Shen, Li-Yong, Pérez-Díaz, Sonia & Yang, Zhengfeng. 2019, “Numerical proper reparametrization of space curves and surfaces”, Computer-Aided Design, vol. 116, article 102732

Available at https://doi.org/10.1016/j.cad.2019.07.001

(Article begins on next page)

Numerical Proper Reparametrization of Space Curves and Surfaces

Li-Yong Shen[a], Sonia Pérez-Díaz[b], Zhengfeng Yang[c]

[a]*School of Mathematical Sciences, University of University of Chinese Academy of Sciences, 100049, Beijing, China*
[b]*Grupo ASYNACS, Dpto. de Física y Matemáticas, Universidad de Alcalá, 28871-Alcalá de Henares, Madrid, Spain*
[c]*Software Engineering Institute, East China Normal University, Shanghai, 200062, China*

Abstract

Simplifying rational parametrizations of surfaces is a basic problem in CAD (computer-aided design). One important way is to reduce their tracing index, called proper reparametrization. Most existing proper reparametrization work is symbolic, yet in practical environments the surfaces are usually given with perturbed coefficients hence need a numerical technique of reparametrization considering the intrinsic properness of the perturbed surfaces. We present algorithms for reparametrizing a numerically rational space curve or surface. First, we provide an efficient way to find a parametric support transformation and compute a reparametrization with proper parametric support. Second, we develop a numerical algorithm to further reduce the tracing index, where numerical techniques such as sparse interpolation and approximated GCD computations are involved. We finally provide the error bound between the given rational curve/surface and our reparametrization result.

Keywords: Numerical/symbolic reparametrization, Space curve, Rational surface, Approximately improper/proper

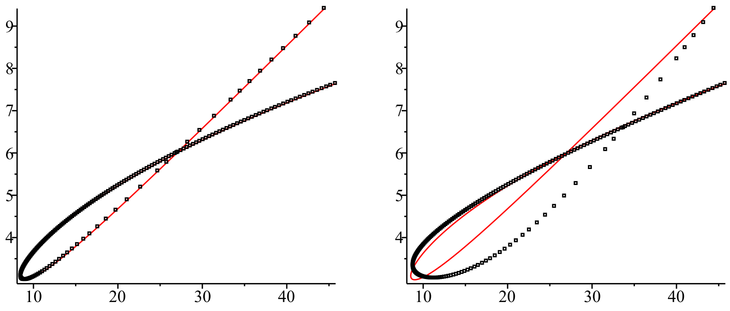
1. Introduction

Rational curves and surfaces are widely used in the original design of engineering, geometric modeling and computer aided design. A significant basic problem is to simplify the parametrization to a proper one, i.e., the rational representation that traces the curve or surface only once. The reason is threefold (see [22]): First, parametrization of smaller degrees can be represented with less data; second, implicitization is easier when the degree is smaller; third, it is easier to find rational curves of smaller degrees on the given surface. Extensive research has considered the problem of proper reparametrization for curves [9, 17, 23, 28] and surfaces [7, 10, 11, 15, 17, 18], mostly in a symbolic computational approach. However, in the practical designing of engineering and modeling applications, the rational curves and surfaces are usually presented by numerical coefficients. Therefore, a proper reparametrization that well "approximates" the original numerical representation is needed.

In the past studies, the parametrization such as splines or Bézier curves, are generally assumed to be proper. However, the parametrization obtained in practical computation may not be proper as expected. To our knowledge, only the recent work [26] considers the proper reparametrization for a given planar curve defined parametrically with perturbed float coefficients. For an approximately improper curve, [26] presents an algorithm of computing a proper reparametrization of the curve, which shows more advantages than the commonly used method presented in [5] of simplifying the curve by degree-reduction

Email addresses: `lyshen@ucas.ac.cn` (Li-Yong Shen), `sonia.perez@uah.es` (Sonia Pérez-Díaz), `zfyang@sei.ecnu.edu.cn` (Zhengfeng Yang)

(see Figure 1). Consideration of the intrinsic properness of the curve leads to the better result than those of the other approximate algorithms. For rational surfaces, there are no papers studying the approximate properness, since the proper reparametrization problem of rational surfaces is not solved completely even in symbolic consideration. In this paper, we extend this idea to numerically improper rational space curves and rational surfaces based on the latest symbolic results.



a) $\mathcal{P}(t)$ v.s. its approx. proper one b) $\mathcal{P}(t)$ v.s. its degree reduced one

Figure 1: Numerical proper reparametrization ([26]) and degree reductions using Eck's method ([5]). The input curve has degree 6 (red color), and its approximately proper one (dash in a)) and degree reduced one by Eck's method (dash in b)) are all of degree 3.

Luroth's theorem [28] shows that a proper rational parametrization always exists for a rational curve, and there are several algorithms on proper reparametrization of exact rational curves [9, 17, 23]. Hence, for numerical rational space curves, we propose a proper reparametrization algorithm (based on the symbolic algorithm presented in [17]) with parallel numerical analysis as in [26]. Unfortunately, much difficulty arises when generalizing the algorithm to the rational surfaces. There are symbolic algorithms that compute the tracing index for the rational surfaces [3, 19], and algorithms of computing the proper reparametrization which are only developed for several special situations of surfaces (see [11, 15, 17, 18]). In particular, the problem of symbolic reparametrization is associated to the problem of functional decomposition of polynomials, which has been the subject of many papers (see e.g. [7], [11], and [10]). In [7], given a set of u polynomials $h = (h_1, \ldots, h_u)$ over a polynomial ring $K[x_1, \ldots, x_n]$ (K is an arbitrary field) the algorithm presented recovers, if any, $f = (f_1, \ldots, f_u) \in K[x_1, \ldots, x_n]$ and $g = (g_1, \ldots, g_u) \in K[x_1, \ldots, x_n]$ such that $h = (f_1(g_1, \ldots, g_u), \ldots, f_u(g_1, \ldots, g_u))$. In [10], it is discussed several notions of decomposition for multivariate polynomials, focussing on the relation with Lüroth's theorem in field theory. These results can be applied to some very particular cases of parametrizations of surfaces as for instance, those that admit a multi–univariate polynomial decomposition. Similar ideas are considered in [11], where algorithms for multi–univariate polynomial decomposition and uni–multivariate polynomial decomposition are obtained. One may check that the results in [18] are not covered by these papers. In fact, in [18], it is presented an algorithm that computes a rational parametrization $\mathcal{Q}$ of a given surface defined parametrically by $\mathcal{P}$ such that the degree of the rational map induced by $\mathcal{Q}$ is less than the degree induced by $\mathcal{P}$. The rational parametrization $\mathcal{P}$ of the surface is defined by any three rational functions (see details in Section 5). Some other results concerning the reparametrization problem for curves and surfaces are presented in [17]. In particular, in [17], the reparametrization problem for surfaces is solved but only for some special surface's parametrizations. Another remarkable work on the proper reparametrization problem is to consider the parametric support and coefficient separately (see [2]). The tracing index of the parametric support is a factor of the tracing index of the rational surface. Moreover, a support transformation naturally induces a reparametrization. Hence, this method gives a simple way to reduce the tracing index of a rational parametrization if the parametric support is not improper.

In this paper, we do not deal with the symbolic reparametrization problem, but with the study of some particular symbolic results of the proper reparametrization problem which can be extended to the numerical situation. For a given rational surface with perturbed float coefficients, we first develop an approach of reparameterizing the surface to a new one with a proper parametric support using the simpler support transformation based on the Hermite form. But even with a proper parametric support, a rational parametrization may still be improper which comes to the classical open problem. To settle this problem, we present the numerical version of the latest proper reparametrization algorithm (see [18]). We remark that we consider the results presented in [18] since they consider general rational parametrizations of surfaces and they are suitable to extend to numerical computations. For practical use, we further examine the numerical error bound between a rational surface and its reparametrization surface.

The paper is organized as follows: in Section 2, some preliminaries and notations are introduced. In Section 3, we give some basic concepts concerning the parametric support transformation and propose an efficient algorithm to find a support transformation (see Subsection 3.1) for an improper support. In Subsection 3.2, we consider general perturbed cases. In Section 4, after reviewing the symbolic proper reparametrization algorithm for an algebraic space curve, we present a numerical algorithm that solve the reparametrization problem from the numerical point of view. In Section 5, we consider the reparametrization problem but for a given algebraic rational surface. We first summarize the algorithm presented in [18], where the symbolic reparametrization problem for surfaces is partially solved, and afterwards we deal with the numerical reparametrization problem of surfaces (see Subsection 5.1). In Subsection 5.2, we prove the correctness of the algorithm and the error analysis.

2. Preliminaries

Let $\mathbb{K}$ be an algebraically closed field of characteristic zero. For an affine rational variety $\mathcal{V}$, let $\mathcal{P}(\,\overline{t}\,)$ be a rational affine parametrization of $\mathcal{V}$ over $\mathbb{K}$, and we write its components as

$$\mathcal{P}(\,\overline{t}\,) = \big(p_1(\,\overline{t}\,),\, p_2(\,\overline{t}\,),\, p_3(\,\overline{t}\,)\big) \in \mathbb{K}(\,\overline{t}\,)^3, \quad p_i(\,\overline{t}\,) = \frac{p_{i,1}(\,\overline{t}\,)}{p_{i,2}(\,\overline{t}\,)}, \quad \gcd(p_{i,1}, p_{i,2}) = 1,\ \ i = 1, 2, 3, \tag{1}$$

where $\overline{t} = t \in \mathbb{K}$ for a rational space curve, and $\overline{t} = (t_1, t_2) \in \mathbb{K}^2$ for a rational surface. Without loss of generality, we assume that p_i, $i = 1, 2, 3$, are not constant. In addition, if $\mathcal{P}$ defines a surface parametrization, we may assume that $\{\nabla p_1,\, \nabla p_2\}$ are linearly independent as vectors in $\mathbb{K}(\,\overline{t}\,)^2$. This assumption is needed to apply the results in [18] (see Section 3).

For a rational function in reduced form $R = M/N \in \mathbb{K}(\,\overline{x}\,)$ where $\overline{x} = (x_1, \ldots, x_n)$, the degree of R with respect to x_i is denoted by $\deg_{x_i}(R) = \max\{\deg_{x_i}(M), \deg_{x_i}(N)\}$. The degree of the rational map induced by $\mathcal{P}$ is denoted by $\deg(\phi_{\mathcal{P}})$. That is, $\phi_{\mathcal{P}} : \mathbb{K}^j \longrightarrow \mathcal{V};\ \overline{t} \longmapsto \mathcal{P}(\,\overline{t}\,)$ (where $j = 1$ for the case of curves, and $j = 2$ for the case of surfaces), and the degree of $\phi_{\mathcal{P}}$ is defined as the degree of the finite field extension $\phi_{\mathcal{P}}(\mathbb{K}(\mathcal{V})) \subset \mathbb{K}(\,\overline{t}\,)$; i.e. $\deg(\phi_{\mathcal{P}}) = [\mathbb{K}(\,\overline{t}\,) : \phi_{\mathcal{P}}(\mathbb{K}(\mathcal{V}))]$ (see e.g. [13] pp.80, or [24] pp.143). As an important result, the properness of $\mathcal{P}(\,\overline{t}\,)$ is characterized by $\deg(\phi_{\mathcal{P}})$ (it is also denoted as $\mathrm{index}(\mathcal{P})$, and we refer to it as the *tracing index*). Precisely, $\mathcal{P}(\,\overline{t}\,)$ is *proper* if and only if $\deg(\phi_{\mathcal{P}}) = 1$, otherwise $\mathcal{P}(\,\overline{t}\,)$ is *improper* (see [13] and [24]). Also, $\deg(\phi_{\mathcal{P}})$ is the cardinality of the fibre of a generic element (see Theorem 7, pp. 76 in [24]). That is,

$$\mathcal{F}_{\mathcal{P}}(P) = \mathcal{P}^{-1}(P) = \{\,\overline{t} \in \mathbb{K}^j \,|\, \mathcal{P}(\,\overline{t}\,) = P\},$$

where $\mathcal{F}_{\mathcal{P}}(P)$ is the fibre of a point $P \in \mathcal{V}$. Associated with the parametrization $\mathcal{P}$, we consider the polynomials

$$H_i^{\mathcal{P}}(\,\overline{t}\,,\,\overline{s}\,) = p_{i,1}(\,\overline{t}\,)p_{i,2}(\,\overline{s}\,) - p_{i,2}(\,\overline{t}\,)p_{i,1}(\,\overline{s}\,) \in (\mathbb{K}[\,\overline{s}\,])[\,\overline{t}\,], \quad i = 1, 2, 3,$$

where $\overline{s}$ is a new variable, and $\overline{s} = s \in \mathbb{K}$ for a rational space curve, and $\overline{s} = (s_1, s_2) \in \mathbb{K}^2$ for a rational surface. Let $\mathbb{F} = \overline{\mathbb{K}(\overline{s})}$ be the algebraic closure of $\mathbb{K}(\overline{s})$. We have that

$$\phi_{\mathcal{P}}^{-1}(\mathcal{P}(\overline{s})) = \{\overline{t} \in \mathbb{F}^j \mid H_i^{\mathcal{P}}(\overline{t}, \overline{s}) = 0,\, i = 1, 2, 3\} \tag{2}$$

and

$$\deg(\phi_{\mathcal{P}}) = \mathrm{Card}(\phi_{\mathcal{P}}^{-1}(\mathcal{P}(\overline{s}))) \tag{3}$$

3. Reparametrization by support transformation

A parametrization $\mathcal{P}(\overline{t})$ always consists on the coefficients and the associated parametric supports. If the parametric support of the parametrization $\mathcal{P}(\overline{t})$ is not improper, then there exists a support transformation that maps the improper support to a proper support. The support transformation naturally induces a reparametrization to reduce the tracing index of a rational parametrization (see [2]). In this section, we introduce the parametric support method but with a new support transformation algorithm with low complexity. For this purpose, we consider two different subsections: first, in Subsection 3.1, we focus on a simple way to compute a support transformation for the improper parametric supports. Afterwards, in Subsection 3.2, we consider general perturbed cases.

3.1. Parametric support transformation

Let $f(\overline{t}) = \sum f_{\overline{i}}\, \overline{t}^{\,\overline{i}}$, $\overline{i} \in \mathbb{Z}^k, k = 1$ or 2 be a polynomial over $\mathbb{K}$. The set of all the integer lattice points $\overline{i}$ such that $f_{\overline{i}} \neq 0$ is called the *parametric support* of f. Since the parametric support of a space curve is simpler than that of the surfaces, we now representatively discuss the parametric support of the surface, i.e., $k = 2$. Then, each lattice point $\overline{i} = (i_1, i_2)$ corresponds to a monomial $t_1^{i_1} t_2^{i_2}$.

Let $\mathcal{P}$ be a rational surface defined by (1). The *parametric support of* $\mathcal{P}$, denoted by $\mathbf{S}$, is defined to be the union set of parametric supports of polynomials $p_{i,1}, p_{i,2},\ i = 1, 2, 3$ of $\mathcal{P}$. $|\mathbf{S}|$ is the number of lattice points in $\mathbf{S}$. The support can be written as a lattice matrix

$$M_{\mathbf{S}} = \left(\overline{i}\right)_{\overline{i} \in \mathbf{S}} \in \mathbb{Z}^{|\mathbf{S}| \times \mathbf{2}},$$

$|S| \geq 2$, and the rank of M_S is two, since the surface reduces to a curve if the rank is one. Furthermore, we assume that $(0, 0) \in \mathbf{S}$ without loss of generality (see Lemma 2 in [25]).

Let $N(\mathbf{S})$ be the normalized area of the Newton polygon of $\mathbf{S}$, and $A(\mathbf{S})$ the algebraic degree of the general surface defined on $\mathbf{S}$. Then, the *tracing index* of $\mathbf{S}$ is defined to be $\mathrm{index}(\mathbf{S}) = \frac{N(\mathbf{S})}{A(\mathbf{S})}$. A support $\mathbf{S}$ is *improper* if $\mathrm{index}(\mathbf{S}) > \mathbf{1}$ and is *proper* if $\mathrm{index}(\mathbf{S}) = \mathbf{1}$. Considering a parametric support transformation

$$\tau : (i_1, i_2) \in \mathbb{Z}^2 \mapsto (a \cdot i_1 + b \cdot i_2,\ c \cdot i_1 + d \cdot i_2) \in \mathbb{Z}^2$$

where a, b, c, d are rational numbers, the support transformation induces a reparametrization transformation for $\mathcal{P}$ as

$$t_1 = u_1^a u_2^c, \qquad t_2 = u_1^b u_2^d$$

with new parameters u_1 and u_2.

Under these conditions, we have the following theorem summarized from Theorem 1, 2 and 3 in [2].

Theorem 1. *Let* $\mathbf{S}$ *be a parametric support of* $\mathcal{P}$*. Then*

(1) $\mathrm{index}(\mathbf{S}) = \gcd\{N(\mathbf{S}') \mid \mathbf{S}' \subseteq \mathbf{S}, |\mathbf{S}'| = 3\}$.

(2) There exists a support transformation τ *mapping* $\mathbf{S}$ *to* $\mathbf{S}_\tau$ *such that* $\mathrm{index}(\mathbf{S}_\tau) = 1$.

(3) If $\operatorname{index}(\mathbf{S}_\tau) = 1$ *then* $\deg(\phi_{\mathcal{P}}) = \deg(\phi_{\mathcal{P}_\tau}) \operatorname{index}(\mathbf{S})$*, where* $\mathcal{P}_\tau$ *is the reparametrization of* $\mathcal{P}$*.*

(4) If $\operatorname{index}(\mathbf{S}) = 1$*, then with a generic coefficient set* $\deg(\phi_{\mathcal{P}}) = 1$*.*

For an improper support $\mathbf{S}$, an algorithm was proposed to compute the tracing index of $\mathbf{S}$ and a support transformation, τ, was found such that $\tau(\mathbf{S})$ is proper. This algorithm computes the gcd of the areas of all the triangular sub-supports in $\mathbf{S}$, and the computation complexity was not less than $O(|\mathbf{S}|^2)$ (see the proof of Theorem 1 in [2]). Here we will compute the tracing index from the Hermite of the integral matrix and then, the support transformation is easily derived. The computational complexity of the Hermite form for an integral $|S| \times 2$-matrix with rank k is $O(4|\mathbf{S}|)$ (see [12]). A support S determines the feature complex of a variety and it is, in general, much greater than the dimension k, i.e., $|\mathbf{S}| >> k$.

