(\mathcal{P}) = \mathcal{Q}$, the barycenters of $\mathcal{P}$ and $\mathcal{Q}$ are also congruent by $\mathcal{T}$. Denote the barycenters of the intersection points of the above systems by $\boldsymbol{P}$ and $\boldsymbol{Q}$, respectively. Thus, we have the following result.

Lemma 3.1. *If $\mathcal{S}_f, \mathcal{S}_g$ are congruent by $\mathcal{T}$, then $\boldsymbol{P}$ and $\boldsymbol{Q}$ are also congruent by $\mathcal{T}$.*

Let us provide a method to determine $\boldsymbol{P}$ and $\boldsymbol{Q}$ efficiently. We explain the method only for $\boldsymbol{P}$, as the computation for $\boldsymbol{Q}$ is the same. First, we need to compute a Gröbner basis for the system $\{f, \Delta f, \|\nabla f\|^2\}$ using a lexicographic order with $z > y > x$ in order to eliminate the variables y, z. We denote the first element of the basis by $p_1(x) = \sum_{i=0}^{m_1} \alpha_i x^i$. Since the first coordinates of the intersection points are the roots of $p_1(x)$, using Cardano-Vieta's formulae, we can determine the first coordinate of $\boldsymbol{P}$ as $-\frac{\alpha_{m_1-1}}{m_1\alpha_{m_1}}$. We then choose a bivariate polynomial $q(x, y)$ in x, y in the basis and set $p_2(y) = Res_x(p_1(x), q(x, y)) := \sum_{i=0}^{m_2} \beta_i y^i$. Similarly, the second coordinate of $\boldsymbol{P}$ can be found as $-\frac{\beta_{m_2-1}}{m_2\beta_{m_2}}$ and the last one can be found by a Gröbner basis of $\{f, \Delta f, \|\nabla f\|^2, p_1(x), p_2(y)\}$ using a lexicographic order with $x > y > z$. We denote the first element of the last basis by $p_3(z) := \sum_{i=0}^{m_3} \gamma_i z^i$, then the third coordinate of $\boldsymbol{P}$ is $-\frac{\gamma_{m_3-1}}{m_3\gamma_{m_3}}$. Finally, we get

$$\boldsymbol{P} = -\left(\frac{\alpha_{m_1-1}}{m_1\alpha_{m_1}}, \frac{\beta_{m_2-1}}{m_2\beta_{m_2}}, \frac{\gamma_{m_3-1}}{m_3\gamma_{m_3}}\right). \tag{3.5}$$

Remark 3.1. *According to the extension Theorem [14], one needs to check whether or not the roots obtained by eliminating y and z can be extended. To do this, we need to extend these solutions to x,y and then to x,y,z. The key question is whether or not the leading coefficients vanish. If not, then the solution can be extended.*

Thus, whenever the intersection of the systems $\{\mathcal{S}_f, \mathcal{L}_f, \mathcal{N}_f\}$ and $\{\mathcal{S}_g, \mathcal{L}_g, \mathcal{N}_g\}$ is nonempty and finite, we can determine the barycenters $\boldsymbol{P}$ and $\boldsymbol{Q}$ efficiently. If the intersection is either empty or not finite, we say that we fall in a `FAIL` case. We will provide an alternative method for these bad cases in Section 5.

Now let us assume that we are not in a fail case. Once we find $\boldsymbol{P}$ and $\boldsymbol{Q}$, we apply translations $\boldsymbol{x}+\boldsymbol{P}$ and $\boldsymbol{x}+\boldsymbol{Q}$ to the surfaces $\mathcal{S}_f$ and $\mathcal{S}_g$ so that $\boldsymbol{P}$ and $\boldsymbol{Q}$ are moved to the origin. These translated surfaces are congruent by $\mathcal{T}(\boldsymbol{x}) = A\boldsymbol{x}$. We will use the following notation for the translated implicit equations of the new surfaces:

$$\begin{aligned} F(\boldsymbol{x}) &:= f(\boldsymbol{x}+\boldsymbol{P}), \\ G(\boldsymbol{x}) &:= g(\boldsymbol{x}+\boldsymbol{Q}). \end{aligned} \tag{3.6}$$

Thus, we aim to find the isometries satisfying

$$F(u, v, w) = \lambda G(x, y, z), \tag{3.7}$$

where $(u, v, w) = (a_{11}x + a_{12}y + a_{13}z, a_{21}x + a_{22}y + a_{23}z, a_{31}x + a_{32}y + a_{33}z)$.

