NATIONAL AND KAPODISTRIAN UNIVERSITY OF ATHENS

SCHOOL OF SCIENCE
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

GRADUATE PROGRAM

MSC THESIS

Implicitization, Interpolation, and Syzygies

Konstantinos D. Gavriil

Supervisors: **Ioannis Emiris,** Professor U. Athens
Vassilis Zissimopoulos, Professor U. Athens

ATHENS

SEPTEMBER 2016

ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αλγεβρικοποίηση Επιφανειών, Παρεμβολή και Αλγεβρικές Συζυγίες

Κωνσταντίνος Δ. Γαβριήλ

Επιβλέποντες: **Ιωάννης Εμίρης,** Καθηγητής Ε.Κ.Π.Α.
Βασίλειος Ζησιμόπουλος, Καθηγητής Ε.Κ.Π.Α.

ΑΘΗΝΑ

ΣΕΠΤΕΜΒΡΙΟΣ 2016

MSC THESIS

Implicitization, Interpolation, and Syzygies

Konstantinos D. Gavriil
S.N.: M1366

SUPERVISORS: **Ioannis Emiris,** Professor U. Athens
Vassilis Zissimopoulos, Professor U. Athens

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αλγεβρικοποίηση Επιφανειών, Παρεμβολή και Αλγεβρικές Συζυγίες

Κωνσταντίνος Δ. Γαβριήλ
Α.Μ.: Μ1366

ΕΠΙΒΛΕΠΟΝΤΕΣ: **Ιωάννης Εμίρης**, Καθηγητής Ε.Κ.Π.Α.
Βασίλειος Ζησιμόπουλος, Καθηγητής Ε.Κ.Π.Α.

ABSTRACT

Implicitization is a fundamental change of representation of geometric objects from a parametric or point cloud representation to an implicit form, namely as the zero set of one (or more) polynomial equation. This thesis examines three questions related to expressing the implicit equation of a curve or a surface.

First, we consider a sparse interpolation method for implicitization: When the basis of the kernel of the interpolation matrix is in reduced row echelon form, the implicit equation can be readily obtained, without demanding computations such as multivariate polynomial GCD or factoring. As a second contribution, a numeric method that computes a multiple of the implicit equation based on the power method is tested and evaluated.

The third contribution of this thesis is to provide a method for computing a matrix representation of a rational planar or space curve, or a rational surface, when we are only given a sufficiently large sample of points (point cloud) on the object in such a way that the value of the parameter is known per point. Our method extends the approach of algebraic syzygies for implicitization to the case where the parameterization is not given but only assumed.

SUBJECT AREA: Algebraic Geometry

KEYWORDS: geometric representation, implicitization, linear algebra, syzygies, matrix representation

ΠΕΡΙΛΗΨΗ

Η αλγεβρικοποίηση καμπυλών και επιφανειών είναι μία θεμελιώδης μετατροπή στην αναπαράσταση γεωμετρικών αντικειμένων από παραμετρική μορφή ή αναπαράσταση νέφους σημείων σε μία αλγεβρική αναπαράσταση, και ειδικότερα ως το μηδενοσύνολο ενός (ή περισσότερων) πολυωνυμικών εξισώσεων. Αυτή η διπλωματική εργασία ερευνά τρία ερωτήματα σχετικά με την έκφραση αυτής της αλγεβρικής αναπαράστασης καμπύλης ή επιφάνειας.

Αρχικά, θεωρούμε τη μέθοδο της αραιής παρεμβολής για την αλγεβρικοποίηση: Όταν η βάση του πυρήνα του πίνακα παρεμβολής είναι σε ανοιγμένη κλιμακωτή μορφή, η αναλυτική εξίσωση μπορεί να ληφθεί άμεσα, χωρίς να απαιτούμε υπολογισμούς όπως ΜΚΔ πολυωνύμων πολλών μεταβλητών ή παραγοντοποίηση. Ως δεύτερη συνεισφορά, εξετάζουμε και αξιολογούμε μία αριθμητική μέθοδο που υπολογίζει ένα πολλαπλάσιο της αναλυτικής εξίσωσης, η οποία βασίζεται στη μέθοδο των δυνάμεων.

Η τρίτη συνεισφορά αυτής της διπλωματικής εργασίας είναι να προσφέρουμε μία μέθοδο για τον υπολογισμό μίας αναπαράστασης μητρώου μίας ρητής δισδιάστατης ή τρισδιάστατης καμπύλης, ή μίας τρισδιάστατης επιφάνειας, όταν μας δίνεται μόνο ένα επαρκές σύνολο σημείων (νέφος σημείων) πάνω στο αντικείμενο με τέτοιον τρόπο ώστε η τιμή της παραμέτρου να είναι γνωστή ανά σημείο. Η μέθοδός μας επεκτείνει την προσέγγιση των αλγεβρικών συζυγιών για το πρόβλημα της αλγεβρικοποίησης επιφανειών και καμπυλών στην περίπτωση που η παραμετροποίηση δεν δίνεται αλλά υποτίθεται.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Αλγεβρική Γεωμετρία

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: γεωμετρική αναπαράσταση, αλγεβρικοποίηση, γραμμική άλγεβρα, συζυγίες, αναπαράσταση μητρώου

Στους γονείς μου.

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω τον επιβλέποντα κ. Ιωάννη Εμίρη για τη συνεργασία, την εμπιστοσύνη και τη βοήθεια κατά την εκπόνηση αυτής της διπλωματικής.

Θα ήθελα, επίσης, να ευχαριστήσω το φίλο μου Χρήστο Κοναξή για τις συνεχόμενες πολύτιμες παρατηρήσεις του σε προκαταρκτικές εκδόσεις του κειμένου.

CONTENTS

LIST OF FIGURES

LIST OF ALGORITHMS

1. INTRODUCTION

This thesis is part of the Master of Science Graduate Program of the Department of Informatics and Telecommunications of the University of Athens in the Computational Science specialization.

Implicitization is a fundamental change of representation of geometric objects from a parametric or point cloud representation to an implicit form, namely as the zero set of one (or more) polynomial equation. This thesis examines three questions related to expressing the implicit equation of a curve or a surface.

We will need to establish some concepts of algebraic geometry before continuing to the algorithms related to the problems. Therefore, Section 2 will introduce the reader to the necessary background and definitions needed in the following sections. We will define the implicitization problem and describe two of the methods for the implicitization problem. The first one is the sparse implicitization by interpolation method using predicted support and the second is the implicit matrix representation method. We will define what an implicit matrix representation is and provide the existing method for its computation.

In Section 3, the first contribution of this work, we consider the sparse interpolation method for implicitization. After constructing the interpolation matrix M of a given curve or surface, the method demands the computation of a basis of the kernel of M. Each of the basis elements corresponds to a polynomial and the last step of the method involves the multivariate polynomial GCD computation of these polynomials or factoring computation of one of the polynomials followed by a polynomial evaluation to acquire the implicit polynomial. We show that when the basis of the kernel of the interpolation matrix is in reduced row echelon form, the implicit equation can be readily obtained, without demanding such computations, namely multivariate polynomial GCD or factoring. This contribution speeds up the final step of the interpolation method for the implicitization problem and uses basic linear algebra and matrix computations instead of number theoretic approaches, such as Hensel lifting.

In Section 4, we present a numeric method that computes an approximation of a multiple of the implicit equation based on the power method. Given a matrix M which has a nontrivial kernel, we compute a nontrivial element of its kernel by using the power iteration method on the Gram matrix of M. This approximate kernel vector corresponds to an approximation of a multiple of the implicit polynomial. The method is tested and evaluated. We show that our initial approach has some drawbacks that render the method impractical for most scenarios. Despite its drawbacks, we see this contribution as an initial step towards a numeric method for computing an approximation of the implicit polynomial when we are given the interpolation matrix of a curve or surface. We propose several questions for future work.

Section 5 is the third and final contribution of this thesis. We provide a method for computing a matrix representation of a rational planar or space curve, or a rational surface, when we are only given a sufficiently large sample of points on the object in such a way that the value of the parameter is known per point. Our method extends the approach of algebraic syzygies for implicitization to the case where the parameterization is not given but only assumed. Additionally, we show how we can compute the degree of the parameterization of the curve or surface when the parameterization is not given explicitly.

2. BACKGROUND

We will need to establish some concepts of algebraic geometry before continuing to the algorithms related to the problems. In this section, we will introduce the reader to the necessary background and definitions needed in the following sections. We will define the implicitization problem and examine the previous work for two of the methods for the implicitization problem.

