Research article

Computation of symmetries of rational surfaces

Juan Gerardo Alcázar[1],*, Carlos Hermoso[1], Hüsnü Anıl Çoban[2] and Uğur Gözütok[3]

[1] Departamento de Física y Matemáticas, Universidad de Alcalá, E-28871 Madrid, Spain

[2] Department of Mathematics, Karadeniz Technical University, Trabzon, Türkiye

[3] Department of Natural Sciences, National Defence University, Istanbul, Türkiye

* **Correspondence:** Email: juange.alcazar@uah.es.

Abstract: In this paper, we provided, first, a general symbolic algorithm for computing the symmetries of a given rational surface, based on the classical differential invariants of surfaces, i.e., *Gauss curvature* and *mean curvature*. In practice, the algorithm works well for sparse parametrizations (e.g., toric surfaces) and PN surfaces. Additionally, we provided a specific, and symbolic, algorithm for computing the symmetries of ruled surfaces. This algorithm works extremely well in practice, since the problem is reduced to that of rational space curves, which can be efficiently solved by using existing methods. The algorithm for ruled surfaces is based on the fact, proven in the paper, that every symmetry of a rational surface must also be a symmetry of its *line of striction*, which is a rational space curve. The algorithms have been implemented in the computer algebra system Maple, and the implementations have been made public. Evidence of their performance is given in the paper.

Keywords: rational surfaces; ruled surfaces; symmetry detection; symbolic computation; differential invariants

1. Introduction

The topic addressed in this paper is the computation of the Euclidean symmetries of a certain type of algebraic object, namely rational surfaces. Thus, in the rest of this paper, a "symmetry" will mean an isometry of the ambient space that leaves the embedded surface invariant; in particular, such a mapping also leaves invariant the Gauss and mean curvatures of the surface. Capturing the information on the Euclidean symmetries of an object is a natural step if one wants to understand its geometry, visualize it accurately, or store it efficiently. Although in applied mathematics the computation of symmetries has been addressed for decades, with a very wide variety of techniques often involving numerics and statistical methods, in the past fifteen years, there has been a growing interest in applying methods from symbolic computation to this problem. In this sense, in applied mathematics the object to be

analyzed is usually a 2D or 3D image, and in practice often a 2D or 3D point cloud, and one seeks *approximate* symmetries, rather than *exact* ones. When the problem is seen from the point of view of symbolic computation, which is the perspective we use in this paper, one is attracted toward exact objects with more structure, and looks for their exact symmetries. Typically, the objects considered are (algebraic) varieties defined by polynomials with integer coefficients, either implicitly given, i.e., as zeros of a given set of polynomials, or defined by means of a parametrization with rational components, i.e., quotients of polynomials.

Although implicitly defined varieties have been addressed (see [1–3] for implicit planar curves, and [4–6] for implicit surfaces), the case that has attracted more attention has been that of rational curves and surfaces, probably because the rational representation is preferred in computer-aided geometric design, a field where the problem has some relevance. For rational parametric curves, the problem is quite well-understood, and in fact a more general question, namely identifying whether or not two rational curves are affinely or projectively equivalent, has been investigated in many papers. Although a complete list would be long, some notable contributions are [7], on symmetries of space curves, and [5, 8], on projective equivalences of rational curves in any dimension; see the bibliographies of these papers for other references. Observe also that isometries and affine equivalences are special instances of projective equivalences.

However, for rational surfaces the problem is harder, both theoretically and computationally. Here we have two types of contributions, a first group providing more general algorithms, and a second group focused on special types of surfaces whose properties one can exploit. Among the first group, [9] considers involutions of parametric polynomial surfaces, [10] addresses affine and projective equivalences of rational surfaces without projective base points (although base points are not uncommon; note that rational curves do not have base points, which makes the problem easier for curves), and [11], which improves on the results of [10] and provides an algorithm for computing the projective equivalences between two rational surfaces without additional requirements. The algorithm in [11] is certainly very general, but still has some drawbacks: on one hand, the problem is reduced to five possible cases, but only two of them are solved in the paper; on the other hand, the algorithm is involved (check the partial implementation in [12]) and requires a strong background of algebraic geometry, so it is not so easily accessible for readers not familiar with such a topic. Let us also mention that in [10], the input is a polynomial or rational parametrization (in projective form) of a surface, with real coefficients; in [11], the input is a rational parametrization of a surface with coefficients over an algebraically closed field.

Among the second group of papers, namely papers related to rational surfaces with special properties, one has [13], for canal surfaces, [14], for translational and minimal surfaces, and [5, 15], for ruled surfaces; in fact, [14, 15] address the more general question of computing affine equivalences, and [5] studies projective equivalences.

In this paper, we present two contributions to the problem. The first one is a general algorithm which uses the two natural invariants for surfaces, namely Gauss curvature and mean curvature, therefore generalizing to surfaces the idea of [7] for space curves. The algorithm is very easy to implement, although it can be computationally costly. The reason is that the algorithm requires one to compute the resultant of two polynomials derived from the two curvatures; however, while the Gauss curvature of a rational surface is always rational, the mean curvature in general has a square root, so we need to use the square of the mean curvature instead, which increases the degree. If we work with a *PN surface*, namely a rational surface where the modulus of the normal vector is rational (see for instance [16]),

we can skip that part and reduce the computational cost, so the algorithm works better; otherwise, we need very sparse parametrizations. As for a comparison with existing algorithms, compared to [10], projective base points are not a problem for our algorithm, although unlike ours, [10] also works in higher dimension. Compared to [11], our algorithm is much simpler. However, it is true that for non-sparse, non-PN surfaces, the computational cost can be high, so in the cases covered by [10, 11] and without further advantages that make our algorithm competitive (e.g., the cases already mentioned or ruled surfaces, see the next paragraph), using [10, 11] could be preferrable.

The second contribution is a specific algorithm for rational ruled surfaces. We prove that every symmetry of a rational ruled surface is a symmetry of the *line of striction* of the surface, which is a rational space curve. Thus, we reduce the problem to that of space curves, which can be efficiently solved. The resulting algorithm provides better timings than the general algorithm described in the previous paragraph and the algorithm in [15], as we will show with several examples. Compared to [5], it is worth noticing that the algorithm in [5] also reduces the problem to the case of curves. However, there are two main differences with our algorithm: (i) [5] uses rational curves in $\mathbb{P}^5$, so the dimension of the curves is higher; (ii) the algorithm in [5] requires one to solve a quadratic system of equations in 7 parameters, which may cause difficulties in some cases (see page 8 of [5]). It is true though that the algorithm in [5] can solve the more general problem of finding the projective equivalences between two ruled surfaces, while our algorithm only aims to compute the symmetries of a given surface.

The two algorithms presented in the paper have been implemented in Maple, and are available on the website of one of the authors [17]; a *Zenodo* version is also available [18].

The structure of the paper is the following. In Section 2, we provide the necessary background to develop our results. The general algorithm is presented in Section 3. The algorithm for ruled surfaces is given in Section 4, where we also analyze the special case of developable surfaces. Section 5 reports on the experimentation carried out on different types of surfaces, general and ruled, using the implementation carried out in Maple; here we also offer some comparative results with the algorithms in [10, 11]. We close with our conclusion in Section 6.

2. General background, notation, and hypotheses

Let $x(t, s)$ be a rational parametrization of a surface $S \subset \mathbb{R}^3$. We say that $x(t, s)$ is a *proper* parametrization of S if the parametrization x, seen as a rational mapping $x : \mathbb{R}^2 \to \mathbb{R}^3$, is birational onto its image, so that the inverse x^{-1} exists and is also rational. Thus, proper mappings are injective except perhaps at a subset of S of dimension at most 1. Checking whether or not a surface parametrization is proper, and reparametrizing it to be proper in the case when it is not, is not easy, but one can always heuristically check properness by taking a random point (t_0, s_0), and solving $x(t, s) = x(t_0, s_0)$: if the only solution is $t = t_0, s = s_0$, then $x(t, s)$ is proper with high probability.

The notion of properness also applies to rational parametrizations $\mathbf{c}(t)$ of a curve $C \subset \mathbb{R}^3$. In this case, however, checking properness and reparametrizing non-proper parametrizations is much easier: once all the components of $\mathbf{c}(t)$ have the same denominator, checking properness amounts to computing the gcd of the components of the numerators of $\mathbf{c}(t) - \mathbf{c}(s)$. If the degree of this gcd is 1, then $\mathbf{c}(t)$ is proper; if it is not, one can compute a proper reparametrization using the information provided by this gcd. The interested reader can read further on this topic in [19].