3.3. Determining isometries fixing the origin

The polynomials F, G in Eq (3.6) satisfy $F(\mathcal{T}(\boldsymbol{x})) = \lambda G(\boldsymbol{x})$, with $\mathcal{T}(\boldsymbol{x}) = A\boldsymbol{x}$. Denoting $A\boldsymbol{x} = (u(x,y,z), v(x,y,z), w(x,y,z))$, we get $F(u,v,w) = \lambda G(x,y,z)$. We can write F and G in terms of their homogeneous forms as $F(x,y,z) = \sum_{i=0}^{n} F_d(x,y,z)$ and $G(x,y,z) = \sum_{i=0}^{n} G_d(x,y,z)$, where F_d, G_d are homogeneous of degree i. In turn, we get $F_d(u,v,w) = \lambda G_d(x,y,z)$. Let d_0 be the smallest integer satisfying $0 \le d_0 \le n$ such that F_{d_0} and G_{d_0} are not zero. Isolating λ for d_0, we obtain $\lambda = \dfrac{F_{d_0}(u,v,w)}{G_{d_0}(x,y,z)}$. Substituting the latter in Eq (3.2) and clearing the denominators, we have equations like

$$\begin{aligned} &\Delta F_i(u,v,w) G_{d_0}(x,y,z) - F_{d_0}(u,v,w) \Delta G_i(x,y,z) = 0, \\ &\|\nabla F_i(u,v,w)\|^2 G_{d_0}^2(x,y,z) - F_{d_0}^2(u,v,w) \|\nabla G_i(x,y,z)\|^2 = 0, \end{aligned} \tag{3.8}$$

where $0 \le i \le n$. Let $\mathcal{R}$ be the polynomial system consisting of the equations of the type in Eq (3.8) with the smallest i, where the corresponding equations in Eq (3.8) are nonzero. We compute a Gröbner basis for $\mathcal{R}$ using the lexicographic order with $w > v > u$. Let us denote the first element of the basis by Φ_1, which can be considered as a polynomial in u with coefficients in $\mathbb{R}[x,y,z]$, then we have the following result, which essentially will follow from Eq (3.7) and the Elimination Theory.

Proposition 3.1. *Let $\mathcal{S}_F$ and $\mathcal{S}_G$ be congruent by $(u,v,w) = (a_{11}x + a_{12}y + a_{13}z, a_{21}x + a_{22}y + a_{23}z, a_{31}x + a_{32}y + a_{33}z)$, then $\eta_1(u) := u - (a_{11}x + a_{12}y + a_{13}z)$ is a linear factor of Φ_1.*

Let Ψ_1, Ψ_2 be the result of substituting $u := (a_{11}x + a_{12}y + a_{13}z)$ into two polynomials of $\mathcal{R}$; notice that Ψ_1, Ψ_2 are polynomials in v, w, x, y, z. Let

$$R(v) := Res_w(\Psi_1(x,y,z,v,w), \Psi_2(x,y,z,v,w)), \tag{3.9}$$

then we get the following result, which, again, follows from resultant properties and Eq (3.7).

Proposition 3.2. *Let $\mathcal{S}_F$ and $\mathcal{S}_G$ be congruent by $(u, v, w) = (a_{11}x + a_{12}y + a_{13}z, a_{21}x + a_{22}y + a_{23}z, a_{31}x + a_{32}y + a_{33}z)$, then $\eta_2(v) := v - (a_{21}x + a_{22}y + a_{23}z)$ is a linear factor of $R(v)$.*

Now, substituting the zeros of the linear factors η_1 and η_2 in Ψ_1 and Ψ_2, we get polynomials $\tilde{\Psi}_1, \tilde{\Psi}_2$ in w, x, y, z.

Proposition 3.3. *Let $\mathcal{S}_F$ and $\mathcal{S}_G$ be congruent by $(u, v, w) = (a_{11}x + a_{12}y + a_{13}z, a_{21}x + a_{22}y + a_{23}z, a_{31}x + a_{32}y + a_{33}z)$, then $\eta_3(w) := w - (a_{31}x + a_{32}y + a_{33}z)$ is a linear factor of the common gcd of $\tilde{\Psi}_1$ and $\tilde{\Psi}_2$.*

The following example illustrates these results.

Example 3.1. *Consider the surfaces implicitly defined by the following polynomials.*

$$\begin{aligned}
f(x,y,z) &= x^4 + y^4 - z^4 - \frac{1}{2}x^2 + \frac{1}{2}y^2 + 2x^2y^2 + \frac{1}{16},\\
g(x,y,z) &= x^4 - 8x^3y + 16x^3z + 8x^2y^2 + 8x^2yz + 2x^2z^2 - 8xy^3 + 16xy^2z - 8xyz^2\\
&\quad + 16xz^3 + 7y^4 + 8y^3z + 8y^2z^2 + 8yz^3 + z^4 + \frac{20}{3}x^3 - 32x^2y + 4x^2z + 32xy^2\\
&\quad - 32xyz - 4xz^2 - \frac{128}{3}y^3 - 32y^2z - 32yz^2 - \frac{20}{3}z^3 + \frac{45}{2}x^2 - 42xy + 24xz\\
&\quad + 96y^2 + 54yz + \frac{51}{2}z^2 + 15x - 96y - 33z + \frac{585}{16}.
\end{aligned}$$

The first surface is called a Cassini surface, whose plotting can be seen in Figure 1. The second surface is obtained by composing the first one with $\mathcal{T}(\boldsymbol{x}) = A\boldsymbol{x} + \boldsymbol{b}$ and multiplying the resulting polynomial by 9, where

$$A = \frac{1}{3}\begin{pmatrix} 2 & -2 & 1\\ 1 & 2 & 2\\ 2 & 1 & -2 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 1\\ -1\\ 0 \end{pmatrix}. \tag{3.10}$$

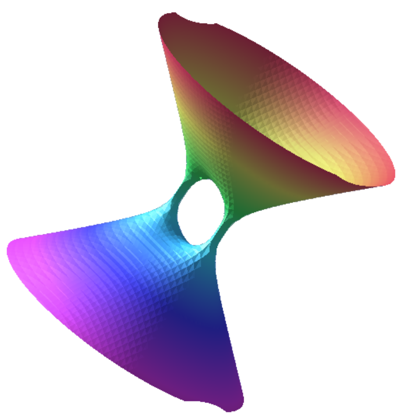


Figure 1. Cassini Surface.