The first one is the sparse implicitization by interpolation method using predicted support. We will briefly present the method and focus on the definitions and background needed for the contribution that we present in Section 3.

The second is the implicit matrix representation method based on syzygies. We will define what an implicit matrix representation is and provide the existing method for its computation. A brief focused introduction to the theory of syzygies is provided. The presented method will be the foundation for the contribution of Section 5.

For an in-depth study of modules and syzygies see [9], while more information about the sparse interpolation method for the implicitization problem can be found in [15] and [13].

Before defining the implicitization problem formally, we will introduce some needed concepts. Firstly, we define the concept of a rational parameterization of a geometric object since this will be one of the two forms of input, the other being a point cloud, to the implicitization algorithms.

2.1 Rational parameterization of curves and surfaces

Let k be a polynomial ring. A *rational function* f is a function that can be expressed as a quotient of polynomials of k.

A *parameterization* of a geometric object is the description of the object by parametric functions. We will be interested in rational parameterizations in the following sections. Let $k = \mathbb{R}[\mathbf{t}]$, where $\mathbf{t} = (t_1, \ldots, t_n)$ A *rational parameterization* φ is of the form

$$\varphi : (\mathbf{t}) \to \left(\frac{f_1(\mathbf{t})}{f_s(\mathbf{t})}, \ldots, \frac{f_{s-1}(\mathbf{t})}{f_s(\mathbf{t})} \right)$$

where $f_1(\mathbf{t}), \ldots, f_s(\mathbf{t}) \in \mathbb{R}[\mathbf{t}]$.

When $n = 1$, φ corresponds generically to a $(s-1)-$dimensional curve and when $n = 2$, it is a $(s-1)-$dimensional surface. The values of (n, s) that we will examine and the corresponding geometric object are:

- $(1, 3)$: planar curve
- $(1, 4)$: space curve
- $(2, 4)$: space surface

2.2 Newton polytope of a polynomial

Another important concept for the sparse approach to the implicitization problem is the Newton polytope of a given polynomial. When studying a polynomial of a certain degree, we can take into account all the possible monomials that the polynomial can contain. The Newton polytope of that polynomial allows us to consider only the monomials that appear (have a nonzero coefficient) in the polynomial.

Given polynomial

$$f(\mathbf{t}) = \sum_{\boldsymbol{\alpha}} c_{\boldsymbol{\alpha}} \mathbf{t}^{\boldsymbol{\alpha}} \in \mathbb{R}[t_1, \ldots, t_n]$$

where $\mathbf{t}^{\boldsymbol{\alpha}} = t_1^{\alpha_1} \cdots t_n^{\alpha_n}$, $\boldsymbol{\alpha} \in \mathbb{N}^n$, $c_{\boldsymbol{\alpha}} \in \mathbb{R}$, we define its *support* as the set of the exponents of the monomials of f with nonzero coefficient in vector form, i.e.

$$\mathcal{SUP}(f) = \{\boldsymbol{\alpha} \in \mathbb{N}^n : c_{\boldsymbol{\alpha}} \neq 0\}$$

We use the notation $\mathcal{CH}$ for the convex hull of a given set of points. We define the *Newton polytope* of the polynomial f as

$$\mathcal{N}(f) = \mathcal{CH}(\mathcal{SUP}(f))$$

Example 1. *Consider polynomial*

$$f(t_1, t_2) = t_1^4 + t_1^2 t_2^2 + t_1^2 + t_1 t_2^4 + t_1 + t_2^3 \in \mathbb{R}[t_1, t_2]$$

Then, its support is the set

$$\mathcal{SUP}(f) = \{(4,0), (2,2), (2,0), (1,4), (1,0), (0,3)\}$$

Its Newton polytope is the convex hull of the points in $\mathcal{SUP}(f)$.

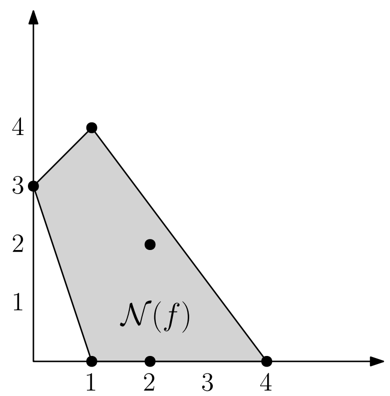

Figure 1: Newton polytope of *f*

Newton polytopes are the basic tool in sparse elimination theory. They are used in the setting of sparse implicitization to predict the support of the implicit polynomial. The interpolation matrix of the curve or surface, has columns indexed by this support. An accurate support prediction reduces the size of the interpolation matrix compared to one built using only degree bounds.

Let us now give the formal definition of the implicitization problem.

2.3 Implicitization problem definition

Implicitization is a fundamental operation with applications in computer-aided geometric design (CAGD) and geometric modelling. It is the process of changing the representation of a geometric object from parametric to implicit. Various approaches have been studied for the implicitization problem, including resultants, Gröbner bases, syzygies and interpolation techniques. We will restrict ourselves to the last two methods, namely syzygies and interpolation.

Consider the parameterization $\varphi : \mathbb{R}^n \to \mathbb{R}^m$ of a geometric object,

$$\varphi : \mathbf{t} = (t_1, \dots, t_n) \to (x_1 = f_1(\mathbf{t}), \dots, x_m = f_m(\mathbf{t}))$$

where the f_i, $i = 1, \dots, m$ are continuous functions, including polynomial, rational, and trigonometric functions.

The *implicitization problem* asks for the smallest algebraic variety containing the closure of the image of the parametric map φ. This image is contained in the variety defined by the ideal of all polynomials $p(x_1, \dots, x_m)$ such that $p(f_1(\mathbf{t}), ..., f_m(\mathbf{t})) = 0$, for all $\mathbf{t}$ in the domain of φ. We restrict ourselves to the case when this is a principal ideal, and we wish to compute its unique defining polynomial

$$p(x_1, \dots, x_m) = 0$$

As we already mentioned, we restrict ourselves to two methods for the implicitization problem, namely sparse implicitization using interpolation, and syzygies. In the following paragraphs, we will make a brief introduction to both these methods before continuing to the following sections and the main contributions of this work.

2.4 Sparse interpolation method using predicted support

We will not provide a detailed description of the underlying theory and method for the implicitization problem using support prediction but rather provide an overview of the method and direct the reader towards [15, 13] for a detailed description. We focus our attention to the algorithm used in [15, 13] since we will be making a contribution towards the improvement of its final step.

We assume we have knowledge of the *implicit polytope*, which is defined as the Newton polytope $\mathcal{N}(p(x_1,\ldots,x_m))$ of the implicit polynomial $p(x_1,\ldots,x_m)$, or a superset S of its implicit support, where S is defined as the set of all lattice points in $\mathcal{N}(p)$. This is achieved by the algorithm provided in [15, 13]. Given the superset S, we construct a $\mu \times |S|$ matrix M whose columns are indexed by monomials with exponents in S and whose rows are indexed by values of $\mathbf{t}$ at which the monomials are being evaluated. The number of rows μ must be greater or equal $|S|$.

Let $v_1,\ldots,v_N \in \mathbb{R}^{|S|}$ be a basis of the kernel of M and $k_1,\ldots,k_N \in \mathbb{R}[x_1,\ldots,x_m]$ be the corresponding polynomials, i.e. $k_i = v_i^T S$, which we call *kernel polynomials*. Then, we have the following result

$$\mathrm{gcd}(k_1,\ldots,k_N) = p(x_1,\ldots,x_m),$$

where $p \in \mathbb{R}[x_1,\ldots,x_m]$ is the implicit polynomial. Therefore, a GCD computation of the kernel polynomials is needed to obtain the implicit polynomial p. The first contribution of this work is to avoid such computations and instead use basic linear algebra computations to obtain the implicit polynomial. We direct the reader to Section 3 for this improvement.

This concludes the introduction of the first method for the implicitization problem. Below, we will define the concept of the implicit matrix representation of a curve or surface, describe the method for constructing such a representation and comment on how this method will be the basis for the contribution of Section 5.

As before, we will begin by introducing the reader to the basic theory of modules and syzygies and by defining homogeneous polynomials. A more detailed introduction and further topics in modules and syzygies can be found in [9]

2.5 Modules

Let $\Bbbk$ be a commutative ring with identity. Polynomials rings, which are of interest for this thesis, are examples of such rings. As defined in [9], a $\Bbbk-$*module* is a set M together with a binary operation, usually denoted as addition, and an operation of $\Bbbk$ on M, called scalar multiplication, satisfying the following properties.