A *Euclidean symmetry* (also called *isometry*) of $\mathbb{R}^3$ is a mapping $f : \mathbb{R}^3 \to \mathbb{R}^3$, $f(\mathbf{x}) = A\mathbf{x} + \mathbf{b}$,

where A is an orthogonal 3×3 matrix, so $A^T A = I$ with I the identity matrix, and $\boldsymbol{b} \in \mathbb{R}^3$; notice that $\det(A) = \pm 1$. We say that such a mapping f is a *symmetry* of a surface S (resp., a curve C) if $f(S) = S$ (reap., $f(C) = C$). When we work with proper, rational parametrizations of curves or surfaces, one can reduce the computation of the symmetries of the curve or surface, if any, to the existence of birational parametrizations in the parameter space, $\mathbb{R}$ in the case of curves, and $\mathbb{R}^2$ in the case of surfaces, satisfying a certain condition; see the result that follows. For curves, we can go a bit further taking into account that the birational mappings of the real line (and in fact of the projective complex line) are the *Möbius transformations*, i.e., the mappings

$$\varphi(t) = \frac{at + b}{ct + d}, \ ad - bc \neq 0.$$

More precisely, we have the following result; see for instance [7] for a proof.

Proposition 1. *Let $C \subset \mathbb{R}^3$ be a rational curve defined by a rational, proper parametrization $\mathbf{c}(t)$, and let $f : \mathbb{R}^3 \to \mathbb{R}^3$ be a symmetry of C. Then there exists a Möbius transformation $\varphi : \mathbb{R} \to \mathbb{R}$ such that*

$$f \circ \boldsymbol{x} = \boldsymbol{x} \circ \varphi, \tag{2.1}$$

i.e., making commutative the following diagram:

$$
\begin{array}{ccc}
C & \xrightarrow{\ f\ } & C \\
{\scriptstyle \mathbf{c}}\big\uparrow & & \big\uparrow{\scriptstyle \mathbf{c}} \\
\mathbb{R} & \xrightarrow[\ \varphi\ ]{} & \mathbb{R}
\end{array}
. \tag{2.2}
$$

For properly parametrized surfaces, we also have a commutative diagram analogous to Eq (2.2), an observation used in papers like [9, 10]. Indeed, if $S \subset \mathbb{R}^3$ is a rational surface defined by a rational, proper parametrization $\boldsymbol{x}(t, s)$, and $f : \mathbb{R}^3 \to \mathbb{R}^3$ is a symmetry of S, then there exists a birational mapping $\psi : \mathbb{R}^2 \to \mathbb{R}^2$ making commutative the diagram

$$
\begin{array}{ccc}
S & \xrightarrow{\ f\ } & S \\
{\scriptstyle \boldsymbol{x}}\big\uparrow & & \big\uparrow{\scriptstyle \boldsymbol{x}} \\
\mathbb{R}^2 & \xrightarrow[\ \psi\ ]{} & \mathbb{R}^2
\end{array}
, \tag{2.3}
$$

i.e., such that

$$f \circ \boldsymbol{x} = \boldsymbol{x} \circ \psi. \tag{2.4}$$

Remark 1. In fact, Proposition 1 holds whenever we have a birational transformation, not necessarily a Euclidean symmetry, $f : \mathbb{R}^n \to \mathbb{R}^n$ mapping two rational curves C_1, C_2 properly parametrized by $\boldsymbol{x}_1, \boldsymbol{x}_2$. Similarly, the diagram in Eq (2.3) works also when we have two surfaces S_1, S_2, properly parametrized, and a birational transformation mapping one of them onto the other. The hypothesis on the properness of the parametrizations is key in both cases.

Birational mappings of the (projective) plane are called *Cremona transformations*, and are known to be generated by linear projective mappings and quadratic transformations. However, while the

birational mappings of the complex and real line are, exactly, the Möbius transformations, and therefore we have a closed form for them, no closed form is known for the Cremona transformations. This is the reason why in several papers related to the problem treated here, e.g., [9, 10, 15], one studies rational parametrizations with additional properties so that a general form for the Cremona transformations behind f can be guessed. In [9], under the considered hypotheses, the transformation is linear; in [10], it is linear projective and in [15], the first component of the transformation is a univariate Möbius mapping, while the second component is more complicated, but with a known general form. Notice that, in general, Möbius transformations or Cremona transformations do not preserve the metric.

2.1. Background on differential geometry

Let us briefly recall some notions from differential geometry that will be needed later; we refer to [20–22] for more information on this topic.

Let $x(t, s)$ be a parametrization of a surface $S \subset \mathbb{R}^3$. We say that $x(t, s)$ is *regular* at $(t_0, s_0) \in \mathbb{R}^2$ if the partial derivatives x_t, x_s are linearly independent at the point. At regular points $P = x(t, s)$, the *tangent plane* to S is well-defined and corresponds to

$$T_P(S) = P + \mathcal{L}(x_t, x_s),$$

where $\mathcal{L}(x_t, x_s)$ denotes the linear variety spanned by x_t, x_s. The normal direction to the tangent plane is therefore that of

$$N_x(t, s) = x_t \times x_s. \tag{2.5}$$

Additionally, the unitary *normal vector* is defined as

$$n_x(t, s) = \frac{x_t \times x_s}{\|x_t \times x_s\|}. \tag{2.6}$$

Notice that if $x(t, s)$ is rational, in general $n_x(t, s)$ is not rational, since a square root is expected in the denominator. Additionally, since $n_x(t, s)$ is unitary, one can also see n_x as a mapping from the surface S onto the unit sphere $\mathbb{S}^2$, i.e., $n_x : S \to \mathbb{S}^2$; this mapping is called the *Gauss map*.

The *first fundamental form* and the *second fundamental form* are two symmetric quadratic forms defined at each tangent plane $T_P(S)$ of S, with P being regular. The first fundamental form captures the metric of the surface, and is defined by the matrix

$$\mathbf{I}_x = \begin{bmatrix} E & F \\ F & G \end{bmatrix} = \begin{bmatrix} x_t \cdot x_t & x_t \cdot x_s \\ x_t \cdot x_s & x_s \cdot x_s \end{bmatrix}. \tag{2.7}$$

The second fundamental form captures the local shape of the surface around the point, and is defined by the matrix

$$\mathbf{II}_x = \begin{bmatrix} L & M \\ M & N \end{bmatrix} = \begin{bmatrix} x_{tt} \cdot n_x & x_{ts} \cdot n_x \\ x_{ts} \cdot n_x & x_{ss} \cdot n_x \end{bmatrix}. \tag{2.8}$$

The *Weingarten map* $\mathcal{W}_x$ is the (linear) mapping $\mathcal{W}_x : T_P(S) \to T_P(S)$ defined as $\mathcal{W}_x = -dn_x$, i.e., the differential of the Gauss map with a changed sign. The matrix defining $\mathcal{W}_x$ is

$$\mathbf{W}_x = \mathbf{I}_x^{-1} \mathbf{II}_x. \tag{2.9}$$

The determinant of $\mathbf{W}_x$ is the *Gauss curvature* of the surface, $\mathbf{K}_x$, and the trace of $\mathbf{W}_x$ is the *mean curvature*, $\mathbf{H}_x$. In terms of the entries of the matrices $\mathbf{I}_x$ and $\mathbf{II}_x$ (see for instance Section 3.4 of [23]),

$$\mathbf{K}_x = \frac{LN - M^2}{EG - F^2}, \quad \mathbf{H}_x = \frac{EN + GL - 2FM}{2(EG - F^2)}. \tag{2.10}$$

Observe that if $x(t, s)$ is rational, $\mathbf{K}_x$ is always a rational function. However, in general $\mathbf{H}_x$ is not rational because of the presence of a square root (due to the functions L, M, and N in the numerator of the formula for $\mathbf{H}_x$), but $\mathbf{H}_x^2$ is certainly rational.

3. A general algorithm

Let $x(t, s)$ be a rational, proper parametrization of a surface $S \subset \mathbb{R}^3$. Our goal is to provide an algorithm for computing the symmetries f of S, if any. In order to compute the symmetries f, the idea, from the diagram in Eq (2.3), is to first find the Cremona transformations ψ satisfying $f \circ x = x \circ \psi$, and then compute the f themselves. Notice that if $f(x) = Ax + b$ is a symmetry of S, Eq (2.4) yields $Ax(t, s) + b = (x \circ \psi)(t, s)$; if ψ is known, one can compute A, b by directly solving the linear system stemming from this equality, or by solving a linear system computed from $Ax(t, s) + b = (x \circ \psi)(t, s)$ after giving enough values of (t_i, s_i).