Applying our algorithm, we find the barycenters $\boldsymbol{P} = (0, 0, 0)$ *and* $\boldsymbol{Q} = (-\frac{1}{3}, \frac{4}{3}, \frac{1}{3})$. *Translating* f *and* g *by* $\boldsymbol{P}$ *and* $\boldsymbol{Q}$, *we get*

$$\begin{aligned}F(x,y,z) &= x^4 + y^4 - z^4 - \frac{1}{2}x^2 + \frac{1}{2}y^2 + 2x^2y^2 + \frac{1}{16},\\ G(x,y,z) &= x^4 - 8x^3y + 16x^3z + 8x^2y^2 + 8x^2yz + 2x^2z^2 - 8xy^3 + 16xy^2z - 8xyz^2\\ &\quad + 16xz^3 + 7y^4 + 8y^3z + 8y^2z^2 + 8yz^3 + z^4 - \frac{3}{2}x^2 + 6xy + 6yz + \frac{3}{2}z^2 + \frac{9}{16}.\end{aligned}$$

Forming the system $\mathcal{R}$ *for* F *and* G *and computing a Gröbner basis, we have that*

$$\Phi_1(u) = 9u^2 - 4x^2 + 8xy - 4xz - 4y^2 + 4yz - z^2,$$

which admits two linear factors

$$\begin{aligned}\eta_{11} &= 3u + 2x - 2y + z,\\ \eta_{12} &= 3u - 2x + 2y - z.\end{aligned}$$

Substituting the above factors into $\mathcal{R}$, *we get* Ψ_1, Ψ_2. *Taking the resultant for each linear factor, we obtain* η_{1i}, $i \in \{1, 2\}$

$$\begin{aligned}\eta_{21} &= 3v - x - 2y - 2z,\\ \eta_{22} &= 3v + x + 2y + 2z,\\ \eta_{23} &= 3v - x - 2y - 2z,\\ \eta_{24} &= 3v + x + 2y + 2z.\end{aligned}$$

Substituting these factors into Ψ_1, Ψ_2, *we get* $\tilde{\Psi}_1, \tilde{\Psi}_2$. *Taking their gcd, we obtain*

$$\begin{aligned}\eta_{31} &= 3w - 2x - y + 2z,\\ \eta_{32} &= 3w + 2x + y - 2z.\end{aligned}$$

All the above solutions together yields eight general solutions with $\lambda = \frac{1}{9}$

$$\frac{1}{3}\begin{pmatrix}2 & -2 & 1\\ 1 & 2 & 2\\ 2 & 1 & -2\end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix}1\\ -1\\ 0\end{pmatrix}; \quad \frac{1}{3}\begin{pmatrix}-2 & 2 & -1\\ 1 & 2 & 2\\ 2 & 1 & -2\end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix}-1\\ -1\\ 0\end{pmatrix};$$

$$\frac{1}{3}\begin{pmatrix}2 & -2 & 1\\ -1 & -2 & -2\\ 2 & 1 & -2\end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix}1\\ 1\\ 0\end{pmatrix}; \quad \frac{1}{3}\begin{pmatrix}-2 & 2 & -1\\ -1 & -2 & -2\\ 2 & 1 & -2\end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix}-1\\ 1\\ 0\end{pmatrix};$$

$$\frac{1}{3}\begin{pmatrix}2 & -2 & 1\\ 1 & 2 & 2\\ -2 & -1 & -2\end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix}1\\ -1\\ 0\end{pmatrix}; \quad \frac{1}{3}\begin{pmatrix}-2 & 2 & -1\\ 1 & 2 & 2\\ -2 & -1 & 2\end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix}-1\\ -1\\ 0\end{pmatrix};$$

$$\frac{1}{3}\begin{pmatrix}2 & -2 & 1\\ -1 & -2 & -2\\ -2 & -1 & 2\end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix}1\\ 1\\ 0\end{pmatrix}; \quad \frac{1}{3}\begin{pmatrix}-2 & 2 & -1\\ -1 & -2 & -2\\ -2 & -1 & 2\end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix}-1\\ 1\\ 0\end{pmatrix}.$$

Note that while the computation of the barycenters took 0.063 *seconds, it took* 0.125 *seconds to compute the whole example.*

IsoSurf

Input: Two surfaces $\mathcal{S}_f, \mathcal{S}_g$ defined by polynomials f, g without multiple factors, which are neither cylindrical nor surfaces of revolution.