1. M is an abelian group under addition.
2. For all $a \in \Bbbk$ and all $f, g \in M$, $a(f+g) = af + ag$.
3. For all $a, b \in \Bbbk$ and all $f \in M$, $(a+b)f = af + bf$.
4. For all $a, b \in \Bbbk$ and all $f \in M$, $(ab)f = a(bf)$.
5. If $1_\Bbbk$ is the multiplicative identity in $\Bbbk$, $1_\Bbbk f = f$ for all $f \in M$.

Since rings are a generalization of fields, modules over a ring can be seen as a generalization of vector spaces over a field.

We provide below some examples of modules before continuing to define the module of syzygies of a set of polynomials.

Example 2. *Let $\mathbb{k}$ be polynomial ring. An example of a $\mathbb{k}-$module is $\mathbb{k}^s$, $s \in \mathbb{N}$, i.e. the set of $s \times 1$ vectors consisting of elements of $\mathbb{k}$. Addition and scalar multiplication are defined similar to vector spaces.*

Another example of a $\mathbb{k}-$module is the set of $\mathbb{k}-$linear combinations of a finite set of vectors $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f_m} \in \mathbb{k}^s$.

$$\{a_1\mathbf{f}_1 + a_2\mathbf{f}_2 + \ldots + a_m\mathbf{f_m} \in \mathbb{k}^s,\ \textit{where } a_1, \ldots, a_m \in \mathbb{k}\}$$

Let M be a $\mathbb{k}-$module and $N \subset M$. Then, N is called a $\mathbb{k}-$*submodule* of M if the following properties are satisfied.

1. For all $f, g \in N$, $f + g \in N$.
2. For all $a \in \mathbb{k}$ and $f \in N$, $af \in N$.

In the following section, we will introduce a submodule of interest for this work, namely syzygies of given polynomials.

2.6 Syzygies

Let $\mathbb{k}$ be a polynomial ring and consider polynomials $f_1, \ldots, f_s \in \mathbb{k}$. Some of the polynomial rings that we will be using in the following sections are $\mathbb{R}[t]$ and $\mathbb{R}[t_1, t_2]$. An $s-$tuple $(h_1, \ldots, h_s) \in \mathbb{k}^s$ of polynomials $h_1, \ldots, h_s \in \mathbb{k}$ that verifies the $\mathbb{k}-$linear relation

$$h_1f_1 + \cdots + h_sf_s = 0$$

is called a *syzygy* on the polynomials $f_1, \ldots, f_s$. The term syzygy comes from the Greek word συζυγία which is used in astronomy to express an alignment of celestial bodies. We can think of a polynomial syzygy as an algebraic null alignment of the polynomials $f_1, \ldots, f_s$.

The set of all $(h_1, \ldots, h_s) \in \mathbb{k}^s$ such that

$$\sum_{i=1}^{s} h_if_i = 0,$$

is a $\mathbb{k}-$submodule of $\mathbb{k}^s$, called the (first) syzygy module of $(f_1, \ldots, f_s)$, and denoted

$$Syz(f_1, \ldots, f_s).$$

Example 3. *Consider the polynomial ring $\mathbb{R}[x, y]$ and consider two of its polynomials*

$$f_1 = x^2 \ \ and \ \ f_2 = y$$

Then $(y, -x^2)$ *is a syzygy of polynomials* f_1, f_2 *since*

$$yf_1 + (-x^2)f_2 = 0$$

This syzygy is also called a trivial syzygy since for any two polynomials $f_1, f_2 \in \mathbb{R}[x, y]$*, it holds that*

$$f_2 f_1 + (-f_1)f_2 = 0.$$

The concepts of homogeneous polynomials and the homogeneous degree will be useful for our description of the implicit matrix representation method. Let us begin by defining homogeneous polynomials and the concept of a graded ring that we will use to define the module of syzygies of a certain homogeneous degree.

2.7 Homogeneous polynomials

A polynomial is *homogeneous of total degree d* if every term appearing in it has total degree d. Consider a non-homogeneous polynomial $f \in \mathbb{R}[t_1, \ldots, t_n]$ of degree d. The *homogenization* of f with respect to a homogenizing variable t_{n+1} is the introduction of t_{n+1} to f such that

$$f^h(t_1, \ldots, t_n, t_{n+1}) = t_{n+1}^d f(\frac{t_1}{t_{n+1}}, \ldots, \frac{t_n}{t_{n+1}}).$$

We can *dehomogenize* a homogenized polynomial by setting the homogenizing variable $t_{n+1} = 1$.

Example 4. *Consider the polynomial*

$$f(x, y) = 3x^2y + xy - 2y^5 \in \mathbb{R}[x, y].$$

The homogenization of f with respect to the variable z gives us

$$f^h(x, y, z) = 3x^2yz^2 + xyz^3 + 2y^5 \in \mathbb{R}[x, y, z].$$

Homogeneous polynomials provide a grading to a polynomial ring. A *graded ring* is a ring $\Bbbk$ that is expressible as $\oplus_{n\geq 0}\Bbbk_n$ where $\Bbbk_n$ are additive subgroups such that $\Bbbk_m\Bbbk_n \subseteq \Bbbk_{m+n}$. We call $\Bbbk_n$ the n^{th} *graded piece* and the elements of $\Bbbk_n$ *homogeneous of degree n*.

Similarly, we can speak of a *graded module*, namely a $\mathbb{k}-$module M is expressible as $\oplus_{m\geq 0} M_m$ and $\mathbb{k}_n M_m \subseteq M_{n+m}$ for all $m, n \geq 0$. Additionally, N is a graded submodule of M if N is a graded module, N is a submodule of M and $N_m = N \cap M_m$ for all $m \geq 0$.

The set of syzygies of the homogeneous polynomials $f_1, \dots, f_s$, which is of interest for this work, is a graded submodule. Thus, we can speak of its graded piece $Syz(f_1, \dots, f_s)_v$, namely the set of syzygies of polynomials $f_1, \dots, f_s$ of homogeneous degree $v \geq 0$. As we will mention again in the following sections, $Syz(f_1, \dots, f_s)_v$ is an $\mathbb{R}-$vector space, which means that knowledge of its basis is adequate for its expression.

Now, we are ready to describe the second method for the implicitization problem which is of interest to this work. The implicit matrix representation method via syzygies does not compute the implicit polynomial of the curve or surface but rather construct a matrix which has a property that can define the curve. The basic method is well known and sketched below, for details see [3, 5].

2.8 Matrix representations of planar curves

We first describe the general method for computing a matrix representation of a rational planar curve via syzygy computations. Consider the parameterization $\varphi : \mathbb{P}^1 \to \mathbb{P}^2$:

$$t = (t_1, t_2) \to (f_1(t), f_2(t), f_3(t)),$$

where $f_i \in \mathbb{R}[t_1, t_2]$ are homogeneous polynomials of the same degree d and for simplicity we assume $\gcd(f_1, f_2, f_3) = 1$, i.e. φ has no base points. In our setting, a point $t \in \mathbb{P}^1$ is called a *base point* if $f_i(t) = 0$ for all $i = 1, 2, 3$. Extensions for addressing base points are well-established [3].

The dehomogenization of φ gives the rational planar curve $\mathcal{C}$ parameterized by

$$\left(\frac{f_1(t_1)}{f_3(t_1)}, \frac{f_2(t_1)}{f_3(t_1)}\right) \subset \mathbb{R}^2 \tag{1}$$

where $f_i(t_1)$ is short for $f_i(t_1, 1)$. A syzygy on the polynomials f_i is a triple (h_1, h_2, h_3) of homogeneous polynomials $h_i \in \mathbb{R}[t_1, t_2]$ that verify the linear relation $\sum_{i=1}^{3} h_i f_i = 0$. We write

$$(h_1, h_2, h_3) \in Syz(f_1, f_2, f_3).$$

By homogenization, the h_1, h_2, h_3 have the same degree, which is the degree of their syzygy. As mentioned before, the set of syzygies $Syz(f_1, f_2, f_3)$ is a graded module; it can be partitioned according to the degree of the syzygies. We fix a degree $v \geq 0$ and consider the set of syzygies of degree v, namely $Syz(f_1, f_2, f_3)_v$, which is known to be an $\mathbb{R}$-vector space. Let $L_1, \dots, L_{N_v}$ be an $\mathbb{R}$-basis of $Syz(f_1, f_2, f_3)_v$, where N_v denotes the basis cardinality.

