Minkowski Decomposition and Geometric Predicates in Sparse Implicitization

Ioannis Z. Emiris
University of Athens,Greece
emiris@di.uoa.gr

Christos Konaxis
University of Athens,Greece
ckonaxis@di.uoa.gr

Zafeirakis Zafeirakopoulos
University of Athens,Greece
zafeirakopoulos@gmail.com

ABSTRACT

Based on the computation of a polytope Q, called the predicted polytope, containing the Newton polytope P of the implicit equation, implicitization of a parametric hypersurface is reduced to computing the nullspace of a numeric matrix. Polytope Q may contain P as a Minkowski summand, thus jeopardizing the efficiency of sparse implicitization. Our contribution is twofold. On one hand we tackle the aforementioned issue in the case of 2D curves and 3D surfaces by Minkowski decomposing Q thus detecting the Minkowski summand relevant to implicitization: we design and implement in `Sage` a new, public domain, practical, potentially generalizable and worst-case optimal algorithm for Minkowski decomposition in 3D based on integer linear programming. On the other hand, we formulate basic geometric predicates, namely membership and sidedness for given query points, as rank computations on the interpolation matrix, thus avoiding to expand the implicit polynomial. This approach is implemented in `Maple`.

Categories and Subject Descriptors

I.1.2 [**Symbolic and Algebraic Manipulation**]: Algorithms—*Algebraic algorithms*; J.6 [**Computer-aided engineering**]: Computer-aided design

General Terms

Algorithms, Theory

Keywords

matrix representation; sparse implicitization; Newton polytope; interpolation; Minkowski decomposition; integer linear programming; membership; sidedness

1. INTRODUCTION

A fundamental question in changing representation of geometric objects is implicitization, namely the process of changing the representation of a geometric object from parametric to implicit. It is a basic operation with several applications in computer-aided geometric design (CAGD) and geometric modeling. There have been numerous approaches for implicitization, including resultants, Gröbner bases, moving lines and surfaces, and interpolation techniques.

In this work, we restrict attention to hypersurfaces and exploit a matrix representation of hypersurfaces without developing the actual implicit equation. Our approach is based on potentially interpolating the unknown coefficients of the implicit polynomial, but we shall avoid actually computing these coefficients when defining our geometric predicates. The basis of this approach is a sparse interpolation matrix, sparse in the sense that it is constructed when one is given a superset of the monomials in the implicit polynomial.

We call the support and the Newton polytope of the implicit equation, *implicit support* and *implicit polytope*, respectively. Its vertices are called *implicit vertices*. The implicit polytope is computed from the Newton polytope of the sparse (or toric) resultant, or *resultant polytope*, of polynomials defined by the parametric equations, thus exploiting the input and output sparseness, in other words, the structure of the parametric equations as well as the implicit polynomial. Under certain genericity assumptions, the implicit polytope coincides with a projection of the resultant polytope, which we call *predicted polytope*, see Section 2. In general, the predicted polytope contains the Minkowski sum of the implicit polytope and an extraneous polytope, which may be a single point. The set of lattice points in the predicted polytope, called the predicted support, is a superset of the implicit support, modulo the Minkowski summand.

The predicted support is used to build a numerical matrix whose kernel is, ideally, 1-dimensional, thus yielding (up to a nonzero scalar multiple) the coefficients corresponding to the predicted implicit support. This is a standard case of *sparse interpolation* of the polynomial from its values. When dealing with hypersurfaces of high dimension, or when the support contains a large number of lattice points, then exact solving is expensive. Since the kernel can be computed numerically, our approach also yields an approximate sparse implicitization method. Our results also apply when the hypersurface is given as a point cloud but in this case a d-simplex is used as the predicted polytope, where d is an estimation of the total degree of the implicit equation.

Contribution.

The contribution of this work is twofold: first we show the usefulness of Minkowski decomposition of the predicted polytope to improve the efficiency of interpolating the im-

ISSAC'15, July 6–9, 2015, Bath, United Kingdom.

DOI: http://dx.doi.org/10.1145/2755996.2756661.

plicit equation. A full-dimensional indecomposable Minkowski summand of the predicted polytope, when it exists, may be the exact implicit polytope. We offer an algorithm, and a public-domain implementation in `Sage`, for computing the 1-skeleton and the V-representation of such summand in $\mathbb{R}^2$ and $\mathbb{R}^3$, by using integer linear programming (ILP), while excluding homothetic summands. Second, we utilize sparse interpolation matrices to formulate some geometric problems as questions in numerical linear algebra. In particular, we reduce the membership test $p(q) = 0$, for a query point q and a hypersurface defined implicitly by $p(x) = 0$, to a rank test on an interpolation matrix for $p(x)$. Moreover, when this matrix is non-singular, we use the (nonzero) sign of its determinant to decide sidedness for query points q with nonzero coordinates that do not lie on the surface $p(x) = 0$. We have implemented these algorithms in `Maple 14`. A beta-version is publicly available[1].

The rest of the paper is organized as follows: Next, we overview previous work. Section 2 describes the overall approach. Section 3 presents an algorithm for Minkowski decomposition, focusing on the 3-dimensional case. Section 4 presents matrix constructions that reduce membership and sidedness predicates to numerical linear algebra. We conclude with future work and open questions.

Previous work.

If S is a superset of the implicit support, then the most direct method to reduce implicitization to linear algebra is to construct a $|S| \times |S|$ matrix M, indexed by monomials with exponents in S evaluated at $|S|$ different values. Then the vector of coefficients of the implicit equation is in the kernel of M. This idea was used in [4, 10, 14]; it is also the starting point of this paper.

Our method of sparse implicitization was introduced in [4], where the overall algorithm was presented together with a preliminary implementation, including the case of approximate sparse implicitization. The emphasis of that work was on sampling and oversampling the parametric object so as to create a numerically stable matrix, and examined evaluating the monomials at random integers, random complex numbers of modulus 1, and complex roots of unity.

One issue was that the kernel of the matrix might be of high dimension, in which case the equation obtained may be a multiple of the implicit equation. In [5] they show that if the kernel is not 1-dimensional then the predicted polytope is the Minkowski sum of the implicit polytope and an extraneous one. The true implicit polynomial is obtained as the greatest common divisor (GCD) of the polynomials corresponding to at least two and at most all of the kernel vectors, or via multivariate polynomial factoring.

There are methods for the computing the implicit polytope based on tropical geometry [13, 14], cf. [3]. Sparse implicitization relies on computing the Newton polytope of the sparse resultant or its orthogonal projection along a given direction [6], implemented in `ResPol`[2].