In order to find ψ, we will use the Gauss curvature $\mathbf{K}_x$ and the mean curvature $\mathbf{H}_x$ of the parametrization. The first step is to find the relationships between: (i) $\mathbf{K}_x, \mathbf{H}_x$ and $\mathbf{K}_{f \circ x}, \mathbf{H}_{f \circ x}$; (ii) $\mathbf{K}_x, \mathbf{H}_x$ and $\mathbf{K}_{x \circ \psi}, \mathbf{H}_{x \circ \psi}$. In turn, this requires one to see how $n_x, \mathbf{I}_x$ and $\mathbf{II}_x$ are affected when moving from x to $f \circ x$ or $x \circ \psi$. We start with $f \circ x$.

Lemma 2. *Let $x(t, s)$ be a rational parametrization of a surface S, and let $f(x) = Ax + b$ be a Euclidean symmetry.*

1) $\mathbf{I}_{f \circ x} = \mathbf{I}_x$.
2) $n_{f \circ x} = \det(A)An_x$.
3) $\mathbf{II}_{f \circ x} = \det(A)\mathbf{II}_x$.

Proof. To see 1), let $\tilde{x} = f \circ x$, so $\tilde{x}(t, s) = Ax(t, s) + b$. Then $\tilde{x}_t = Ax_t$ and $\tilde{x}_s = Ax_s$. Since A is orthogonal, $\tilde{x}_t \cdot \tilde{x}_t = x_t \cdot x_t, \tilde{x}_t \cdot \tilde{x}_s = x_t \cdot x_s, \tilde{x}_s \cdot \tilde{x}_s = x_t \cdot x_t$, and 1) follows. To see 2), we observe that

$$\tilde{x}_t \times \tilde{x}_s = (Ax_t) \times (Ax_s) = \det(A)A^{-T}(x_t \times x_s).$$

Since A is orthogonal, $A^{-T} = A$, and therefore $\tilde{x}_t \times \tilde{x}_s = \det(A)A(x_t \times x_s)$. Then 2) follows by using Eq (2.6), taking into account that since A is orthogonal, norms are preserved. Finally, for 3), we observe first that $\tilde{x}_{tt} = Ax_{tt}, \tilde{x}_{ts} = Ax_{ts}$ and $\tilde{x}_{ss} = Ax_{ss}$; and then the result follows from the definition of the second fundamental form and statement 2), again taking into account that since A is orthogonal, dot products are preserved.

Corollary 3. *Let $x(t, s)$ be a rational parametrization of a surface S, and let $f(x) = Ax + b$ be a Euclidean symmetry. Then*

$$\mathbf{W}_{f \circ x} = \det(A)\mathbf{W}_x, \quad \mathbf{K}_{f \circ x} = \mathbf{K}_x \text{ and } \mathbf{H}_{f \circ x} = \det(A)\mathbf{H}_x. \tag{3.1}$$

Proof. The equality $\mathbf{W}_{f \circ x} = \det(A)\mathbf{W}_x$ follows from Eq (2.9), taking into account the statements 1) and 3) of Lemma 2. The remaining equalities follow from the fact that $\mathbf{K}_{f \circ x}$ and $\mathbf{H}_{f \circ x}$ are the determinant and trace of the matrix $\mathbf{W}_{f \circ x}$.

Now let us analyze the relationship between $\mathbf{K}_x, \mathbf{H}_x$ and $\mathbf{K}_{x \circ \psi}, \mathbf{H}_{x \circ \psi}$. In order to do this, we need to recall first the relationship between the first and second fundamental forms of x and $x \circ \psi$; this is essentially known, but we include it here for the convenience of the reader. Let us denote $y := x \circ \psi$, $\psi(t, s) = (\psi_1(t, s), \psi_2(t, s))$. Using the Chain Rule,

$$
\begin{aligned}
y_t &= x_t(\psi)\frac{\partial \psi_1}{\partial t} + x_s(\psi)\frac{\partial \psi_2}{\partial t}, \\
y_s &= x_t(\psi)\frac{\partial \psi_1}{\partial s} + x_s(\psi)\frac{\partial \psi_2}{\partial s},
\end{aligned}
\tag{3.2}
$$

where y_t, y_s are evaluated at (t, s), and $x_t(\psi), x_s(\psi)$ denote the evaluations of x_t, x_s at $\psi(t, s)$. In matrix notation, we have

$$
\begin{bmatrix} y_t \\ y_s \end{bmatrix} = J^T \cdot \begin{bmatrix} x_t(\psi) \\ x_s(\psi) \end{bmatrix},
$$

where J represents the Jacobian of ψ. Taking Eq (2.7) into account, we get that

$$
\mathbf{I}_y = J^T \cdot \mathbf{I}_x(\psi) \cdot J,
$$

where, again, the entries of the first fundamental form at the left-hand side must be evaluated at (t, s), while at the right-hand side, the entries of the form are evaluated at $\psi(t, s)$. To relate n_y and n_x we also use Eq (3.2). Finally, to relate $\mathbf{II}_y$ and $\mathbf{II}_x$ we need to differentiate Eq (3.1) again, applying the Chain Rule; although the calculations are lengthy, we get the same result that we obtained for the first fundamental form. So we have the following lemma.

Lemma 4. *Let $x(t, s)$ be a rational parametrization of a surface S, let ψ be a planar rational mapping, and let J be the Jacobian of ψ.*

1) $\mathbf{I}_{x \circ \psi} = J^T \cdot \mathbf{I}_x(\psi) \cdot J$.
2) $n_{x \circ \psi} = \frac{\det(J)}{|\det(J)|} n_x(\psi)$.
3) $\mathbf{II}_{x \circ \psi} = J^T \cdot \mathbf{II}_x(\psi) \cdot J$.

Now we can prove what we need.

Corollary 5. *Let $x(t, s)$ be a rational parametrization of a surface S, let ψ be a planar birational mapping, and let J be the Jacobian of ψ. Then*

$$
\mathbf{W}_{x \circ \psi} = J^{-1} \cdot \mathbf{W}_x(\psi) \cdot J,
\tag{3.3}
$$

and

$$
\mathbf{K}_{x \circ \psi} = \mathbf{K}_x(\psi), \ \mathbf{H}_{x \circ \psi} = \mathbf{H}_x(\psi).
\tag{3.4}
$$

Proof. From Eq (2.9) and statements 1) and 3) of Lemma 4,

$$
\mathbf{W}_{x \circ \psi} = \left(J^T \cdot \mathbf{I}_x(\psi) \cdot J\right)^{-1} \left(J^T \cdot \mathbf{II}_x(\psi) \cdot J\right) = J^{-1} \cdot \mathbf{I}_x^{-1}(\psi) \cdot J^{-T} \cdot J^T \cdot \mathbf{II}_x(\psi) \cdot J,
$$

and Eq (3.3) is proved. Since from Eq (3.3) the matrices $\mathbf{W}_{x \circ \psi}, \mathbf{W}_x(\psi)$ are similar, they have the same determinant and trace, and Eq (3.4) follows.

Finally we reach the following result.

Proposition 6. *Let $x(t, s)$ be a proper parametrization of a surface S, and let $f(x) = Ax + b$ be a Euclidean symmetry of S. Then there exists a birational planar mapping ψ satisfying that $f \circ x = x \circ \psi$, and*

$$\mathbf{K}_x = \mathbf{K}_x(\psi), \ \mathbf{H}_x = \det(A)\mathbf{H}_x(\psi). \tag{3.5}$$

Proof. From the diagram in Eq (2.3), there exists a binational mapping ψ such that $f \circ x = x \circ \psi$. By Corollary 3, $\mathbf{K}_{f \circ x} = \mathbf{K}_x$, and by Eq (3.4), $\mathbf{K}_{x \circ \psi} = \mathbf{K}_x(\psi)$. Since $f \circ x = x \circ \psi$, we get the result for $\mathbf{K}_x$. For $\mathbf{H}_x$, the idea is the same.