Assuming (x_1, x_2, x_3) are the homogeneous coordinates of $\mathbb{P}^2$, an equation of the form $\sum_{i=1}^{3} h_i x_i$ is called a *moving line*. We associate each $L_j = (h_1^{(j)}, h_2^{(j)}, h_3^{(j)})$ to its moving line and we develop it in terms of the t_i as follows:

$$\sum_{k=1}^{3} h_k^{(j)} x_k = \sum_{i=1}^{v+1} \Lambda_{i,j}(x_1, x_2, x_3) t_1^{i-1} t_2^{v+1-i},$$

where $\Lambda_{i,j}(x_1, x_2, x_3)$ is a linear polynomial in $\mathbb{R}[x_1, x_2, x_3]$. Now, we can define $\mathbb{M}_v(\varphi)$ as a $(v+1) \times N_v$-matrix whose entry (i,j) is the linear polynomial $\Lambda_{i,j}(x_1, x_2, x_3)$.

After describing the construction method of matrix $\mathbb{M}_v(\varphi)$, we describe the needed property for the matrix to be an implicit matrix representation, i.e. having the property to define the given curve of parameterization φ.

Basic property: For $v \geq d-1$, the matrix $\mathbb{M}_v(\varphi)$ is an implicit matrix representation of the curve $\mathcal{C}$, since it holds the following property: for any point $p = (x_1, x_2, x_3) \in \mathbb{P}^2$ the rank of $\mathbb{M}_v(\varphi)$ evaluated at p drops if and only if p belongs to the algebraic closure of $\mathrm{Im}(\varphi)$ [5].

We dehomogenize by setting $x_3 = 1$ and have the equivalent property for the non-homogeneous setting, that a point $(X, Y) \in \mathbb{R}^2$ belongs to $\mathcal{C}$ if and only if the rank of $\mathbb{M}_v(X, Y)$ drops; the latter denotes the matrix in the non-homogeneous setting.

Thus, we have described the general method of constructing an implicit matrix representation (a matrix having the above property) of a planar curve. The above method can be slightly adapted to compute implicit matrix representations of space curves and space surfaces. [3] provides an in-depth analysis of the method for these additional cases.

We will use this method as our foundation to construct an implicit matrix representation of a curve or surface when the parameterization φ is unknown. Instead, we are given a set of parametric points which will be used to interpolate a basis for the set of syzygies of a certain degree. We direct the reader to Section 5 for the continuation of this contribution.

3. AN IMPROVEMENT TO THE SPARSE INTERPOLATION METHOD

Continuing from the introduction of the implicitization problem and the sparse interpolation method given in Section 2, we focus our attention to the algorithm used in [15, 13] and provide an improvement to its final step, obtaining the implicit polynomial from the kernel polynomials.

3.1 Avoiding GCD computation of kernel polynomials

Our contribution in this section is the avoidance of calculating the *GCD* of the kernel polynomials; a costly action in the case of multivariate polynomials. Instead, we prove that when the basis of the kernel of M is in reduced row echelon form (RREF), then one of the kernel polynomials is the implicit polynomial, and it can be found in linear time on the cardinality of the kernel. This improvement takes advantage of the already built-in functionality of many computer algebra packages of producing the basis of the kernel of a matrix in RREF. Additionally, it avoids multivariate GCD computations and instead relies on standard matrix calculations.

Input: Kernel polynomials of M in RREF k_i, $i = 1, \ldots, N$
Output: Implicit polynomial $p \in \mathbb{R}[x_0, \ldots, x_n]$

// divide by common factor
for $i = 1$ **to** N **do**
  // find in k_i the minimum degree for each variable
  $\alpha_0, \ldots, \alpha_n \leftarrow \min_{\boldsymbol{\alpha} \in \mathcal{SUP}(k_i)}\{\alpha_0\}, \ldots, \min_{\boldsymbol{\alpha} \in \mathcal{SUP}(k_i)}\{\alpha_n\}$;
  $k_i \leftarrow k_i / x_0^{\alpha_0} \cdots x_n^{\alpha_n}$;
end
// find implicit polynomial
$best \leftarrow k_1$;
for $i = 2$ **to** N **do**
  if $\exists j \in \{0, \ldots, n\} : \max_{\boldsymbol{\alpha} \in \mathcal{SUP}(k_i)}\{\alpha_j\} < \max_{\boldsymbol{\alpha} \in \mathcal{SUP}(best)}\{\alpha_j\}$ **then**
    $best \leftarrow k_i$;
  end
  return $best$;
end

Alg. 1: FindImpl

Given the kernel polynomials of M, Algorithm 1 returns the implicit polynomial p. The correctness of this claim is proven in Lemmas 1 and 2. In Lemma 1 we show that one of the kernel polynomials is a multiple of the implicit polynomial p by some monomial, while Lemma 2 shows that Algorithm 1 returns a kernel polynomial of this form.

Lemma 1 (Existence)**.** *Let $v_1, \ldots, v_N$ be a kernel basis of M in RREF. Then there exists $j \in \{1, \ldots, N\}$ such that v_j corresponds to a polynomial*

$$k_j = x^\alpha \cdot p,\ \alpha \in \mathbb{N}^{n+1}$$

i.e. a monomial multiple of the implicit polynomial p.

Proof. Assume monomial order *ord* for the corresponding monomials of the columns of M and that all kernel polynomials are monic, i.e. their leading coefficient is 1.

Case 1: Assume, initially, that corank(M)=1. We shall show that the unique kernel polynomial is $k = \mathbf{x}^{\boldsymbol{\alpha}} \cdot p$. We use the notation $\mathbf{x}^{\boldsymbol{\alpha}}$ to denote a monomial and p is the implicit polynomial.

Assume, towards contradiction, that $k = q \cdot p$ for some $q \in \mathbb{R}[\mathbf{x}]$ with $|\mathcal{SUP}(q)| > 1$ and $\mathcal{SUP}(q) = \{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_{|\mathcal{SUP}(q)|} \mid \boldsymbol{\alpha}_1 >_{ord} \ldots >_{ord} \boldsymbol{\alpha}_{|\mathcal{SUP}(q)|}\}$.

Then, we construct a new polynomial $k' = q \cdot p - \mathbf{x}^{\boldsymbol{\alpha}_1} \cdot p$ and show that it corresponds to kernel vector of M

$$\mathcal{N}(\mathbf{x}^{\boldsymbol{\alpha}_1} \cdot p) \subseteq \mathcal{N}(q \cdot p) \subseteq \mathcal{CH}(S) \Rightarrow \mathcal{SUP}(\mathbf{x}^{\boldsymbol{\alpha}_1} \cdot p) \subseteq S$$

where S is a superset of the support of the implicit polynomial. Here, the second inclusion is derived from the fact the construction of the interpolation matrix, whose kernel vector corresponds to $q \cdot p$. Then,

$$\mathcal{SUP}(q \cdot p), \mathcal{SUP}(\mathbf{x}^{\boldsymbol{\alpha}_1} \cdot p) \subseteq S \Rightarrow \mathcal{SUP}(q \cdot p - \mathbf{x}^{\boldsymbol{\alpha}_1} \cdot p) \subseteq S$$

That is, $k' = q \cdot p - \mathbf{x}^{\boldsymbol{\alpha}_1} \cdot p \neq 0$ is a polynomial with support in S and therefore, corresponds to a kernel vector of M (it is a multiple of p). Additionally, since $\mathcal{LT}(k') <_{ord} \mathcal{LT}(k)$, k' corresponds to a kernel vector of M not spanned by the kernel vector corresponding to k. Therefore, corank$(M) > 1$ which is a contradiction. Therefore, the unique kernel polynomial k is of the form $\mathbf{x}^{\boldsymbol{\alpha}} \cdot p$

Case 2: Now, assume that corank$(M) > 1$ or, equivalently, we have the kernel polynomials $k_1, \ldots, k_i$, $i > 1$. Let the corresponding kernel vectors be in RREF following *ord*. Below, we will describe a procedure that, assuming we begin by an arbitrary kernel vector, constructs a polynomial of strictly lower degree that also belongs to the kernel of the interpolation matrix.