Matrix representations in geometric modeling are not new. A major current direction is based on the theory of moving curves and surfaces. In [2], they use generalized matrix-based representations of parameterized surfaces in order to represent the intersection curve of two such surfaces as the zero set of a matrix determinant. In [1] they introduce a new implicit representation of rational Bézier curves and surfaces in $\mathbb{R}^3$, namely a matrix whose entries depend on the space variables and whose rank drops exactly on this curve or surface.

Previous work on Minkowski decomposition algorithms mainly focuses on $\mathbb{R}^2$. In [8] the authors give an algorithm deciding decomposability in $\mathbb{R}^2$ (in pseudo-polynomial time) and a generalization in higher dimensions. A subset-sum based pseudo-polynomial time algorithm for decomposition of polygons is given in [7]. Smilansky [12] offers decomposability criteria by considering the space of affine dependences of the vertices of the dual polytope. In [9] they reduce the problem of deciding absolute irreducibility of multivariate polynomials to Minkowski decomposability of a lattice polytope which in turn is reduced to ILP. Our approach differs in the construction of the ILP: we exclude homothetic summands and favor balanced-size summands. Moreover, we employ a combinatorial algorithm to obtain a V-representation of the summand.

2. IMPLICITIZATION BY SUPPORT PREDICTION

This section describes how sparse elimination can be used to compute the implicit polytope by exploiting sparseness and how this can reduce implicitization to linear algebra. We also discuss how the quality of the predicted support affects the implicitization algorithm and develop the necessary constructions that allow us to formulate the membership and sidedness criteria in the next sections.

A *parameterization* of a geometric object of co-dimension one, in a space of dimension $n+1$, can be described by a set of parametric functions:

$$x_i = f_i(t_1, \ldots, t_n) : \Omega \to \mathbb{R}$$

where $i = 0, 1, 2, \ldots, n$, $\Omega := \Omega_1 \times \cdots \times \Omega_n$, $\Omega_i \subseteq \mathbb{R}$ and $t := (t_1, t_2, \ldots, t_n)$ is the vector of parameters and $f := (f_0, \ldots, f_n) : \Omega \to \mathbb{R}^{n+1}$ is a vector of continuous functions, also called *coordinate 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 $f : \mathbb{R}^n \to \mathbb{R}^{n+1} : t \mapsto f(t)$. Implicitization of planar curves and surfaces in three dimensional space corresponds to $n = 1$ and $n = 2$ respectively. The image of f is contained in the variety defined by the ideal of all polynomials $p(x_0, \ldots, x_n)$ such that $p(f_0(t), \ldots, f_n(t)) = 0$, for all t in Ω. We restrict ourselves to the case when this is a principal ideal, and we wish to compute its unique (up to a constant multiple) defining polynomial $p(x_0, \ldots, x_n) \in \mathbb{R}[x_0, \ldots, x_n]$, given its Newton polytope $P = N(p) \subset \mathbb{R}^{n+1}$. We can regard the variety in question as the (closure of the) projection of the graph of map f to the last $n+1$ coordinates. Assuming the rational parameterization

$$x_i = f_i(t)/g_i(t),\ i = 0, \ldots, n, \tag{1}$$

implicitization is reduced to eliminating t from the polynomials in $(\mathbb{R}(x_0, \ldots, x_n))[t, y]$:

$$\begin{aligned} F_i &:= x_i g_i(t) - f_i(t),\ i = 0, \ldots, n, \\ F_{n+1} &:= 1 - y g_0(t) \cdots g_n(t), \end{aligned} \tag{2}$$

[1] `http://ergawiki.di.uoa.gr/index.php/Implicitization`
[2] `http://sourceforge.net/projects/respol`

where y is a new variable and F_{i+1} assures that all $g_i(t) \neq 0$. If one omits F_{n+1}, the generator of the corresponding (principal) ideal would be a multiple of the implicit equation. Then the extraneous factor corresponds to the g_i. Eliminating t, y is done by the *resultant* of the polynomials in (2).

Let $A_i \subset \mathbb{Z}^{n+1}$, $i = 0, \ldots, n+1$ be the supports of the polynomials F_i and consider the generic polynomials

$$F'_0, \ldots, F'_n, F'_{n+1} \tag{3}$$

with the same supports A_i and symbolic coefficients c_{ij}.

DEFINITION 1. *Their sparse resultant $Res(F'_0, \ldots, F'_{n+1})$ is a polynomial in the c_{ij} with integer coefficients, namely*

$$\mathcal{R} \in \mathbb{Z}[c_{ij} : i = 0, \ldots, n+1, j = 1, \ldots, |A_i|],$$

which is unique up to sign and vanishes if and only if the system $F'_0 = F'_1 = \cdots = F'_{n+1} = 0$ has a common root in a specific variety. This variety is the projective variety $\mathbb{P}^n$ over the algebraic closure of the coefficient field in the case of projective (or classical) resultants, or the toric variety defined by the A_i's.

The implicit equation of the parametric hypersurface defined in (2) equals the resultant $\mathrm{Res}(F_0, \ldots, F_{n+1})$, provided that the latter does not vanish identically. $\mathrm{Res}(F_0, \ldots, F_{n+1})$ can be obtained from $\mathrm{Res}(F'_0, \ldots, F'_{n+1})$ by specializing the symbolic coefficients of the F'_i's to the actual coefficients of the F_i's, provided that this specialization is generic enough. Then the implicit polytope P equals the projection Q of the resultant polytope to the space of the implicit variables, i.e. the Newton polytope of the specialized resultant, up to some translation. We shall call Q the *predicted (implicit) polytope*. When the specialization of the c_{ij} is not generic enough, then Q contains a translate of P. This follows from the fact that the method computes the same resultant polytope as the tropical approach, see [13, Prop.5.3]. Note that there is no exception even in the presence of base points.

Our method is based on computing the predicted polytope Q, given the Newton polytopes of the polynomials in (2). Then the implicit support is a subset of the set of lattice points contained in the predicted polytope, modulo the Minkoswki summand. For computing Q we employ [6] and software `ResPol`.

EXAMPLE 1. *Eight-surface parameterization:*

$$\left(\frac{4s(-1+t^2)(-1+s^2)}{(1+t^2)(1+s^2)^2}, \frac{-8st(-1+s^2)}{(1+t^2)(1+s^2)^2}, \frac{2s}{(1+s^2)} \right).$$

`ResPol` predicts a polytope Q with vertices $(0,0,8)$, $(0,0,12)$, $(0,2,2)$, $(0,4,0)$, $(0,4,4)$, $(2,2,0)$, $(4,0,4)$. The true polytope of the implicit equation $-4x_2^2+x_1^2+x_0^2+4x_2^4$ is smaller.