Let us see how to derive an algorithm from Proposition 6 in order to compute the symmetries of S. First, we will use $\widehat{\mathbf{H}}_x = \mathbf{H}_x^2$, which is a rational function, instead of $\mathbf{H}_x$. Writing

$$\mathbf{K}_x(t, s) = \mathbf{K}_x(u, v), \ \widehat{\mathbf{H}}_x(t, s) = \widehat{\mathbf{H}}_x(u, v),$$

we observe that the above equalities are satisfied, under the hypotheses of Proposition 6, for $u = \psi_1(t, s)$, $v = \psi_2(t, s)$. Next $\mathbf{K}_x$ and $\widehat{\mathbf{H}}_x$ are rational functions, so after clearing denominators, the above equations lead to

$$\xi_1(t, s, u, v) = 0, \ \xi_2(t, s, u, v) = 0, \tag{3.6}$$

where ξ_1 and ξ_2 are two polynomials in the variables $t, s, u,$ and v, the first one coming from $\mathbf{K}_x$, and the second one from $\widehat{\mathbf{H}}_x$. By well-known properties of resultants, if t, s, u and v satisfy Eq (3.6) then t, s and u satisfy

$$\eta_1(t, s, u) = 0, \ \text{with } \eta_1 := \text{Res}_v(\xi_1, \xi_2), \tag{3.7}$$

and t, s and v satisfy

$$\eta_2(t, s, v) = 0, \ \text{with } \eta_2 := \text{Res}_u(\xi_1, \xi_2). \tag{3.8}$$

Now let

$$\psi_1(t, s) = \frac{\psi_{1,n}(t, s)}{\psi_{1,d}(t, s)}, \ \psi_2(t, s) = \frac{\psi_{2,n}(t, s)}{\psi_{2,d}(t, s)}, \tag{3.9}$$

where $\psi_{i,n}, \psi_{i,d}$, for $i = 1, 2$, are polynomials, and $\gcd(\psi_{i,n}, \psi_{i,d}) = 1$ for $i = 1, 2$. Then we have the following result.

Theorem 7. *Let $x(t, s)$ be a proper parametrization of a surface S, and let $f(x) = Ax + b$ be a Euclidean symmetry of S. Then there exists a binational planar mapping $\psi = (\psi_1, \psi_2)$, with components as in Eq (3.9), such that $\psi_{1,d}(t, s)u - \psi_{1,n}(t, s)$ is a factor of $\eta_1(t, s, u)$, and $\psi_{2,d}(t, s)v - \psi_{2,n}(t, s)$ is a factor of $\eta_2(t, s, v)$.*

Proof. From the diagram in Eq (2.3), there exists a birational mapping ψ such that $f \circ x = x \circ \psi$. Then for any (t, s), we have that $(t, s, \psi_1(t, s), \psi_2(t, s))$ satisfies Eq (3.6), and therefore $(t, s, \psi_1(t, s))$ fulfills Eq (3.7), and $(t, s, \psi_2(t, s))$ fulfills Eq (3.8). Hence, all the points of the surface (in the (t, s, u) space) defined by $\psi_{1,d}(t, s)u - \psi_{1,n}(t, s) = 0$ are also points of the surface $\eta_1(t, s, u) = 0$. Furthermore, since $\gcd(\psi_{1,n}, \psi_{1,d}) = 1$ the polynomial $\psi_{1,d}(t, s)u - \psi_{1,n}(t, s)$ is irreducible, so by Study's Lemma (see Section 6.13 of [24]), $\psi_{1,d}(t, s)u - \psi_{1,n}(t, s)$ divides $\eta_1(t, s, u)$. For $\psi_{2,d}(t, s)v - \psi_{2,n}(t, s)$, we argue in the same way.

Thus, if, say, $\eta_1(t, s, u)$ is not identically zero, we can look for the factors of η_1 which are linear in u, which provides the functions $\psi_1(t, s)$, substitute $u = \psi_1(t, s)$ into $\xi_1(t, s, u, v)$, $\xi_2(t, s, u, v)$, and recover v from the linear factor in v of the gcd of the polynomials in t, s, v resulting from that substitution. We might also work with $\eta_2(t, s, v)$, but we do not need to compute both resultants. Notice that in this process it is not necessary to know a priori the form of $\psi(t, s)$. Furthermore, once ψ is known, we can compute f from the equality $f \circ \boldsymbol{x} = \boldsymbol{x} \circ \psi$. This leads to an algorithm, Algorithm 1, for computing the symmetries of S. The algorithm works whenever both resultants η_1, η_2 are not identically zero, if η_1, η_2 are both identically zero then either ξ_1, ξ_2 share a factor, or some of them are identically zero. We can identify some cases where this can happen:

- *Gauss curvature being constant:* It is well-known that the only surfaces with nonzero constant Gauss curvature $\mathbf{K}$ are isometric to the sphere, if $\mathbf{K} > 0$, or to the *pseudosphere*, if $\mathbf{K} < 0$. However, the pseudosphere is a surface of revolution whose directrix is a trascendental curve, the *tractix*, so the pseudosphere is not algebraic. This implies that the only irreducible algebraic surface with constant nonzero Gaussian curvature is the sphere. However, there are algebraic, and in fact rational, surfaces with vanishing Gauss curvature; these surfaces are called *developable* surfaces, and will come up in the next section.
- *Mean curvature being constant:* These surfaces certainly exist, and among them we have the remarkable class of surfaces where $\mathbf{H} = 0$, called *minimal surfaces*. Symmetries of rational minimal surfaces are treated in [14].
- *Certain functional relationships between the Gauss curvature and the mean curvature:* An example is the case of *linear Weingarten surfaces* (see for instance [25]), which are surfaces where $a\mathbf{H} + b\mathbf{K} = c$, for constants a, b and c; this family includes as particular cases surfaces with constant Gauss or mean curvature.

In these cases the algorithm fails, and an alternative must be used; nevertheless, for developable surfaces we can use the ideas in the next section, and for minimal surfaces we can use [14].

Example 1. Consider the toy example of the ellipsoid parametrized by

$$\boldsymbol{x}(t, s) = \left(\frac{2(-s^2 - t^2 + 1)}{s^2 + t^2 + 1}, \frac{-2t}{s^2 + t^2 + 1}, \frac{8s}{s^2 + t^2 + 1} \right).$$

In this case, Algorithm 4 provides 8 symmetries, including the trivial symmetry (the identity), corresponding to the Cremona transformations:

$$\left(\frac{t}{t^2 + s^2}, \frac{-s}{t^2 + s^2} \right), \ \left(\frac{t}{t^2 + s^2}, \frac{s}{t^2 + s^2} \right), \ \left(\frac{-t}{t^2 + s^2}, \frac{-s}{t^2 + s^2} \right), \ \left(\frac{-t}{t^2 + s^2}, \frac{s}{t^2 + s^2} \right),$$

and

$$(-t, -s), \ (-t, s), \ (t, -s), \ (t, s).$$

For instance, the symmetry corresponding to the first Cremona transformation, $\left(\dfrac{t}{t^2 + s^2}, \dfrac{-s}{t^2 + s^2} \right)$, is $f(\boldsymbol{x}) = A\boldsymbol{x}$ with

$$A = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}.$$

Notice in particular that the above Cremona transformation is not linear projective, i.e., its components are not quotients of linear polynomials in t, s.

Algorithm 1 Rational Surface Symmetries

Require: A proper parametrization $x(t, s)$ of a surface $S \subset \mathbb{R}^3$.

Ensure: $\mathrm{Sym}(S)$.

 1: compute $\mathbf{K}(t, s)$, $\widehat{\mathbf{H}}(t, s)$, and the polynomials ξ_1, ξ_2
 2: compute the resultant $\eta_1(t, s, u) = \mathrm{Res}_v(\xi_1, \xi_2)$
 3: **if** η_1 is not identically zero **then**
 4:     compute the univariate factors in u of $\eta_1(t, s, u)$, and the corresponding functions $u = \psi_1^{(j)}(t, s)$
 5:     **for** each function $\psi_1^{(j)}(t, s)$ **do**
 6:         substitute $u = \psi^{(j)}(t, s)$ into ξ_1, ξ_2
 7:         compute the gcd of the polynomials, after the above substitution
 8:         compute the univariate factors in v of the gcd
 9:         compute $v = \psi_1^{(j)}(t, s)$ by solving for v in each univariate factor
10:     **end for**
11: **else**
12:     compute the resultant $\eta_2(t, s, v) = \mathrm{Res}_u(\xi_1, \xi_2)$
13:     **if** η_2 is not identically zero **then**
14:         proceed as before, with η_2
15:     **end if**
16: **end if**
17: **if** η_1, η_2 are both identically zero **then**
18:     **return** `method fails: zero resultants`
19: **end if**
20: **for** each couple $\psi^{(j)}(t, s) = (\psi_1^{(j)}(t, s), \psi_2^{(j)}(t, s))$ computed in the previous process **do**
21:     compute the symmetry f from $f \circ x = x \circ \psi^{(j)}$ and **return** f
22: **end for**

---

4. Symmetries of ruled surfaces

4.1. Background on ruled surfaces

We say that $S \subset \mathbb{R}^3$ is *ruled* if S is covered by lines, called the *rulings* of the surface. These surfaces are well-known in differential geometry; one can check classical texts like [20–22] for further reading on the topic, and also for several facts that we will be using in this subsection.