Output: The isometries between $\mathcal{S}_f, \mathcal{S}_g$, or the text `The surfaces are not congruent`.

procedure IsoSurf(f, g)
  Compute the barycenters $\boldsymbol{P}$ and $\boldsymbol{Q}$.
  if It fails to compute $\boldsymbol{P}$ and $\boldsymbol{Q}$ **then**
    return `FAIL`
  Assing $L := \emptyset$ and $M := \emptyset$
  Assign $F := f(\boldsymbol{x} + \boldsymbol{P})$ and $G := g(\boldsymbol{x} + \boldsymbol{P})$
  Form the system $\mathcal{R}$ using F and G
  Compute a Gröbner basis $\mathcal{G}$ for $\mathcal{R}$
  Compute $\Phi_1(u)$ using $\mathcal{G}$
  Compute the linear factors $\eta_1 = u - a_{11}x + a_{12}y + a_{13}z$ of $\Phi_1(u)$
  for each linear factor η_1 **do**
    Compute $R(v)$ using Eq (3.9)
    Compute the linear factors $\eta_2 = v - a_{21}x + a_{22}y + a_{23}z$ of $R(v)$
    for each linear factor η_2 **do**
      Compute $\tilde{\Psi}_1$ and $\tilde{\Psi}_2$ substituting $v = a_{21}x + a_{22}y + a_{23}z$
      Compute the linear factors $\eta_3 = w - a_{31}x + a_{32}y + a_{33}z$ from the gcd of $\tilde{\Psi}_1$ and $\tilde{\Psi}_2$
      for each linear factor η_3 **do**
        Check if

$$u = a_{11}x + a_{12}y + a_{13}z,$$
$$v = a_{21}x + a_{22}y + a_{23}z,$$
$$w = a_{31}x + a_{32}y + a_{33}z,$$

satisfy $F(u, v, w) = \lambda G(x, y, z)$

In the affirmative case, append (u, v, w) to L.
  if $L \neq \emptyset$ **then**
    for each (u, v, w) in L **do**
      Solve the system $f(u + b_1, v + b_2, w + b_3) = \lambda g(x, y, z)$ for b_1, b_2, b_3
      if there is no solution **then**
        next
      else
        Append $[(u, v, w), \lambda, (b_1, b_2, b_3)]$ to M
    return M
  else
    return The surfaces are not congruent.

4. Experimentation.

The algorithm IsoSurf was implemented in the computer algebra system Maple™ [20] and was tested on a PC with a 2.4 GHz Intel Core i5 processor and 8 GB RAM. Technical details, examples and source codes of the procedures are provided in the first author's personal website [16].

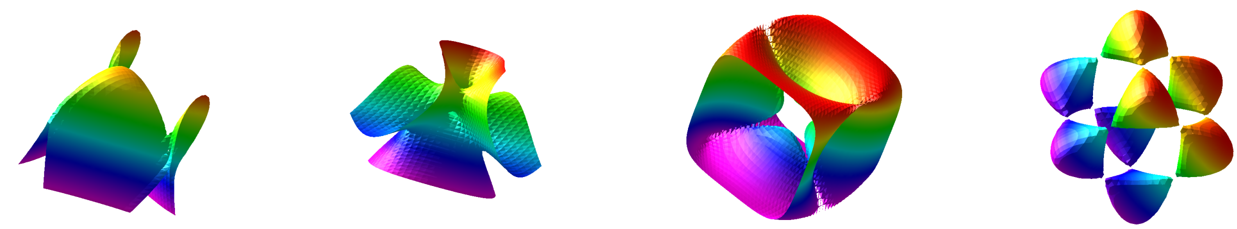
In all of our tests and examples, in order to generate congruent surfaces, we take a surface $\mathcal{S}_f$ defined by the implicit equation $f(x, y, z) = 0$ and apply to $\mathcal{S}_f$ the isometry in Eq (3.10) (see Section 3.3), so that $\mathcal{S}_g$ is the surface implicitly defined by $g(x, y, z) = 0$, with $g = f \circ \mathcal{T}$.

We start by surfaces whose implicit equations are provided in Table 1.

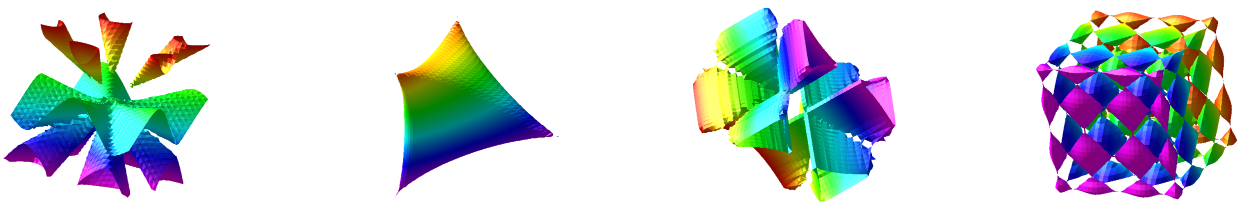
Plots of the surfaces given in Table 1 are presented in Figure 2.