Sparse elimination theory works over the ring of Laurent polynomials $\mathbb{C}[t_1^{\pm 1}, \ldots, t_n^{\pm 1}]$ which means that points in the supports of the polynomials may have negative coordinates. As a consequence, evaluation points of polynomials cannot have zero coordinates. In the sequel we assume that P and Q are translated to the positive orthant and have non-empty intersection with all coordinate axes. This allows us to consider points with zero coordinates.

Let $S := \{s_1, \ldots, s_{|S|}\}$ be the set of lattice points in Q. S is used in our implicitization algorithm to construct a numerical matrix M: each $s_j = (s_{j0}, \ldots, s_{jn})$, $j = 1, \ldots, |S|$ is an exponent of a (potential) monomial $m_j := x^{s_j} = x_0^{s_{j0}} \cdots x_n^{s_{jn}}$ of the implicit polynomial, where x_i is defined in (1). We denote by $\boldsymbol{m} = (m_1, \ldots, m_{|S|})$ the vector of potential monomials and evaluate m_j at *generic* points $\tau_k \in \mathbb{C}^n$, $k = 1, \ldots, \mu$, $\mu \geq |S|$, avoiding values that make the denominators of the parametric expressions close to 0. Let

$$m_j|_{t=\tau_k} := \prod_{i=0}^{n} \left(\frac{f_i(\tau_k)}{g_i(\tau_k)} \right)^{s_{ji}}, \quad j = 1, \ldots, |S|$$

denote the evaluated j-th monomial m_j at τ_k. Thus, we construct an $\mu \times |S|$ matrix M with rows indexed by $\tau_1, \ldots, \tau_\mu$ and columns by $m_1, \ldots, m_{|S|}$:

$$M = \begin{bmatrix} m_1|_{t=\tau_1} & \cdots & m_{|S|}|_{t=\tau_1} \\ \vdots & \cdots & \vdots \\ m_1|_{t=\tau_\mu} & \cdots & m_{|S|}|_{t=\tau_\mu} \end{bmatrix}. \tag{4}$$

The vectors in the kernel of M contain the coefficients of the monomials with exponents in S in multiples of the implicit polynomial $p(x)$, where $x := (x_0, \ldots, x_n)$.

To cope with numerical issues, especially when computation is approximate, we let $\mu \geq |S|$; this overconstrained system increases numerical stability and reduces the probability of obtaining an empty or higher dimensional kernel due to a bad sampling, see below.

When constructing matrix M we assume that the parametric hypersurface is sampled sufficiently generically by evaluating the parametric expressions at random points $\tau_k \in \mathbb{C}^n$. It is possible to check a-posteriori the genericity of the sampling by testing the evaluated matrix. Using more than $|S|$ sample points we reduce the probability that another polynomial vanishes at those points. Let $G \subset \Omega$ be the sampling space, typically the set of lattice points in a hypercube in $\mathbb{R}^n$ of size $|S|^2$, and by abuse of notation, let τ_k, $k = 1 \ldots, \mu$, denote the parameter value and its image via the parameterization (1). Let $h(x)$ be a nonzero polynomial in the basis S, of total degree $d \leq |S|^{2n}$. By the Schwartz-Zippel lemma [11]:

$$Prob[h(\tau_k) = 0] \leq d/|G| = d/|S|^{2n},$$

hence for μ (independently chosen) lattice sample points

$$Prob[h(\tau_k) = 0, \text{ for all } k = 1, \ldots, \mu] \leq \left(d/|S|^2\right)^\mu.$$

It follows that we can obtain a good sample by choosing suitable values for μ and $|G|$. Hence:

LEMMA 2. [5] *Any polynomial in the basis of monomials S indexing M, with coefficient vector in the kernel of M, is a multiple of the implicit polynomial $p(x)$.*

As in [4], one of the main difficulties is to build M whose corank, or kernel dimension, equals 1, i.e. its rank is 1 less than its column dimension. For some inputs we obtain a matrix of corank > 1 when the predicted polytope Q is significantly larger than P. It can be explained by the nature of our method: we rely on a *generic* resultant to express the implicit equation, whose symbolic coefficients are then specialized to the actual coefficients of the parametric equations. If this specialization is not generic, then the implicit equation divides the specialized resultant. The following theorem establishes the relation between the dimension of the kernel of M and the accuracy of the predicted support. It remains valid even in the presence of base points. In fact,

it also accounts for them since then P is expected to be much smaller than Q.

THEOREM 3. [5] *Let $P = N(p(x))$ be the implicit polytope, and Q be the predicted polytope. Assuming M has been built using sufficiently generic evaluation points, the dimension of its kernel equals $r = \#\{a \in \mathbb{Z}^{n+1} : a + P \subseteq Q\} = \#\{a \in \mathbb{Z}^{n+1} : N(x^a \cdot p(x)) \subseteq Q\}$. In particular, corank$(M) \geq 1$.*

The formula for the corank of the matrix also implies that the coefficients of the polynomials $x^a p(x)$ such that $N(x^a p(x)) \subseteq Q$, form a basis of the kernel of M (see [5, Proof of Thm. 10]). This observation will be useful in Lem. 14 but also implies the following.

COROLLARY 4. [5] *Let M be the matrix from (4), built with sufficiently generic evaluation points, and suppose the specialization of the polynomials in (3) to the parametric equations is sufficiently generic. Let $\{\boldsymbol{c_1}, \ldots, \boldsymbol{c_\lambda}\}$ be a basis of the kernel of M and $g_1(x), \ldots, g_\lambda(x)$ be the polynomials obtained as the inner product $g_i = \boldsymbol{c_i} \cdot \boldsymbol{m}$. Then the greatest common divisor (GCD) of $g_1(x), \ldots, g_\lambda(x)$ equals the implicit equation up to a monomial factor x^e.*

REMARK 5. *The extraneous monomial factor x^e in the previous corollary is always a constant when the predicted polytope Q is of the form $Q = P + E$ and, as we assume throughout this paper, it is translated to the positive orthant and touches the coordinate axes. However, it is possible that Q strictly contains $P + E$ and the extraneous polytope E is a point $e \in \mathbb{R}^{n+1}$, or it is the Minkowski sum of point e and a polytope E' which touches the axis. Let $\sum_\beta c_\beta x^\beta$ be the GCD of the polynomials g_i in Cor. 4, and let $\gamma = (\gamma_0, \ldots, \gamma_n)$, where $\gamma_i = \min_\beta(\beta_i)$, $i = 0, \ldots, n$. We can efficiently remove the extraneous monomial x^e by dividing $\sum_\beta c_\beta x^\beta$ with x^γ, i.e. the GCD of monomials x^β.*

3. MINKOWSKI DECOMPOSITION

In this section, we develop a method for Minkowski decomposition in $\mathbb{R}^3$, which also works in $\mathbb{R}^2$. A predicted polytope may correspond to a polynomial containing extraneous factors whose Newton polytope may be Minkowski summands of the predicted polytope. We shall briefly examine the case where $P + E \subsetneq Q$ later in this section.