If S is rational, then we can always find [26], a rational parametrization of the form

$$x(t, s) = u(t) + sv(t), \tag{4.1}$$

where $u(t), v(t)$ parametrize space rational curves. The curve defined by $u(t)$ is called the *directrix*. We refer to Eq (4.1) as a *standard parametrization* of S, and we will assume the ruled surface S we work with to be parametrized in this way; notice that, regardless of potential reparametrizations of $u(t), v(t)$, a standard parametrization of S is not necessarily unique, since different curves may serve as directrices of the surface.

A special type of ruled surface, called *developable surfaces*, is classical in differential geometry, and widely used in applications. Intuitively speaking, developable surfaces are the surfaces that can

be "unfolded" onto the plane, so that we can find a dipheomorphic mapping $\phi : S \rightarrow \mathbb{R}^2$ preserving distances.

Developable surfaces can be characterized as the ruled surfaces with vanishing Gauss curvature. Also, if S is parametrized by Eq (4.1) then S is developable if and only if the mixed product $[\boldsymbol{u}'(t), \boldsymbol{v}(t), \boldsymbol{v}'(t)]$ is identically zero. This allows us to classify developable surfaces in three types:

(i) *Cylindrical surfaces*: Ruled surfaces where all the rulings are parallel to the same vector. Thus, they can be parametrized as

$$\boldsymbol{x}(t, s) = \boldsymbol{u}(t) + s\boldsymbol{v}_0, \tag{4.2}$$

where $\boldsymbol{v}_0$ is a constant vector.

(ii) *Conical surfaces*: These are ruled surfaces where all the rulings intersect at the same point, called the *vertex* of the surface. By applying, if necessary, a translation so that the vertex coincides with the origin, these surfaces can be parametrized as

$$\boldsymbol{x}(t, s) = s\boldsymbol{v}(t). \tag{4.3}$$

(iii) *Tangential surfaces*: Ruled surfaces consisting of the union of tangent lines to the directrix, which can be parametrized as

$$\boldsymbol{x}(t, s) = \boldsymbol{u}(t) + s\boldsymbol{u}'(t). \tag{4.4}$$

Cylindrical surfaces and conical surfaces can be recognized, for example, using the algorithm in [27]; furthermore, in those cases we can find the constant direction of the rulings, for cylindrical surfaces, and the vertex, for conical surfaces.

4.2. Computation of symmetries of ruled surfaces

In order to compute the symmetries of a ruled surface, we also need to recall the notion of a *line of striction* (see for instance Sections 3–5 of [20] for further information on this notion and its properties). Given a ruled surface S, parametrized as in Eq (4.1), the line of striction $\mathcal{E}$ is a parametric space curve $\mathbf{c}(t)$, contained in S, such that $\mathbf{c}'(t)$ is orthogonal to $\boldsymbol{v}'(t)$. The expression for the line of striction is

$$\mathbf{c}(t) = \boldsymbol{u}(t) - \frac{(\boldsymbol{v}(t) \times \boldsymbol{v}'(t)) \cdot (\boldsymbol{v}(t) \times \boldsymbol{u}'(t))}{\|\boldsymbol{v}(t) \times \boldsymbol{v}'(t)\|^2} \boldsymbol{v}(t), \tag{4.5}$$

and can be proven to be independent of the directrix considered in Eq (4.1). In particular, if Eq (4.1) is rational then $\mathbf{c}(t)$ is also rational. This expression does not make sense for cylindrical surfaces; also, for conical surfaces, the line of striction degenerates into a point. Excluding cylindrical surfaces, we can always assume that S is parametrized as in Eq (4.1), where the directrix is the line of striction. One can check that for tangential developable surfaces, the directrix appearing in Eq (4.4) is already the line of striction.

The reason why the line of striction is useful in our context is that any symmetry of the surface is also a symmetry of the line of striction. In order to see this, let the directrix $\boldsymbol{u}(t)$ be the line of striction, and let $f(\boldsymbol{x}) = A\boldsymbol{x} + \boldsymbol{b}$ be a symmetry of S. Then

$$\tilde{\boldsymbol{x}}(t, s) = (f \circ \boldsymbol{x})(t, s) = A\boldsymbol{x}(t, s) + \boldsymbol{b} = \underbrace{(A\boldsymbol{u}(t) + \boldsymbol{b})}_{\tilde{\boldsymbol{u}}(t)} + s \underbrace{A\boldsymbol{v}(t)}_{\tilde{\boldsymbol{v}}(t)} \tag{4.6}$$

also parametrizes S, and we can derive the following theorem.

Theorem 8. *Let $f(\boldsymbol{x}) = A\boldsymbol{x} + \boldsymbol{b}$ be a symmetry of a ruled surface S, and let $\mathcal{E}$ be the line of striction of S. Then f is also a symmetry of $\mathcal{E}$.*

Proof. Assume that the directrix $\boldsymbol{u}(t)$ is the line of striction $\mathcal{E}$. Let us see that the parametrization $\tilde{\boldsymbol{u}}(t)$ in Eq (4.6), which parametrizes the image of $\mathcal{E}$ under f, is also a parametrization of $\mathcal{E}$. Indeed,

$$\tilde{\boldsymbol{u}}'(t) \cdot \tilde{\boldsymbol{v}}'(t) = (A\boldsymbol{u}'(t)) \cdot (A\boldsymbol{v}'(t)).$$

Since A is an orthogonal matrix, $(A\boldsymbol{u}'(t)) \cdot (A\boldsymbol{v}'(t)) = \boldsymbol{u}'(t) \cdot \boldsymbol{v}'(t) = 0$. Thus, $\tilde{\boldsymbol{u}}(t)$ satisfies the condition defining the line of striction, parametrized by $\boldsymbol{u}(t)$, so $\tilde{\boldsymbol{u}}(t)$ also parametrizes $\mathcal{E}$. Thus $f(\mathcal{E}) = \mathcal{E}$, and the result follows.

The symmetries of the line of striction can be computed using known algorithms for space rational curves. Thus, whenever the line of striction has finitely many symmetries, we just need to test which ones are also symmetries of S. If the line of striction has infinitely many symmetries, i.e., if the line of striction is either a line or a circle, then we can use the algorithm in the previous section, or other algorithms known for ruled surfaces. Notice that cylindrical and conical surfaces are excluded in this approach.

4.3. Developable surfaces

Since developable surfaces have zero Gauss curvature, the general algorithm presented in Section 3 is not applicable to them. Also, for two of the subfamilies of developable surfaces, cylindrical and conical surfaces, using the line of striction is not an option, either.

For cylindrical surfaces, the symmetries of the surface follow from the symmetries of a normal section of the surface, i.e., the intersection of the surface with a plane Π normal to the direction $\mathbf{v}$ of the rulings. Because Eq (4.1) is linear in s, the intersection of S with a plane Π normal to the rulings is a rational planar curve, so its symmetries can be computed using known algorithms for rational curves. Notice also that cylindrical surfaces are invariant under translations by any vector parallel to $\mathbf{v}$.

For conical surfaces, parametrized as in Eq (4.3), we just need to consider symmetries fixing the origin, i.e., symmetries $f(\boldsymbol{x}) = A\boldsymbol{x}$, since in Eq (4.3) the origin is the vertex of the surface, which must be kept invariant under any symmetry. Then we can just apply the method in Section 4.1 of [15]. Notice that any symmetry of the curve defined by $\boldsymbol{v}(t)$ (see Eq (4.3)) will also be a symmetry of the surface, but the converse is false (e.g., a cone of revolution, with $\boldsymbol{v}(t)$ corresponding to an ellipse contained in the cone).

Finally, in the case of tangential developable surfaces, Theorem 8 works without any problem. However, we can go a bit further.