Name	Implicit Eq
Cayley's Cubic	$2x^3 - 6xy^2 - zx^2 - zy^2 + 3x^2 + 3y^2 + z^3 - 1/3z^2 + z - 1$
Clebsch Cubic	$81x^3 - (189 * y + 189 * z + 9)x^2 + (-189y^2 + (54z + 126)y - 189z^2 + 126z - 9)x + 81(-3z + \frac{1}{3} + y)(z - \frac{1}{3} + y)(-\frac{1}{3}z - \frac{1}{9} + y)$
Bohemian Dome	$-x^4 - 2x^2y^2 + 2x^2z^2 - y^4 - 2y^2z^2 - z^4 + 4y^2$
Chubs Surface	$x^4 + y^4 + z^4 - x^2 - y^2 - z^2 + \frac{1}{2}$
Togliatti Surface	$64(x - 1)(x^4 - 10x^2y^2 + 5y^4 - 4x^3 - 20xy^2 - 4x^2 - 20y^2 + 16x + 16) - 5\sqrt{5 - \sqrt{5}}(2z - \sqrt{5 - \sqrt{5}})(4x^2 + 4y^2 - 4z^2 + 1 + 3\sqrt{5})^2$
Kusner Schmitt	$x^2y^2z^2 + 3x^2y^2 + 3x^2z^2 + 3y^2z^2 - 32xyz + 9x^2 + 9y^2 + 9z^2 - 5$
Bohemian Star	$x^4y^4 + 2x^2y^6 + 2x^2y^4z^2 + y^8 + 2y^6z^2 + y^4z^4 - 4x^2y^4 - 16x^2y^2z^2 - 4y^4z^2 + 16x^2z^2$
$C8$ Surface	$64x^8 + 64y^8 + 64z^8 - 128x^6 - 128y^6 - 128z^6 + 80x^4 + 80y^4 + 80z^4 - 16x^2 - 16y^2 - 16z^2 + 2$

Table 1. The implicit equations of various surfaces.



(a) Cayley's Cubic **(b)** Clebsch Cubic **(c)** Bohemian Dome **(d)** Chubs Surface



(e) Togliatti Surface **(f)** Kusner Schmitt **(g)** Bohemian Star **(h)** $C8$ Surface

Figure 2. Plots of the surfaces given in Table 1.

In Table 2, we provide the computation times for computing all isometries and symmetries of the surfaces given in Table 1. The table reveals that computing the symmetries is quite faster than computing the isometries. The reason is that the isometries make the first polynomial denser so that the algorithm deals with a more complicated system. Still, the algorithm handles isometries in a few seconds, as seen in the table. In addition to this, notice that we choose examples so that the surfaces admit lots of isometries/symmetries, including surfaces with 48 isometries/symmetries.

Surface	♯ of Isom./Symm.	t Isom.	t Symm.
Cayley's Cubic	2	0.063	0.047
Clebsch Cubic	6	0.047	0.047
Bohemian Dome	16	0.172	0.031
Chubs Surface	48	0.359	0.063
Togliatti Surface	2	8.672	3.219
Kusner Schmitt	24	0.984	0.172
Bohemian Star	16	12.000	0.253
$C8$ Surface	48	7.891	0.485

Table 2. CPU time in seconds for isometries and symmetries of the surfaces defined by the polynomials in Table 1.

We continue with computing symmetries of surfaces implicitly defined by random dense polynomials. To guarantee that the randomly generated surfaces have several symmetries, we apply the change of variables $(x, y, z) \to (x^2, y^2, z^2)$ to the random dense polynomial $f(x, y, z)$ defining the surface. The surfaces generated by this method have exactly 8 symmetries

$$\mathcal{T}_{ijk} = \begin{pmatrix} (-1)^i & 0 & 0 \\ 0 & (-1)^j & 0 \\ 0 & 0 & (-1)^k \end{pmatrix}$$

for all $i, j, k \in \{0, 1\}$, where $\mathcal{T}_{000}$ is the trivial symmetry. We test these surfaces according to various degrees and coefficient bitsizes. Recalling that the bitsize τ of an integer m is defined to be the integer $\tau = \lceil log_2 k \rceil + 1$, we call the maximum bitsize of the coefficients of the polynomial defining a surface as the bitsize of the surface. In Table 3, we present the timings for computing symmetries of the randomly generated implicit surfaces with degrees ranging from 4 to 12 and bitsizes ranging from 4 to 64. In all of the cases, our algorithm can compute all 8 symmetries in a couple of seconds.

n	$\tau = 4$	$\tau = 8$	$\tau = 16$	$\tau = 32$	$\tau = 64$
4	0.015	0.010	0.010	0.010	0.010
6	0.016	0.016	0.015	0.016	0.015
8	0.063	0.078	0.063	0.078	0.422
10	0.125	0.125	0.281	0.703	0.453
12	1.906	2.047	2.344	3.656	7.250

Table 3. CPU time in seconds for computing symmetries of implicit surfaces implicitly defined by random dense polynomials.