Given two polytopes A and B in $\mathbb{R}^d$, we define their Minkowski sum by

$$A + B = \{a + b \mid a \in A, \in B\} \subseteq \mathbb{R}^d;$$

A and B are called Minkowski summands.

PROBLEM 6 (MINKOWSKI DECOMPOSITION). *Given a polytope $Q \in \mathbb{R}^d$, find polytopes A and B in $\mathbb{R}^d$, such that $A + B = Q$ and neither A nor B are homothetic to Q.*

A polytope $Q \in \mathbb{R}^d$ is homothetic to the polytope $A \in \mathbb{R}^d$ if there exist $\lambda \in \mathbb{R}$ and $t \in \mathbb{R}^d$ such that $Q = t + \lambda A$.

In general polytope Q may be written as a sum of more than two summands and one may examine decomposition into indecomposable summands. By applying recursion, it suffices to consider decomposition into two summands.

Let us restrict to 3-dimensional polytopes: $d = 3$. Our goal is to construct an ILP expressing Minkowski decomposition so as to compute the 1-skeleton of a summand and its V-representation.

We represent polytopes by a combination of their face lattice and a list of primitive edges. Since the face lattice contains no information about coordinates, we will use the primitive edges for the computation of the V-representation of the polytope. For representing edges we follow a classical method, cf. [8, 9]. Let Q be a lattice polytope and $V = [v_1, v_2, \ldots, v_m]$ its vertices. For every edge $\mathcal{E}_i = (v_{i,1}, v_{i,2})$, where $v_{i,j} \in V$, denote by e_i' the vector $v_{i,2} - v_{i,1}$. Let ℓ_i, called the integer length of $\mathcal{E}_i$, be the gcd of the entries in e_i', and let e_i be the primitive vector obtained by dividing each entry of e_i' by ℓ_i. Any facet F of Q is a polygon in $\mathbb{R}^2$, determined by a set of primitive vectors and their integer lengths.

Determining the edges of a summand.

The following lemma is the starting point of the algorithm.

LEMMA 7. *Given a Minkowski summand A of a polytope Q, for every facet $F = \langle (e_1, \ell_1), (e_2, \ell_2), \ldots (e_k, \ell_k) \rangle$ of Q there exist a_i with $0 \leq a_i \leq \ell_i$ for $i = 1, 2, \ldots, k$, such that $F' = \langle (e_1, a_1), (e_2, a_2), \ldots (e_k, a_k) \rangle$ is a face of A. Moreover, every facet of A can be obtained from a facet of Q by an appropriate choice of a_i's.*

PROOF. If F is a facet of Q, there are 3 cases:

- $F' = F$ is a facet both of Q and A. Then $a_i = \ell_i$ for all i.
- Only 0-dimensional faces of A appear as Minkowski summands of F. Then $a_i = 0$ for all i.
- Face F' of A is a Minkowski summand of F. Then there is a choice of a_i such that $F' = \langle (e_1, a_1), (e_2, a_2), \ldots (e_k, a_k) \rangle$, due to the theory of Minkowski decomposition in $\mathbb{R}^2$.

By the construction of Minkowski sum, every facet of A, appears as a Minkowski summand of a facet of Q. □

The previous lemma considers a single facet at a time. But we fix the integers a_i globally, i.e., if an edge $\ell_i e_i$ appears in facets F_1 and F_2 of Q, then fixing a_i means that $a_i e_i$ appears in both F_1' and F_2', possibly with opposite sign. We call this sign the orientation of the edge in the face and illustrate it in Figure1, showing the development in $\mathbb{R}^2$ of 3 adjacent oriented facets of a unit cube intersecting on a vertex. Note that at least one edge (e.g., the one shown in red/grey in the electronic/printed version) will have inconsistent orientation in the two facets it belongs to, i.e., the signs of this edge in the two facets are opposite.

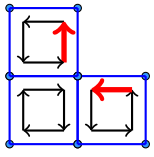

Figure 1: Edge orientation.

For a face $F \subset Q$, we call a face $F' \subset A$ the corresponding face if F' is obtained by F through an appropriate choice of a_i's. Since the a_i are global, we obtain:

LEMMA 8 (PRESERVING ADJACENCY). *Let F_1, F_2 be adjacent, i.e., they have a non-empty intersection, facets of Q and F_1' and F_2' be corresponding faces in A, a Minkowski summand of P. Then either F_1' and F_2' are adjacent faces in A or they are 0-dimensional and $F_1' = F_2'$.*

Due to Lem. 7, given Q, we know that any facet of a Minkowski summand is obtained by a facet of Q. Moreover, every facet $F \subset Q$ is a 2D polygon, thus $\sum_{i \text{ s.t. } e_i \in F} \sigma_{i,F} \ell_i e_i = 0$, where $\sigma_{i,F}$ is the sign of e_i and depends on the orientation of the edge $\ell_i e_i$ in the facet F. Similarly, every facet $F' = \langle (e_1, a_1), (e_2, a_2), \dots (e_k, a_k) \rangle$ of A corresponding to facet $F = \langle (e_1, \ell_1), (e_2, \ell_2), \dots (e_k, \ell_k) \rangle$ of Q, needs to satisfy the same condition, i.e.,

$$\sum_{i \text{ s.t. } e_i \in F} \sigma_{i,F} a_i e_i = 0 \tag{5}$$

Lem. 8 implies that the edges of a Minkowski summand A are obtained by choosing integers $0 \le a_i \le \ell_i$ satisfying relations (5) for every facet of Q. In other words, the set of edges $\mathcal{E}_A = [a_1 e_1, a_2 e_2, \dots, a_n e_n]$ corresponds to the set of edges of a polytope.

In the definition of Minkowski decomposition, we exclude summands homothetic to Q. In order to avoid such summands we require that the ratios a_i/ℓ_i are not all equal, i.e., $\sum_{i \neq j} (a_i \ell_j - a_j \ell_i) r_i \neq 0$ for some sufficiently random $r_i \in \mathbb{R}^*$.
More formally, we have:

THEOREM 9. *Let Q be a polytope given as a set of M facets, where each facet is a list of edges denoted by pairs (e_i, ℓ_i). Let N be the total number of edges. Then there exists a linear Diophantine system with $N+1$ variables and $M+2$ inequalities such that:*

- *The system is infeasible iff there is no non-homothetic Minkowski summand of Q, i.e., Q is indecomposable.*
- *Substituting ℓ_i by a_i in the facet representation of Q, where $(a_1, a_2, \dots, a_N)$ is a solution to the system, we obtain the facet representation of a (non-homothetic) Minkowski summand of Q.*

In particular, one such linear Diophantine system is:

$$x_i \in \mathbb{N}, \quad b \in \{0, 1\}, \tag{6}$$

$$\sum_{i e_i \in F} \sigma_{i,F} x_i e_{i,k} = 0 \text{ for every facet } F \text{ and } k = 1, 2, 3, \tag{7}$$

$$bM + \sum_{i \ge j} (a_i \ell_j - a_j \ell_i) r_i \ge \epsilon, \tag{8}$$

$$bM + \sum_{i \ge j} (a_i \ell_j - a_j \ell_i) r_i \le M - \epsilon, \tag{9}$$

where M bounds the absolute value of $\sum_{i \ge j} (a_i \ell_j - a_j \ell_i) r_i$.