Let $M_{\mathbf{S}} \in \mathbb{Z}^{|\mathbf{S}| \times 2}$ be the integer support matrix consisting of all the lattice points in $\mathbf{S}$, and let $H_{\mathbf{S}}$ be the Hermite form of $M_{\mathbf{S}}$. Then we have

$$H_{\mathbf{S}}^T = \begin{pmatrix} h_{11} & 0 & 0 & \cdots & 0 \\ h_{12} & h_{22} & 0 & \cdots & 0 \end{pmatrix},$$

where $h_{11}h_{22} \neq 0$ since the rank of $M_{\mathbf{S}}$ is two.

Theorem 2. $\operatorname{index}(\mathbf{S}) = h_{11}h_{22}$ *and*

$$\tau : (i, j) \mapsto (i, j) \begin{pmatrix} h_{11} & h_{12} \\ 0 & h_{22} \end{pmatrix}^{-1}$$

defines a support transformation mapping $\mathbf{S}$ *to* $\mathbf{S}_\tau$ *with* $\operatorname{index}(\mathbf{S}_\tau) = 1$*.*

Proof. By Theorem 1 and linear algebra, one can find that the index is equivalent to the product of diagonal elements of the Smith form of $M_{\mathbf{S}}$. Hence the index is also equivalent to the product of diagonal elements of the Hermite form H and $T = \begin{pmatrix} h_{11} & h_{12} \\ 0 & h_{22} \end{pmatrix}^{-1}$ defines a support transformation for $\mathbf{S}$ such that $\operatorname{index}(\mathbf{S}_\tau) = 1$. □

3.2. STLN-based sparse approximation

The reparametrization based on support transformation is determined by the support structure. Hence the existence of terms of $\mathcal{P}$ plays an essential role. When we consider the general perturbed cases, dense terms are often brought into, but the target rational surfaces may not need the dense structure. In this subsection, a method is proposed to remove some tiny influent terms from the parametrization $\mathcal{P}(\bar{t})$ and refine the coefficients of the remaining support such that the sparse parametrization $\mathcal{P}_e(\bar{t}) = (\tilde{p}_1(\bar{t}), \tilde{p}_2(\bar{t}), \tilde{p}_3(\bar{t})) \in \mathbb{K}(\bar{t})^3$, $\tilde{p}_i(\bar{t}) = \frac{\tilde{p}_{i,1}(\bar{t})}{\tilde{p}_{i,2}(\bar{t})}$, $i = 1, 2, 3$, is close to the given $\mathcal{P}(\bar{t})$.

To find a sparse approximation in an interval, a simple way is to approximate the numerators and the denominators of the rational functions within an interval respectively, i.e.,

$$\left\| \tilde{p}_{i,1}(\bar{t}) - p_{i,1}(\bar{t}) \right\|_2 \leq \epsilon \quad \text{and} \quad \left\| \tilde{p}_{i,2}(\bar{t}) - p_{i,2}(\bar{t}) \right\|_2 \leq \epsilon \quad \text{for } i = 1, 2, 3,$$

where $\| \cdot \|_2$ returns the 2-norm of the coefficient vector of a polynomial. We can also consider the approximations for the rational functions, i.e.

$$\left\| \frac{\tilde{p}_{i,1}(\bar{t})}{\tilde{p}_{i,2}(\bar{t})} - \frac{p_{i,1}(\bar{t})}{p_{i,2}(\bar{t})} \right\|_2 \leq \epsilon \quad \text{for } i = 1, 2, 3. \tag{4}$$

Here $\| \cdot \|_2$ returns the 2-norm of the numerator coefficient vector of a rational function. To solve the above problems, we need to consider two sub-problems:

$\mathbf{P}_1$ Get the sparse support by removing some tiny influent terms for the given tolerance ϵ.

$\mathbf{P}_2$ Refine the coefficients corresponding to the resulted support from $\mathbf{P}_1$ with a given tolerance ϵ.

The problems $\mathbf{P}_1$ and $\mathbf{P}_2$ are both similar to bivariate rational function interpolation problem. Problem $\mathbf{P}_1$ is simpler and can be solved explicitly since the candidate support of the rational function is known. After normalizing the rational function $\frac{p_{i,1}(\bar{t})}{p_{i,2}(\bar{t})}$, the sparse support can be obtained by removing terms whose coefficients are less than the given tolerance ϵ.

As mentioned above, $\mathbf{P}_2$ is a subproblem of rational function interpolation. Our goal is to find a sparse parametrization such that $\frac{\tilde{p}_{i,1}(\bar{t})}{\tilde{p}_{i,2}(\bar{t})}$ is as sparse as possible under $\left\|\frac{\tilde{p}_{i,1}(\bar{t})}{\tilde{p}_{i,2}(\bar{t})} - \frac{p_{i,1}(\bar{t})}{p_{i,2}(\bar{t})}\right\|_2 \leq \epsilon, i = 1, 2, 3$ for each $\bar{t} = (t_1, t_2) \in \mathbb{K}^2$. Therefore, the problem of computing the coefficients can be transformed to that of preserving matrix structure (Vandermonde-block) total least squares (see Section 4 of [14]). Problem $\mathbf{P}_2$ can also be solved by the STLN method (see [16]).

Example 1. *Let $\mathcal{V}$ be the surface defined parametrically by*

$$\mathcal{P}(\bar{t}) = \left(\frac{2.002\,{t_1}^2+0.003\,{t_2}^4-0.005\,t_1t_2}{0.997+{t_1}^4+t^2+0.001\,{t_1}^2{t_2}^2}, \frac{2\,{t_2}^2+0.005\,{t_2}^4-0.002\,t_1t_2}{1.001+{t_1}^4+{t_2}^2-0.001\,{t_1}^2t}, \frac{0.999-{t_1}^4-{t_2}^2+0.001\,{t_2}^4-0.001\,t_1t_2}{1.002+{t_1}^4+{t_2}^2+0.001\,{t_1}^2{t_2}^2}\right)$$

(see Figure 2.a)). We compute a sparse approximation with tolerance $\epsilon = 0.001$. The discussion region is $[0,1] \times [0,1]$. We get a sparse approximation $\mathcal{P}_S(t_1, t_2) = \big(\tilde{p}_1(\bar{t}), \tilde{p}_2(\bar{t}), \tilde{p}_3(\bar{t})\big) \in \mathbb{K}(\bar{t})^3$, where

$$\begin{aligned}
\tilde{p}_1(\bar{t}) &= \frac{-1.997257414797780994{t_1}^2}{-0.997475460442092454-1.0\,{t_1}^4-0.996661205265358174\,{t_2}^2},\\
\tilde{p}_2(\bar{t}) &= \frac{1.98718765468330627{t_2}^2}{1.0+0.992465458325022309\,{t_1}^4+0.983860429454241792\,{t_2}^2},\\
\tilde{p}_3(\bar{t}) &= \frac{0.993875570416674292-0.997053277031201168\,{t_1}^4-0.993971171354035232\,{t_2}^2}{0.997743315877603565+1.0\,{t_1}^4+0.993828950897961527\,{t_2}^2}.
\end{aligned}$$

The parametric support of the sparse approximation $\mathcal{P}_S$ is $\mathbf{S} = \{(0,0), (2,0), (0,2), (0,4), (4,0)\}$ and this support is improper with $\operatorname{index}(\mathbf{S}) = 2$. *The Hermite form of $M_{\mathbf{S}}$ is $H = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \end{pmatrix}$. Using the support transformation defined by*

$$\tau : (i,j) \mapsto (i,j)T, \textit{where } T = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}^{-1} = \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix}.$$

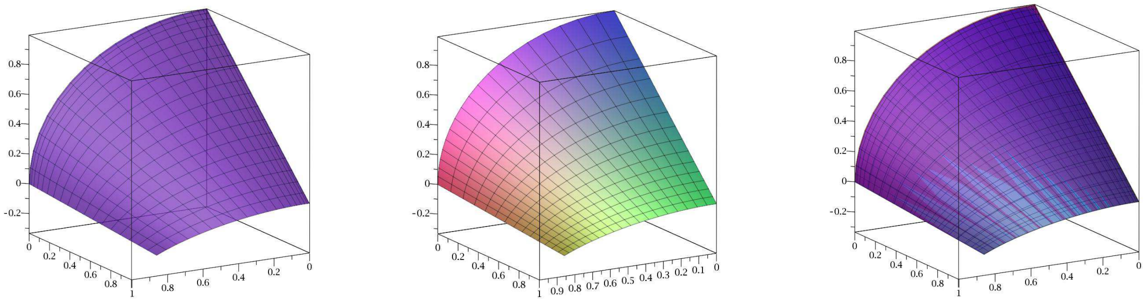
This parametric support transformation induces a reparametrization $u_1 = \sqrt{t_1}, u_2 = \sqrt{t_2}$ (See Figure 2.b).

The reparameterization includes two steps: sparse approximation and reparameterization induced by the support transformation. For the first step, the difference between the given surface and the approximated surface is given in equation (4). In the second step, the reparameterization induced by the support transformation does not involve any error.

According to Theorem 1, a rational parametrization with generic coefficients is proper if the parametric support is proper. However, a parametrization with perturbed float coefficients can be still improper even the parametrization has a proper parametric support. Hence, the proper reparametrization problem comes to the open general situation. In this section 4 and in Section 5, we consider the problem of proper reparametrization of space curves and surfaces which are assumed to have proper parametric supports.

4. Proper Reparametrization of Space Curves

In this section, we deal with the proper reparametrization of a given parametric space curve from the symbolic and also from the numerical point of view. For this purpose, we first summarize the symbolic



a) Given surface

b) Approx. surface with proper support

c) Given surface v.s. its approx. one

Figure 2: The numerical input surface (a) and the reparametrization (b) generated by Theorem 2 from a sparse approximation of (a). In the right figure (c), one may check that both surfaces are very close.

algorithm presented in [17] (Section 3) for computing a proper reparametrization of an algebraic space curve. Afterwards, in Subsection 4.1, we present a numerical algorithm that solve the reparametrization problem from the numerical point of view.

Review the tracing index formulas of (2) and (3) for a rational space curve $\mathcal{P}(\bar{t}) = \mathcal{P}(t) \in \mathbb{K}(t)^3$, we have that

$$\deg(\phi_{\mathcal{P}}) = \deg_t(\gcd(H_1^{\mathcal{P}}(t,s), H_2^{\mathcal{P}}(t,s), H_3^{\mathcal{P}}(t,s))). \tag{5}$$

The following symbolic algorithm is presented in [17] (Section 3) for computing a proper reparametrization of a parametric algebraic space curve. That is, *given a parametrization $\mathcal{P}(t) \in \mathbb{K}(t)^3$ of an algebraic space curve $\mathcal{C}$ over a field $\mathbb{K}$, we find a rational proper parametrization $\mathcal{Q}(t) \in \mathbb{K}(t)^3$ of $\mathcal{C}$, and a rational function $R(t) \in \mathbb{K}(t) \setminus \mathbb{K}$ such that $\mathcal{P}(t) = \mathcal{Q}(R(t))$.*

Algorithm 1 (Proper Reparametrization for Space Curves (PRSC)).
INPUT: *a field $\mathbb{K}$, a rational affine parametrization*

$$\mathcal{P}(t) = (p_1(t), p_2(t), p_3(t)) \in \mathbb{K}(t)^3, \quad p_i(t) = p_{i,1}(t)/p_{i,2}(t), \ \gcd(p_{i,1}, p_{i,2}) = 1$$

of an algebraic space curve $\mathcal{C}$.
OUTPUT: *a rational proper parametrization, $\mathcal{Q}(t) = (q_1(t), q_2(t), q_3(t)) \in \mathbb{K}(t)^3, q_i(t) = q_{i,1}(t)/q_{i,2}(t)$, of $\mathcal{C}$, and $R(t) \in \mathbb{K}(t)$ such that $\mathcal{P}(t) = \mathcal{Q}(R(t))$.*

```
1. Compute $H_j^{\mathcal{P}}(t,s) = p_{j,1}(t)p_{j,2}(s) - p_{j,1}(s)p_{j,2}(t), \quad j = 1,2,3$.
2. Determine $S^{\mathcal{P}}(t,s) = \gcd(H_1^{\mathcal{P}}(t,s), H_2^{\mathcal{P}}(t,s), H_3^{\mathcal{P}}(t,s)) = C_m(t)s^m + \cdots + C_0(t)$, and $\deg(\phi_{\mathcal{P}}) = \deg_t(S^{\mathcal{P}})$.
3. If $\deg(\phi_{\mathcal{P}}) = 1$, RETURN $\mathcal{Q}(t) = \mathcal{P}(t)$, and $R(t) = t$. Otherwise go to Step 4. [The algorithm finishes here if $\mathcal{P}$ is proper.]
4. Compute a rational function $R(t) = C_i(t)/C_j(t) \in \mathbb{K}(t) \setminus \mathbb{K}$, where $C_j(t), C_i(t)$ are two of the polynomials obtained in Step 2 such that $\gcd(C_j, C_i) = 1$ and $C_jC_i \notin \mathbb{K}$ (see Section 2 in [17]).
5. For $i = 1,2,3$, consider the polynomials $G_i^{\mathcal{P}}(t, x_i) = x_i p_{i,2}(t) - p_{i,1}(t)$, and compute
       $L_i(s, x_i) = \mathrm{Res}_t(G_i^{\mathcal{P}}(t, x_i), sC_j(t) - C_i(t)) = (q_{i,2}(s)x_i - q_{i,1}(s))^{\deg(R)}$.
6. RETURN the rational function $R(t) \in \mathbb{K}(t) \setminus \mathbb{K}$, and the proper parametrization
       $\mathcal{Q}(t) = (q_{1,1}(t)/q_{1,2}(t), q_{2,1}(t)/q_{2,2}(t), q_{3,1}(t)/q_{3,2}(t)) \in \mathbb{K}(t)^3$.
   [The algorithm returns a proper parametrization, $\mathcal{Q}(t)$, of $\mathcal{C}$ and $R(t)$ such that $\mathcal{P}(t) = \mathcal{Q}(R(t))$.]
```

Remark 1. *Observe that the coefficient field $\mathbb{K}$ is not necessarily an algebraically closed field when we consider the reparametrization of a space curve. Thus, we also may apply the algorithm for $\mathcal{P}(s,t) \in (\mathbb{K}(s))(t)^3$ (that is, for a parametrization of a space curve defined over the field $\mathbb{K}(s)$). The output parametrization, $\mathcal{Q}(s,t)$ is in $(\mathbb{K}(s))(t)^3$ (see Section 5).*

We give an example to illustrate Algorithm PRSC.

Example 2. *Let $\mathcal{C}$ be a rational space curve over $\mathbb{C}$ defined by the parametrization*

$$\mathcal{P}(t) = \left(\frac{(t+1)^2(t^2+1)}{2(t^4+4t^2+1+2t^3+2t)}, \frac{2(t^2+t+1)}{(t+1)^2}, \frac{-(t^2+1)(t^2+1+4t)}{(t+1)^4}\right) \in \mathbb{C}(t)^3.$$

In Step 1 of the algorithm, we compute the polynomials $H_i^{\mathcal{P}}(t,s)$, for $i=1,2,3$. Next, we compute $S^{\mathcal{P}}(t,s)$. We obtain $S^{\mathcal{P}}(t,s) = C_0(t)+C_1(t)s+C_2(t)s^2$, where $C_0(t)=t$, $C_1(t)=-t^2-1$, and $C_2(t)=t$. Since $\deg(\phi_{\mathcal{P}}) = \deg_t(S^{\mathcal{P}}) > 1$, we go to Step 4 of the algorithm, and we consider the rational function

$$R(t) = -(t^2+1)/t \in \mathbb{C}(t).$$

Note that $\gcd(C_0,\, C_1) = 1$. Now, we compute the polynomials

$$L_1(s,x_1) = \mathrm{Res}_t(G_1^{\mathcal{P}}(t,x_1), sC_1(t)-C_0(t)) = (4x_1+4sx_1-2s-s^2+2s^2x_1)^2,$$

$$L_2(s,x_2) = \mathrm{Res}_t(G_2^{\mathcal{P}}(t,x_2), sC_1(t)-C_0(t)) = (2x_2-2-2s+sx_2)^2,$$

$$L_3(s,x_3) = \mathrm{Res}_t(G_3^{\mathcal{P}}(t,x_3), sC_1(t)-C_0(t)) = (4x_3+4s+4sx_3+s^2+s^2x_3)^2,$$

where $G_i^{\mathcal{P}}(t,x_i) = x_ip_{i,2}(t) - p_{i,1}(t)$ (see Step 5). Finally, in Step 6, the algorithm outputs the proper parametrization $\mathcal{Q}(t)$, and the rational function $R(t)$:

$$\mathcal{Q}(t) = \left(\frac{t(t+2)}{2(2+2t+t^2)}, \frac{2(t+1)}{2+t}, \frac{-t(t+4)}{4+4t+t^2}\right) \in \mathbb{C}(t)^3, \qquad R(t) = -\frac{t^2+1}{t} \in \mathbb{C}(t).$$

4.1. Numerical Algorithm for Proper Reparametrization for Space Curves

The problem of numerical proper reparametrization for space curves solved in this subsection can be stated as follows: *given the field $\mathbb{C}$ of complex numbers, a tolerance $\epsilon > 0$, and a rational parametrization $\mathcal{P}(t) \in \mathbb{C}(t)^3$ of an algebraic space curve $\mathcal{C}$ that is approximately improper (see Definition 1), we find a rational parametrization $\mathcal{Q}(t) \in \mathbb{C}(t)^3$ of an algebraic space curve $\mathcal{D}$, and a rational function $R(t) \in \mathbb{C}(t) \setminus \mathbb{C}$ such that $\mathcal{Q}$ is an ϵ-proper reparametrization of $\mathcal{P}$ (see Definition 4). In addition, we measure the closeness between $\mathcal{C}$ and $\mathcal{D}$.*

In the following we consider a tolerance $\epsilon > 0$, and a rational parametrization of a given algebraic space curve $\mathcal{C}$

$$\mathcal{P}(t) = (p_1(t), p_2(t), p_3(t)) \in \mathbb{C}(t)^3, \quad p_i(t) = \frac{p_{i,1}(t)}{p_{i,2}(t)}, \quad \epsilon\text{-}\gcd(p_{i,1}, p_{i,2}) = 1,\ i = 1,2,3,$$

where ϵ-$\gcd(p_{j,1}, p_{j,2})$ denotes the approximate gcd for the polynomials $p_{i,1}$ and $p_{i,2}$ (we remind that $\mathcal{P}$ is expected to be given with perturbed float coefficients). The function ϵ-$\gcd(f,g)$ returns a univariate numerical polynomial d such that d is an ϵ-gcd for the input polynomials f and g, i.e., there are univariate numerical polynomials u, v satisfy

$$\|uf+vg-d\| < \epsilon\|(f,g,u,v,d)\|, \quad \|f-df_1\| < \epsilon\|f\|, \quad \text{and} \quad \|g-dg_1\| < \epsilon\|g\|,$$

where the polynomials f_1 and f_2 are cofactors of f and g with respect to the divisor d, $\|\cdot\|$ denotes the infinity norm, and $\|(f,g,u,v,d)\| := \max\{\|f\|, \|g\|, \|u\|, \|v\|, \|d\|\}$.

