Computer-Aided Design 45 (2013) 252–261

Contents lists available at SciVerse ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Sparse implicitization by interpolation: Characterizing non-exactness and an application to computing discriminants

Ioannis Z. Emiris [a], Tatjana Kalinka [a,*], Christos Konaxis [b], Thang Luu Ba [a,c]

[a] Department of Informatics and Telecoms, University of Athens, Greece

[b] Archimedes Center for Modeling, Analysis and Computation (ACMAC), University of Crete, Heraklio, Greece

[c] Department of Mathematics, Hanoi National University of Education, Viet Nam

ARTICLE INFO

Keywords:
Geometric representation
Implicitization
Linear algebra
Sparse polynomial
Discriminant

ABSTRACT

We revisit implicitization by interpolation in order to examine its properties in the context of sparse elimination theory. Based on the computation of a superset of the implicit support, implicitization is reduced to computing the nullspace of a numeric matrix. The approach is applicable to polynomial and rational parameterizations of curves and (hyper)surfaces of any dimension, including the case of parameterizations with base points. Our support prediction is based on sparse (or toric) resultant theory, in order to exploit the sparsity of the input and the output. Our method may yield a multiple of the implicit equation: we characterize and quantify this situation by relating the nullspace dimension to the predicted support and its geometry. In this case, we obtain more than one multiple of the implicit equation; the latter can be obtained via multivariate polynomial GCD (or factoring). All of the above techniques extend to the case of approximate computation, thus yielding a method of sparse approximate implicitization, which is important in tackling larger problems. We discuss our publicly available Maple implementation through several examples, including the benchmark of a bicubic surface. For a novel application, we focus on computing the discriminant of a multivariate polynomial, which characterizes the existence of multiple roots and generalizes the resultant of a polynomial system. This yields an efficient, output-sensitive algorithm for computing the discriminant polynomial.

© 2012 Published by Elsevier Ltd

1. Introduction

Implicitization is the process of changing the representation of a geometric object from parametric to algebraic, or implicit. It is a fundamental operation with several applications in computer-aided design (CAD) and geometric modeling. There have been numerous approaches for implicitization, including resultants, Groebner bases, and moving lines and surfaces. In this paper, we restrict attention to hypersurfaces: Our approach is based on interpolating the unknown coefficients of the implicit polynomial given a superset of its monomials. The latter is computed by means of sparse (or toric) resultant theory, so as to exploit the input and output sparseness. Here is the main notion that formalizes sparseness (see also Fig. 1).

Definition 1. Given a polynomial $f = \sum_a c_a t^a \in \mathbb{R}[t_1, \ldots, t_n]$, $t^a = t_1^{a_1} \cdots t_n^{a_n}$, $a \in \mathbb{N}^n$, $c_a \in \mathbb{R}$, its *support* is the set $\{a \in \mathbb{N}^n : c_a \neq 0\}$; its *Newton polytope* $N(f)$ is the convex hull of its support. All concepts extend to the case of Laurent polynomials, i.e. with integer exponent vectors $a \in \mathbb{Z}^n$.

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. Under certain genericity assumptions, the implicit polytope coincides with a projection of the resultant polytope, see Section 2. In general, the implicit polytope is contained in the projected resultant polytope, in other words, a superset of the implicit support is given by the lattice points contained in the projected resultant polytope. A superset of the implicit support can also be obtained by other methods, see Section 1.1; the rest of our approach does not depend on the method used to compute this support.

The predicted support is used to build a numerical matrix whose kernel is, ideally, one 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

* Corresponding author.
E-mail addresses: emiris@di.uoa.gr (I.Z. Emiris), kalinkat@di.uoa.gr (T. Kalinka), ckonaxis@acmac.uoc.gr (C. Konaxis), thanglb@di.uoa.gr (T. Luu Ba).