PROOF. Conditions (7) ensure that for every facet $F \subset Q$, by substituting ℓ_i with a_i we obtain a 2D polygon. By construction, if an edge (e_i, ℓ_i) of Q is taken as (a_i, ℓ_i), then it has the same length in both faces it appears. By Lem. 8, if two facets are adjacent in Q, the respective faces after substitution are either adjacent or the same 0-dimensional face. Thus, after substituting ℓ_i by a_i in the facets of Q, we obtain facets of A. By Lem. 7 and the discussion following it, if A is a Minkowski summand of Q, then the face list we obtained contains the list of facets of A. Conditions (8)- (9) guarantee that the Minkowski summand A is not homothetic to Q. □

We recurse the decomposition procedure until we obtain lower dimensional or indecomposable summands. Let the size of a summand be the sum of integer length over its edges, i.e., the sum of all a_i's. A heuristic we use is to try obtain balanced summands, i.e., we favor summands with almost equal size. This reduces the depth of the recursion tree. Thus we add to the system of Thm. 9 the constraint $\sum_{i=1}^n a_i \le \frac{1}{2} \sum_{i=1}^n \ell_i$ and consider the optimization problem maximizing over the linear functional $\sum_{i=1}^n a_i$.

Reconstruction.

From the previous discussion, we can obtain a set of pairs (e_i, a_i) that are edges of a polytope. This is neither an H-representation nor a V-representation of the polytope. We choose to construct a V-representation.

LEMMA 10. *Solving the ILP above we obtain a list $\mathcal{F} = [[(e_i, a_i) \mid a_i \neq 0, (e_i, \ell_i) \in F] \mid F \text{ a facet of } Q]$. Then every $F' \in \mathcal{F}$ belongs to one of three types:*

1. *F' is 0-dimensional (has cardinality 0).*
2. *F' is 1-dimensional (has cardinality 2).*
3. *F' is 2-dimensional (has cardinality more than 2).*

PROOF. Due to the constraint corresponding to F' in the ILP, if F' is not 0-dimensional, then it contains at least 2 edges. If it contains exactly 2 edges, then they must have opposite directions, the same length and correspond to the same edge $\mathcal{E}_i$, i.e., they define an edge. There is no way to choose more than 2 edges of a 2D polygon such that they constitute a polytope of dimension < 2. □

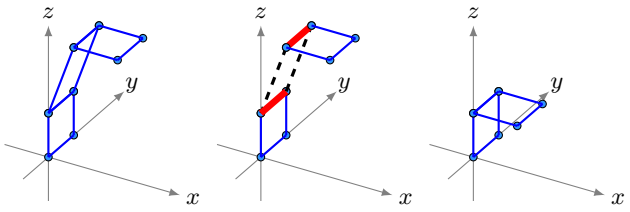

Figure 2: Identifying edges.

Due to Lem. 10, we use possible faces of cardinality 2, in order to "glue" the faces of A together. In Fig. 2 we illustrate a case of cardinality-2 face. Essentially we collapse all edges except the two bold ones, i.e., we collapse two sequences of edges (the ones shown dashed). In the figure, each sequence consists of exactly one edge. From Lem. 10, we have that the two bold edges define the same edge in the polytope we are constructing. Thus we translate one of the facets accordingly, so that the two edges coincide.

More generally, assume we are given a list of faces $L' = \{F_1, F_2, \dots, F_k\}$. Let $\mathcal{I}$ be the set of all pairs (e_i, e_j) such that there exists an F_k containing only edges corresponding to e_i and e_j. Let $\mathcal{D}$ be a dictionary mapping any e_i appearing in any F_k in L' to a pair of points. This dictionary fixes the endpoints of an edge in the coordinate system. In order to obtain a polytope, we require that edges corresponding to a pair in $\mathcal{I}$, are mapped to the same pair of points in $\mathcal{D}$. Now let L be the list containing all elements of L' with cardinality greater than 2. Let $F = [(e_1, a_1), (e_2, a_2), \dots, (e_n, a_n)]$ be an element of L. Starting from the origin, fix the endpoints of e_1, i.e., put in the dictionary the key-value pair $(e_1 : [(0,0,0), a_1 e_1])$. We call e_1 a fixed edge, since its endpoints have fixed coordinates. Continue with e_2 starting from the point $a_1 e_1$, and so on for all e_i's in F. Since this is the description of a 2-dimensional face, the endpoint will be the origin. Remove F from L.

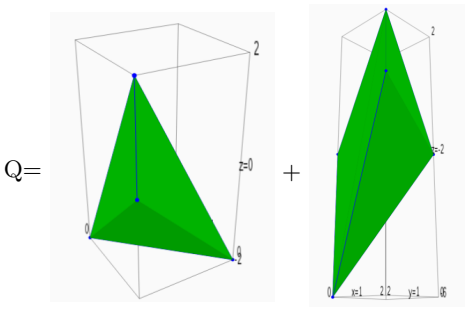

Figure 3: The decomposition of polytope Q in Exam. 2 into indecomposable summands.

Pick an element of L that contains an e_i already mapped via the dictionary $\mathcal{D}$. If this is not possible, pick an element of L containing an e_j, such that there exists a pair (e_i, e_j) (or (e_j, e_i)) in $\mathcal{I}$ with e_i already mapped via $\mathcal{D}$. Repeat the procedure as above, using as starting point the starting point of e_i given by the dictionary instead of the origin. Note that orientation needs to be taken care of, i.e., if e_k is backwards, then use as starting point the endpoint of e_k as fixed in the dictionary. The procedure described will produce all vertices of a polytope defined by the faces described in L. Actually, it will also produce all edges, i.e., the 1-skeleton of the polytope. Thus, it allows for a recursive application of the algorithm in order to obtain indecomposable factors.