At this point, we need to generalize the concept of tracing index to the numerical situation. Intuitively speaking, the *approximate tracing index of* $\mathcal{P}$ can be defined as the common ϵ-roots (see e.g. [20]) of the polynomials $H_i^{\mathcal{P}}(t,s) = p_{i,1}(t)p_{i,2}(s) - p_{i,1}(s)p_{i,2}(t)$, $i = 1,2,3$, w.r.t the variable t (see Subsection 3.1. in [26]). The formal notion of this intuitive idea will be done in Definitions 1, 2 and 3. Note that the approximate gcd can also be defined as the approximate common factors associated to the common ϵ-roots. However, in our implementations we use the widely used ϵ-gcd algorithms [1, 4] which can be generalized to multivariate polynomials [8, 29].

Definition 1. *We define the* approximate tracing index *of* $\mathcal{P}$ *as* $\deg_t(S_\epsilon^{\mathcal{P}})$, *where*

$$S_\epsilon^{\mathcal{P}}(t,s) = \epsilon\text{-gcd}(H_1^{\mathcal{P}}, H_2^{\mathcal{P}}, H_3^{\mathcal{P}}), \quad H_i^{\mathcal{P}}(t,s) = p_{i,1}(t)p_{i,2}(s) - p_{i,1}(s)p_{i,2}(t),\ i = 1,2,3.$$

We denote it as ϵ-index$(\mathcal{P})$. *Furthermore,* $\mathcal{P}$ *is said to be* approximately improper *or* ϵ-improper *if* ϵ-index$(\mathcal{P}) > 1$. *Otherwise,* $\mathcal{P}$ *is said to be* approximately proper *or* ϵ-proper.

In Definition 2, we present the notion of ϵ-point for points of the form $(t, R(t)) \in \mathbb{C}(t)^2$. Furthermore, throughout the paper, we use the infinity norm denoted as $\|\cdot\|$, and num$(\cdot)$ returns the numerator of a rational function. Note that since in our situation all the norms are equivalent, one may reason similarly with a different norm.

Definition 2. *Given a non-zero polynomial* $A \in \mathbb{C}[t,s]$ *and* $R(t) \in \mathbb{C}(t)$, *we say that* $A(t,R(t)) \approx_\epsilon 0$ *if* $\|\text{num}(A(t,R(t)))\| \leq \epsilon\|A\|$.

Definition 3. *Let* $R(t) \in \mathbb{C}(t)$. *Let* $\mathcal{Q}(t) = (q_1(t), q_2(t), q_3(t)) \in \mathbb{C}(t)^3$, $q_i(t) = \frac{q_{i,1}(t)}{q_{i,2}(t)}$, ϵ-gcd$(q_{i,1}, q_{i,2}) = 1$, $i = 1,2,3$, *be a rational parametrization of a space curve. We say that* $\mathcal{P}(t) \sim_\epsilon (\mathcal{Q} \circ R)(t)$ *if* $H_j^{\mathcal{P}\mathcal{Q}}(t,R(t)) \approx_\epsilon 0$, *where*

$$H_i^{\mathcal{P}\mathcal{Q}}(t,s) = p_{i,1}(t)q_{i,2}(s) - q_{i,1}(s)p_{i,2}(t),\ i = 1,2,3.$$

Now, we are ready to introduce the notions of ϵ-*numerical reparametrization* and ϵ-*proper reparametrization*. For this purpose, we use Definition 3.

Definition 4. *Let* $\mathcal{P}(t) \in \mathbb{C}(t)^3$ *be a rational parametrization of a given space curve* $\mathcal{C}$. *We say that a parametrization* $\mathcal{Q}(t) \in \mathbb{C}(t)^3$ *is an* ϵ-numerical reparametrization *of* $\mathcal{P}(t)$ *if there exists* $R(t) = M(t)/N(t) \in \mathbb{C}(t) \setminus \mathbb{C}$, ϵ-gcd$(M,N) = 1$, *such that* $\mathcal{P} \sim_\epsilon \mathcal{Q} \circ R$. *In addition, if* ϵ-index$(\mathcal{Q}) = 1$, *then we say that* $\mathcal{Q}$ *is an* ϵ-proper reparametrization *of* $\mathcal{P}$.

In the following, we present a result that characterizes whether an ϵ-numerical reparametrization of $\mathcal{P}(t)$, namely $\mathcal{Q}$, is ϵ-proper (see Corollary 1 in [26]). This theorem shows that results obtained in the numerical case are similar and consistent with those obtained in the symbolic case.

Theorem 3. *Let* $\mathcal{Q}(t) \in \mathbb{C}(t)^3$ *be an* ϵ*-numerical reparametrization of* $\mathcal{P}(t)$. *It holds that* ϵ-index$(\mathcal{P}) = \epsilon$-index$(\mathcal{Q}) \deg(R)$.

In the following, we give the numerical reparametrization algorithm for rational space curve and we summarize the error analysis. The algorithm and related results can be proved similarly to those of the rational planar curve in [26] (see Sections 4 and 5 and in particular, Theorem 5 and the *Numeric Algorithm Reparametrization for Curves*).

Algorithm 2 (Numerical Proper Reparametrization for Space Curves (NPRSC)).
INPUT: *a tolerance* $\epsilon > 0$, *and a rational affine parametrization*

$$\mathcal{P}(t) = (p_1(t),\, p_2(t),\, p_3(t)) \in \mathbb{C}(t)^3, \quad p_i(t) = \frac{p_{i,1}(t)}{p_{i,2}(t)}, \quad \epsilon\text{-gcd}(p_{i,1}, p_{i,2}) = 1$$

of an algebraic space curve $\mathcal{C}$.
OUTPUT: *a rational parametrization* $\mathcal{Q}(t) = (q_1(t),\, q_2(t), q_3(t)) \in \mathbb{C}(t)^3, q_i(t) = q_{i,1}(t)/q_{i,2}(t)$, $\epsilon\text{-gcd}(q_{i,1}, q_{i,2}) = 1$, $i = 1, 2, 3$, *such that* $\epsilon\text{-index}(\mathcal{Q}) = 1$ *and* $\mathcal{P} \sim_\epsilon \mathcal{Q} \circ R$, *where* $R(t) \in \mathbb{C}(t) \setminus \mathbb{C}$.

1. *Compute* $H_j^{\mathcal{P}}(t,s) = p_{j,1}(t)p_{j,2}(s) - p_{j,1}(s)p_{j,2}(t), \quad j = 1, 2, 3.$
2. *Determine*

   $$S_\epsilon^{\mathcal{P}}(t,s) = \epsilon\text{-gcd}(H_1^{\mathcal{P}}(t,s), H_2^{\mathcal{P}}(t,s), H_3^{\mathcal{P}}(t,s)) = C_m(t)s^m + \cdots + C_0(t),$$

   and $\epsilon\text{-index}(\mathcal{P}) := \ell$, *where* $\ell := \deg_t(S_\epsilon^{\mathcal{P}})$.
3. *If* $\epsilon\text{-index}(\mathcal{P}) = 1$, RETURN $\mathcal{Q}(t) = \mathcal{P}(t)$, *and* $R(t) = t$. *Otherwise go to Step 4. [The algorithm finishes here if* $\mathcal{P}$ *is* ϵ*–proper.]*
4. *Consider* $R(t) = C_i(t)/C_j(t) \in \mathbb{C}(t)$, *such that* $C_j(t)$, $C_i(t)$ *are two of the polynomials obtained in Step 2 satisfying that* $C_jC_i \notin \mathbb{C}$, *and* $\epsilon\text{-gcd}(C_j, C_i) = 1$.
5. *For* $i = 1, 2, 3$, *define the polynomials* $G_i^{\mathcal{P}}(t, x_i) = x_ip_{i,2}(t) - p_{i,1}(t)$, *and compute*

   $$L_i(s, x_i) = \mathrm{Res}_t(G_i^{\mathcal{P}}(t, x_i), sC_j(t) - C_i(t)).$$

6. *For* $i = 1, 2, 3$, *compute the root in the variable* x_i *of the polynomial* $\dfrac{\partial^{\ell-1}L_i}{\partial^{\ell-1}x_i}(s, x_i)$. *Let* $\widetilde{q}_i(t) = \widetilde{q}_{i,1}(t)/\widetilde{q}_{i,2}(t)$ *be this root, and let* $\widetilde{\mathcal{Q}}(t) = (\widetilde{q}_1(t),\, \widetilde{q}_2(t),\, \widetilde{q}_3(t)) \in \mathbb{C}(t)^3$.
7. *Simplify* $\widetilde{\mathcal{Q}}(t)$ *by removing the approximate gcd from the numerator and denominator of each component of* $\widetilde{\mathcal{Q}}$. *Let*

   $$\mathcal{Q}(t) = (q_1(t),\, q_2(t),\, q_3(t)) \in \mathbb{C}(t)^3,\, q_k(t) = \frac{q_{k,1}(t)}{q_{k,2}(t)}, \quad \epsilon\text{-gcd}(q_{k,1}, q_{k,2}) = 1,\ k = 1, 2, 3,$$

   be the obtained parametrization.

   7.1. *Check whether the following equality holds*

   $$L_i(s, x_i) = (x_iq_{i,2}(s) - q_{i,1}(s))^\ell + \epsilon^\ell W_i(s, x_i), \qquad \|\mathrm{num}(W_i(R, p_i))\| \leq \|H_i^{\mathcal{P}\mathcal{Q}}\|^\ell \quad (\star)$$

   (that is, check whether L_i *can be written as* $L_i(s, x_i) = (x_iq_{i,2}(s) - q_{i,1}(s))^\ell + \epsilon^\ell W_i(s, x_i)$, *where* $W_i(s, x_i) \in \mathbb{C}[s, x_i]$*).*
   If $(\star)$ *does not hold, look for a new tolerance,* $\overline{\epsilon}$, *such that*

   $$L_k(s, x_k) = (x_kq_{k,2}(s) - q_{k,1}(s))^\ell + \overline{\epsilon}^\ell\, W_k(s, x_k), \quad \|\mathrm{num}(W_k(R, p_k))\| \leq \|H_k^{\mathcal{P}\mathcal{Q}}\|^\ell.$$

8. RETURN *the rational function $R(t) \in \mathbb{C}(t) \setminus \mathbb{C}$, $\mathcal{Q} \in \mathbb{C}(t)^3$, and the message "$\mathcal{Q}$ is an ϵ-proper reparametrization of $\mathcal{P}$" or "$\mathcal{Q}$ is an $\overline{\epsilon}$-proper reparametrization of $\mathcal{P}$", if a new tolerance, $\overline{\epsilon}$, is considered in Step 7.1. [The algorithm returns an ϵ–proper parametrization, $\mathcal{Q}(t)$, of $\mathcal{C}$ and $R(t)$ such that $\mathcal{P} \sim_\epsilon \mathcal{Q} \circ R$.]*

Remark 2. *1. Generalized from the tracing index in symbolic situation, the approximate tracing index ϵ-index$(\mathcal{P})$ of $\mathcal{P}$ can be defined as the number of common roots of $H_i^{\mathcal{P}}$, $i = 1, 2, 3$ for all but a finite number of values of the variable s with respect to a given tolerance ϵ. One can find that the ϵ-index$(\mathcal{P})$ can be obtained by computing the ϵ-gcd.*

2. For the simplification of $\widetilde{\mathcal{Q}}$ in Step 7, we compute ϵ-gcd$(\widetilde{q}_{k,1}(t), \widetilde{q}_{k,2}(t))$, $k = 1, 2, 3$ under the given tolerance ϵ, and we remove it from $\widetilde{q}_{k,1}(t)$ and $\widetilde{q}_{k,2}(t)$. For this purpose, one may apply well known ϵ-gcd algorithms proposed for inexact polynomials (see for instance, [1, 4, 29]). Also, the ϵ-gcd computation in Step 2 can be computed by these algorithms. In addition, since the polynomials $H_i^{\mathcal{P}}$, $i = 1, 2, 3$ and the gcd are symmetric with two variables, we can also use the specially designed approximate gcd algorithm (see Subsection 5.2.1).

3. Similarly as in the symbolic case, it holds that $\deg(\mathcal{P}) = \deg(\mathcal{Q})\deg(R)$, and one may prove that $\ell = \deg(R)$ (see Corollary 4 in [26]).

In order to study the closeness of the curves $\mathcal{C}$ and $\mathcal{D}$, where $\mathcal{D}$ is the curve defined by the simplified parametrization $\mathcal{Q}$, and $\mathcal{C}$ is the curve defined by $\mathcal{P}$ (we remind that $\mathcal{P} \sim_\epsilon \mathcal{Q} \circ R$, where $R(t) \in \mathbb{C}(t) \setminus \mathbb{C}$, and ϵ-index$(\mathcal{P}) = \deg(R) = \deg_t(S_\epsilon^{\mathcal{P}}) = \ell$ and ϵ-index$(\mathcal{Q}) = 1$), we analyze the behavior at affine points. For this purpose, it is assumed that $\deg(p_{i,1}) = \deg(p_{i,2})$, and $\deg(q_{i,1}) = \deg(q_{i,2})$, $i = 1, 2, 3$ (otherwise, one can apply on both parametrizations a birational parameter transformation). In addition, let

$$\|p\| := \max\{\|p_{1,1}\|, \|p_{2,1}\|, \|p_{1,2}\|, \|p_{2,2}\|, \|p_{1,3}\|, \|p_{2,3}\|\},$$

$$\|q\| := \max\{\|q_{1,1}\|, \|q_{2,1}\|, \|q_{1,2}\|, \|q_{2,2}\|, \|q_{1,3}\|, \|q_{2,3}\|\}.$$

Finally, we also assume that equality $(\star)$ in Step 7.1 of the algorithm holds and then $\mathcal{Q}$ is an ϵ-proper reparametrization of $\mathcal{P}$. If equality $(\star)$ does not hold, then $\mathcal{Q}$ an $\overline{\epsilon}$-proper reparametrization of $\mathcal{P}$ (see Step 7.1 of Algorithm NPRSC). In this case, the formula obtained in Theorem 4 remains unchanged except that ϵ becomes $\overline{\epsilon}$.

Under these conditions, in order to analyze the behavior at affine points, we shall restrict to an interval where the parametrizations $\mathcal{P} = (p_1, p_2, p_3)$, $p_i = p_{i,1}/p_{i,2}$, and $\mathcal{Q} = (q_1, q_2, q_3)$, $q_i = q_{i,1}/q_{i,2}$, are both well defined. For this purpose, we consider the interval $I := (d_1, d_2) \subset \mathbb{R}$ satisfying that for all $t_0 \in I$, there exists $M \in \mathbb{N}$ such that $|q_{i,2}(R(t_0))| \geq M$, and $|p_{i,2}(t_0)| \geq M, i = 1, 2, 3$. Note that we can decompose $\mathbb{R}$ to a union of finitely many intervals, $I_j, j = 1, \ldots, n$, satisfying the above condition (that is, the interval without any root of the denominators of the parametrizations; see [21]). Then we shall reason similarly as in Theorem 4 for each interval $I_j, j = 1, \ldots, n$. The following theorem and Corollary 1 follow from Theorem 5 and Corollary 7 in Section 4 of [26].