We choose an arbitrary kernel polynomial k. Then, if $k = \mathbf{x}^{\boldsymbol{\alpha}} \cdot p$ for some $\boldsymbol{\alpha} \in \mathbb{R}^n$ we terminate the procedure and consider this a success. Otherwise, if $k = q \cdot p$ for some $q \in \mathbb{R}[\mathbf{x}]$ with $|\mathcal{SUP}(q)| > 1$ and $\mathcal{SUP}(q) = \{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_{|\mathcal{SUP}(q)|} \mid \boldsymbol{\alpha}_1 >_{ord} \ldots >_{ord} \boldsymbol{\alpha}_{|\mathcal{SUP}(q)|}\}$

we construct a polynomial $k' = q \cdot p - \mathbf{x}^{\boldsymbol{\alpha}_1} \cdot p \neq 0$, such that $\mathcal{LT}(k') <_{ord} \mathcal{LT}(k)$, in a similar manner to Case 1. This means that k' corresponds to a vector that belongs to the kernel of M but has a leading term strictly lower than k, with respect to *ord*. Thus, this new kernel vector is not spanned by k. But since it belongs to the kernel of M it must be spanned by the kernel vectors which are in RREF.

If the new polynomial $k' = \mathbf{x}^{\boldsymbol{\alpha}'} \cdot p$ for some $\boldsymbol{\alpha}' \in \mathbb{R}^n$ we again terminate the procedure. If not, we can repeat the procedure and construct a new polynomial, whose corresponding vector belongs to the kernel of M, and which has a leading term strictly lower than the previous polynomial, meaning it is not spanned by any of the previous polynomials.

Notice that this procedure must terminate with a success, meaning we have achieved to construct a polynomial which belongs to the kernel of M, and is of the form $k = \mathbf{x}^{\boldsymbol{\alpha}} \cdot p$. This is because, at each step, if the current polynomial is a multiple of the implicit polynomial by some polynomial (not monomial), we showed that we are able to construct a polynomial with a leading term of strictly smaller degree. By assuming that we do not get a polynomial which is of the form $k = \mathbf{x}^{\boldsymbol{\alpha}} \cdot p$ at any step, we are able to construct an infinite chain of descending leading terms for the constructed polynomials. Since the number of terms $|S|$ is finite and the leading terms belong to S, the previous assumption leads to a contradiction. ■

Lemma 2 (Validity)**.** *The Algorithm 1 (FindImpl) returns the implicit polynomial.*

Proof. Let $k_1, \ldots, k_N$ be the kernel polynomials in RREF. From Lemma 1, we know that at least one is of the form $\mathbf{x}^{\boldsymbol{\alpha}} \cdot p$.

The first step of Algorithm 1 divides each of the kernel polynomials by its common factor. After the first step, we have that $\forall i \in \{1, \ldots, N\} : k_i = p$ or $k_i = q \cdot p$.

We observe a useful property; that $\deg_{x_j}(p) < \deg_{x_j}(q \cdot p)$ for some $j \in \{0, \ldots, n\}$. Thus, if for some kernel polynomial k it holds that $\deg_{x_i}(k) \leq \deg_{x_i}(k_j)$ for all $i \in \{0, \ldots, n\}$ and $j \in \{1, \ldots, N\}$, then $k = p$. ■

3.2 Conclusion

The proposed method for finding the implicit polynomial p when given the kernel polynomials of matrix M has some advantages when compared to any method of multivariate GCD or factoring. Firstly, Algorithm 1 has lower time complexity, i.e. $O(\text{corank}(M) \cdot |S|)$. Secondly, it is based on basic linear algebra and matrix operations instead of number theoretic approaches, like the Chinese remainder theorem and Hensel lifting.

4. APPROXIMATING A KERNEL VECTOR

As we described in the previous sections, the sparse implicitization by interpolation method involves the construction of an interpolation matrix M, whose kernel elements correspond to multiples of the implicit polynomial p by some polynomial q or a monomial $\mathbf{x}^{\alpha}$. The subsequent steps of the method involved the computation of a kernel basis of M. The motivation behind this section is to further speed up the method and to be able to obtain an approximation of just one element of the kernel of the interpolation matrix M without describing the entire kernel. We will test a method for approximating an element of the kernel of a matrix based on a variation of the power method (or power iteration). This kernel element will again correspond to a multiple of the implicit polynomial p by some polynomial q. Finally, we will examine its drawbacks, some of which are inherent in the power method.

4.1 Analysis of the method

Let $M \in \mathbb{R}^{|S|\times|S|}$ be the interpolation matrix from the sparse implicitization by interpolation method, constructed as described in previous sections, where the dimension of the matrix is given by the cardinality of S, i.e. the superset of the predicted support of the implicit polynomial p.

Input: Matrix M, iterations r of PowerMethod
Output: Vector $v \in \ker(M)$

$G \leftarrow MM^T$;
$\lambda \leftarrow$ PowerMethod(G, r);
$G' \leftarrow -G + \lambda I$;
$\lambda, w \leftarrow$ PowerMethod(G',r);
$v \leftarrow M^T w$;
if $v \in \ker(M)$ **then**
  return v;
else
  $G \leftarrow M^T M$;
  $\lambda \leftarrow$ PowerMethod(G, r);
  $G' \leftarrow -G + \lambda I$;
  $\lambda, w \leftarrow$ PowerMethod(G',r);
  return w;
end

Alg. 2: AKV

We will be basing our method on the *power method*, which is a numerical iterative method for computing an approximation of the eigenvalue with the largest absolute value, called the *dominant* eigenvalue and its corresponding eigenvector. One of the disadvantages of the power method is that it fails when the matrix has complex eigenvalues. For this reason, instead of M, we will be using a variation of its Gram matrix. The *Gram matrix* of M is defined as M^TM but we will instead use a variation, namely

$$G = MM^T.$$

The matrix G is real symmetric since $G^T = (MM^T)^T = (M^T)^TM^T = MM^T = G$, and positive-semidefinite since $\forall z,\ z^TGz = z^TMM^Tz = (M^Tz)^T(M^Tz) = \|M^Tz\|_2^2 \geq 0$. Thus, it has real non-negative eigenvalues. By using matrix G instead of M we overcome the possibility of M having complex eigenvalues.

Using the power method iteration, we can compute an approximation of the dominant eigenvalue of G, i.e. $\lambda = \max_i\{|\lambda_i|, i = 1, \ldots, n\}$, where λ_i are the eigenvalues of G. Now, we consider the shifted matrix $G' = -G + \lambda I$. In Lemma 3 below, we shall show that G' has the same dominant eigenvalue as G. Thus, we can apply the power method iteration to G' in order to compute an approximation of the eigenpair (λ, w) of G', where $\lambda \in \mathbb{R}$ is the dominant eigenvalue of both matrix G and G', and $w \in \mathbb{R}^{n\times 1}$ the corresponding eigenvector.

Since (λ, w) is an (approximate) eigenpair of G', we have that

$$G'w = \lambda w \Rightarrow (-G + \lambda I - \lambda I)w = 0 \Rightarrow Gw = 0,$$

meaning vector w is a nontrivial (nonzero) element of the kernel of G. Since $G = MM^T$, we have that

$$MM^Tw = 0 \Rightarrow M(M^Tw) = 0,$$

which means that $M^Tw \in \ker(M)\setminus\{0\}$ or $w \in \ker(M^T)$. We are interested in the first case, i.e. M^Tw is an approximation of a kernel element of matrix M.

If that is not the case, and instead we have that $w \in \ker(M^T)$, we repeat the entire procedure using the Gram matrix $G = M^TM$, instead of the above variation. The real symmetric and positive-semidefinite properties hold for this case too. Thus, the procedure remains the same with the difference being that we are interested in vector w instead of M^Tw. That is because

$$Gw = 0 \Rightarrow M^T(Mw) = 0.$$

The entire procedure is summarized in Algorithm 2 and denoted AKV for approximate kernel vector.

Lemma 3. *If A is a singular positive-semidefinite matrix with dominant eigenvalue λ, then λ is the dominant eigenvalue of matrix $-A + \lambda I$.*

Proof. Since A is singular and positive-semidefinite, all its eigenvalues are non-negative and one of its eigenvalues is $\lambda_1 = 0$. Let A have k eigenvalues and λ be the dominant eigenvalue, so we have

$$0 = \lambda_1 < \lambda_2 < \ldots < \lambda_{k-1} < \lambda$$

where λ_i, $i = 1, \ldots, k-1$ are the eigenvalues other than λ. Let $q(A) = -A + \lambda I$ be a polynomial of matrix A. We know from the properties of eigenvalues that if λ is an eigenvalue of A then $q(\lambda)$ is an eigenvalue of $q(A)$. We can easily show that the eigenvalues of $q(A)$ are arranged as

$$0 = q(\lambda) < q(\lambda_{k-1}) < \ldots < q(\lambda_2) < q(\lambda_1),$$

meaning that $q(\lambda_1) = q(0) = \lambda$ is the dominant eigenvalue of $q(A) = -A + \lambda I$. ■

4.2 Drawbacks of AKV

The method described above is a first attempt to approximate a nontrivial element of the kernel of a given matrix M under the motivation of avoiding any kernel basis computations and further speeding up the method of sparse implicitization by interpolation. As such, there are drawbacks that we will present in this section.