EXAMPLE 2 (CONT'D FROM EXAM. 1). *Given the predicted polytope Q, we construct an ILP in 12 variables and 22 constraints. Our method obtains the decomposition (see Fig. 3):*

$$\begin{aligned} Q &= CH\left((0,0,6),(0,0,8),(0,2,0),(0,2,4),(2,0,4)\right) \\ &+ CH\left((0,2,0),(2,0,0),(0,0,2),(0,0,4)\right), \end{aligned}$$

where CH denotes convex hull. Both summands are indecomposable; the second one is the exact implicit polytope P. Using Q in the implicitization algorithm yields a 67×67 matrix, while P a 10×10 matrix.

REMARK 11. *Minkowski decomposition fails to extract the implicit polytope P from the predicted polytope Q, if $Q \subsetneq P + E$. Then Q might be indecomposable, see Exam. 3, or less often, Q is decomposable but none of the summands is exactly P. We can address these cases by either taking smaller homothetic copies of Q and trying to decompose them, or by removing a few vertices from Q and using the resulting polytope as input to our algorithm.*

EXAMPLE 3. *For the Bohemian Dome*

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

`ResPol` predicts a polytope with vertices $(0,0,0)$, $(0,0,4)$, $(0,4,0)$, $(4,0,0)$, $(4,0,4)$, while the true implicit polytope has vertices $(0,0,4)$, $(0,4,0)$, $(4,0,0)$, $(0,2,0)$. The predicted polytope is indecomposable.

Analysis.

Let Q be given by M facets, where each facet is a list of edges denoted by pairs (e_i, ℓ_i) and let N be the total number of edges. Our algorithm has two steps.

Algorithm 1: Minkowski Decomposition

Input : A polytope Q, given as a list of its facets $[F_1, F_2, \ldots, F_k]$ where F_i is a list of pairs (ℓ_i, e_i) (length, primitive edge)
Output: A V-rep of a Minkowski summand of Q.

$f \leftarrow$MAXIMIZE$\left(\sum_{i=1}^n x_i\right)$
Choose random r_i
$B \leftarrow \sum_{i \geq j} \left(\ell_j^2 - \ell_i\right) r_i$
$C \leftarrow 0 < \sum_{i=1}^n x_i < \frac{1}{2} \sum_{i=1}^n \ell_i$
for $i \leftarrow 1$ **to** $|F|$ **do**
  $C \leftarrow \sum_{i \text{ s.t. } e_i \in F} \sigma_{i,F} a_i e_i = 0$
$C \leftarrow bB + \sum_{i \geq j} \left(a_i \ell_j - a_j \ell_i\right) r_i \geq \epsilon$
$C \leftarrow bB + \sum_{i \geq j} \left(a_i \ell_j - a_j \ell_i\right) r_i \leq B - \epsilon$
$a =$ solution to the ILP given by (f, C)
for $f \in F$ **do**
  $L \leftarrow f|_{\ell_i = a_i}$
$\mathcal{I} =$ the set of (e_i, e_j) such that there exists $F_k \in L$ containing only edges corresponding to e_i and e_j
$\mathcal{D} =$ an empty dictionary.
Remove from L all elements of cardinality less than 3
$F =$ the first element of L ; startpoint$=(0,0,0)$
while *true* **do**
  for $i \leftarrow 1$ **to** $|F|$ **do**
    if $e_i \notin \mathcal{D}$ **then**
      if $e_i \in \mathcal{I}$ **then** $e = e_j$ such that (e_i, e_j) in $\mathcal{I}$
      else $e = e_i$
      endpoint$=$ startpoint$+ a_i e$
      $\mathcal{D}\,[e_i] =$ [startpoint , endpoint]
      startpoint$=$endpoint
    else startpoint$= \mathcal{D}\,[e_i][2]$
  if $L \neq \emptyset$ **then**
    Pick F containing an edge either fixed or in $\mathcal{I}$
  else Break
$A =$ the convex hull of all points where $\mathcal{D}$ is mapping to
return A

The first consists in defining and solving an ILP. In order to define the program, we need a number of operations in $O(M)$. We then solve a system in $N+1$ variables with $M+2$ inequalities. ILP is strongly NP-complete, thus we cannot expect a pseudo-polynomial algorithm. Nevertheless, there exist very efficient implementations.

The second step concerns the computation of a V-representation of a Minkowski summand, given a solution of the ILP. In Alg 1, we use a dictionary and the complexity becomes $\mathcal{O}(NM^2)$. A detailed analysis of further data-structures and optimizations to increase performance is out of scope.

4. GEOMETRIC OPERATIONS

In this section we formulate the membership and sidedness operations on the hypersurface $p(x) = 0$ as matrix operations. This is done by modifying slightly the construction of the interpolation matrix M in Section 2 to obtain matrix $M(x)$ which is numeric except for its last row.

Recall S is the predicted support and $\boldsymbol{m}$ the row vector of predicted monomials. Fix a set of *generic* distinct values τ_k, $k = 1, \ldots, |S|-1$ and recall that $m_j|_{t=\tau_k}$ denotes the j-th monomial m_j evaluated at τ_k. Let M' be the $(|S|-1) \times |S|$

numeric matrix obtained by evaluating $\boldsymbol{m}$ at the $|S|-1$ points τ_k, i.e., M' is obtained from M in (4) for $\mu = |S|-1$. Finally, let $M(x)$ be the $|S| \times |S|$ matrix obtained by appending row vector $\boldsymbol{m}$ to matrix M':

$$M(x) = \begin{bmatrix} M' \\ \boldsymbol{m} \end{bmatrix}. \tag{10}$$

Given a point $q \in \mathbb{R}^{n+1}$, let $M(q) = \begin{bmatrix} M' \\ \boldsymbol{m}|_{x=q} \end{bmatrix}$, where $\boldsymbol{m}|_{x=q}$ denotes vector $\boldsymbol{m}$ evaluated at q. We assume that $q \neq x(\tau_k)$, for all $k = 1, \ldots, |S|-1$, which implies that the rows of $M(q)$ are distinct. This can be checked efficiently. Obviously, when $p(q) = 0$, $M(q)$ is equivalent to matrix M in (4) in the sense that they both have the same kernel.