Theorem 4. *The following statements hold:*

1. *Let $I := (d_1, d_2)$ be an interval in $\mathbb{R}$, and $M \in \mathbb{N}$ be such that for every $t_0 \in I$, it holds that $|q_{i,2}(R(t_0))| \geq M$, and $|p_{i,2}(t_0)| \geq M$ for $i = 1, 2, 3$. Let $d := \max\{|d_1|, |d_2|\}$. Then, for every $t_0 \in I$,*

$$|p_i(t_0) - q_i(R(t_0))| \leq 2/M^2 \epsilon\, \zeta \|p\|\|q\|, \quad i = 1, 2, 3, \quad \textit{where}$$

$$\zeta = \begin{cases} \dfrac{d^{\deg(\mathcal{P})+1}}{(d-1)^{1/\ell}} & \textit{if } d > 1, \\ \dfrac{1}{(1-d)^{1/\ell}} & \textit{if } d < 1, \\ \ell^{1/\ell}\deg(\mathcal{P})^{1/\ell} & \textit{if } d = 1. \end{cases}$$

2. *$\mathcal{C}_{t\in I}$ is contained in the offset region of $\mathcal{D}_{s\in J}$ at distance $4\sqrt{2}/M^2\epsilon\,\zeta\,\|p\|\|q\|$, where $J = R(I)$.*
3. *$\mathcal{D}_{s\in J}$ is contained in the offset region of $\mathcal{C}_{t\in I}$ at distance $4\sqrt{2}/M^2\epsilon\,\zeta\,\|p\|\|q\|$, where $J = R(I)$.*

From Theorem 4, we deduce the following corollary:

Corollary 1. *Under the conditions of Theorem 4, it holds that:*

1. *If $d \geq 2$, then $\zeta \leq d^{\deg(\mathcal{P})+1}$.*
2. *If $1 < d < 2$, then $\zeta \leq 2^{\deg(\mathcal{P})+1}$.*

In the following, we illustrate Algorithm NPRSC by one example.

Example 3. *Let $\epsilon = 0.0001$, and the rational curve $\mathcal{C}$ defined by the parametrization*

$\mathcal{P}(t) = (t^6 - 3.0002t^5 + 6.0001t^4 - 7t^3 + 6t^2 - 3.0001t + 6.0001,\ t^6 - 3.0001t^5 + 6.0002t^4 - 7t^3 + 4.0001t^2 - t,\ t^4 - 2.0001t^3 + 3t^2 - 2.0001t) \in \mathbb{C}(t)^3.$

First, one may check that $\epsilon\text{-gcd}(p_{i,1}, p_{i,2}) = 1,\ i = 1,2,3,$ using statement 2 in Remark 2. In Step 1 of the algorithm, we compute the polynomials $H_i^{\mathcal{P}}(t,s),\ i = 1,2,3.$ Now, we determine the polynomial $S_\epsilon^{\mathcal{P}}$, and we get that $S_\epsilon^{\mathcal{P}}(t,s) = C_0(t) + C_1(t)s + C_2(t)s^2$, where

$C_0(t) = t(2530603171874t - 2530771894500),\ C_1(t) = -1125t + 2530771894500,\ C_2(t) = -2530603170749.$

Then, $\epsilon\text{-index}(\mathcal{P}) = \deg_t(S_\epsilon^{\mathcal{P}}) = 2$ (see Definition 1). Now, we apply Step 4 of the algorithm, and we consider

$$R(t) = \frac{C_0(t)}{C_3(t)} = \frac{-2t(1265301585937t - 1265385947250)}{2530603170749}.$$

In Steps 5 and 6 of the algorithm, we determine the polynomials $L_k(s, x_k)$, and we compute the root in the variable x_k of the polynomial $\frac{\partial L_k}{\partial x_k}(s, x_k)$, $k = 1,2,3$. We get the rational parametrization $\widetilde{\mathcal{Q}}(t)$. We simplify $\widetilde{\mathcal{Q}}$ by removing the approximate gcd from the numerator and denominator of each component of $\widetilde{\mathcal{Q}}$ (see statement 2 in Remark 2), and we return the curve $\mathcal{D}$ defined by the ϵ-numerical reparametrization

$\mathcal{Q}(t) = (t^3 + 2.999699996t^2 + 2.999899978t + 6.000099998,$
$t^3 + 3.000050012t^2 + 1.000450055t + 0.0001333620053, t^2 + 1.999916672t - 0.00003333377739).$

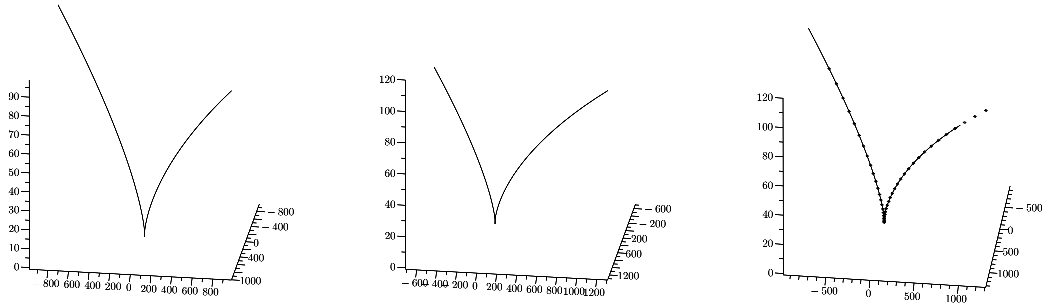
One may check that the equality of Step 7.1 holds. Then, $\mathcal{Q}$ is an $\overline{\epsilon}$-proper reparametrization of $\mathcal{P}$. In Figure 3, we plot the input curve $\mathcal{C}$ and the output curve $\mathcal{D}$.

We next perform error analysis by Theorem 4. Let the interval $I = (0, 0.5)$. Thus, $d = 0.5$. Let $M \in \mathbb{N}$ be such that for every $t_0 \in I$, it holds that $|q_{i,2}(R(t_0))| \geq M$ and $|p_{i,2}(t_0)| \geq M$, for $i = 1,2$. We have that $M = 1$. Then, by Theorem 4 we deduce that

$$\zeta = \frac{1}{(1-d)^{1/\ell}} = 1.414213562,$$

and for every $t_0 \in I$, it holds that

$$|p_i(t_0) - q_i(R(t_0))| < 2/M^2\overline{\epsilon}\,\zeta\|p\|\|q\| = 0.01187959191, \quad i = 1,2,3.$$



a) Input curve $\mathcal{C}$

b) Output curve $\mathcal{D}$

c) Both curves $\mathcal{C}$ and $\mathcal{D}$ (dash)

Figure 3: The numerical input space curve (a) and the approximated proper reparametrization (b) computed using Algorithm NPRSC. In the right figure (c), one may check that both curves are very close.

5. Partially Proper Reparametrization of Surfaces

In Section 4, we dealt with both, the symbolic and numerical problem of computing a rational proper reparametrization of a given improperly parametrized algebraic space curve. The symbolic case is based on the results presented in Sections 2 and 3 of [17] and the numerical case is deduced from the results in Sections 4 and 5 of [26].

In this section, we consider the reparametrization problem (also from both points of view, the symbolic and the numerical situations) but for a given algebraic rational surface. For this case, although it is known from Castelnuovo's Theorem that unirationality and rationality are equivalent over algebraically closed fields, the problem is not solved computationally. That is, there does not exist an algorithm that computes the proper reparametrization. Only some algorithms to find proper parametrization in several special cases are proposed (see [11, 15, 17, 18]).

In particular, in [18], *given an algebraically closed field* $\mathbb{K}$, *and* $\mathcal{P}(\overline{t}) \in \mathbb{K}(\overline{t})^3$, $\overline{t} = (t_1, t_2)$, *a rational parametrization of a surface* $\mathcal{V}$, *it is computed a parametrization of* $\mathcal{V}$, $\mathcal{Q}(\overline{t}) \in \mathbb{K}(\overline{t})^3$, *and* $R(\overline{t}) = (S(\overline{t}), T(S(\overline{t}), t_2))$, $S, T \in \mathbb{K}(\overline{t})$, *such that*

$$\mathcal{P}(\overline{t}) = \mathcal{Q}(R(\overline{t})), \qquad \textit{and} \quad \deg(\phi_{\mathcal{P}}) = \deg(\phi_{\mathcal{Q}}) \deg_{t_1}(S) \deg_{t_2}(T)$$

(see Theorem 4 in [18]). Note that from the above equality, it is deduced that if $\mathcal{Q}$ is not proper and $\deg_{t_1}(S)\deg_{t_2}(T) \neq 1$, then $\deg(\phi_{\mathcal{Q}}) < \deg(\phi_{\mathcal{P}})$ (see Corollary 2 in [18]). In addition, it is established under which conditions $\mathcal{Q}$ is proper (see statement 3 in Remark 3).

The method developed in Section 3 of [18] is obtained by applying the Algorithm PRSC to some *partial parametrizations associated to* $\mathcal{P}$ which are defined as the parametrizations $\mathcal{P}_i(t_j) := \mathcal{P}(\overline{t}) \in (\mathbb{K}(t_i))(t_j)^3$ (that is, $\mathcal{P}$ is defined over $\mathbb{K}(t_i)$), for $i, j \in \{1, 2\}$ and $i \neq j$ (see statement 4 in Remark 3). Observe that the partial parametrization $\mathcal{P}_i(t_j)$ $(i \neq j)$ defines a space curve over $\overline{\mathbb{K}(t_i)}$ (we refer to this space curve as the partial space curve). Note that, since $\mathcal{P}(\overline{t})$ is a surface parametrization its jacobian has rank 2, and therefore the gradient of $\mathcal{P}_i(t_j)$ (with respect to t_j) must have rank 1.

Hence, the goal of the algorithm is to properly reparametrize the partial parametrizations, $\mathcal{P}_i$ defined over the field $\mathbb{K}(t_i)$, $i = 1, 2$, by applying the partial parametrizations Algorithm PRSC. The algorithm outputs a rational parametrization $\mathcal{Q}(\overline{t}) \in \mathbb{K}(\overline{t})^3$ of $\mathcal{V}$, and $R(\overline{t}) \in \mathbb{K}(\overline{t})^2$ such that $\mathcal{P}(\overline{t}) = \mathcal{Q}(R(\overline{t}))$, and $\deg(\phi_{\mathcal{Q}}) < \deg(\phi_{\mathcal{P}})$. If some additional properties hold, then $\mathcal{Q}$ is proper (see statement 3 in Remark 3).

By generalizing the tracing index formula (5) of a given rational space curve for the parametrization $\mathcal{P}_i(t_j)$, we get that

$$\deg(\phi_{\mathcal{P}_i}) = \deg_{t_j}(\gcd(H_1^{\mathcal{P}_i}(t_j,s_j), H_2^{\mathcal{P}_i}(t_j,s_j), H_3^{\mathcal{P}_i}(t_j,s_j))), i,j \in \{1,2\} \text{ and } i \neq j$$

since $\mathcal{P}_i(t_j)$ is regarded as a rational space curve with respect to t_j over $\overline{\mathbb{K}(t_i)}$. Note that $\deg(\phi_{\mathcal{P}_1})\deg(\phi_{\mathcal{P}_2})$ is a factor of $\deg(\phi_{\mathcal{P}})$. In addition, $\deg(\phi_{\mathcal{P}_1})\deg(\phi_{\mathcal{P}_2}) = \deg(\phi_{\mathcal{P}})$ if some additional properties hold (see statement 3 in Remark 3).

Algorithm 3 (Reparametrization for Surfaces (RS)).
<u>Input:</u> *an algebraically closed field* $\mathbb{K}$*, and a rational affine parametrization*

$$\mathcal{P}(\,\overline{t}\,) = \big(p_1(\,\overline{t}\,), p_2(\,\overline{t}\,), p_3(\,\overline{t}\,)\big) \in \mathbb{K}(\,\overline{t}\,)^3, \quad p_i(\,\overline{t}\,) = \frac{p_{i,1}(\,\overline{t}\,)}{p_{i,2}(\,\overline{t}\,)}, \quad \gcd(p_{i,1}, p_{i,2}) = 1,\ i = 1,2,3$$

of an algebraic surface $\mathcal{V}$*.*
<u>Output:</u> *a rational parametrization* $\mathcal{Q}(\,\overline{t}\,) = \big(q_1(\,\overline{t}\,), q_2(\,\overline{t}\,), q_3(\,\overline{t}\,)\big) \in \mathbb{K}(\,\overline{t}\,)^3$*,* $\quad q_i(\,\overline{t}\,) = \frac{q_{i,1}(\,\overline{t}\,)}{qi,2(\,\overline{t}\,)}$*, of* $\mathcal{V}$*, and* $R(\,\overline{t}\,) \in (\mathbb{K}(\,\overline{t}\,) \setminus \mathbb{K})^2$ *such that* $\mathcal{P}(\,\overline{t}\,) = \mathcal{Q}(R(\,\overline{t}\,))$*, and* $1 \leq \deg(\phi_{\mathcal{Q}}) < \deg(\phi_{\mathcal{P}})$*.*

1. *Check whether* $\mathcal{P}_1$ *and* $\mathcal{P}_2$ *are proper (apply Steps 1 and 2 of* Algorithm PRSC*). In the affirmative case,* Return *the message "you cannot apply the algorithm". Otherwise, go to Step 2 if* $\mathcal{P}_2$ *is not proper, or go to Step 3 if* $\mathcal{P}_1$ *is not proper.*
2. *If* $\mathcal{P}_2$ *is not proper do:*
   - 2.1. *Apply* Algorithm PRSC *to* $\mathcal{P}_2$*. [This algorithm returns a parametrization* $\mathcal{M}(\,\overline{t}\,) \in \mathbb{K}(\,\overline{t}\,)^3$*, and* $S(\,\overline{t}\,) \in \mathbb{K}(\,\overline{t}\,)$ *such that the partial parametrization associated to* $\mathcal{M}$*,* $\mathcal{M}_2(t_1) \in (\mathbb{K}(t_2))(t_1)^3$*, is proper and* $S_2(t_1) \in (\mathbb{K}(t_2))(t_1)$ *satisfies* $\mathcal{P}_2(t_1) = \mathcal{M}_2(S_2(t_1))$*.]*
   - 2.2. *Check whether the partial parametrization associated to* $\mathcal{M}$*,* $\mathcal{M}_1(t_2) \in (\mathbb{K}(t_1))(t_2)^3$*, is proper (apply Steps 1 and 2 of* Algorithm PRSC*). In the affirmative case,* Return $\mathcal{Q} := \mathcal{M}$*, and* $R(\,\overline{t}\,) := (S(\,\overline{t}\,), t_2)$*. Otherwise, go to Step 2.3.*
   - 2.3. *Apply* Algorithm PRSC *to the partial parametrization associated to* $\mathcal{M}$*,* $\mathcal{M}_1(t_2)$*. [This algorithm returns a parametrization* $\mathcal{Q}(\,\overline{t}\,) \in \mathbb{K}(\,\overline{t}\,)^3$*, and* $T(\,\overline{t}\,) \in \mathbb{K}(\,\overline{t}\,)$ *such that the partial parametrization associated to* $\mathcal{Q}$*,* $\mathcal{Q}_1(t_2) \in (\mathbb{K}(t_1))(t_2)^3$*, is proper and* $T_1(t_2) \in (\mathbb{K}(t_1))(t_2)$ *satisfies* $\mathcal{M}_1(t_2) = \mathcal{Q}_1(T_1(t_2))$*.]*
   - 2.4. *Check whether the partial parametrization associated to* $\mathcal{Q}$*,* $\mathcal{Q}_2(t_1) \in (\mathbb{K}(t_2))(t_1)^3$*, is proper (apply Steps 1 and 2 of* Algorithm PRSC*). In the affirmative case,* Return *the reparametrization* $\mathcal{Q}$*, and* $R(\,\overline{t}\,) := (S(\,\overline{t}\,), T(S(\,\overline{t}\,), t_2))$*. Otherwise,* Return *the reparametrization* $\mathcal{Q}$*,* $R(\,\overline{t}\,) := (S(\,\overline{t}\,), T(S(\,\overline{t}\,), t_2))$*, and the message "you may apply the algorithm again (Step 2) to* $\mathcal{Q}_2$*".*
3. *If* $\mathcal{P}_1$ *is not proper, apply Step 2.3 to* $\mathcal{P}$ *and* $\mathcal{P}_1$*.*

Remark 3. *Observe that:*

1. *If* $\deg(\phi_{\mathcal{P}_i}) = 1$ *for* $i = 1,2$*, the algorithm does not start.*
2. *The algorithm returns a rational reparametrization* $\mathcal{Q}$ *of* $\mathcal{P}$*. More precisely, we get a rational parametrization* $\mathcal{Q}$*, and* $R(\,\overline{t}\,) \in \mathbb{K}(\,\overline{t}\,)^2$ *such that* $\mathcal{P} = \mathcal{Q}(R)$*. The output parametrization* $\mathcal{Q}$ *may not be proper (see statement 3) but it holds that* $1 \leq \deg(\phi_{\mathcal{Q}}) < \deg(\phi_{\mathcal{P}})$ *(see Corollary 2 in [18]), if* $\deg(\phi_{\mathcal{P}_i}) \neq 1$ *for some* $i = 1,2$ *(see statement 1).*
3. *Let* $\mathcal{Q}$ *be the output parametrization obtained by* Algorithm RS*, and let us assume that* $\deg(\phi_{\mathcal{Q}_i}) = 1$*, for* $i = 1,2$*. Then, note that the algorithm cannot be applied again. Under these conditions,* $\deg(\phi_{\mathcal{Q}}) = 1$ *if and only if* $\deg(\phi_{\mathcal{M}}) = \deg_{t_2}(T)$ *if and only if* $\deg(\phi_{\mathcal{P}}) = \deg_{t_1}(S)\deg_{t_2}(T)$ *(see Corollary 3 in [18]).*

4. *In order to apply* Algorithm PRSC, *we need that the space curves defined by the auxiliary parametrizations $\mathcal{P}_i$ are over a field (see Remark 1). Observe that in our situation, they are defined over the updated field $\mathbb{K} := \mathbb{K}(t_i)$.*
5. *Note that the algorithm does not loop forever. If it starts, it always outputs a reparametrization $\mathcal{Q}$ satisfying $1 \leq \deg(\phi_{\mathcal{Q}}) < \deg(\phi_{\mathcal{P}})$ (see Corollary 2 in [18]). In addition, if $\mathcal{Q}_2$ is not proper, one may apply the algorithm again to further decrease $\deg(\phi_{\mathcal{Q}})$ (see Step 2.4). If the algorithm can be applied (see statement 1) till we get $\deg(\phi_{\mathcal{Q}}) = 1$, the we obtain a proper reparametrization and then, the reparametrization problem is solved.*

Next, we illustrate Algorithm RS with an example, where different steps are applied.