5. Backup method

In this section, we provide a method similar to the one presented in [9] for computing isometries whenever the method presented in Section 3 fails; this happens when the solution set of the systems $\{\mathcal{S}_f, \mathcal{L}_f, \mathcal{N}_f\}$ and $\{\mathcal{S}_g, \mathcal{L}_g, \mathcal{N}_g\}$ is either empty or not finite. In that case, we exploit the relationship between the homogeneous forms of maximum degree of the polynomials defining the surfaces we want to compare. We will consider a projective setup.

Let $\mathcal{S}_f$ and $\mathcal{S}_g$ be two surfaces implicitly defined by $f(x, y, z)$ and $g(x, y, z)$. Let $\tilde{\mathcal{S}}_F$ and $\tilde{\mathcal{S}}_G$ be their projective closures, implicitly defined by $F(x, y, z, t)$ and $G(x, y, z, t)$, where t is the homogenization variable. If $\tilde{\mathcal{S}}_F$ and $\tilde{\mathcal{S}}_G$ are congruent, then

$$F(M\bar{\boldsymbol{x}}) = \lambda G(\bar{\boldsymbol{x}}), \tag{5.1}$$

where $\bar{\boldsymbol{x}} = (x, y, z, t)^\intercal$ and

$$M = \left(\begin{array}{c:c} A & \boldsymbol{b} \\ \hdashline \boldsymbol{0} & 1 \end{array}\right), \tag{5.2}$$

with $\boldsymbol{b} \in \mathbb{R}^3$ and A is orthogonal. Collecting F and G, with respect to the homogenization variable, we get

$$\begin{aligned} F(x, y, z, t) &= f_n(x, y, z) + f_{n-1}(x, y, z)t + \cdots + f_0(x, y, z)t^n, \\ G(x, y, z, t) &= g_n(x, y, z) + g_{n-1}(x, y, z)t + \cdots + g_0(x, y, z)t^n, \end{aligned} \tag{5.3}$$

where f_d and g_d correspond to the homogeneous forms of degree d of f and g. Let us denote

$$F(M\bar{\boldsymbol{x}}) = \tilde{f}_n(x, y, z) + \tilde{f}_{n-1}(x, y, z)t + \cdots + \tilde{f}_0(x, y, z)t^n. \tag{5.4}$$

Thus, if Eq (5.1) holds, then we have for all $0 \le d \le n$,

$$\tilde{f}_d(x, y, z) = \lambda g_d(x, y, z). \tag{5.5}$$

For $d = n$, it is clear that $\tilde{f}_n(x, y, z) = \lambda g_n(x, y, z)$ implies that $f_n(A\boldsymbol{x}) = \lambda g_n(\boldsymbol{x})$, so $f_n(x, y, z)$ and $g_n(x, y, z)$ are congruent by $\mathcal{T}(\boldsymbol{x}) = A\boldsymbol{x}$, where $\boldsymbol{x} = (x, y, z)^\intercal$. Since $\mathcal{T}$ is an isometry fixing the origin, we can develop a method similar to the method given in Section 3.3. Denoting $\mathcal{T}(\boldsymbol{x}) = (u, v, w)$, we get a system

$$\begin{aligned} f_n(u, v, w) - \lambda g_n(x, y, z) &= 0, \\ \Delta f_n(u, v, w) - \lambda \Delta g_n(x, y, z) &= 0, \\ \|\nabla f_n(u, v, w)\|^2 - \lambda^2 \|\nabla g_n(x, y, z)\|^2 &= 0. \end{aligned} \tag{5.6}$$

Here, we find another equality in λ. Let us denote

$$\Delta^k p(x, y, z) = \underbrace{\Delta \circ \Delta \circ \cdots \circ \Delta}_{k \text{ times}} p(x, y, z),$$

for a polynomial p. Consider the polynomial $\Delta^k \|\nabla f_n\|^2$. Here, since $\|\nabla f_n\|^2$ is a polynomial of degree $2n - 2$, we get, for all $k \ge n$, $\Delta^k \|\nabla f_n\|^2 = 0$. Let k_0 be the greatest integer satisfying $1 \le k_0 \le n - 1$ such that $\Delta^{k_0} \|\nabla f_n\|^2$ is not a zero polynomial. Using the second equality in system (5.6), we get that

$$\lambda^2 = \frac{\Delta^{k_0} \|\nabla f_n(u, v, w)\|^2}{\Delta^{k_0} \|\nabla g_n(x, y, z)\|^2}. \tag{5.7}$$

Substituting the above equality into the system (5.6) and clearing denominators, we get a system free from the parameter λ.

$$\begin{aligned} f_n^2(u,v,w)\Delta^{k_0}\|\nabla g_n\|^2 - g_n^2(x,y,z)\Delta^{k_0}\|\nabla f_n\|^2 &= 0,\\ (\Delta f_n)^2(u,v,w)\Delta^{k_0}\|\nabla g_n\|^2 - (\Delta g_n)^2(x,y,z)\Delta^{k_0}\|\nabla f_n\|^2 &= 0,\\ \|\nabla f_n(u,v,w)\|^2\Delta^{k_0}\|\nabla f_n\|^2 - \|\nabla g_n(x,y,z)\|^2\Delta^{k_0}\|\nabla g_n\|^2 &= 0. \end{aligned} \tag{5.8}$$

Finally, we proceed as in Section 3.3 but with replacing the system $\mathcal{R}$ by the above system. Once we determine the isometry $\mathcal{T}(\boldsymbol{x}) = A\boldsymbol{x}$, we substitute the matrix A in M defined in Eq (5.2). Therefore, it remains to determine $\boldsymbol{b}$. To do this, we use the polynomial equality corresponding $d = n-1$ in Eq 5.5, which is a linear system. Let us provide an example below to illustrate the idea.