REMARK 12. *Let M be a matrix as in (4) and $\boldsymbol{c}$ be a vector in the kernel of M. Since the kernel is a vector space, then $\lambda\boldsymbol{c}$ is also in the kernel of M, for any $0 \neq \lambda \in \mathbb{R}$. This also follows from the fact that the implicit polynomial is defined up to a non-zero scalar multiple. Hence we can set an arbitrary non-zero coordinate of $\boldsymbol{c}$ equal to 1. As a consequence the matrices M', M and $M(q)$, for $p(q) = 0$, have the same kernel of corank r, where r is given in Thm. 3.*

Matrix $M(x)$ has an important property:

LEMMA 13. *Assuming M' is of full rank, the $\det M(x)$ equals the implicit polynomial $p(x)$ up to a constant.*

PROOF. Suppose that M' is of full rank equal to $|S|-1$. Then there exists a non-singular $(|S|-1)\times(|S|-1)$ submatrix of M'. Without loss of generality we assume that it is the submatrix $M'' = M'_{-|S|}$ obtained from M' by removing its last column. By Rmk 12, M' and M have the same kernel consisting of a single vector $\boldsymbol{c} = (c_1, \ldots, c_{|S|})$, where we can assume that $c_{|S|} = 1$. Let N denote $|S|$, then

$$\begin{bmatrix} m_1|_{t=\tau_1} & \cdots & m_N|_{t=\tau_1} \\ \vdots & \cdots & \vdots \\ m_1|_{t=\tau_{N-1}} & \cdots & m_N|_{t=\tau_{N-1}} \end{bmatrix} \begin{bmatrix} c_1 \\ \vdots \\ c_{N-1} \\ 1 \end{bmatrix} = \mathbf{0} \Leftrightarrow \tag{11}$$

$$\begin{bmatrix} m_1|_{t=\tau_1} & \cdots & m_{N-1}|_{t=\tau_1} \\ \vdots & \cdots & \vdots \\ m_1|_{t=\tau_{N-1}} & \cdots & m_{N-1}|_{t=\tau_{N-1}} \end{bmatrix} \begin{bmatrix} c_1 \\ \vdots \\ c_{N-1} \\ 1 \end{bmatrix} = - \begin{bmatrix} m_k|_{t=\tau_1} \\ \vdots \\ m_k|_{t=\tau_{N-1}} \end{bmatrix},$$

which, by applying Cramer's rule yields

$$c_k = \det M''_k / \det M'', \quad k = 1, \ldots, N-1, \tag{12}$$

where M''_k is the matrix obtained by replacing the kth column of M'' by the $|S|$th column of M', which plays the role of the constant vector in (11). Note that M''_k equals (up to reordering of the columns) M'_{-k}, where M'_{-k} is the matrix obtained by removing the kth column of M'. Hence, $\det M''_k$ equals (up to sign) $\det M'_{-k}$.

Now, the assumption that M' is of full rank in conjunction with Thm. 3 and Cor. 4 implies that

$$p(x) = \boldsymbol{m} \cdot \boldsymbol{c} = \sum_{i=1}^{|S|} m_i \cdot c_i = \sum_{i=1}^{|S|-1} m_i \cdot c_i + m_{|S|},$$

which combined with (12) gives

$$\begin{aligned} p(x) &= \sum_{i=1}^{|S|-1} m_i \cdot \frac{\det M''_k}{\det M''} + m_{|S|} \\ &= \pm \sum_{i=1}^{|S|-1} m_i \cdot \frac{\det M'_{-k}}{\det M'_{-|S|}} + m_{|S|} - = \pm \frac{\det M(x)}{\det M'_{-|S|}}. \quad \square \end{aligned}$$

Membership predicate.

Given parameterization (1) and query point $q \in \mathbb{R}^{n+1}$, we wish to decide whether $p(q) = 0$ or not, where $p(x)$ is the unknown implicit equation of the parametric hypersurface. We formulate this using the interpolation matrix in (10).

Working with matrices instead of polynomials, we cannot utilize Cor. 4 and Rmk 5 to process the kernel polynomials. To avoid false positives we restrict membership testing to points $q \in (\mathbb{R}^*)^{n+1}$, where $\mathbb{R}^* = \mathbb{R} \setminus \{0\}$.

LEMMA 14. *Let $M(x)$ be as in (10) and $q = (q_0, \ldots, q_n)$ be a query point in $(\mathbb{R}^*)^{n+1}$. Then q lies on the hypersurface defined by $p(x) = 0$ if and only if corank$(M(q))$ = corank(M').*

PROOF. For every point q, since M' is an $(|S|-1) \times |S|$ submatrix of the $|S| \times |S|$ matrix $M(q)$, we have that $\text{rank}(M(q)) \geq \text{rank}(M')$ which implies that $\text{corank}(M(q)) \leq \text{corank}(M')$. Moreover, it holds that

$$\text{kernel}(M(q)) \subseteq \text{kernel}(M'). \tag{13}$$

($\rightarrow$) Assume that q lies on the hypersurface defined by p, hence $p(q) = 0$. Then by Rmk 12 the matrices $M(q)$ and M' have the same corank.
($\leftarrow$) Suppose that $\text{corank}(M(q)) = \text{corank}(M')$. Then the last row $\boldsymbol{m}|_{x=q}$ of $M(q)$ is linearly dependent on the first $|S|-1$ rows, hence there exist $l_k \in \mathbb{R}, k = 1, \ldots, |S|$, not all zero, such that $\boldsymbol{m}|_{x=q} = \sum_{k=1}^{|S|} l_k \boldsymbol{m}|_{t=\tau_k}$. Let $\boldsymbol{c} \in \text{kernel}(M')$. Then $\boldsymbol{m}|_{x=q} \cdot \boldsymbol{c} = (\sum_{i=k}^{|S|} l_k \boldsymbol{m}|_{t=\tau_k}) \cdot \boldsymbol{c} = \sum_{i=k}^{|S|} l_k (\boldsymbol{m}|_{t=\tau_k} \cdot \boldsymbol{c}) = 0$, so $\boldsymbol{c} \in \text{kernel}(M(q)$, which, given relation (13), implies that M' and $M(q)$ have the same kernel.

Every vector $\boldsymbol{c}$ in the kernel of M', hence, also of $M(q)$, is a linear combination of the coefficient vectors of the polynomials $x^a p(x)$, where $a \in \mathbb{Z}^{n+1}$ such that $N(x^a p(x)) \subseteq Q$, (see also the discussion following Thm. 3). So we have $\boldsymbol{m}|_{x=q} \cdot \boldsymbol{c} = \sum_a \lambda_a q^a p(q) = 0$, where $\lambda_a \in \mathbb{R}$ are not all equal to zero, which, since $q \in (\mathbb{R}^*)^{n+1}$, implies that $p(q) = 0$. $\square$

Lemma 14 readily yields an algorithm that reduces the membership test $p(q) = 0$ for a query point $q \in (\mathbb{R}^*)^{n+1}$, to the comparison of the ranks of the matrices M' and $M(q)$. Note that the lemma is valid even if $\text{corank}(M')>1$.

Sidedness predicate.

The sidedness operation is defined as follows.

DEFINITION 15. *Given a hypersurface in $\mathbb{R}^{n+1}$ with defining equation $p(x) \in \mathbb{R}[x]$, and a point $q \in \mathbb{R}^{n+1}$ such that $p(q) \neq 0$, we define side(q) = sign$(p(q)) \in \{-1, 1\}$.*

See Fig. 4 for an example of applying Def. 15 to the Eight-curve defined by $x^4 - x^2 + y^2 = 0$.