Example 4. *Let $\mathcal{V}$ be a rational surface defined over the field of the complex numbers, $\mathbb{C}$, by the parametrization*

$$\mathcal{P}(\bar{t}\,) = \big(-(3t_2^8t_1^4 + 2t_2^6t_1^6 + 2t_2^{10}t_1^2 + t_2^4t_1^8 + t_2^{12} - t_2^4t_1^2 - t_2^2t_1^4 - t_2^6 + 2t_2^2)/t_2^2,$$

$$-t_1^2t_2^2 - t_1^4 - t_2^4 + 3t_1^4t_2^4 + 2t_1^6t_2^2 + 2t_1^2t_2^6 + t_1^8 + t_2^8 + t_2^2 + t_2^6, 3 + t_1^2t_2^2 + t_1^4 + t_2^4\big) \in \mathbb{C}(\bar{t}\,)^3.$$

We apply Algorithm RS. *For this purpose, in Step 1, we apply* Algorithm PRSC, *and we find that*

$$S^{\mathcal{P}_2}(t_1, s_1) = (s_1 - t_1)(s_1 + t_1)(s_1^2 + t_2^2 + t_1^2) \in (\mathbb{C}[t_2])[t_1, s_1]$$

which implies that $\mathcal{P}_2(t_1)$ is not proper (in fact, $\deg(\phi_{\mathcal{P}_2}) = \deg_{t_1}(S^{\mathcal{P}_2}) = 4$). Thus, we go to Step 2 and we apply Algorithm PRSC *to $\mathcal{P}_2$. We obtain $S_2(t_1) = -t_1^2t_2^2 - t_1^4 \in (\mathbb{C}[t_2])[t_1]$. Furthermore, we determine the polynomials*

$$L_i(s_1, x_i) = \mathrm{Res}_{t_1}(G_i^{\mathcal{P}_2}(\bar{t}\,, x_i), s_1 - S_2(t_1)) = (m_{i,2}(s_1)x_i - m_{i,1}(s_1))^{\deg_{t_1}(S_2)},$$

where $G_i^{\mathcal{P}_2}(\bar{t}\,, x_i) = x_ip_{i,2}(\bar{t}\,) - p_{i,1}(\bar{t}\,)$, for $i = 1, 2, 3$, and we get

$$\mathcal{M}(\bar{t}\,) = \left(-2 - t_2^{10} + t_2^4 + 2t_1t_2^6 - t_1 - t_1^2t_2^2, -t_2^4 + t_2^8 + t_2^2 + t_2^6 + t_1 - 2t_2^4t_1 + t_1^2, 3 + t_2^4 - t_1\right).$$

Now, in Step 2.2 of the algorithm, we apply Algorithm PRSC *to $\mathcal{M}_1(t_2) \in (\mathbb{C}(t_1))(t_2)^3$, and we find that*

$$S^{\mathcal{M}_1}(t_2, s_2) = (t_2 - s_2)(t_2 + s_2) \in (\mathbb{C}[t_1])[t_2, s_2].$$

Thus, since $\deg(\phi_{\mathcal{M}_1}) = \deg_{t_2}(S^{\mathcal{M}_1}) = 2$, we get that $\mathcal{M}_1$ is not proper. Then, we go to Step 2.3. We apply Algorithm PRSC *to $\mathcal{M}_1$, and we compute $T_1(t_2) = t_2^2 \in (\mathbb{C}[t_1])[t_2]$, and the polynomials*

$$L_i(s_2, x_i) = \mathrm{Res}_{t_2}(G_i^{\mathcal{M}_1}(\bar{t}\,, x_i), s_2 - T_1(t_2)) = (q_{i,2}(s_2)x_i - q_{i,1}(s_2))^{\deg_{t_2}(T_1)},$$

where $G_i^{\mathcal{M}_1}(\bar{t}\,, x_i) = x_im_{i,2}(\bar{t}\,) - m_{i,1}(\bar{t}\,)$, for $i = 1, 2, 3$. We obtain

$$\mathcal{Q}(\bar{t}\,) = \left(-2 - t_1 - t_2t_1^2 + t_2^2 + 2t_1t_2^3 - t_2^5, t_1 + t_1^2 + t_2 - t_2^2 - 2t_1t_2^2 + t_2^3 + t_2^4, 3 - t_1 + t_2^2\right).$$

Finally, in Step 2.4 of the algorithm, we apply Algorithm PRSC *to $\mathcal{Q}_2(t_1) \in (\mathbb{C}(t_2))(t_1)^3$. We get that $S^{\mathcal{Q}_2}(t_1, s_1) = s_1 - t_1 \in (\mathbb{C}[t_2])[t_1, s_1]$ which implies that $\mathcal{Q}_2$ is proper. Therefore,* Algorithm RS *outputs the parametrization $\mathcal{Q}(\bar{t}\,)$, and*

$$R(\bar{t}\,) = (S(\bar{t}\,), T(S(\bar{t}\,), t_2)) = (-t_1^2t_2^2 - t_1^4,\, t_2^2) \in \mathbb{C}(\bar{t}\,)^2.$$

5.1. Numerical Algorithm of Partially Reparametrization for Surfaces

In this subsection, we deal with the problem of proper reparametrization for surfaces from the numerical point of view. More precisely, the goal is to generalize Algorithm RS, using the ideas and definitions presented in Subsection 4.1, so that it can be applied to a given parametrization $\mathcal{P}$ with perturbed float coefficients.

The problem of numerical proper reparametrization for surfaces can be stated as follows: *given the field $\mathbb{C}$ of complex numbers, a tolerance $\epsilon > 0$, and a rational parametrization $\mathcal{P}(t_1, t_2) \in \mathbb{C}(t_1, t_2)^3$ of an algebraic surface $\mathcal{V}$ that is approximately improper (see Definition 5), compute a rational parametrization $\mathcal{Q}(t_1, t_2) \in \mathbb{C}(t_1, t_2)^3$ of an algebraic surface $\mathcal{W}$, and $R(t_1, t_2) \in (\mathbb{C}(t_1, t_2) \setminus \mathbb{C})^2$ such that $\mathcal{Q}$ is an ϵ-proper reparametrization of $\mathcal{P}$ (see Definition 8).* In addition, the closeness between $\mathcal{V}$ and $\mathcal{W}$ should be measured.

We do not intent to solve the above problem but a particular case using Algorithm RS, and the results presented in Subsection 4.1. In particular, the purpose of this subsection is to decrease the tracing index of the given parametrization $\mathcal{P}$ (see Definition 5) by analyzing the tracing index of the partial parametrizations associated to $\mathcal{P}$, $\mathcal{P}_1$ and $\mathcal{P}_2$ (see Algorithm RS and Subsection 4.1).

For this purpose, we first consider a tolerance $\epsilon > 0$, and a rational parametrization of a given algebraic surface $\mathcal{V}$

$$\mathcal{P}(\bar{t}\,) = \big(p_1(\bar{t}\,), p_2(\bar{t}\,), p_3(\bar{t}\,)\big) \in \mathbb{C}(\bar{t}\,)^3, \quad p_i(\bar{t}\,) = p_{i,1}(\bar{t}\,)/p_{i,2}(\bar{t}\,),$$

where ϵ-gcd$(p_{i,1}, p_{i,2}) = 1,\ i = 1, 2, 3$ and $\bar{t} = (t_1, t_2)$ (we remind that $\mathcal{P}$ is expected to be given with perturbed float coefficients).

Therefore, we need to give the concept of tracing index to the numerical situation for a given parametrization of an algebraic rational surface. Actually, we generalize Definition 1 and the *approximate tracing index of* $\mathcal{P}$ is defined as the common ϵ-roots in $\overline{\mathbb{C}(\bar{s})}$ of the polynomials $H_j^{\mathcal{P}}(\bar{t}\,,\bar{s}) = p_{j,1}(\bar{t}\,)p_{j,2}(\bar{s}) - p_{j,1}(\bar{s})p_{j,2}(\bar{t}\,),\ j = 1, 2, 3,$ w.r.t the variables $\bar{t}\,$.

Definition 5. *We define the* approximate tracing index *of $\mathcal{P}$ as the number of the common ϵ-roots in $\overline{\mathbb{C}(\bar{s})}$ of the polynomials $H_j^{\mathcal{P}}(\bar{t}\,,\bar{s}) = p_{j,1}(\bar{t}\,)p_{j,2}(\bar{s}) - p_{j,1}(\bar{s})p_{j,2}(\bar{t}\,),\ j = 1, 2, 3,$ w.r.t the variables $\bar{t}\,$. We denote it as ϵ-*index$(\mathcal{P})$*. Furthermore, $\mathcal{P}$ is said to be* approximately improper *or* ϵ-improper *if* ϵ-index$(\mathcal{P}) > 1$. *Otherwise, $\mathcal{P}$ is said to be* approximately proper *or* ϵ-proper.

In the following definitions, we generalize Definitions 2 and 3.

Definition 6. *Given a non-zero polynomial $A \in \mathbb{C}[\bar{t}\,,\bar{s}]$ and $R(\bar{t}\,) = (r_1(\bar{t}\,), r_2(\bar{t}\,)) \in \mathbb{C}(\bar{t}\,)^2$, we say that $A(\bar{t}\,, R(\bar{t}\,)) \approx_\epsilon 0$ if $\|\mathrm{num}(A(\bar{t}\,, R(\bar{t}\,)))\| \leq \epsilon \|A\|$.*

Definition 7. *Let $R(\bar{t}\,) \in \mathbb{C}(\bar{t}\,)^2$ and $\mathcal{Q}(\bar{t}\,) = \big(q_1(\bar{t}\,), q_2(\bar{t}\,), q_3(\bar{t}\,)\big) \in \mathbb{C}(\bar{t}\,)^3$, $q_i(\bar{t}\,) = q_{i,1}(\bar{t}\,)/q_{i,2}(\bar{t}\,)$, ϵ-gcd$(q_{j,1}, q_{j,2}) = 1,\ j = 1, 2, 3,$ be a rational parametrization of a surface. We say that $\mathcal{P}(\bar{t}\,) \sim_\epsilon (\mathcal{Q} \circ R)(\bar{t}\,)$ if $H_j^{\mathcal{P}\mathcal{Q}}(\bar{t}\,, R(\bar{t}\,)) \approx_\epsilon 0$, where $H_j^{\mathcal{P}\mathcal{Q}}(\bar{t}\,,\bar{s}) = p_{j,1}(\bar{t}\,)q_{j,2}(\bar{s}) - q_{j,1}(\bar{s})p_{j,2}(\bar{t}\,),\ j = 1, 2, 3.$*

To introduce the notions of *ϵ-numerical reparametrization* and *ϵ-proper reparametrization*, we generalize Definition 4.

Definition 8. *Let $\mathcal{P}(\bar{t}\,) \in \mathbb{C}(\bar{t}\,)^3$ be a rational parametrization of a given surface $\mathcal{V}$. We say that a parametrization $\mathcal{Q}(\bar{t}) \in \mathbb{C}(\bar{t})^3$ is an ϵ-numerical reparametrization of $\mathcal{P}(\bar{t}\,)$ if there exists $R(\bar{t}\,) \in (\mathbb{C}(\bar{t}\,) \setminus \mathbb{C})^2$, such that $\mathcal{P} \sim_\epsilon \mathcal{Q} \circ R$. In addition, if* ϵ-index$(\mathcal{Q}) = 1$, *then we say that $\mathcal{Q}$ is an ϵ-proper reparametrization of $\mathcal{P}$.*

Now, we present the numerical algorithm that is based on the Algorithm RS.

Algorithm 4 (Numerical Reparametrization for Surfaces (NRS))**.**
<u>INPUT:</u> *a tolerance $\epsilon > 0$, and a rational affine parametrization*

$$\mathcal{P}(\overline{t}\,) = \big(p_1(\overline{t}\,),\, p_2(\overline{t}\,),\, p_3(\overline{t}\,)\big) \in \mathbb{C}(\overline{t}\,)^3, \quad p_i(\overline{t}\,) = p_{i,1}(\overline{t}\,)/p_{i,2}(\overline{t}\,), \quad \epsilon\text{-gcd}(p_{i,1}, p_{i,2}) = 1,\ i = 1,2,3$$

of an algebraic surface $\mathcal{V}$.
<u>OUTPUT:</u> *a rational parametrization $\mathcal{Q}(\overline{t}\,) = \big(q_1(\overline{t}\,),\, q_2(\overline{t}\,),\, q_3(\overline{t}\,)\big) \in \mathbb{C}(\overline{t}\,)^3$, $\quad q_i(\overline{t}\,) = \frac{q_{i,1}(\overline{t}\,)}{q_{i,2}(\overline{t}\,)}$, of $\mathcal{V}$, and $R(\overline{t}\,) \in (\mathbb{C}(\overline{t}\,) \setminus \mathbb{C})^2$ such that $\mathcal{P} \sim_\epsilon \mathcal{Q} \circ R$, and $1 \leq \epsilon\text{-index}(\mathcal{Q}) < \epsilon\text{-index}(\mathcal{P})$.*

1. *Check whether $\epsilon\text{-index}(\mathcal{P}_1) = 1$ ($\mathcal{P}_1$ is ϵ-proper) and $\epsilon\text{-index}(\mathcal{P}_2) = 1$ ($\mathcal{P}_2$ is ϵ-proper). For this purpose, apply Steps 1 and 2 of* Algorithm NPRSC*. In the affirmative case,* RETURN *the message "you cannot apply the algorithm". Otherwise, go to Step 2 if $\epsilon\text{-index}(\mathcal{P}_2) \neq 1$, or go to Step 3 if $\epsilon\text{-index}(\mathcal{P}_1) \neq 1$.*
2. *If $\mathcal{P}_2$ is not ϵ-proper do:*
   - *2.1. Apply* Algorithm NPRSC *to $\mathcal{P}_2$. [This algorithm returns a parametrization $\mathcal{M}(\overline{t}\,) \in \mathbb{C}(\overline{t}\,)^3$, and $S(\overline{t}\,) \in \mathbb{C}(\overline{t}\,)$ such that the partial parametrization associated to $\mathcal{M}$, $\mathcal{M}_2(t_1) \in (\mathbb{C}(t_2))(t_1)^3$, is ϵ-proper and $S(\overline{t}\,) \in \mathbb{C}(\overline{t}\,)$ satisfies $\mathcal{P} \sim_\epsilon \mathcal{M} \circ (S, t_2)$.]*
   - *2.2. Check whether the partial parametrization associated to $\mathcal{M}$, $\mathcal{M}_1(t_2) \in (\mathbb{C}(t_1))(t_2)^3$, is ϵ-proper (apply Steps 1 and 2 of* Algorithm NPRSC*). In the affirmative case,* RETURN *$\mathcal{Q} := \mathcal{M}$, and $R(\overline{t}\,) := (S(\overline{t}\,), t_2)$. Otherwise, apply Step 3 to $\mathcal{M}$ and $\mathcal{M}_1$.*
3. *If $\mathcal{P}_1$ is not ϵ-proper do:*
   - *3.1. Apply* Algorithm NPRSC *to $\mathcal{P}_1$. [This algorithm returns a parametrization $\mathcal{N}(\overline{t}\,) \in \mathbb{C}(\overline{t}\,)^3$, and $T(\overline{t}\,) \in \mathbb{C}(\overline{t}\,)$ such that the partial parametrization associated to $\mathcal{P}$, $\mathcal{P}_1(t_2) \in (\mathbb{C}(t_1))(t_2)^3$, is ϵ-proper and $T(\overline{t}\,) \in \mathbb{C}(\overline{t}\,)$ satisfies $\mathcal{P} \sim_\epsilon \mathcal{N} \circ (t_1, T)$.]*
   - *3.2. Check whether the partial parametrization associated to $\mathcal{N}$, $\mathcal{N}_2(t_1) \in (\mathbb{C}(t_2))(t_1)^3$, is ϵ-proper (apply Steps 1 and 2 of* Algorithm NPRSC*). In the affirmative case,* RETURN *the reparametrization $\mathcal{Q} := \mathcal{N}$, and $R(\overline{t}\,) := (t_1, T(\overline{t}\,))$. Otherwise,* RETURN *the reparametrization $\mathcal{Q} := \mathcal{N}$, and $R(\overline{t}\,) := (t_1, T(\overline{t}\,))$, and the message "you may apply the algorithm again (Step 2) to $\mathcal{Q}$ and $\mathcal{Q}_2$)".*

We here illustrate the above algorithm with an example.