The first drawback is the slow convergence rate of the power method iteration when $\frac{|\lambda_1|}{|\lambda_2|} \approx 1$, where $\lambda_1 > \lambda_2$ the two eigenvalues with the largest absolute values. In our experiments, matrices of large dimensions, which is the case when the predicted support of the implicit polynomial is contained in a large superset, the slow convergence rate led to computational times that exceeded the time for the exact computation of a kernel basis of M. Moreover, since the power method is used twice in algorithm AKV, any of the two cases $\frac{\lambda}{\lambda_{k-1}} \approx 1$ and $\frac{q(\lambda_1)}{q(\lambda_2)} \approx 1$, following the notation of Lemma 3, can lead to slow convergence rates.

The second drawback is that algorithm AKV computes a multiple of the implicit polynomial p by another polynomial. Since the computations are numeric and we are working in the approximate setting, we must use approximate factoring if we wish to extract the approximate implicit polynomial. This further increases the computational cost of the method.

Another drawback of the proposed method is that it is not evident beforehand whether the computed vector $v = M^T w$ belongs to $\ker(M)$ or $w \in \ker(M^T)$.

4.3 Conclusion and future work

We provided an initial attempt to the problem of computing a nontrivial element of the kernel of a matrix M using power method. The drawbacks of the method, as we described in the previous section, are far from minor and render the method not practical in most settings.

In numerical analysis, there exist various methods other than the power method, such as the Rayleigh quotient method and the QR method etc, to compute eigenvalues (dominant or not) of a given matrix. Every method is sensitive to the type of the input matrix, and as such can be tested against the structured Vandermonde-type interpolation matrix M of the implicitization by interpolation method.

The initial motivation of this method was to compute an approximation of a particular element of the kernel of matrix M, namely a kernel vector which corresponds to a polynomial which is a multiple of the implicit polynomial p by some monomial (and not polynomial). We already showed how one can readily obtain such a vector when a basis of the kernel is known in RREF. Methods for the computation of the sparsest kernel vector can be tested to achieve the desired result [17]

5. SYZYGIES AND INTERPOLATION

In this section, we will provide a method for interpolating the syzygies of a set of parametric points. After computing a basis for the $\mathbb{R}-$vector space of the syzygies of a certain homogeneous degree, we will rely on the method we described in Section 2 for constructing an implicit matrix representation. We will also provide a method for computing the degree of the parameterization since we will be using it for the construction of the implicit matrix representation.

In particular, we will provide a method for constructing an implicit matrix representation under the above assumptions for each of the following cases: planar curves, space curves and space surfaces. Each method has some minor differences that we will describe in detail below.

We begin with the case of planar curves.

5.1 Matrix representations of planar curves using interpolation

Consider a planar curve $\mathcal{C}$ for which there exists a rational parameterization $\varphi : \mathbb{R}^1 \to \mathbb{R}^2$,

$$\varphi : t \to \left(X(t) = \frac{f_1(t)}{f_3(t)}, Y(t) = \frac{f_2(t)}{f_3(t)} \right), \tag{2}$$

where φ is not known. Instead, the input is a set of triplets of the form

$$(\tau_1; X_1, Y_1), (\tau_2; X_2, Y_2), \ldots$$

such that $\varphi(\tau_k) = (X_k, Y_k)$, for a range of k to be defined below. This means that we have as input a set of points that belong to $\mathcal{C}$ along with the value of the parameter t for each point. We will call this set a parametric set of points and each of these points a parametric point. These triplets are sampled following the scenarios described in Section 2 , for instance when φ is an arc-length parameterization and the triplets are sampled by a scanner following $\mathcal{C}$. We now provide an algorithm for computing an implicit matrix representation of the curve $\mathcal{C}$ described by this parametric trail of points.

Initially, the algorithm fixes a degree $\nu \geq 0$ for the degree of the syzygies it will compute. Then, the algorithm shall compute an $\mathbb{R}$-basis for $Syz(X, Y, 1)_\nu$. Since the rational functions of $X(t), Y(t)$ are not explicitly known, we compute the basis in the following manner.

Consider the moving line $h_1X + h_2Y + h_3 = 0$. The expanded form of each h_i is

$$h_i = \sum_{\delta=0}^{\nu} h_{i,\delta} t^\delta \in \mathbb{R}[t], \; i = 1, 2, 3, \tag{3}$$

where the $h_{i,\delta}$ are (unknown) coefficients. These are exactly the coefficients we need to compute in order to gain knowledge of the syzygies that are needed to construct the implicit matrix representation of $\mathcal{C}$. This set of coefficients is an $\mathbb{R}-$vector space and we will immediately show how it corresponds to the set of syzygies. The moving line can be rewritten as

$$\sum_{\delta=0}^{v} t^{\delta} X h_{1,\delta} + \sum_{\delta=0}^{v} t^{\delta} Y h_{2,\delta} + \sum_{\delta=0}^{v} t^{\delta} h_{3,\delta} = 0.$$

Such equations are going to be used to determine the $3(v+1)$ unknown coefficients $h_{i,\delta}$ by interpolation at the known input triplets. For this, we define a $3(v+1) \times 3(v+1)$ matrix H whose rows are indexed by evaluations $t = \tau_k$, and each row expresses the above equation as follows:

$$[X_k, \tau_k X_k, \ldots, \tau_k^v X_k,\ Y_k, \tau_k Y_k, \ldots, \tau_k^v Y_k,\ 1, \tau_k, \ldots, \tau_k^v]\,.$$

We compute a basis of the kernel of matrix H and rewrite each kernel basis vector

$$(h_{1,0}^{(j)}, \ldots, h_{1,v}^{(j)}, h_{2,0}^{(j)}, \ldots, h_{2,v}^{(j)}, h_{3,0}^{(j)}, \ldots, h_{3,v}^{(j)})$$

as $(h_1^{(j)}, h_2^{(j)}, h_3^{(j)})$ following equation (3). We can easily show that the triplets $(h_1^{(j)}, h_2^{(j)}, h_3^{(j)})$, $j = 1, \ldots, N_v$ form an $\mathbb{R}$-basis of $Syz(X, Y, 1)_v$. This way, we have achieved to compute the needed $\mathbb{R}$-basis of $Syz(X, Y, 1)_v$ without knowing the polynomials f_i, $i = 1, 2, 3$ "behind" the parametric points. We use this basis to construct the matrix $\mathbb{M}_v(X, Y)$.

We focus our attention on the basic property of an implicit matrix representation described in Section 2.8. In order for the matrix $\mathbb{M}_v(X, Y)$ to be a matrix representation of $\mathcal{C}$, the implicit curve $\mathcal{C}$, v must verify the inequality $v \geq d - 1$. Since the degree d of the parameterization is unknown in our setting, we establish the following lemma.

Lemma 4. *Consider a rational parametric curve $\mathcal{C}$ of the form (2). Let d be the homogeneous degree of the (unknown) f_i, $i = 1, 2, 3$, $v \geq 0$ be the degree chosen by the algorithm and $h = \dim \ker(H)$ be the cardinality of the kernel basis of H. Then,*

1. *$h < v + 1$ if and only if $v < d - 1$.*
2. *$h = v + 1$ if and only if $v = d - 1$.*
3. *$h > v + 1$ if and only if $v > d - 1$.*

Proof Sketch. By construction, the kernel basis of H corresponds to an $\mathbb{R}$-basis of $Syz(f_1, f_2, f_3)_v$ and, thus, $N_v = h$. The idea behind the proof is that $N_{d-1} = d$ [9] and that N_v is an increasing function of v.

As a consequence, in the case $h \geq v + 1$, the algorithm yields a valid implicit matrix representation $\mathbb{M}_v(X, Y)$ of $\mathcal{C}$. Lemma 4 is a very useful tool for our setting. Furthermore, Lemma 4 allows us to compute d by constructing matrix H and comparing h with the selected v.