Example 5.1. *Consider the surfaces implicitly defined by the polynomials*

$$\begin{aligned} f(x,y,z) &= x^4 + 4x^2y^2 - 2x^2z^2 + 4y^4 - 4y^2z^2 + z^4 + 2x^2 + y^2 - 1,\\ g(x,y,z) &= \frac{4}{9}x^4 - \frac{16}{9}x^3y + \frac{80}{9}x^3z + \frac{20}{3}x^2y^2 - \frac{32}{3}x^2yz + \frac{140}{3}x^2z^2 - \frac{88}{9}xy^3\\ &+ \frac{104}{3}xy^2z + \frac{200}{3}xyz^2 + \frac{200}{9}xz^3 + \frac{121}{9}y^4 + \frac{352}{9}y^3z + \frac{122}{3}y^2z^2\\ &+ \frac{160}{9}yz^3 + \frac{25}{9}z^4 - 16x^2y - 8x^2z + 32xy^2 - 144xyz - 80xz^2\\ &- 88y^3 - 172y^2z - 104yz^2 - 20z^3 + 21x^2 - 36xy + 132xz + 222y^2\\ &+ 240yz + 72z^2 + 18x - 252y - 108z + 99. \end{aligned} \tag{5.9}$$

For these surfaces the intersections of the systems $\{\mathcal{S}_f, \mathcal{L}_f, \mathcal{N}_f\}$ and $\{\mathcal{S}_g, \mathcal{L}_g, \mathcal{N}_g\}$ are empty. Thus, the algorithm IsoSurf fails to reduce the problem using barycenters $\boldsymbol{P}$ and $\boldsymbol{Q}$. Applying the method given in this section gives exactly 8 *isometries between $\mathcal{S}_f$ and $\mathcal{S}_g$. These isometries are, with $\lambda = \frac{1}{9}$,*

$$\frac{1}{3}\begin{pmatrix} 2 & -2 & 1\\ 1 & 2 & 2\\ -2 & -1 & 2 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 1\\ -1\\ 0 \end{pmatrix}; \quad \frac{1}{3}\begin{pmatrix} -2 & 2 & -1\\ 1 & 2 & 2\\ -2 & -1 & 2 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} -1\\ -1\\ 0 \end{pmatrix};$$

$$\frac{1}{3}\begin{pmatrix} 2 & -2 & 1\\ -1 & -2 & -2\\ -2 & -1 & 2 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 1\\ 1\\ 0 \end{pmatrix}; \quad \frac{1}{3}\begin{pmatrix} -2 & 2 & -1\\ -1 & -2 & -2\\ -2 & -1 & 2 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} -1\\ 1\\ 0 \end{pmatrix};$$

$$\frac{1}{3}\begin{pmatrix} 2 & -2 & 1\\ 1 & 2 & 2\\ 2 & 1 & -2 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 1\\ -1\\ 0 \end{pmatrix}; \quad \frac{1}{3}\begin{pmatrix} -2 & 2 & -1\\ 1 & 2 & 2\\ 2 & 1 & -2 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} -1\\ -1\\ 0 \end{pmatrix};$$

$$\frac{1}{3}\begin{pmatrix} 2 & -2 & 1\\ -1 & -2 & -2\\ 2 & 1 & -2 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} 1\\ 1\\ 0 \end{pmatrix}; \quad \frac{1}{3}\begin{pmatrix} -2 & 2 & -1\\ -1 & -2 & -2\\ 2 & 1 & -2 \end{pmatrix}, \quad \boldsymbol{b} = \begin{pmatrix} -1\\ 1\\ 0 \end{pmatrix}.$$

The whole computation to determine all isometries between $\mathcal{S}_f$ and $\mathcal{S}_g$ using the direct method took 0.187 seconds.

6. Conclusions

We have provided a new and computationally efficient algorithm for computing the isometries between two implicit algebraic surfaces, as well as computing symmetries when the input surfaces coincide. We have implemented the algorithm in the computer algebra system Maple to test it on various aspects. The results of the tests demonstrate that the algorithm works efficiently. Even for highly symmetric surfaces with higher degrees, the algorithm can compute all the isometries/symmetries in a few seconds.

The algorithm consisted of two main steps. The first step reduced the problem to finding isometries fixing the origin. Once we reduced the problem, the second step computed the isometries by determing their components as linear factors from polynomials originated from a Gröbner elimination process. If the reduction step failed, we used a backup alternative method, inspired by [9].

The reason why the method was efficient is that the reduction step allowed us to replace the original polynomials by polynomials of much smaller degrees, which are in fact homogeneous forms of the polynomials obtained after appropriately translating the original polynomials. The main computational tools that we employed were Gröbner basis computations, polynomial factoring and gcds. In particular, we completely avoided multivariate polynomial system solving.