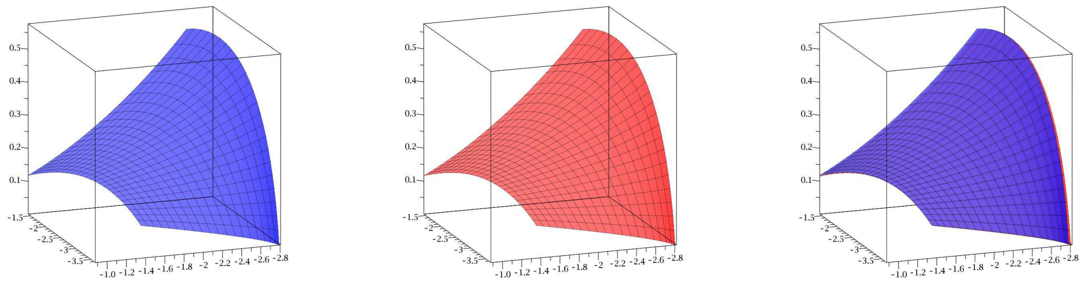
Example 5. *Let $\epsilon = 0.001$, and the rational surface $\mathcal{V}$ defined by the rational parametrization $\mathcal{P}(\overline{t}\,) = \big(p_1(\overline{t}\,),\, p_2(\overline{t}\,),\, p_3(\overline{t}\,)\big) \in \mathbb{C}(\overline{t}\,)^3$ where*

$$\begin{aligned}
p_1(\overline{t}\,) &= \frac{-3\,{t_1}^4-4\,{t_1}^2{t_2}^2-6.003\,{t_1}^3-2\,{t_1}^2t_2-4.003\,t_1\,{t_2}^2-4\,{t_2}^3-3\,{t_1}^2-2.003\,t_1\,t_2-7\,{t_2}^2-0.006\,{t_1}^3t_2}{3\,{t_1}^4+5.995\,{t_1}^3+3\,{t_1}^2+3\,{t_2}^2+0.002\,{t_1}^3{t_2}^2-0.004\,t_1\,{t_2}^2-0.003\,t_1\,t_2},\\
p_2(\overline{t}\,) &= \frac{-{t_1}^4t_2+{t_1}^4-2\,{t_1}^3t2+2\,{t_1}^2{t_2}^2+1.999\,{t_1}^3-11\,t1^2t_2+2\,t_1\,{t_2}^2+3\,{t_2}^3+t1^2-10\,t_1\,t_2-2.997\,{t_2}^2-0.001\,t1^3{t_2}^2+0.009}{2\,{t_1}^4+3.999\,{t_1}^3-{t_2}^3+2\,{t_1}^2+2.003\,{t_2}^2-0.001\,{t_1}^3{t_2}^2+0.009},\\
p_3(\overline{t}\,) &= \frac{-8\,{t_2}^3+8\,{t_2}^2-0.001\,{t_1}^4{t_2}^2+0.001\,{t_1}^4-0.001\,{t_1}^2{t_2}^2+0.002\,{t_1}^2}{2.001\,{t_1}^4+4\,{t_1}^3+{t_2}^3+2.002\,{t_1}^2+2\,{t_2}^2-0.001\,{t_1}^4{t_2}^2-0.001\,{t_1}^2{t_2}^2}.
\end{aligned}$$

This parametrization is approximate improper. We can show this fact by showing the figures of two patches with intervals $(t_1, t_2) \in (0.5, 1.0) \times (0.5, 1.0)$ and $(t_1, t_2) \in (-2.0, -1.5) \times (0.5, 1.0)$ and noting that the two patches are very close (see Figure 4).

This numerical surface has proper support even after sparse approximation, hence the support transformation method (see Section 3) is not applicable and we will apply Algorithm NRS*. For this purpose, in Step 1 , we apply* Algorithm NPRSC*, and we find that $\mathcal{P}_2$ is not ϵ-proper. More precisely, we compute the polynomials $H_i^{\mathcal{P}_2}(t_1, s_1)$, $i = 1, 2, 3$, and we have that*

$$S_\epsilon^{\mathcal{P}_2}(t_1, s_1) = C_0(t_1, t_2) + C_1(t_1, t_2)s_1 + C_2(t_1, t_2)s_1^2 \in (\mathbb{C}[t_2])[t_1, s_1],$$



a) A patch at $(-2.0,-1.5)\times(0.5,1.0)$ b) A patch at $(0.5,1.0)\times(0.5,1.0)$ c) Two patches of a) and b)

Figure 4: An approximate improper parametrization: the patches with intervals $(t_1,t_2)\in(0.5,1.0)\times(0.5,1.0)$ (a) and $(t_1,t_2)\in(-2.0,-1.5)\times(0.5,1.0)$ (b) are very close (c).

where

$$C_0(t_1,t_2) = -1.0\ (-1.64515\,t_1 + 0.00677173\,t_2 - 1.57307)\,t_1,$$

$$C_1(t_1,t_2) = -1.57307 + 0.00677173\,t_2, \quad C_2(t_1,t_2) = -1.64515.$$

Then, ϵ-index$(\mathcal{P}_2) = \deg_{t_1}(S_\epsilon^{\mathcal{P}_2}) = 2$ *(see Definition 1). Now, we apply Step 2 of* Algorithm NRS *which consists essentially in applying* Algorithm NPRSC *to* $\mathcal{P}_2$. *For this purpose, from* $S_\epsilon^{\mathcal{P}_2}(t_1,s_1)$, *we consider*

$$S(\bar{t}\,) = C_0(\bar{t}\,)/C_2(\bar{t}\,) = 0.607848\ (-1.64515\,t_1 + 0.00677173\,t_2 - 1.57307)\,t_1.$$

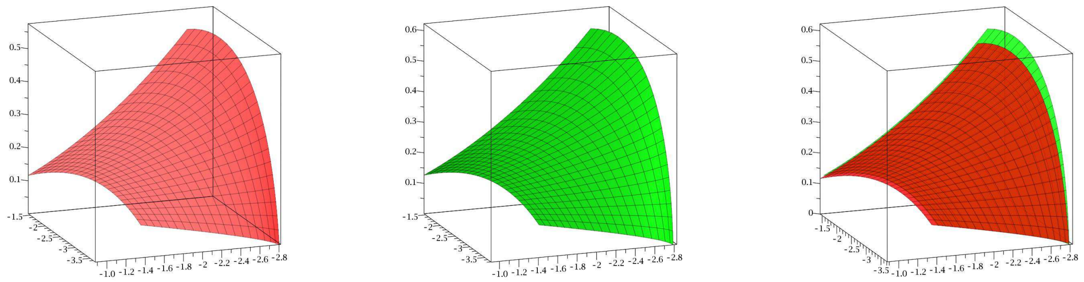
We determine the polynomials $L_k(s_1,x_k)\in(\mathbb{C}[t_2])[s_1,x_k]$, *and we compute the root in the variable* x_k *of the polynomial* $\frac{\partial L_k}{\partial x_k}(s_1,x_k)$, $k=1,2,3$. *We get the* ϵ*-numerical reparametrization* $\mathcal{M}(\bar{t}\,) = (m_1(\bar{t}\,),\ m_2(\bar{t}\,),\ m_3(\bar{t}\,))$, *where*

$m_1(\bar{t}\,) = (-0.3794280\,t_2{}^3 + 0.3808158\,t_1\,t_2{}^2 - 0.0001517840\,t_1{}^2 t_2 - 0.00009970054\,t_1{}^3 - 0.6568573\,t_2{}^2 + 0.1891787\,t_1\,t_2 - 0.2852680\,t_1{}^2 + 0.003779722\,t_2 - 0.01098717\,t_1 - 0.0004537563)/(0.2851180\,t_2{}^2 + 0.001025204\,t_1\,t_2 + 0.2852858\,t_1{}^2 + 0.0003979236\,t_2 + 0.01070202\,t_1 + 0.0005952313)$,
$m_2(\bar{t}\,) = (-0.2609498\,t_2{}^3 + 0.1746983\,t_1\,t_2{}^2 + 0.08685465\,t_1{}^2 t_2 - 0.00004887348\,t_1{}^3 + 0.2613865\,t_2{}^2 - 0.8657369\,t_1\,t_2 - 0.08729423\,t_1{}^2 - 0.01913675\,t_2 - 0.004494467\,t_1 - 0.001014182)/(0.08703259\,t_2{}^3 - 0.00008747335\,t_1\,t_2{}^2 + 0.0000002466221\,t_1{}^2 t_2 + 0.000005685722\,t_1{}^3 - 0.1741250\,t_2{}^2 - 0.001190926\,t_1\,t_2 - 0.1739044\,t_1{}^2 - 0.0002466680\,t_2 - 0.007163162\,t_1 - 0.001016453)$,
$m_3(\bar{t}\,) = (0.6835082\,t_2{}^3 - 0.0003170363\,t_1\,t_2{}^2 + 0.00004202926\,t_1{}^2 t_2 + 0.00009821240\,t_1{}^3 - 0.6834193\,t_2{}^2 - 0.0007039878\,t_1\,t_2 - 0.0001782237\,t_1{}^2 + 0.0001590481\,t_2 + 0.0009338094\,t_1 - 0.0002863107)/(-0.08545207\,t_2{}^3 - 0.0003265848\,t_1\,t_2{}^2 - 0.00003466410\,t_1{}^2 t_2 + 0.00006390510\,t_1{}^3 - 0.1707625\,t_2{}^2 - 0.0004194026\,t_1\,t_2 - 0.1710274\,t_1{}^2 - 0.00008676312\,t_2 - 0.006719450\,t_1 - 0.0002456190)$.

Now, in Step 2.2 of the algorithm, we apply Algorithm PRSC *to* $\mathcal{M}_1(t_2)\in(\mathbb{C}(t_1))(t_2)^3$, *and we find that* $S_\epsilon^{\mathcal{M}_1}(t_2,s_2) = t_2 - s_2 \in (\mathbb{C}[t_1])[t_2,s_2]$. *Thus, since* $\deg(\phi_{\mathcal{M}_1}) = \deg_{t_2}(S_\epsilon^{\mathcal{M}_1}) = 1$, *we get that* $\mathcal{M}_1$ *is proper and thus,* Algorithm NRS *returns* $\mathcal{Q}(\bar{t}\,) := \mathcal{M}(\bar{t}\,)$ *and* $R(\bar{t}\,) = (S(\bar{t}\,), t_2)$. *The result is shown in Figure 5.*

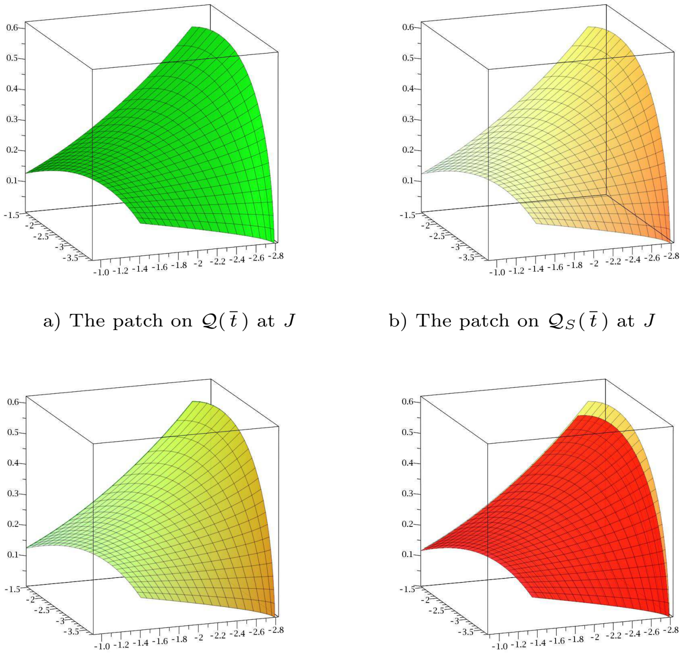
The output parametrization can be further simplified by sparse approximation in some situations. In this example, a sparse approximation of $\mathcal{Q}(\bar{t}\,)$ *is* $\mathcal{Q}_S(\bar{t}\,) = \big(q_1(\bar{t}\,), q_2(\bar{t}\,), q_3(\bar{t}\,)\big)$, *where*

$$q_1(\bar{t}\,) = \frac{-0.0385129\,t_1 + 0.0132489\,t_2 + 1.33486\,t_1\,t_2{}^2 + 0.663120\,t_1\,t_2 - 0.999938\,t_1{}^2 - 2.30245\,t_2{}^2 - 1.32999\,t_2{}^3}{0.00208644 + 0.0375133\,t_1 + 0.00359360\,t_1\,t_2 + 1.0\,t_1{}^2 + 0.999412\,t_2{}^2},$$
$$q_2(\bar{t}\,) = \frac{-0.00582445 - 0.0258117\,t_1 - 0.109902\,t_2 + 0.498806\,t_1{}^2 t_2 + 1.00329\,t_1\,t_2{}^2 - 4.97193\,t_1\,t_2 - 0.501331\,t_1{}^2 + 1.50114\,t_2{}^2 - 1.49863\,t_2{}^3}{-0.00583749 - 0.0411381\,t_1 - 0.00683949\,t_1\,t_2 - 0.998733\,t_1{}^2 - 1.0\,t_2{}^2 + 0.499828\,t_2{}^3},$$
$$q_3(\bar{t}\,) = \frac{0.00546000\,t_1 - 0.00411623\,t_1\,t_2 - 3.99596\,t_2{}^2 + 3.99648\,t_2{}^3}{-0.0392887\,t_1 - 1.0\,t_1{}^2 - 0.998451\,t_2{}^2 - 0.499640\,t_2{}^3}.$$



a) $\mathcal{P}(\bar{t})$ at $I = (0.5, 1.0) \times (0.5, 1.0)$ b) $\mathcal{Q}(\bar{t})$ at $J = R(I) = (-1.954, -0.726) \times (0.5, 1.0)$ c) Two patches of a) and b)

Figure 5: The input parametrization, $\mathcal{P}(\bar{t})$, at $I = (0.5, 1.0) \times (0.5, 1.0)$ (a) and its approximated proper reparametrization, $\mathcal{Q}(\bar{t})$, obtained from Algorithm NRS at $J = R(I) = (-1.954, -0.726) \times (0.5, 1.0)$. Both patches are very close (c).



a) The patch on $\mathcal{Q}(\bar{t})$ at J b) The patch on $\mathcal{Q}_S(\bar{t})$ at J

c) Two patches of a) and b) d) $\mathcal{P}(\bar{t})$ at I v.s. $\mathcal{Q}_S(\bar{t})$ at J

Figure 6: The output parametrization, $\mathcal{Q}(\bar{t})$, at J (a) and the sparse approximation of $\mathcal{Q}(\bar{t})$, $\mathcal{Q}_S(\bar{t})$, at J. Both patches are very close (c) and in fact, the input parametrization $\mathcal{P}(\bar{t})$ and the sparse approximation $\mathcal{Q}_S(\bar{t})$ are very close (d).

We can find that the sparse approximated parametrization $\mathcal{Q}_S(\bar{t})$, with much less terms and lower degrees, is very close to $\mathcal{Q}(\bar{t})$ and it approximates to the original surface well (see Figure 6) .

5.2. *Correctness of the* Algorithm NRS

In order to check the correctness of Algorithm NRS, we check the correctness of Step 2 (note that Step 3 is similar but with another variable). Let us assume that the output of the algorithm is given by the rational parametrization $\mathcal{Q}(\bar{t}\,) := \mathcal{M}(\bar{t}\,)$, and $R(\bar{t}\,) := (S(\bar{t}\,), t_2)$.

Under these conditions, some points have to be verified or clarified:

I. Since Algorithm NPRSC is applied to $\mathcal{P}_2(t_1) \in (\mathbb{C}(t_2))(t_1)$:
   (a) We have to clarify Steps 2 and 7, where an ϵ-gcd for polynomials with coefficients in $\mathbb{C}(t_2)$ has to be applied (see Subsection 5.2.1).
   (b) We have to prove that $\mathcal{P} \sim_\epsilon \mathcal{Q} \circ (S, t_1)$ (see Subsection 5.2.2).
   (c) We have to prove that $\mathcal{Q}_2(t_1) \in (\mathbb{C}(t_2))(t_1)^3$ is ϵ-proper (see Subsection 5.2.2).

II. It should be proved that ϵ-index$(\mathcal{Q}) <$ ϵ-index$(\mathcal{P})$ (see Subsection 5.2.2).

III. The error analysis has to be analyzed (see Subsection 5.2.3).

5.2.1. Computation of ϵ-gcd *(Steps 2 and 7)*

To find the ϵ-index in Step 2 of Algorithm NPRSC, we need the approximate gcd of three polynomials in $(\mathbb{C}(t_2))[t_1, s_1]$ (i.e. the coefficient field is $\mathbb{C}(t_2)$). Precisely, consider the three polynomials $H_1^{\mathcal{P}_2}(t_1, s_1), H_2^{\mathcal{P}_2}(t_1, s_1), H_3^{\mathcal{P}_2}(t_1, s_1) \in (\mathbb{C}(t_2))[t_1, s_1]$, and find an approximate gcd of them with a given error ϵ. Notice that $t_1 - s_1$ must be a factor of the three polynomials. Then, we may focus on $H_i(t_2, t_1, s_1) = H_i^{\mathcal{P}_2}(t_1, s_1)/(t_1 - s_1)(\mathbb{C}(t_2))[t_1, s_1], i = 1, 2, 3$. Let $D(t_2, t_1, s_1) = \epsilon\text{-gcd}(H_1, H_2, H_3)$ be their approximate gcd and $Q_1(t_2, t_1, s_1), Q_2(t_2, t_1, s_1)$, $Q_3(t_2, t_1, s_1)$ be the cofactors of $D(t_2, t_1, s_1)$ with respect to $H_i(t_2, t_1, s_1), i = 1, 2, 3$, respectively. The particular feature yields that our approximate gcd algorithm must satisfy the following requirements:

a). $\|H_i(t_2, t_1, s_1) - Q_i(t_2, t_1, s_1)D(t_2, t_1, s_1)\| \leq \epsilon, i = 1, 2, 3$;

b). $\deg_{t_1}(H_i) = \deg_{t_1}(Q_i) + \deg_{t_1}(D)$, $\deg_{s_1}(H_i) = \deg_{s_1}(Q_i) + \deg_{s_1}(D)$, $\deg_{t_2}(H_i) = \deg_{t_2}(Q_i) + \deg_{t_2}(D)$, $i = 1, 2, 3$;

c). $D(t_2, t_1, s_1) = D(t_2, s_1, t_1)$, $Q(t_2, t_1, s_1) = Q(t_2, s_1, t_1), i = 1, 2, 3$.

Note that the approximate gcd of the original polynomials $H_i^{\mathcal{P}_2}(t_1, s_1)$, $i = 1, 2, 3$ is $(t_1 - s_1)D(t_2, t_1, s_1)$.

Remark 4. *The conditions a) and b) come from the propositions of gcd for numerical case. The polynomials $H_i(t_2, t_1, s_1), i = 1, 2, 3$ are symmetric with respect to t_1 and s_1. This fact leads to conditions c) and helps us to reduce the redundant terms in numerical computations.*

Based on the above observation, we implement a particular approximate gcd algorithm satisfying the above requirements by the following steps:

1. Compute the total degree, k, of the approximate gcd of H_1, H_2, H_3, by several random univariate projection of H_1, H_2, H_3 by looking for the numerical rank of the corresponding univariate Sylvester matrices.