We use matrix $M(x)$ defined in (10) to reduce sidedness as in Def. 15, to the computation of the sign of a numerical determinant. First we show that this determinant is non-zero for relevant inputs.

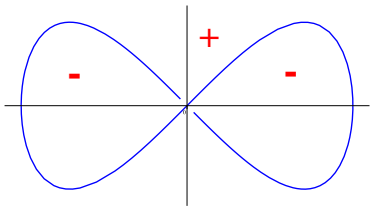

Figure 4: The sign of the Eight-curve polynomial.

LEMMA 16. *Suppose that the predicted polytope Q contains only one translate of the implicit polytope P. Let $M(x)$ be a matrix as in (10) and let $q \in (\mathbb{R}^*)^{n+1}$ such that $p(q) \neq 0$. Then* $\det M(q) \neq 0$.

PROOF. Since Q contains only one translate of the implicit polytope P, Thm. 3 implies that $\operatorname{corank}(M) = 1$ and by Rmk 12 this means that $\operatorname{corank}(M') = 1$, where matrix M is defined in (4). Then since $p(q) \neq 0$, from Lem. 14 we have that $\operatorname{corank}(M') \neq \operatorname{corank}(M(q)$, which implies that the matrix $M(q)$ is of full rank equal to $|S|$. Hence $\det M(q) \neq 0$. □

Next we show that, given matrix $M(x)$ and a point $q \in (\mathbb{R}^*)^{n+1}$ such that $p(x) \neq 0$, the sign of $\det(M(q))$ is consistent with $\operatorname{side}(q)$ in the following sense: for every pair of query points q_1, q_2, whenever $\operatorname{side}(q_1) = \operatorname{side}(q_2)$, we have that $\operatorname{sign}(\det M(q_1)) = \operatorname{sign}(\det M(q_2))$.

THEOREM 17. *Let $M(x)$ be as in (10) and q_1, q_2 be two query points in $(\mathbb{R}^*)^{n+1}$ not lying on the hypersurface defined by $p(x) = 0$. Assuming that Q contains only one translate of the implicit polytope P, then side(q_1) = side(q_2) if and only if sign$(\det M(q_1))$ = sign$(\det M(q_2))$, where sign$(\cdot)$ is an integer in $\{-1, 1\}$.*

PROOF. For points q_1, q_2 as in the statement of the theorem, we have from Lem. 16 that $\det M(q_1))$ and $\det M(q_2))$ are non-zero, hence their sign is an integer in $\{-1, 1\}$. We need to show that $\operatorname{sign}(p(q_1)) = \operatorname{sign}(p(q_2))$ if and only if $\operatorname{sign}(\det M(q_1)) = \operatorname{sign}(\det M(q_2))$. But this is an immediate consequence from Lem. 13, since $\det M(x)$ equals $p(x)$ up to a constant factor. □

We thus obtain an algorithm for deciding sidedness for any two query points. The rank test can be avoided if we directly compute $\operatorname{sign}(\det M(q_i))$ and proceed depending on whether this sign equals 0 (i.e., $\det M(q_i) = 0$) or not.

5. FUTURE WORK

We currently work on further operations on the matrix representation of a hypersurface, most notably ray shooting, either in exact or approximate form. This boils down to computing the smallest positive root of a univariate polynomial in matrix form. We plan to employ state of the art real solvers which rely on evaluating the polynomial in hand.

We plan to study the structure of our matrices, which generalizes Vandermonde: columns are indexed by monomials and rows by values where the monomials are evaluated. To gain an order of magnitude in complexity, though, we need fast multivariate interpolation and evaluation over arbitrary points, for which there are many open questions.

We are currently enhancing our method and code by also interpolating the normal vector to the curve or surface: We add rows to the interpolation matrix expressing the fact that the normal to the parametric hypersurface and the gradient of the implicit equation must be parallel. We thus add new constraints at the same evaluation points.

Acknowledgement. This research has been co-financed by the European Union (European Social Fund - ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: THALIS-UOA (MIS 375891).

6. REFERENCES

[1] L. Busé. Implicit matrix representations of rational Bézier curves and surfaces. *J. CAD*, 46:14–24, 2014.

[2] L. Busé and T. Luu Ba. The surface/surface intersection problem by means of matrix based representations. *J. CAGD*, 29(8):579–598, 2012.

[3] C. D'Andrea and M. Sombra. Rational parametrizations, intersection theory and Newton polytopes. In *Nonlinear Comp. Geom.*, volume 151 of *Volumes in Math. & Appl.*, pages 35–50. IMA, 2009.

[4] I. Emiris, T. Kalinka, C. Konaxis, and T. Luu Ba. Implicitization of curves and surfaces using predicted support. *Theor. Comp. Science*, 479:81–98, 2013.

[5] I. Emiris, T. Kalinka, C. Konaxis, and T. Luu Ba. Sparse implicitization by interpolation: Characterizing non-exactness and an application to computing discriminants. *J. CAD*, 45(2):252–261, 2013.

[6] I. Z. Emiris, V. Fisikopoulos, C. Konaxis, and L. Peñaranda. An oracle-based, output-sensitive algorithm for projections of resultant polytopes. *Int. J. Comp. Geom. App., Special Issue*, 23:397–423, 2013.

[7] I. Z. Emiris and E. P. Tsigaridas. Minkowski decomposition of convex lattice polygons. In M. Elkadi, B. Mourrain, and R. Piene, eds., *Algebraic geometry and geometric modeling*. Springer, 2005.

[8] S. Gao and A. Lauder. Decomposition of polytopes and polynomials. *Disc. Comp. Geom*, 26:89–104, 2001.

[9] D. Kesh and S. K. Mehta. Polynomial irreducibility testing through minkowski summand computation. In *Proc. Canadian Conf. Comp. Geom.*, 2008.

[10] A. Marco and J. Martinez. Implicitization of rational surfaces by means of polynomial interpolation. *J. CAGD*, 19:327–344, 2002.

[11] J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, 1980.

[12] Z. Smilansky. Decomposability of polytopes and polyhedra. *Geometriae Dedicata*, 24(1):29–49, 1987.

[13] B. Sturmfels, J. Tevelev, and J. Yu. The Newton polytope of the implicit equation. *Moscow Math. J.*, 7(2), 2007.

[14] B. Sturmfels and J. Yu. Tropical implicitization and mixed fiber polytopes. In *Software for Algebraic Geometry*, volume 148 of *IMA Volumes in Math. & its Applic.*, pages 111–131. Springer, New York, 2008.