Example 5. *Consider the folium of Descartes curve* affinely *parameterized as:*

$$\mathcal{C} = \left\{ \left(\frac{3t}{t^3+1}, \frac{3t^2}{t^3+1} \right) \in \mathbb{R}^2 : t \in \mathbb{R} \right\} \tag{4}$$

We will be using equation (4) only to sample random points of $\mathcal{C}$ for various values of the parameter t and using them as triplets $(\tau_k; X_k, Y_k)$ to construct the matrix H as described above, so we imply that we have no knowledge of the parametric equation. We try different values of v.

For $v_1 = 1$, the $\mathbb{R}$-basis of $Syz(X, Y)_1$ is $\{(-t, 1, 0)\}$, that is we are in case 1 of Lemma 4 since $N_{v_1} < v_1 + 1$. The kernel basis cardinality is not adequate to construct a valid matrix representation of $\mathcal{C}$.

For $v_2 = 2$, the computed basis of $Syz(X, Y)_2$ is $\{(-t^2, t, 0), (-t, 1, 0), (-1/3, -t^2/3, t)\}$, that is case 2 of Lemma 4. That is to be expected since we picked $v_2 = d - 1$ (notice $d = 3$ for curve $\mathcal{C}$). Any $v \geq v_2$ is a valid choice to construct the implicit representation matrix $\mathbb{M}_v(X, Y)$.

For $v_2 = 2$, the matrix is

$$\mathbb{M}_2(X, Y) = \begin{bmatrix} -X & 0 & -Y/3 \\ Y & -X & 1 \\ 0 & Y & -X/3 \end{bmatrix} \tag{5}$$

Let us test the drop-of-rank property at this point. Notice that $\operatorname{rank} \mathbb{M}_2(3, 3) = 3$ *since* $(3, 3) \notin \mathcal{C}$*. By testing a point that belongs to the curve $\mathcal{C}$, for example $(\frac{3}{2}, \frac{3}{2})$, we have that* $\operatorname{rank} \mathbb{M}_2(\frac{3}{2}, \frac{3}{2}) = 2 < 3$*.*

Additionally, we can test point $(0, 0)$ which is a point of intersection for curve $\mathcal{C}$, i.e. there exist more than one parameter values for t that output the same point. In this case, $\operatorname{rank} \mathbb{M}_2(0, 0) = 1$*. Notice that the drop-of-rank is by a value of 2; the number of corresponding parameter values for this point. This additional property is explored in [3].*

5.2 Matrix representations of space curves using interpolation

The method we have described extends naturally to the case of space curves. In this case, the degree for computing the matrix representation is $2d - 1$, meaning v must be greater or equal to $2d - 1$, where d is similarly defined as the homogeneous degree of the polynomials $f_i(t_1, t_2)$, $i = 1, \ldots, 4$. The same property (”critical” degree) holds for rational surfaces, where d is defined as the total degree of the polynomials $f_i(t_1, t_2, t_3)$, $i = 1, \ldots, 4$. Hence our method reduces the computation of syzygies, and eventually an implicit matrix representation, to interpolation through a point sample obtained by one of the aforementioned scenarios.

Similarly to the previous section, consider a space curve $\mathcal{C}$ for which there exists a rational parameterization $\varphi : \mathbb{R}^1 \to \mathbb{R}^3$,

$$\varphi : t \to \left(X(t) = \frac{f_1(t)}{f_4(t)}, Y(t) = \frac{f_2(t)}{f_4(t)}, Z(t) = \frac{f_3(t)}{f_4(t)} \right), \tag{6}$$

where φ is unknown. The input in this case are 4-tuples of the form

$$(\tau_1; X_1, Y_1, Z_1), (\tau_2; X_2, Y_2, Z_2), \ldots$$

Again, we fix the degree $v \geq 0$ of the syzygies and consider the moving plane $h_1 X + h_2 Y + h_3 Z + h_4 = 0$. Each of the h_i, $i = 1, \ldots, 4$ can be written as

$$h_i = \sum_{\delta=0}^{v} h_{i,\delta} t^{\delta} \in \mathbb{R}[t],\ i = 1, 2, 3, 4, \tag{7}$$

where the $h_{i,\delta}$ are (unknown) coefficients. Thus, the moving plane can be rewritten as

$$\sum_{\delta=0}^{v} t^{\delta} X h_{1,\delta} + \sum_{\delta=0}^{v} t^{\delta} Y h_{2,\delta} + \sum_{\delta=0}^{v} t^{\delta} Z h_{3,\delta} + \sum_{\delta=0}^{v} t^{\delta} h_{4,\delta} = 0.$$

We, again, will use the above equations to determine the $4(v+1)$ unknown coefficients $h_{i,\delta}$ by interpolation at the known input triplets. We define a $4(v+1) \times 4(v+1)$ matrix H whose rows are indexed by evaluations $t = \tau_k$, and each row expresses the above equation as follows:

$$[X_k, \tau_k X_k, \ldots, \tau_k^v X_k,\ Y_k, \tau_k Y_k, \ldots, \tau_k^v Y_k, Z_k, \tau_k Z_k, \ldots, \tau_k^v Z_k,\ 1, \tau_k, \ldots, \tau_k^v]\,.$$

We compute a basis of the kernel of matrix H and rewrite each kernel basis vector

$$(h_{1,0}^{(j)}, \ldots, h_{1,v}^{(j)}, h_{2,0}^{(j)}, \ldots, h_{2,v}^{(j)}, h_{3,0}^{(j)}, \ldots, h_{3,v}^{(j)}, h_{4,0}^{(j)}, \ldots, h_{4,v}^{(j)})$$

as $(h_1^{(j)}, h_2^{(j)}, h_3^{(j)}, h_4^{(j)})$ following equation (7). The 4-tuples $(h_1^{(j)}, h_2^{(j)}, h_3^{(j)}, h_4^{(j)})$, $j = 1, \ldots, N_v$ form an $\mathbb{R}$-basis of $Syz(X, Y, Z, 1)_v$. We use this basis to construct the matrix $\mathbb{M}_v(X, Y, Z)$ in a similar manner.

The basic difference between planar curves and space curves (and space surfaces as we will later see) is the ”critical” degree for the chosen degree v for the degree of the syzygies. For the case of planar curves, that was $d - 1$, where d was the degree of the parameterization. For space curves, that ”critical” degree is $2d - 1$. We do not give a detailed analysis for this result and direct the reader to [3] for more information on the topic.

5.3 Matrix representations of space surfaces using interpolation

In this section, a similar method for computing an implicit matrix representation will be given for the case of space surfaces. The input consists of parametric points that belong to the surface and the method is based on computing the syzygies of a certain homogeneous degree of the underlying polynomials of the rational surface.

Consider a space surface $\mathcal{S}$ for which there exists a rational parameterization $\varphi : \mathbb{R}^2 \to \mathbb{R}^3$,

$$\varphi : \mathbf{t} = (t_1, t_2) \to \left(X(\mathbf{t}) = \frac{f_1(\mathbf{t})}{f_4(\mathbf{t})}, Y(\mathbf{t}) = \frac{f_2(\mathbf{t})}{f_4(\mathbf{t})}, Z(\mathbf{t}) = \frac{f_3(\mathbf{t})}{f_4(\mathbf{t})} \right), \tag{8}$$

where φ is unknown and instead the input consists of 5-tuples of the form

$$(\boldsymbol{\tau}_1; X_1, Y_1, Z_1), (\boldsymbol{\tau}_2; X_2, Y_2, Z_2), \ldots$$

where $\boldsymbol{\tau}_k \in \mathbb{R}^2$ is an abbreviation of the $k-$th 2-dimensional parameter value of (t_1, t_2) that corresponds to point $(X_k, Y_k, Z_k) \in \mathbb{R}^3$. Again, we fix the degree $v \geq 0$ of the syzygies and consider the moving plane $h_1 X + h_2 Y + h_3 Z + h_4 = 0$. Each of the $h_i \in \mathbb{R}[\mathbf{t}]$, $i = 1, \ldots, 4$ can be written as

$$h_i = \sum_{\substack{\boldsymbol{\delta}=(\delta_1,\delta_2)\\ \delta_1,\delta_2 \geq 0 \\ \delta_1+\delta_2 \leq v}} h_{i,\boldsymbol{\delta}} \mathbf{t}^{\boldsymbol{\delta}} \in \mathbb{R}[\mathbf{t}], \; i = 1, 2, 3, 4, \tag{9}$$