2. Construct the k-th Sylvester matrix of H_1, H_2, H_3, and then perform singular values decomposition (SVD) technique to obtain their cofactors satisfying the symmetric constraint, denoted by Q_1, Q_2, Q_3.

3. Apply least squares based polynomial division algorithm, to obtain the approximate gcd D from H_1, H_2, H_3 and the cofactors Q_1, Q_2, Q_3 satisfying the symmetric constraint.

Remark 5. *As pointed out in [29], there is an explicit relation between the total degree of gcd for multivariate polynomials and the dimension of null space of the associated generalized Sylvester matrix. Particularly, for univariate polynomials, the degree of the gcd is equivalent to the dimension of null space of* *the Sylvester matrix. In Step 1, we wish to determine the total degree of ϵ-gcd(H_1, H_2, H_3). Similar to [8], here we also exploit univariate projection technique to determine the total degree of ϵ-gcd(H_1, H_2, H_3). Concretely, we start with univariate projection to get the associated univariate polynomials, denoted by $\tilde{H}_1, \tilde{H}_2, \tilde{H}_3$, and to construct the Sylvester matrix $S_1 = S(\tilde{H}_1, \tilde{H}_2, \tilde{H}_3)$, proceed by determining what the degree of gcd, k, should be, that is, the numerical rank of S_1. Typically, when determining the rank numerically, one would specify a tolerance ϵ and find a gap in the singular values of S_1:*

$$\sigma_m \geq \cdots \geq \sigma_{m-k-1} > \epsilon \geq \sigma_{m-k} \geq \cdots \geq \sigma_1.$$

As shown in [8], randomized univariate projection technique can obtain the exact degree of ϵ-gcd(H_1, H_2, H_3) with probability one. To obtain the exact degree for our problem, we can try randomized projection for several times similar to [8]. Precisely, here we exploit randomized univariate projection for several times, to obtain the associated univariate gcds. Among the degrees of gcds, we select the maximal degree as the degree of gcd.

Sensitivity. Stated in Section 4.2 of [29], computing the approximate gcd is a regular problem and is not ill-posed. In this paper, the theoretical analysis provided in [29], including the sensitivity and error bound, can be generalized to our case: computing the approximate gcd of three polynomials.

Let $\mathbf{d}$,$\mathbf{h}_i$ and $\mathbf{q}_i$ be the coefficient vectors of the polynomial D, H_i, Q_i, respectively. It is seen that the problem for computing a gcd with degree k is to solve a quadratic system as following:

$$F(z) = \mathbf{b}, \tag{6}$$

with the form

$$F(\mathbf{z}) = \begin{pmatrix} \mathbf{r}^T\mathbf{d} - 1 \\ C(\mathbf{q}_1)\,\mathbf{d} \\ C(\mathbf{q}_2)\,\mathbf{d} \\ C(\mathbf{q}_3)\,\mathbf{d} \end{pmatrix}, \quad \mathbf{z} = \begin{pmatrix} \mathbf{d} \\ \mathbf{q}_1 \\ \mathbf{q}_2 \\ \mathbf{q}_3 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 0 \\ \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \end{pmatrix},$$

where the matrix C is the Cauchy matrix corresponding to the given coefficient vector, and $\mathbf{r}$ is a scaling vector. The least squares solution to $\|F(\mathbf{z}) - \mathbf{b}\|_2$ will be attained at $\mathbf{z}$ which satisfies

$$J(\mathbf{z})^H\,(F(\mathbf{z}) - \mathbf{b}) = 0,$$

where $J(\mathbf{z})$ is the Jacobian of $F(\mathbf{z})$,

$$J(\mathbf{z}) = \begin{pmatrix} \mathbf{r}^H & & & \\ C(\mathbf{q}_1) & C(\mathbf{d}) & & \\ C(\mathbf{q}_2) & & C(\mathbf{d}) & \\ C(\mathbf{q}_3) & & & C(\mathbf{d}) \end{pmatrix}. \tag{7}$$

The fact that $Q_i, i = 1, 2, 3$ are approximate coprime yields that the Jacobian $J(\mathbf{z})$ is of full rank, which means that its smallest singular value is strictly positive. In this stage, Proposition 2 in [29] can also be adapted to our case.

Proposition 1. *Let (H_1, H_2, H_3) and $(\tilde{H}_1, \tilde{H}_2, \tilde{H}_3)$ be two pairs of polynomials. Suppose D, Q_1, Q_2, Q_3 and $\tilde{D}, \tilde{Q}_1, \tilde{Q}_2, \tilde{Q}_3$ are the gcds and cofactors, respectively. Let $J(\mathbf{z}), \mathbf{z}$ and $\mathbf{r}$ be given in (6) and (7). Assume that $\epsilon \in \mathbb{R}_{>0}$ is a sufficiently small tolerance, and $J(\mathbf{z}), \mathbf{z}$ and $\mathbf{r}$ satisfy $\|F(\mathbf{z}) - \mathbf{b}\|_2 \leq \epsilon$ and $J(\mathbf{z})^H[F(\mathbf{z}) - b] = 0$. Let $\sigma_{\min}$ be the smallest singular value of $J(\mathbf{z})$. Assume that (H_1, H_2, H_3) are sufficiently near $(\tilde{H}_1, \tilde{H}_2, \tilde{H}_3)$. Then*

$$\left\| \begin{pmatrix} \tilde{\mathbf{d}} - \mathbf{d} \\ \tilde{\mathbf{q}}_1 - \mathbf{q}_1 \\ \tilde{\mathbf{q}}_2 - \mathbf{q}_2 \\ \tilde{\mathbf{q}}_3 - \mathbf{q}_3 \end{pmatrix} \right\|_2 \leq \frac{1}{\sigma_{\min}} \left(2\,\epsilon + \left\| \begin{pmatrix} \tilde{\mathbf{h}}_1 - \mathbf{h}_1 \\ \tilde{\mathbf{h}}_2 - \mathbf{h}_2 \\ \tilde{\mathbf{h}}_3 - \mathbf{h}_3 \end{pmatrix} \right\|_2 \right) + h.o.t.$$

where h.o.t. is higher order terms of the ϵ.

Proposition 1 indicates that the problem for computing approximate gcd is well-posed. The sensitivity of approximate gcd computation under perturbation can be measured once the error of the input polynomials is sufficiently small.

5.2.2. Properties of the parametrization $\mathcal{Q}$

In this subsection, we first prove that the output of Algorithm NRS, $\mathcal{Q}(\overline{t})$, is an ϵ-reparametrization of $\mathcal{P}(\overline{t})$ (that is, $\mathcal{P}(\overline{t}) \sim_\epsilon (\mathcal{Q}\circ R)(\overline{t})$, where $R(\overline{t}) = (S(\overline{t}), t_2)$). For this purpose, we prove the following theorem that is based on Theorem 2 in [26]. In this new situation, the main difference is that we are working with polynomials in $\mathbb{C}[t_1, t_2]$ and thus, the norm and the notion of ϵ-reparametrization is different. However, as we prove in Theorem 5, we can adapt Theorem 2 in [26] to these new conditions.

Theorem 5. *For $k = 1, 2, 3$, let*

$$L_k(s, t_2, x_k) = \mathrm{Res}_{t_1}(G_k(\overline{t}, x_k), sC_j(\overline{t}) - C_i(\overline{t})), \text{ where } G_k(\overline{t}, x_k) = x_k p_{k,2}(\overline{t}) - p_{k,1}(\overline{t}).$$

If for $k = 1, 2, 3$,

$$L_k(s, t_2, x_k) = (x_k q_{k,2}(s, t_2) - q_{k,1}(s, t_2))^\ell + \epsilon^\ell W_k(s, t_2, x_k), \quad \|\mathrm{num}(W_k(S, t_2, p_k))\| \le \|H_k^{\mathcal{P}\mathcal{Q}}\|^\ell,$$

where $\ell := \deg_{t_1}(S)$, $S(\overline{t}) = C_i(\overline{t})/C_j(\overline{t})$ and ϵ-$\mathrm{gcd}(q_{k,1}, q_{k,2}) = 1$, then $\mathcal{Q}(\overline{t}) = \big(q_1(\overline{t}),\, q_2(\overline{t}),\, q_3(\overline{t})\big)$, $q_i(\overline{t}) = q_{i,1}(\overline{t})/q_{i,2}(\overline{t})$, $i = 1, 2, 3$, is an ϵ-numerical reparametrization of $\mathcal{P}(\overline{t})$.

Proof. First, we observe that $L_k \neq 0$ (otherwise, G_k and $sC_j(\overline{t}) - C_i(\overline{t})$ have a common factor depending on $\overline{t}$, which is impossible because $\mathrm{gcd}(C_i, C_j) = 1$). In addition, it holds that $\deg_{x_k}(L_k) = \deg_{t_1}(S)$. Indeed, since

$$L_k(s, t_2, x_k) = \mathrm{Res}_{t_1}(G_k(\overline{t}, x_k), sC_j(\overline{t}) - C_i(\overline{t})),$$

we get that, up to constants in $\mathbb{C}(s, t_2) \setminus \{0\}$,

$$L_k(s, t_2, x_k) = \prod_{\{\alpha_\ell \,|\, sC_j(\alpha_\ell, t_2) - C_i(\alpha_\ell, t_2) = 0\}} G_k(\alpha_\ell, t_2, x_k),$$

(see Sections 5.8 and 5.9 in [27]), and thus

$$\deg_{x_k}(L_k) = \deg_{t_1}(sC_j(\overline{t}) - C_i(\overline{t}))\deg_{x_k}(G_k(\overline{t}, x_k)) = \deg_{t_1}(S).$$

In addition, from $\deg_{x_k}(L_k) = \deg_{t_1}(S)$, we deduce that $\deg_{x_k}(W_k) \le \ell$. In fact, since we are working with approximate mathematical objects, we may assume without loss of generality that $\deg_{x_k}(W_k) = \ell$. Now, taking into account the properties of the resultant (see [27]), one has that

$$0 = L_k(S(\overline{t}), t_2, p_k(\overline{t})) = (p_k(\overline{t})q_{k2}(S(\overline{t}), t_2) - q_{k1}(S(\overline{t}), t_2))^\ell + \epsilon^\ell W_k(S(\overline{t}), t_2, p_k(\overline{t})).$$

Then, for $k = 1, 2, 3$,

$$\mathrm{num}(H_k^{\mathcal{P}\mathcal{Q}}(\overline{t}, S(\overline{t}), t_2))^\ell = \epsilon^\ell e_k(\overline{t}), \text{ where } e_k := -W_k(S(\overline{t}), t_2, p_k(\overline{t}))p_{k,2}(\overline{t})^\ell C_j^{\ell \deg(q_k)}.$$

Since $\deg_{x_k}(W_k) = \ell$, and $\deg_s(W_k) = \ell \deg(q_k)$ (see statement 3 in Remark 2), one has that

$$e_k = -\mathrm{num}(W_k(S(\overline{t}), t_2, p_k(\overline{t}))) \in \mathbb{C}[\overline{t}]$$

(i.e. the denominator of $W_k(S(\overline{t}), t_2, p_k(\overline{t}))$ is canceled with $p_{k,2}(\overline{t})^\ell C_j(\overline{t})^{\ell \deg_{t_1}(q_k)}$). Therefore, from $\mathrm{num}(H_k^{\mathcal{P}\mathcal{Q}}(\overline{t}, S(\overline{t}), t_2))^\ell = \epsilon^\ell e_k(\overline{t})$, and using that $\|\mathrm{num}(W_k(S(\overline{t}), t_2, p_k))\| \le \|H_k^{\mathcal{P}\mathcal{Q}}\|^\ell$, we get that

$$\|\mathrm{num}(H_k^{\mathcal{P}\mathcal{Q}}(\overline{t}, S(\overline{t}), t_2))\|^\ell = \epsilon^\ell \|e_k\| \le \epsilon^\ell \|H_k^{\mathcal{P}\mathcal{Q}}\|^\ell,$$

which implies that $H_k^{\mathcal{P}\mathcal{Q}}(\overline{t}, S(\overline{t}), t_2) \approx_\epsilon 0$ (see Definition 2), and then $\mathcal{P}(\overline{t}) \sim_\epsilon \mathcal{Q} \circ (S(\overline{t}), t_2)$ (see Definitions 7 and 8). □

In the following, we prove that $\mathcal{Q}_2(t_1)$ is ϵ-proper. For this purpose, we observe that if $\mathcal{Q}(t) \in \mathbb{C}(t)^3$ is an ϵ-numerical reparametrization of $\mathcal{P}(t)$, it holds that ϵ-index$(\mathcal{P}) = \epsilon$-index$(\mathcal{Q}) \deg(R)$ (see Theorem 3). Let us prove now that this property also holds for $\mathcal{P}_2(t_1) \in (\mathbb{C}(t_2))(t_1)^3$ and $\mathcal{Q}_2(t_1) \in (\mathbb{C}(t_2))(t_1)^3$. More precisely, we have the following theorem.

Theorem 6. *Let $\mathcal{Q}(\bar{t}\,) \in \mathbb{C}(\bar{t}\,)^3$ and $R(\bar{t}\,) = (S(\bar{t}\,), t_2) \in \mathbb{C}(\bar{t}\,)^2$ be the output of* Algorithm NRS. *It holds that*

1. ϵ-index$(\mathcal{P}_2) = \epsilon$-index$(\mathcal{Q}_2) \deg_{t_1}(S)$.
2. ϵ-index$(\mathcal{Q}_2) = 1$.

Proof. We consider $t_2 = b \in \mathbb{C}$ such that $\|\mathrm{num}(W_k(S(t_1), b, p_k(t_1, b)))\| \leq \|H_k^{\mathcal{PQ}}(t_1, b, s_1, b)\|^\ell$ (note that since $\|\mathrm{num}(W_k(S, t_2, p_k))\| \leq \|H_k^{\mathcal{PQ}}\|^\ell$, this $b \in \mathbb{C}$ always exists). Now, taking into account the behaviour of the resultant under specialization, we have that

$$L_k(s, b, x_k) = \mathrm{Res}_{t_1}(G_k(t_1, b, x_k), sC_j(t_1, b) - C_i(t_1, b)) = (x_k q_{k,2}(s, b) - q_{k,1}(s, b))^\ell + \epsilon^\ell W_k(s, b, x_k).$$

Since $\|\mathrm{num}(W_k(S(t_1), b, p_k(t_1, b)))\| \leq \|H_k^{\mathcal{PQ}}(t_1, b, s_1, b)\|^\ell$, by applying Theorem 2 in [26], we conclude that $\mathcal{Q}(t_1, b)$ is an ϵ-reparametrization of $\mathcal{P}(t_1, b)$ ($\mathcal{P}(t_1, b) \sim_\epsilon \mathcal{Q} \circ (S(t_1, b), b)$), and ϵ-index$(\mathcal{P}(t_1, b)) =$ ϵ-index$(\mathcal{Q}(t_1, b)) \deg_{t_1}(S(t_1, b))$, and ϵ-index$(\mathcal{Q}(t_1, b)) = 1$.
Since we can reason as above for infinitely many values of $t_2 = b \in \mathbb{C}$, and the ϵ-index of a parametrization is invariant for all points in a non-empty Zariski open subset of the curve, we conclude that the equalities of the theorem hold. That is, ϵ-index$(\mathcal{P}_2) = \epsilon$-index$(\mathcal{Q}_2) \deg_{t_1}(S)$ and ϵ-index$(\mathcal{Q}_2) = 1$. □

Finally, we prove that ϵ-index$(\mathcal{Q}) < \epsilon$-index$(\mathcal{P})$. For this purpose, we note that clearly statement 1 in Theorem 6 implies that ϵ-index$(\mathcal{P}) = \epsilon$-index$(\mathcal{Q}) \deg_{t_1}(S)$. Therefore, one easily deduces that ϵ-index$(\mathcal{Q}) < \epsilon$-index$(\mathcal{P})$.

Theorem 7. *Let $\mathcal{Q}(\bar{t}\,) \in \mathbb{C}(\bar{t}\,)^3$ and $R(\bar{t}\,) = (S(\bar{t}\,), t_2) \in \mathbb{C}(\bar{t}\,)^2$ be the output of* Algorithm NRS. *It holds that*

$$\epsilon\text{-index}(\mathcal{P}) = \epsilon\text{-index}(\mathcal{Q}) \deg_{t_1}(S).$$

Therefore, ϵ-index$(\mathcal{Q}) < \epsilon$-index$(\mathcal{P})$. □

5.2.3. Error Analysis

In this subsection, we show how the input surface and the output surface are related using a similar idea to the one presented in Subsection 4.1. For this purpose, we shall restrict to a region where the parametrizations $\mathcal{P}$ and $\mathcal{Q}$ are both well defined. Hence, we consider the intervals $I_j := (d_{1,j}, d_{2,j}) \subset \mathbb{R}$, $j = 1, 2$, satisfying that for all $(a, b) \in I := I_1 \times I_2 \subset \mathbb{R}^2$, there exists $M \in \mathbb{N}$ such that $|q_{i,2}(S(a, b), b)| \geq M$, and $|p_{i,2}(a, b)| \geq M$, $i = 1, 2, 3$. Note that we can reason similarly as in Theorem 8 for each $I \subset \mathbb{R}^2$ we consider. In the following, we denote by $\ell = \deg_{t_1}(S)$.