Proposition 9. *Let S be a rational tangential developable surface. Then $f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is a symmetry of S if and only if f is a symmetry of its line of striction.*

Proof. The implication ($\Rightarrow$) is Theorem 8, so let us address ($\Leftarrow$). Thus, let $f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, $f(\boldsymbol{x}) = A\boldsymbol{x} + \boldsymbol{b}$ be a symmetry of the line of striction of S, let $\boldsymbol{u}(t)$ be a rational parametrization of the line of striction, and let us assume without loss of generality that $\boldsymbol{u}(t)$ is proper. By Proposition 1, $A\boldsymbol{u}(t) + \boldsymbol{b} = (\boldsymbol{u} \circ \varphi)(t)$, with $\varphi(t)$ a Möbius transformation. Considering the parametrization of S provided by Eq (4.4),

$$(f \circ \boldsymbol{x})(t, s) = (A\boldsymbol{u}(t) + \boldsymbol{b}) + sA\boldsymbol{u}'(t) = (\boldsymbol{u} \circ \varphi)(t) + sA\boldsymbol{u}'(t). \tag{4.7}$$

Differentiating $A\boldsymbol{u}(t) + \boldsymbol{b} = (\boldsymbol{u} \circ \varphi)(t)$ and using the Chain Rule, we deduce that $A\boldsymbol{u}'(t) = \boldsymbol{u}'(\varphi(t))\varphi'(t)$. Substituting this into Eq (4.7), we obtain

$$(f \circ \boldsymbol{x})(t, s) = \boldsymbol{u}(\varphi(t)) + s\boldsymbol{u}'(\varphi(t))\varphi'(t). \tag{4.8}$$

The above equation can be written as

$$(f \circ \boldsymbol{x})(t, s) = \boldsymbol{u}(\varphi(t)) + \psi_2(t, s)\boldsymbol{u}'(\varphi(t)) = \boldsymbol{x}(\varphi(t), \psi_2(t, s)), \tag{4.9}$$

with $\psi_2(t, s) = s\varphi'(t)$. Eq (4.9) means, by the diagram in Eq (2.3), that f is a symmetry of S.

Observe that, for tangential surfaces, the line of striction $\boldsymbol{u}(t)$ cannot be planar, since in that case the surface S is a plane, which is a trivial case. Thus, the number of symmetries of $\boldsymbol{u}(t)$ must be finite.

5. Experimentation

In this section, we provide some experimentation on the algorithms given in the previous sections. We have implemented both algorithms in the CAS Maple [28], and tried several examples, using an Intel(R) Core(TM) $i7$ with a 3.6 GHz processor and 32 Gb RAM. All the examples and implementation can be found in the last author's personal website [17], and also in [18]. We start with the general algorithm described in Section 3, and then move to ruled surfaces and the algorithm given in Section 4.

5.1. General algorithm

In order to test the algorithm for the general case, we consider several surfaces of three different types: *toric* surfaces, *PN surfaces*, and *conoids*; in all these cases, the surfaces have projective base points, so the algorithm in [10] is not applicable. Additionally, we have also considered some examples from the papers [10, 11]. As an overall comment, let us mention that most of the time is consumed by the computation of the resultant in Step 2 of Algorithm 1, which is the bottleneck of the algorithm.

Torics. A toric surface is a surface parametrized by $\boldsymbol{x}(t, s) = (t^{m_1} s^{n_1}, t^{m_2} s^{n_2}, t^{m_3} s^{n_3})$, where $m_i, n_i \in \mathbb{Z}$. In Table 1, we provide the computation time (t, in seconds) for detecting symmetries of toric surfaces with various degrees.

Table 1. CPU times t (seconds) for toric parametrizations of various degrees.

Degree	Parametrization	Symmetries	t
2	$\left(t^2, \frac{t}{s}, s\right)$	8	0.125
3	$\left(\frac{1}{s^2 t}, \frac{s}{t^2}, \frac{1}{s}\right)$	4	3.391
4	$\left(t^3 s, \frac{t^3}{s}, s\right)$	12	0.156
5	$\left(t^4 s, \frac{t^5}{s}, s\right)$	4	0.188
6	$\left(t^5 s, \frac{t^2}{s}, s\right)$	4	0.094
7	$\left(t^5 s^2, t^3, s^3\right)$	12	1.594
8	$\left(t^5 s^3, t^8, s^3\right)$	12	0.094
9	$\left(s, \frac{t^2}{s}, t^9\right)$	8	0.734

PN surfaces. A PN surface is a rational surface with a rational normal vector field. In the case of PN surfaces, we replace in our algorithm the equation $\widehat{\mathbf{H}}_x(t, s) = \widehat{\mathbf{H}}_x(u, v)$ by $\mathbf{H}_x(\mathbf{t}, \mathbf{s}) = \pm \mathbf{H}_x(\mathbf{u}, \mathbf{v})$. This reduces the degree of the equations in the resultants. In order to generate nontrivial PN surfaces we use the strategy given in [16], where the authors provide a method using quaternions. First, we consider the cubic PN surface (see Figure 1b) given by the following rational parametrization.

$$\boldsymbol{x}(t, s) = \begin{pmatrix} -\frac{1}{3}s^3 - \frac{1}{3}s^2 t + \frac{7}{3}st^2 + t^3 - \frac{5}{3}s^2 + \frac{26}{3}st + \frac{25}{3}t^2 \\ -\frac{1}{3}s^3 - \frac{5}{3}s^2 t - \frac{5}{3}st^2 + t^3 - \frac{13}{3}s^2 - 10st + \frac{13}{3}t^2 - \frac{46}{3}s + \frac{2}{3}t \\ -\frac{4}{3}s^3 - 4s^2 t - \frac{4}{3}st^2 - 4t^3 - 9s^2 + \frac{14}{3}st + \frac{5}{3}t^2 - \frac{10}{3}s + \frac{26}{3}t \end{pmatrix}.$$

This PN surface has one projective base point, $[-3 : 1 : 0]$, and admits two symmetries corresponding to the Cremona transformations

$$\left(\frac{3}{5}t - \frac{4}{5}s + \frac{1}{5}, \frac{3}{5}t - \frac{4}{5}s - \frac{3}{5}\right), (t, s),$$

with symmetries

$$f_1(\boldsymbol{x}) = \begin{pmatrix} \frac{24}{25} & \frac{7}{25} & 0 \\ \frac{7}{25} & -\frac{24}{25} & 0 \\ 0 & 0 & 1 \end{pmatrix} \boldsymbol{x} + \begin{pmatrix} -\frac{98}{75} \\ \frac{686}{75} \\ 0 \end{pmatrix},$$

$$f_2(\boldsymbol{x}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \boldsymbol{x} + \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

The whole computation of this example took 0.203 second. Now, consider the cubic PN surface

(see Figure 1a) given by the following rational parametrization.

$$y(t, s) = \begin{pmatrix} -\frac{4}{13}s + \frac{1}{13}s^2 - \frac{11}{156}s^3 - \frac{3}{32}s^4 - \frac{4}{13}t - \frac{6}{13}ts - \frac{9}{52}ts^2 - \frac{7}{13}t^2 - \frac{9}{52}t^2s - \frac{3}{16}t^2s^2 - \frac{59}{156}t^3 - \frac{3}{32}t^4 \\ \frac{4}{13}s^2 - \frac{11}{52}s^3 + \frac{1}{8}s^4 + \frac{25}{52}ts^2 + \frac{4}{13}t^2 - \frac{1}{52}t^2s + \frac{1}{4}t^2s^2 + \frac{19}{52}t^3 + \frac{1}{8}t^4 \\ -\frac{4}{13}s + s^2 - \frac{17}{78}s^3 + \frac{3}{8}s^4 + \frac{28}{13}t - \frac{8}{13}ts + \frac{22}{13}ts^2 + \frac{37}{13}t^2 - \frac{4}{13}t^2s + \frac{3}{4}t^2s^2 + \frac{131}{78}t^3 + \frac{3}{8}t^4 \end{pmatrix}$$

This PN surface has two projective base points $[\mathbf{i} : 1 : 0]$ and $[-\mathbf{i} : 1 : 0]$. The surface admits two symmetries corresponding to the Cremona transformations

$$\left(-s - \frac{12}{13}, -t - \frac{12}{13}\right), (t, s),$$

with symmetries

$$f_1(\mathbf{x}) = \frac{1}{169}\begin{pmatrix} 137 & -96 & 24 \\ -96 & -119 & 72 \\ 24 & 72 & 151 \end{pmatrix}\mathbf{x} + \frac{1}{28561}\begin{pmatrix} 3840 \\ 11520 \\ -2880 \end{pmatrix},$$

$$f_2(\mathbf{x}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}\mathbf{x} + \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

The whole computation of this example took 0.125 second.
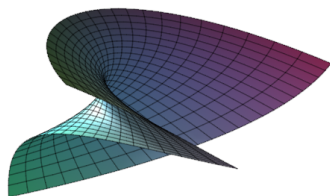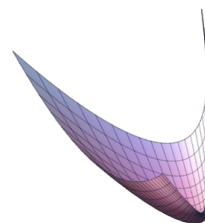
(a) Cubic PN surface   (b) Quartic PN surface

Figure 1. Plots of the PN surfaces.