One can wonder whether our ideas can be extended to detecting other equivalences between implicit surfaces, e.g., similarities, affine equivalences, or projective equivalences. For similarities, the behavior of the invariants used in this paper, namely, the gradient and the laplacian, were also good, so there might be some hope; however, for similarities, we have a new unknown, the similarity ratio, which needs to be considered. Affine or projective equivalences are more challenging, since the behavior of the invariants used in this paper were not good anymore.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

The authors would like to thank the reviewers for their valuable comments that helped us improve the earlier version of the paper. Uğur Gözütok is supported by the grant $121C421$, in the scope of 2218-National Postdoctoral Research Fellowship Program, from TUBITAK (The Scientific and Technological Research Council of Türkiye).

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. J. G. Alcázar, C. Hermoso, G. Muntingh, Symmetry detection of rational space curves from their curvature and torsion, *Comput. Aided Geom. D.*, **33** (2015), 51–65. https://doi.org/10.1016/j.cagd.2015.01.003

2. J. G. Alcázar, M. Lávička, J. Vršek, Symmetries and similarities of planar algebraic curves using harmonic polynomials, *J. Comput. Appl. Math.*, **357** (2019), 302–318. https://doi.org/10.1016/j.cam.2019.02.036

3. J. G. Alcázar, E. Quintero, Affine equivalences of trigonometric curves, *Acta Appl. Math.*, **170** (2020), 691–708. https://doi.org/10.1007/s10440-020-00354-6

4. J. G. Alcázar, E. Quintero, Affine equivalences, isometries and symmetries of ruled rational surfaces, *J. Comput. Appl. Math.*, **364** (2020), 112339. https://doi.org/10.1016/j.cam.2019.07.004

5. J. G. Alcázar, C. Hermoso, Computing projective equivalences of planar curves birationally equivalent to elliptic and hyperelliptic curves, *Comput. Aided Geom. D.*, **91** (2021), 102048. https://doi.org/10.1016/j.cagd.2021.102048

6. J. G. Alcázar, M. Lávička, J. Vršek, Computing symmetries of implicit algebraic surfaces, *Comput. Aided Geom. D.*, **104** (2023), 102221. https://doi.org/10.1016/j.cagd.2023.102221

7. J. G. Alcázar, U. Gözütok, H. A. Çoban, C. Hermoso, Detecting affine equivalences between implicit planar algebraic curves, *Acta Appl. Math.*, **182** (2022), 2. https://doi.org/10.1007/s10440-022-00539-1

8. J. G. Alcázar, C. Hermoso, Involutions of polynomially parametrized surfaces, *J. Comput. Appl. Math.*, **294** (2016), 23–38. https://doi.org/10.1016/j.cam.2015.08.002

9. M. Bizzarri, M. Làvička, J. Vršek, Computing projective equivalences of special algebraic varieties, *J. Comput. Appl. Math.*, **367** (2020), 112438. https://doi.org/10.1016/j.cam.2019.112438

10. M. Bizzarri, M. Làvička, J. Vršek, Approximate symmetries of planar algebraic curves with inexact input, *Comput. Aided Geom. D.*, **76** (2020), 101794. https://doi.org/10.1016/j.cagd.2019.101794

11. M. Bizzarri, M. Làvička, J. Vršek, Symmetries of discrete curves and point clouds via trigonometric interpolation, *J. Comput. Appl. Math.*, **408** (2021), 114124. https://doi.org/10.1016/j.cam.2022.114124

12. M. Bizzarri, M. Làvička, J. Vršek, Approximate symmetries of perturbed planar discrete curves, *Comput. Aided Geom. D.*, **96** (2022), 102115. https://doi.org/10.1016/j.cagd.2022.102115

13. H. S. M. Coxeter, *Introduction to geometry*, John Wiley & Sons, 1969.

14. D. Cox, J. Little, D. O'Shea, *Ideals, varieties and algorithms*, 2007.

15. U. Gözütok, H. A. Çoban, Y. Sağiroğlu, J. G. Alcázar, A new method to detect projective equivalences and symmetries of rational 3*D* curves, *J. Comput. Appl. Math.*, **419** (2023), 114782. https://doi.org/10.1016/j.cam.2022.114782

16. Uğur Gözütok, `https://www.ugurgozutok.com`.

17. M. Hauer, B. Jüttler, Projective and affine symmetries and equivalences of rational curves in arbitrary dimension, *J. Symb. Comput.*, **87** (2018), 68–86. https://doi.org/10.1016/j.jsc.2017.05.009

18. M. Hauer, B. Jüttler, J. Schicho, Projective and affine symmetries and equivalences of rational and polynomial surfaces, *J. Comput. Appl. Math.*, **349** (2018), 424–437. https://doi.org/10.1016/j.cam.2018.06.026

19. B. Jüttler, N. Lubbes, J. Schicho, Projective isomorphisms between rational surfaces, *J. Algebra*, **594** (2022), 571–596. https://doi.org/10.1016/j.jalgebra.2021.11.045

20. Maple™, 2021. Maplesoft, a division of Waterloo Maple Inc. Waterloo, Ontario.