Theorem 8. *Let $\mathcal{Q}(\bar{t}\,) \in \mathbb{C}(\bar{t}\,)^3$ and $R(\bar{t}\,) = (S(\bar{t}\,), t_2) \in \mathbb{C}(\bar{t}\,)^2$ be the output of the* Algorithm NRS. *The following statements hold:*

1. *Let $I := (d_{1,1}, d_{2,1}) \times (d_{1,2}, d_{2,2}) \subset \mathbb{R}^2$, and $M \in \mathbb{N}$ be such that for every $(a, b) \in I$, it holds that $|q_{i,2}(S(a, b), b)| \geq M$, and $|p_{i,2}(a, b)| \geq M$ for $i = 1, 2, 3$. Let $d := \max\{|d_{1,1}|, |d_{1,2}|, |d_{2,1}|, |d_{2,2}|\}$. Then, for every $(a, b) \in I$,*

$$|p_i(a, b) - q_i(S(a, b), b)| \leq 2/M^2 \epsilon\, \zeta \|p\| \|q\|, \quad i = 1, 2, 3, \quad \textit{where}$$

$$\zeta = \begin{cases} \dfrac{d^{\mu/\ell+1}\,\mu^{1/\ell}}{(d-1)^{2/\ell}} & \textit{if } d>1, \\ \dfrac{1}{(d-1)^{2/\ell}} & \textit{if } d<1, \\ \\ (1/2(\mu+2)(\mu+1))^{1/\ell} & \textit{if } d=1, \end{cases}$$

and $\mu := \deg(S)\deg(\mathcal{P})$.

2. $\mathcal{V}_{\overline{t}\in I}$ *is contained in the offset region of* $\mathcal{W}_{\overline{s}\in J}$ *at distance* $6\sqrt{3}/M^2\epsilon\,\zeta\,\|p\|\|q\|$, *where* $J = R(I)$.
3. $\mathcal{W}_{\overline{s}\in J}$ *is contained in the offset region of* $\mathcal{V}_{\overline{t}\in I}$ *at distance* $6\sqrt{3}/M^2\epsilon\,\zeta\,\|p\|\|q\|$, *where* $J = R(I)$,

Proof. Firstly, statement (1) implies statements (2) and (3). To see this, we note that for almost all real affine points $Q \in \mathcal{W}$ there exists an affine real point $P \in \mathcal{V}$ such that

$$\|P-Q\|_2 \le 2\sqrt{3}/M^2\epsilon\,\zeta\,\|p\|\|q\|.$$

Indeed, using statement (1), we have

$$\|P-Q\|_2 = \sqrt{(p_1(a,b)-q_1(S(a,b),b))^2 + (p_2(a,b)-q_2(S(a,b),b)^2 + (p_3(a,b)-q_3(S(a,b),b)^2} \le$$

$$\sqrt{(2/M^2\epsilon\,\zeta\|p\|\|q\|)^2 + (2/M^2\epsilon\,\zeta\|p\|\|q\|)^2 + (2/M^2\epsilon\,\zeta\|p\|\|q\|)^2} \le 2\sqrt{3}/M^2\epsilon\,\zeta\,\|p\|\|q\|.$$

Now, reasoning as in Section 2.2 in [6], we deduce statements (2) and (3).

We next prove statement (1). By the proof of Theorem 5, we have

$$H_i^{\mathcal{PQ}}(\overline{t}, S(\overline{t}), t_2)^\ell = (p_{i,1}(\overline{t})q_{1,2}(S(\overline{t}),t_2) - q_{i,1}(S(\overline{t}),t_2)p_{1,2}(t))^\ell = \epsilon^\ell e_i(\overline{t}),$$

where

$$e_i(\overline{t}) = -\mathrm{num}(W_i(S(\overline{t}),t_2,p_i(\overline{t}))) = e_{i,0} + e_{i,1,1}t_1 + e_{i,1,2}t_2 + \ldots + e_{i,n_{i,1},1}t_1^{n_{i,1}} + e_{i,n_{i,2},2}t_2^{n_{i,2}} \in \mathbb{C}[\overline{t}],$$

and

$$\|e_i\| = \|\mathrm{num}(W_i(S(\overline{t}),t_2,p_i(\overline{t})))\| \le \|H_i^{\mathcal{PQ}}\|^\ell.$$

In addition, since $e_i(\overline{t}) = -\mathrm{num}(W_i(S(\overline{t}),t_2,p_i(\overline{t})))$, we have $n_{i,1} := \deg_{t_1}(e_i) \le \ell\deg_{t_1}(\mathcal{P})$ for $i = 1,2,3$. Indeed, since $\deg_{x_i}(W_i) = \ell$ (from the proof of Theorem 5, we get that $\deg_{x_i}(L_i) = \deg_{x_i}(W_i) = \deg_{t_1}(S) = \ell$), we deduce that

$$n_{i,1} = \deg_{t_1}(e_i) \le \max\{\deg_{t_1}(S)\deg_{t_1}(W_i), \ell\deg_{t_1}(\mathcal{P})\} \le$$

$$\max\{\ell\deg_{t_1}(\mathcal{P}), \ell\deg_{t_1}(\mathcal{P})\} = \ell\deg_{t_1}(\mathcal{P}) \le \deg(S)\deg(\mathcal{P}),$$

$$n_{i,2} = \deg_{t_2}(e_i) \le \max\{\deg_{t_2}(S)\deg_{t_1}(W_i), \deg_{t_2}(W_i), \ell\deg_{t_2}(\mathcal{P})\} \le$$

$$\max\{\deg_{t_2}(S)\deg_{t_1}(\mathcal{P}), \deg(S)\deg(\mathcal{P}), \ell\deg_{t_2}(\mathcal{P})\} = \deg(S)\deg(\mathcal{P}).$$

Under these conditions, let $\mu := \deg(S)\deg(\mathcal{P})$, and for every $(a,b) \in I$, if $d \ne 1$, it holds that

$$|H_i^{\mathcal{PQ}}(a,b,S(a,b),b)^\ell| = \epsilon^\ell|e_i(a,b)| \le$$

$$\epsilon^\ell(|e_{i,0}| + |e_{i,1,1}||a| + |e_{i,1,2}||b| + \ldots + |e_{i,n_{i,1},1}||a|^{n_{i,1}} + |e_{i,n_{i,1},2}||b|^{n_{i,2}}) \le$$

$$\epsilon^\ell(|e_{i,0}| + |e_{i,1,1}|d + |e_{i,1,2}|d + \ldots + |e_{i,n_{i,1},1}|d^{n_{i,1}} + |e_{i,n_{i,1},2}|d^{n_{i,2}}) \le$$

$$\epsilon^\ell\|H_i^{\mathcal{PQ}}\|^\ell(1+2d+3d^2+\ldots+(\mu+1)d^\mu) = \epsilon^\ell\|H_i^{\mathcal{PQ}}\|^\ell\,\frac{d^{\mu+1}(d(\mu+1)-\mu-2)+1}{(d-1)^2}, \quad i=1,2,3. \quad (8)$$

If $d = 1$, then

$$|H_i^{\mathcal{PQ}}(a, b, S(a, b), b)^\ell| \leq \epsilon^\ell \|H_i^{\mathcal{PQ}}\|^\ell (1 + |a| + |b| \ldots + |a|^{n_{i,1}} + |b|^{n_{i,2}}) \leq$$

$$\epsilon^\ell \|H_i^{\mathcal{PQ}}\|^\ell (1 + 2 + 3 + \ldots + (\mu + 1)) = \epsilon^\ell \|H_i^{\mathcal{PQ}}\|^\ell \, 1/2(\mu + 2)(\mu + 1). \tag{9}$$

Therefore, we conclude that:

1. If $d > 1$, by (8), and taking into account that $|q_{i,2}(S(a, b), b)| \geq M$, and $|p_{i,2}(a, b)| \geq M$ for $i = 1, 2$, we obtain that
$$|p_i(a, b) - q_i(S(a, b), b)| = \frac{|H_i^{\mathcal{PQ}}(a, b, S(a, b), b)|}{|q_{i,2}(S(a, b), b)p_{i,2}(a, b)|} \leq$$
$$1/M^2 \epsilon \|H_i^{\mathcal{PQ}}\| \; \frac{(d^{\mu+1}(d(\mu + 1) - \mu - 2) + 1)^{1/\ell}}{(d - 1)^{2/\ell}} \leq 1/M^2 \, \epsilon \, \|H_i^{\mathcal{PQ}}\| \; \frac{d^{\mu/\ell + 2/\ell} \, \mu^{1/\ell}}{(d - 1)^{2/\ell}}$$
$$\leq 1/M^2 \, \epsilon \, \|H_i^{\mathcal{PQ}}\| \, \frac{d^{\mu/\ell + 1} \, \mu^{1/\ell}}{(d - 1)^{2/\ell}}.$$

2. If $d < 1$, from (8), and taking into account that $1 - d^{n_i+1} < 1$, and $|q_{i2}(S(a, b), b)| \geq M$, and $|p_{i2}(a, b)| \geq M$ for $i = 1, 2$, we obtain that
$$|p_i(a, b) - q_i(S(a, b), b)| = \frac{|H_i^{\mathcal{PQ}}(a, b, S(a, b), b)|}{|q_{i,2}(S(a, b), b)p_{i,2}(a, b)|} \leq 1/M^2 \, \epsilon \, \|H_i^{\mathcal{PQ}}\| \frac{1}{(1 - d)^{2/\ell}}.$$

3. If $d = 1$, from (5.2.3), and taking into account that $|q_{i2}(S(a, b), b)| \geq M$, and $|p_{i2}(a, b)| \geq M$ for $i = 1, 2$, we obtain that
$$|p_i(a, b) - q_i(S(a, b), b)| = \frac{|H_i^{\mathcal{PQ}}(a, b, S(a, b), b)|}{|q_{i,2}(S(a, b), b)p_{i,2}(a, b)|} \leq 1/M^2 \, \epsilon \, \|H_i^{\mathcal{PQ}}\| (1/2(\mu + 2)(\mu + 1))^{1/\ell}.$$

Finally, the theorem is deduced from the above inequalities and taking into account that

$$\|H_i^{\mathcal{PQ}}\| = \|p_{i,1}(\,\bar{t}\,)q_{i,2}(\,\bar{s}\,) - q_{i,1}(\,\bar{s}\,)p_{i,2}(\,\bar{t}\,)\| \leq 2\|p\|\|q\|.$$

□

In the following, we present the error bound obtained in the Example 5.

Example 6. *Let us consider the rational parametrization $\mathcal{P}$ of the surface $\mathcal{V}$ introduced in Example 5, and the output provided by* Algorithm NRS. *First, one may check that the equality of Step 7.1 holds. In fact, this equality holds under $\bar{\epsilon} = 10^{-70}$. Then, $\mathcal{Q}$ is an $\bar{\epsilon}$-proper reparametrization of $\mathcal{P}$. In Figure 5, we plot the input surface $\mathcal{V}$ and the output surface $\mathcal{W}$.*

We next perform the error analysis using Theorem 4. For this purpose, we first consider $I = (0.5, 1) \times (0.5, 1)$ (see Figure 5). Thus, $d = 1$. Let $M \in \mathbb{N}$ be such that for every $(a, b) \in I$, it holds that $|q_{i,2}(S(a, b), b)| \geq M$, and $|p_{i,2}(a, b)| \geq M$ for $i = 1, 2, 3$. We have that $M = 0.2145301953$. Then, by Theorem 8 we deduce that

$$\zeta = 9.539392014,$$

and for every $(a, b) \in I$, it holds that

$$|p_i(t_0) - q_i(R(t_0))| < 2/M^2 \bar{\epsilon} \, \zeta \|p\|\|q\| = 1.013787066 \, 10^{-68}, \quad i = 1, 2, 3.$$

6. Conclusion

The proper reparametrization is a classic problem in computer aided geometric design. This problem is generally studied in the symbolic situation. When approximate objects are being thoroughly studied as a real-world counterparts to exact mathematical objects, a natural way is to extend the symbolic algorithms to numerical situations. Certainly, there are differences between the numerical and symbolic considerations including definitions, algorithms and discussions. Moreover, the error analysis is always an important task in numerical situations. We propose the numerical proper reparametrization algorithms for rational space curves and rational surfaces, as well as the detailed error analysis between the given rational curve/surface and our reparametrized curve/surface. The results propose the approximate proper reparametrizations and are expected to be helpful in computing the approximate implicit equations with lower degrees.

Our algorithms are based on the latest symbolic results. As the further work, another way is to find the reparametrization by interpolation/fitting method. With the consideration of the approximate properness of the curve/surface, we can find a good degree estimation as the division of the given degree over the approximate improper index.

Acknowledgements

This work has been partially supported by FEDER/Ministerio de Ciencia, Innovación y Universidades - Agencia Estatal de Investigación/MTM2017-88796-P (Symbolic Computation: new challenges in Algebra and Geometry together with its applications) and by National Natural Science Foundation of China under Grant 61872332, 11731013. The second author belongs to the Research Group ASYNACS (Ref. CCEE2011/R34).

References

[1] B. Beckermann and G. Labahn. A fast and numerically stable euclidean-like algorithm for detecting relatively prime numerical polynomials. *Journal of Symbolic Computation*, 26(6):691 – 714, 1998.

[2] E.-W. Chionh, X.-S. Gao, and L.-Y. Shen. Inherently improper surface parametric supports. *Computer Aided Geometric Design*, 23(8):629 – 639, 2006.

[3] E.-W. Chionh and R. N. Goldman. Degree, multiplicity, and inversion formulas for rational surfaces using u-resultants. *Computer Aided Geometric Design*, 9(2):93 – 108, 1992.

[4] R. M. Corless, S. M. Watt, and L. Zhi. Qr factoring to compute the gcd of univariate approximate polynomials. *IEEE Transactions on Signal Processing*, 52(12):3394–3402, Dec 2004.

[5] M. Eck. Degree reduction of Bézier curves. *Computer Aided Geometric Design*, 10(3):237 – 251, 1993.

[6] R. Farouki and V. Rajan. On the numerical condition of algebraic curves and surfaces 1. implicit equations. *Computer Aided Geometric Design*, 5(3):215 – 252, 1988.

[7] J.-C. Faugère and L. Perret. An efficient algorithm for decomposing multivariate polynomials and its applications to cryptography. *Journal of Symbolic Computation*, 44(12):1676 – 1689, 2009.

[8] S. Gao, E. Kaltofen, J. May, Z. Yang, and L. Zhi. Approximate factorization of multivariate polynomials via differential equations. In *Proceedings of the 2004 International Symposium on Symbolic and Algebraic Computation*, ISSAC '04, pages 167 – 174, New York, NY, USA, 2004. ACM.

[9] X.-S. Gao and S.-C. Chou. Implicitization of rational parametric equations. *Journal of Symbolic Computation*, 14(5):459 – 470, 1992.

[10] J. v. z. Gathen, J. Gutierrez, and R. Rubio. Multivariate polynomial decomposition. *Applicable Algebra in Engineering, Communication and Computing*, 14(1):11–31, 2003.

[11] J. Gutierrez, R. Rubio, and D. Sevilla. On multivariate rational function decomposition. *Journal of Symbolic Computation*, 33(5):545 – 562, 2002.

[12] J. L. Hafner and K. S. McCurley. Asymptotically fast triangularization of matrices over rings. *SIAM Journal on Computing*, 20(6):1068 – 1083, 1991.

[13] J. Harris. Algebraic geometry. a first course. *Springer-Verlag*, 1995.

[14] E. Kaltofen, Z. Yang, and L. Zhi. On probabilistic analysis of randomization in hybrid symbolic-numeric algorithms. In *Proceedings of the 2007 International Workshop on Symbolic-numeric Computation*, SNC '07, pages 11 – 17, New York, NY, USA, 2007. ACM.

[15] J. Li, L.-Y. Shen, and X.-S. Gao. Proper reparametrization of rational ruled surface. *Journal of Computer Science and Technology*, 23(2):290 – 297, 2008.

[16] H. Park, L. Zhang, and J. B. Rosen. Low rank approximation of a hankel matrix by structured total least norm. *BIT Numerical Mathematics*, 39(4):757 – 779, 1999.

[17] S. Pérez-Díaz. On the problem of proper reparametrization for rational curves and surfaces. *Computer Aided Geometric Design*, 23:307 – 323, 2006.

[18] S. Pérez-Díaz. A partial solution to the problem of proper reparametrization for rational surfaces. *Computer Aided Geometric Design*, 30(8):743 – 759, 2013.

[19] S. Pérez-Díaz and J. R. Sendra. Computation of the degree of rational surface parametrizations. *Journal of Pure and Applied Algebra*, 193(1–3):99 – 121, 2004.

[20] S. Pérez-Díaz, J. R. Sendra, and J. Sendra. Distance bounds of ε-points on hypersurfaces. *Theoretical Computer Science*, 359(1-3):344 – 368, 2006.

[21] S. Pérez-Díaz, J. R. Sendra, and C. Villarino. Finite piecewise polynomial parametrization of plane rational algebraic curves. *Applicable Algebra in Engineering, Communication and Computing*, 18(1):91 – 105, 2007.

[22] J. Schicho. Simplification of surface parametrizations: A lattice polygon approach. *J. Symb. Comput.*, 36(3-4):535 – 554, 2003.

[23] T. W. Sederberg. Improperly parametrized rational curves. *Computer Aided Geometric Design*, 3(1):67 – 75, 1986.

[24] I. Shafarevich. Basic algebraic geometry i and ii. *Springer-Verlag, Berlin New York*, 1994.

[25] L.-Y. Shen, E. Chionh, X.-S. Gao, and J. Li. Proper reparametrization for inherently improper unirational varieties. *Journal of Systems Science and Complexity*, 24(2):367 – 380, 2011.

[26] L.-Y. Shen and S. Pérez-Díaz. Numerical proper reparametrization of parametric plane curves. *Journal of Computational and Applied Mathematics*, 277:138 – 161, 2015.

[27] B. L. v. d. Waerden. *Algebra Volume II.* Springer-Verlag, New York, Berlin, Paris, 1991. Trad. de l'allemand : Algebra II (5. Aufl., 1967).

[28] R. J. Walker. *Algebraic curves.* Springer-Verlag, New York, Paris, 1978. Reprint the edtion of 1950, Princeton University Press.

[29] Z. Zeng and B. H. Dayton. The approximate gcd of inexact polynomials part ii: a multivariate algorithm. In *In ISSAC 2004 Proc. 2004 Internat. Symp. Symbolic Algebraic Comput.*, pages 320 – 327, 2004.