Plücker's conoids. Plücker's conoid is a ruled surface defined by the following parametrization

$$\mathbf{x}(r, \theta) = (r \cos \theta, r \sin\theta, \sin 2\theta).$$

In our tests, we used a generalization of Plücker's conoid, which is again a ruled surface, defined by

$$\mathbf{x}(r, \theta) = (r \cos \theta, r \sin\theta, \sin 2n\theta),$$

where $n > 0$. Using the fact that

$$(\cos\theta, \sin\theta) \longleftrightarrow \left(\frac{1 - t^2}{1 + t^2}, \frac{2t}{1 + t^2}\right),$$

and exploiting the Chebychev polynomials of the second kind U_k, we get a rational parametrization of the generalized Plücker's conoid as

$$\boldsymbol{x}(t, s) = \left(\frac{(1 - t^2)s}{1 + t^2}, \frac{2ts}{1 + t^2}, \frac{2t}{1 + t^2}U_k\left(\frac{1 - t^2}{1 + t^2}\right)\right),$$

where $k > 0$ (see Figure 2 for the plots of the surfaces for $k \in \{1, 2, 3, 4\}$). In Table 2, we provide the computation times for detecting symmetries of the generalized Plücker's conoids with various degrees. Notice that any generalized Plücker's conoid has two projective base points, $[1 : 0 : 0]$ and $[0 : 1 : 0]$.

Table 2. CPU times t (seconds) for the generalized Plücker's conoids of various degrees.

Degree	Symmetries	t
4	16	0.360
6	8	1.203
8	16	6.407
10	8	16.812
12	16	41.500
14	8	107.391
16	16	162.469
18	8	195.578

(a) $k = 1$ **(b)** $k = 2$ **(c)** $k = 3$ **(d)** $k = 4$
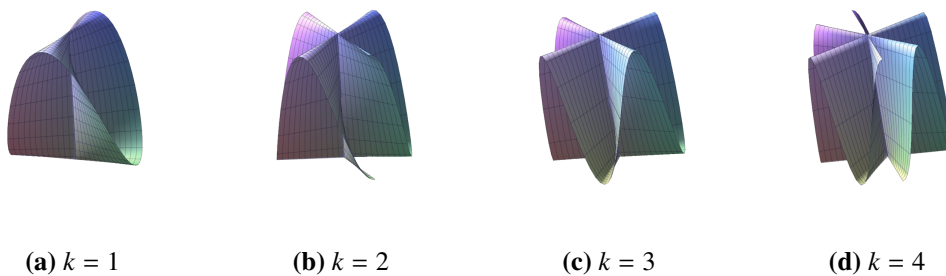
Figure 2. Generalized Plücker's conoids for different values of k.

Some examples from [10, 11].

We have tried examples of *quadratically parametrizable surfaces* considered in Section 5.1 of [10]. Following the notation in [10], they are denoted as $\Sigma_1, \ldots, \Sigma_9$, where $\Sigma_1, \ldots, \Sigma_6$ are base-point free (so the algorithm in [10] can be applied), and $\Sigma_7, \Sigma_8, \Sigma_9$ are not (and the algorithm

in [10] cannot be applied). We list the timings in Table 3. We can also find timings in [10], although in the case of [10], the algorithm finds projective mappings leaving the surface invariant, and not only isometries. For Σ_1 and Σ_2, the timings in [10] are much better. For Σ_3 to Σ_6, the timings for our algorithm and the one in [10] are comparable. Σ_7, Σ_8 and Σ_9, however, cannot be solved by [10]. In all the cases, the parametrizations have small coefficients.

Table 3. Quadratically parametrizable surfaces (see [10]).

The curve	Timing in seconds
Σ_1	50.016
Σ_2	70.344
Σ_3	0.140
Σ_4	0.235
Σ_5	0.688
Σ_6	0.001
Σ_7	1.719
Σ_8	0.053
Σ_9	0.023

We have also tried some concrete examples of quadratically parametrized surfaces appearing in [10] (see Table 6 in [10]). We provide the timings in Table 4, where the first column reproduces the classification as spelt in [10].

Table 4. Some concrete Σ_is.

Class of the curve	The curve	Timing (in seconds)
$3 - 2 - 1a$	$(t^2, s^2, t + s)$	1.243
$3 - 2 - 1c$	$(t^2, s^2 + t, s)$	0.078
$3 - 2 - 3$	$(t^2 - s^2, ts, t)$	0.363
$3 - 3 - 1b$	$(t^2, s^2, ts + t)$	5.327
$3 - 3 - 1c$	$(t^2, s^2, t + s + ts)$	9.243
$3 - 3 - 2a$	$(t^2, s^2 + t, ts)$	0.0781
$3 - 3 - 2b$	$(t^2, s^2, ts - s)$	0.095

As for a comparison with [11], in the GitHub implementation [12] of the algorithm in [11], one can find five examples, concerning the *Roman surface* (Example 1 in [12]), Veronese-Segre surfaces (Example 2), adjoints to double ruled quadrics (Example 4), the rational normal scroll (Example 5), and conic bundles (Example 6). Out of the five examples, our algorithm can find the 24 symmetries of the first one, the Roman surface

$$\boldsymbol{x}(t, s) = \left(\frac{t}{1 + t^2 + s^2}, \frac{s}{1 + t^2 + s^2}, \frac{ts}{1 + t^2 + s^2} \right),$$

in 50.016 seconds. There are no timings in [11], so here we cannot compare. However, for the remaining four examples, the computations with our algorithm were too costly, meaning that the resultant in Step 2 of Algorithm 1 took too long to compute.

5.2. Ruled surfaces

Here we consider the method in Section 4, which requires us to determine first the line of striction of the surface. Then we apply the method in [7] to find the symmetries of the rational curve defined by the line of striction. In Table 5, we compare the timing (t_r) of the algorithm for ruled surfaces in Section 4, the timing (t_e) of the algorithm given in [15], and the the timing (t_g) for the general algorithm given in Section 3. One can see that the method in Section 4 beats the other methods in all cases.

Table 5. CPU times (seconds) for symmetries of ruled surfaces given in Table 6.

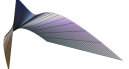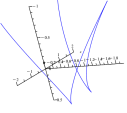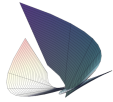| Param.	Deg.	Symm.	t_r	t_e	t_g
x_1	9	8	0.156	9.640	$> 10^3$
x_2	7	1	0.297	1.981	$> 10^3$
x_3	7	2	0.563	1.888	$> 10^3$
x_4	5	2	0.016	1.684	902.213
x_5	2	2	0.047	3.448	54.141
x_6	7	2	0.828	1.935	986.213
x_7	6	2	0.063	1.716	626.012
x_8	17	8	0.172	9.828	$> 10^3$
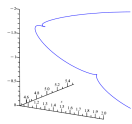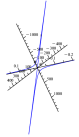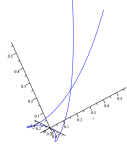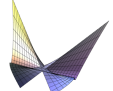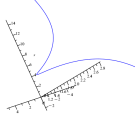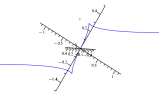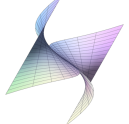
Remark 2. The Plücker conoids analyzed in the previous subsection are also ruled surfaces, but their striction lines are straight lines, and therefore the ideas of Section 4 are not useful for that case.

Remark 3. In Table 5, we have not included the comparison with [5], where no timings are provided. But in any case, as we mentioned in the Introduction to this paper, the algorithm in [5] requires using rational curves in higher dimension ($\mathbb{P}^5$) plus a further step involving the solution of a polynomial quadratic system in a high number of variables, namely 7. Thus, one can expect that the timings of [5] are, in general, much higher than ours. Recall, though, that the algorithm in [5] is aimed to the more general problem of finding projective equivalences between ruled surfaces.

6. Conclusions

While the problem of computing the symmetries of rational curves, and related questions, has received much attention in the literature, similar problems for rational surfaces still pose several challenges. In this paper, we have provided two algorithms related to this question. The first one is simple, very general, and, at least theoretically, can be applied to a wide variety of parametrizations. However, the computations can be hard, so in practice one requires some kind of advantage in the parametrizations: some cases where the algorithm works well, as we have shown in the experimentation section, are the cases of PN surfaces, toric surfaces, and Plücker conoids. The second algorithm we provide is specific for ruled surfaces, and proceeds by reducing the problem to the case of space curves (which we can solve efficiently) by using the line of striction of the surface. As is also shown in the experimentation section, this last algorithm beats other algorithms that have been considered for the same problem.