where the $h_{i,\boldsymbol{\delta}}$ is the unknown coefficient of $\mathbf{t}^{\boldsymbol{\delta}} = t_1^{\delta_1} t_2^{\delta_2}$. Thus, in the surface case, the moving plane can be rewritten as

$$\sum_{\substack{\boldsymbol{\delta}=(\delta_1,\delta_2)\\ \delta_1,\delta_2 \geq 0 \\ \delta_1+\delta_2 \leq v}} \mathbf{t}^{\boldsymbol{\delta}} X h_{1,\boldsymbol{\delta}} + \sum_{\substack{\boldsymbol{\delta}=(\delta_1,\delta_2)\\ \delta_1,\delta_2 \geq 0 \\ \delta_1+\delta_2 \leq v}} \mathbf{t}^{\boldsymbol{\delta}} Y h_{2,\boldsymbol{\delta}} + \sum_{\substack{\boldsymbol{\delta}=(\delta_1,\delta_2)\\ \delta_1,\delta_2 \geq 0 \\ \delta_1+\delta_2 \leq v}} \mathbf{t}^{\boldsymbol{\delta}} Z h_{3,\boldsymbol{\delta}} + \sum_{\substack{\boldsymbol{\delta}=(\delta_1,\delta_2)\\ \delta_1,\delta_2 \geq 0 \\ \delta_1+\delta_2 \leq v}} \mathbf{t}^{\boldsymbol{\delta}} h_{4,\boldsymbol{\delta}} = 0.$$

We will use the above equations to determine the unknown coefficients $h_{i,\boldsymbol{\delta}}$ by interpolation at the known input triplets. The unknown coefficients are $2(v+1)(v+2)$, so we define a $2(v+1)(v+2) \times 2(v+1)(v+2)$ matrix H whose rows are indexed by evaluations $t = \tau_k$, and each row contains the coefficients of all $h_{i,\boldsymbol{\delta}}$ in the above equation.

We compute a basis of the kernel of matrix H and rewrite each kernel basis vector as $(h_1^{(j)}, h_2^{(j)}, h_3^{(j)}, h_4^{(j)})$ following equation (9). The 4-tuples $(h_1^{(j)}, h_2^{(j)}, h_3^{(j)}, h_4^{(j)})$, $j = 1, \ldots, N_v$ form an $\mathbb{R}$-basis of $Syz(X, Y, Z, 1)_v$. We use this basis to construct the matrix $\mathbb{M}_v(X, Y, Z)$.

As with the case of the space curves, the "critical" degree of the syzygies is $2d-1$, meaning that the chosen value of v must be greater or equal to $2d-1$, where d is the homogeneous degree of the f_i, $i = 1, 2, 3, 4$.

5.4 Conclusion

In this section, we provided a method for interpolating syzygies of polynomials from a parameterization, from a set of parametric points, when the underlying parameterization is unknown. We did this for the case of planar and space curves and space surfaces.

Additionally, we provided a method for determining the degree of the parameterization of the curve or surface from the set of parametric points.

This setting can be further studied for the case of curves or surfaces of higher dimension. Furthermore, the numerical approach to this method can be studied. Initial attempts to compute the syzygies from this set of parametric points numerically lead to a kernel of full rank. This makes the determination of the degree of the parameterization a nontrivial task. Ideas for future work at this direction include the use of the numerical rank for the determination of the dimension of the kernel.

6. GENERAL CONCLUSION AND FUTURE WORK

In this thesis, we based our contributions on two of the existing methods for the implicitization problem, namely the sparse implicitization by interpolation method and the method of implicit matrix representations based on the theory of syzygies. In this final section, we discuss the future work for two of the three contributions of this thesis, namely the contributions of sections 4 and 5.

As we already mentioned, the motivation for the contribution of Section 4 was to compute an approximation of a particular element of the kernel of the interpolation matrix M, namely a kernel vector which corresponds to a polynomial which is a multiple of the implicit polynomial p by some monomial (and not polynomial). We provided a method based on the power method and explored its drawbacks, so future work towards this direction would include experimentation with different numerical methods to compute an approximation of the desired kernel vector, by taking into account the Vandermonde-type structure of the interpolation matrix M. Additionally, the desired kernel vector may be closely related to the sparsest kernel vector, so the method used in [17] can be studied for this scenario.

In section 5, we provided a method for interpolating syzygies of polynomials from a set of parametric points, when the underlying parameterization is unknown. We did this for the case of planar and space curves and space surfaces. As we already mentioned in the same section, this setting can be further studied for the case of curves or surfaces of higher dimension. Moreover, the numerical approach to this method can be examined, i.e. the approximation of the required syzygies, which will lead to an approximate matrix representation of the curve or surface. Future work might include the use of the numerical rank for the determination of the dimension of the kernel, which is used for the calculation of the degree of the parameterization. Numerical experiments could show the accepted threshold of the drop-of-rank property for the approximate implicit matrix representation.

REFERENCES

[1] O. J. D. Barrowclough, and T. Dokken. *Approximate implicitization using linear algebra*. J. Appl. Math., 25, 2012.

[2] N. Botbol, and A. Dickenstein. *Implicitization of rational hypersurfaces via linear syzygies: a practical overview*. preprint, 2015.

[3] L. Busé. *Implicit matrix representations of rational Bézier curves and surfaces*. Computer-Aided Design, Elsevier, 2013 SIAM Conference on Geometric and Physical Modeling, 46, pp.14-24, 2014.

[4] L. Buse, D. Cox, and C. D'Andrea. *Implicitization of surfaces in the projective space in the presence of base points*, Journal of Algebra and Its Applications, 2 (2003) vol. 2, 189-214.

[5] L. Busé, and T. Luu Ba. *Matrix-based Implicit Representations of Rational Algebraic Curves and Applications.* Computer Aided Geometric Design, 27(9):681–699, 2010.

[6] D. A. Cox. *Curves, surfaces, and syzygies*. In Topics in algebraic geometry and geometric modeling, volume 334 of Contemp. Math, 131–150. Amer. Math.Soc, Providence, RI, 2003.

[7] D. Cox, R. Goldman, and M. Zhang. *On the validity of implicitization by moving quadrics of rational surfaces with no base points*. J. Symbolic Comput., 29(3):419-440, 2000.

[8] D. A. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra, 3/e (Undergraduate Texts in Mathematics)*, Springer-Verlag, New York, 2007.

[9] D. A. Cox, J. Little, and D. O'Shea. *Using algebraic geometry*. Springer-Verlag, New York, 1998.

[10] T. Dokken. *Approximate implicitization*. In Mathematical methods for curves and surfaces (Oslo, 2000), Innov. Appl. Math., pages 81–102. Vanderbilt Univ. Press, Nashville, TN, 2001.

[11] T. Dokken, and J.B. Thomassen. *Overview of approximate implicitization*. Topics in algebraic geometry and geometric modeling, 334, pages 169-184, 2003.

[12] T. Dokken, and J.B. Thomassen. *Weak approximate implicitization*. In Proceedings of IEEE International Conference on Shape Modeling and Applications 2006, pages 204-214, Los Alamitos, CA, USA, 2006, IEEE Computer Society.

[13] I. Z. Emiris, T. Kalinka, C. Konaxis. *Geometric operations using sparse interpolation matrices*. Graphical Models, Volume 82, November 2015, Pages 99-109.

[14] I. Z. Emiris, T. Kalinka, C. Konaxis, and L. Ba Thang. *Implicitization of curves and (hyper)surfaces using predicted support*. Theoretical Computer Science - Special Issue on Symbolic and Numeric Computation, 2012.

[15] I. Z. Emiris, T. Kalinka, C. Konaxis, and T. Luu Ba. *Sparse implicitization by interpolation: Characterizing non-exactness and an application to computing discriminants*. Computer-Aided Design - Special Issue on Solid and Physical Modeling, 2012, 45 (2). pp. 252-261.

[16] M. S. Floater, and T. Surazhsky. *Parameterization for curve interpolation*. Studies in Computational Mathematics, 2006.

[17] E. L. Kaltofen, and M. Nehring. *Supersparse Black Box Rational Function Interpolation*. Proceedings of the 36th International Symposium on Symbolic and Algebraic Computation, 2011.

[18] T.W. Sederberg, and F. Chen. *Implicitization using moving curves and surfaces*. In R. Cook, editor, Proc. SIGGRAPH, pages 301-308. Addison Wesley, 1995.