Table 6. Some ruled surfaces and their lines of striction.

Parametrization	Line of striction	Plots of surf.
$$\boldsymbol{x}_1(t,s) = \begin{pmatrix} \dfrac{2t^8 - 10t^6 - 10t^4 + 5t^2 + 1}{t^2 + 1} + s(2t^5 - 12t^3 + 2t) \\ -\dfrac{t^9 - 6t^7 + 6t^3 + t^2 - 3t + 1}{t^2 + 1} + s(-t^6 + 7t^4 - 7t^2 + 1) \\ t^7 + 3t^5 + 3t^3 + t + 5 + s(t^2 + 1)^3 \end{pmatrix}$$		
$$\boldsymbol{x}_2(t,s) = \begin{pmatrix} \dfrac{t^7 + 7t^5 + 3t^3 - t^2 - 3t + 1}{t^2 + 1} + s(-t^4 - 6t^2 + 3) \\ \dfrac{8t^6 + 8t^4 + 2t}{t^2 + 1} + 8st^3 \\ t(t^2 + 1) + 2 + s(t^2 + 1)^2 \end{pmatrix}$$		
$$\boldsymbol{x}_3(t,s) = \begin{pmatrix} t^6 - 6t^4 + t^2 + 2t + s(t^5 - 6t^3 + t) \\ -t^7 + 6t^5 - t^3 + t^2 + t + s(-t^6 + 6t^4 - t^2 + 1) \\ t^3 + t + s(t^2 + 1) \end{pmatrix}$$		
$$\boldsymbol{x}_4(t,s) = \begin{pmatrix} \dfrac{t^2}{t^2 + 1} + st \\ \dfrac{t^4}{t^2 + 1} + st^3 \\ \dfrac{t^5}{t^2 + 1} + s \end{pmatrix}$$		
$$\boldsymbol{x}_5(t,s) = \begin{pmatrix} 4 + s(t + 1)^2 \\ 1 + s(t + 1) \\ t + s \end{pmatrix}$$		
$$\boldsymbol{x}_6(t,s) = \begin{pmatrix} \dfrac{t^3}{t^2 + 1} + s(-t^5 + t) \\ \dfrac{t^5}{t^2 + 1} + 3st^7 \\ \dfrac{t^7}{t^2 + 1} 2st^3 \end{pmatrix}$$		
$$\boldsymbol{x}_7(t,s) = \begin{pmatrix} t^4 + t^2 + t + s(t^3 + t) \\ t^6 + t^3 + st^5 \\ t^5 + t^3 + t^2 + 3t + s(t^4 + t^2 + 3) \end{pmatrix}$$		
$$\boldsymbol{x}_8(t,s) = \begin{pmatrix} -\dfrac{t^{17} - 6t^{15} + 6t^{11} - 6t^7 + 6t^3 - t^2 - t + 1}{t^2 + 1} + s(-t^6 + 7t^4 - 7t^2 + 1) \\ -\dfrac{2t^{16} - 10t^{14} - 10t^{12} + 2t^{10} + 2t^8 - 10t^6 - 10t^4 + 2t^2 + 1}{t^2 + 1} + s(2t^5 - 12t^3 + 2t) \\ t(t^2 + 1)^3(t^8 + 1) + s(t^2 + 1)^3 \end{pmatrix}$$		

For the general algorithm, the bottleneck of the algorithm is the computation of the resultant of the polynomials derived from the curvatures of the surface. Using the curvatures of the surface is a quite natural idea, so a possibility for further research is to look for strategies to compute this resultant in a more efficient way. In any case, we do not claim that the first algorithm provided in this paper should always be used. Apparently, all the algorithms presented so far to solve the problem considered in this paper have some kind of limitation or drawback. We just intend to contribute to the problem with an algorithm that can be useful in some cases, and perhaps in cases where other algorithms may fail. As for the second algorithm, for ruled surfaces, we dare say that the algorithm is more efficient than the other algorithms presented to this day. Nevertheless, for this second algorithm, it can certainly happen that the line of striction is a line or a circle, in which case we would have infinitely many symmetries to test. In that situation, one should choose an alternative algorithm among the ones reviewed in this paper.

The ideas presented in this paper can be extended without any effort to compute isometries between rational surfaces, both in the cases of non-ruled and ruled surfaces. As for other generalizations, in order to detect, for instance, affine or projective equivalences between rational surfaces, we would need to replace the Gauss and mean curvature by affine or projective invariants. Finding invariants with the properties that we require, which include a "good" behavior with respect to reparametrizations, is not a trivial task (see [29] for a similar question for curves). So we leave this question here as an open problem.

Use of AI tools declaration

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

Acknowledgments

Conflict of interest

The authors declare there is no conflict of interest.

References

1. 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

2. P. Lebmair, J. Richter-Gebert, Rotations, translations and symmetry detection for complexified curves, *Comput. Aided Geom. Des.*, **25** (2008), 707–719. https://doi.org/10.1016/j.cagd.2008.09.004

3. P. Lebmair, *Feature Detection for Real Plane Algebraic Curves*, Ph.D thesis, Universität München, 2009.

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

5. 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

6. U. Gözütok, H. A. Çoban, Detecting isometries and symmetries of implicit algebraic surfaces, *AIMS Math.*, **9** (2024), 4294–4308. https://doi.org/10.3934/math.2024212

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

8. 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

9. 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

10. M. Hauer, B. Jüttler, J. Schicho, Projective and affine symmetries and equivalences of rational and polynomial surfaces, *J. Comput. Appl. Math.*, **349** (2019), 424–437. https://doi.org/10.1016/j.cam.2018.06.026

11. 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

12. https://github.com/niels-lubbes/surface_equivalence

13. J. G. Alcázar, H. Dahl, G. Muntingh, Symmetries of canal surfaces and Dupin cyclides, *Comput. Aided Geom. Des.*, **59** (2018), 68–85. https://doi.org/10.1016/j.cagd.2017.10.001

14. J. G. Alcázar, G. Muntingh, Affine equivalences of surfaces of translation and minimal surfaces, and applications to symmetry detection and design, *J. Comput. Appl. Math.*, **411** (2022), 114206. https://doi.org/10.1016/j.cam.2022.114206

15. 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

16. J. Kozak, M. Krajnc, V. Vitrih, A quaternion approach to polynomial PN surfaces, *Comput. Aided Geom. Des.*, **47** (2016), 172–188. https://doi.org/10.1016/j.cagd.2016.05.007

17. https://www.ugurgozutok.com/academics/software

18. U. Gözütok, Source Code/Examples for the paper: Computation of symmetries of rational surfaces, *Zenodo*, 2024. https://doi.org/10.5281/zenodo.13970116

19. J. R. Sendra, F. Winkler, S. Pérez-Díaz, *Rational Algebraic Curves—A Computer Algebra Approach*, Springer, 2008. https://doi.org/10.1007/978-3-540-73725-4

20. M. P. Do Carmo, *Differential Geometry of Curves and Surfaces*, 2nd edition, Courier Dover Publications, 2016.

21. A. Gray, *Differential Geometry of Curves and Surfaces with Mathematica*, CRC Press, 1999.

22. D. J. Struik, *Lectures on Classical Differential Geometry*, Dover Publications, 1998.

23. N. M. Patrikalakis, T. Maekawa, *Shape Interrogation for Computer Aided Design and Manufacturing*, Springer, 2002. https://doi.org/10.1007/978-3-642-04074-0

24. G. Fischer, *Plane Algebraic Curves*, American Mathematical Society, 2001.

25. D. Pellis, M. Kilian, H. Pottmann, M. Pauly, Computational design of weingarten surfaces, *ACM Trans. Graphics*, **40** (2021), 1–11. https://doi.org/10.1145/3450626.3459939

26. L. Y. Shen, S. Pérez-Díaz, Characterization of rational ruled surfaces, *J. Symb. Comput.*, **63** (2014), 21–45. https://doi.org/10.1016/j.jsc.2013.11.003

27. J. G. Alcázar, R. Goldman, Detecting when an implicit equation or a rational parametrization defines a conical or cylindrical surface, or a surface of revolution, *IEEE Trans. Visual Comput. Graphics*, **23** (2017), 2550–2559. https://doi.org/10.1109/TVCG.2016.2625786

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

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