0010-4485/$ – see front matter © 2012 Published by Elsevier Ltd
doi:10.1016/j.cad.2012.10.008

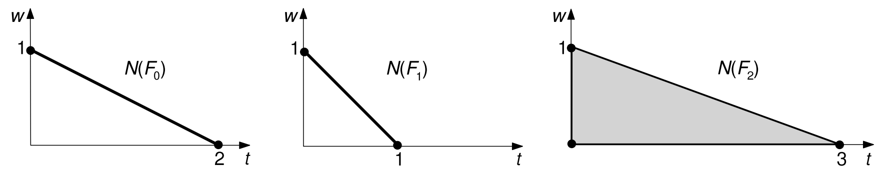

Fig. 1. Newton polytopes of F_0, F_1, F_2 in Example 6.

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 method of sparse implicitization was sketched in [1], where we presented an algorithm and some preliminary results on its implementation. Its main drawback is that the kernel of the matrix may be of high dimension. In this paper, we address this situation by presenting techniques that alleviate this phenomenon. More formally, we relate it to the geometry of the predicted support, which is a superset of the true implicit support. Another reason for obtaining a high-dimensional kernel is that the numeric evaluation of the support monomials may not be sufficiently generic. We study a method to obtain the true implicit polynomial by taking 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.

Furthermore, we present our publicly available Maple implementation by offering several examples. We also explain how it depends on other software, most notably the software computing the resultant polytope and its orthogonal projection required for predicting the implicit polytope.

Our main motivation is in changing the representation of geometric (hyper)surfaces given parametrically by polynomial, rational, or trigonometric parameterizations. Our method automatically handles the case of base points, so the user does not need to examine whether the given parameterization induces base points or not.

Here, we extend our method to a more general geometric problem, namely to computing the discriminant of a multivariate polynomial, which is an important question with several geometric applications. The vanishing of the discriminant characterizes the existence of multiple roots of the given polynomial. This is a hard computation, since explicit formulas only exist for low-degree univariate polynomials. In general, one can reduce discriminant computation to computing the resultant of a rather large system, comprised of the polynomial and its partial derivatives, but this is inefficient. Instead, we reduce discriminant computation to sparse implicitization, thus obtaining an output-sensitive algorithm, whose complexity depends on the size of the discriminant's Newton polytope. Moreover, this technique can be used to compute discriminants of well-constrained systems as well as resultants because the latter can be viewed as a special case of discriminants.

The paper is organized as follows: Section 1.1 overviews previous work, and Section 2 describes our approach to predicting the implicit support while exploiting sparseness. Section 3 presents our implicitization algorithm based on computing a matrix kernel, either exactly or approximately, and focuses on the case of high dimensional kernels. Our Maple implementation is described in Section 4, whereas Section 5 applies our method to computing discriminants. We conclude with future work. The Appendix contains omitted examples and omitted results from examples in Section 5, and further experimental results.

1.1. 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 (columns) and $|S|$ different values (rows) at which all monomials get evaluated. Then the vector p of coefficients of the implicit equation is in the kernel of M. This idea was used in [1–4]; it is also the starting point of this paper.

Our method of sparse implicitization was sketched in [1], where the overall algorithm was presented together with some results on its 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 on random integers, random complex numbers of modulus 1, and complex roots of unity. That paper also proposed ways to obtain a smaller implicit polytope by downscaling the original polytope when the corresponding kernel dimension was higher than one.

A similar approach was based on integrating matrix $M = SS^\top$, over each parameter $t_1, \ldots, t_n$ [5]. Then p is in the kernel of M. In fact, the authors propose to consider successively larger supports in order to capture sparseness. This method covers polynomial, rational, and trigonometric parameterizations, but the matrix entries take big values (e.g. up to 10^{28}), so it is difficult to control its numeric corank, i.e. the dimension of its nullspace. Thus, the accuracy of the approximate implicit polynomial is unsatisfactory. When it is computed over floating-point numbers, the implicit polynomial does not necessarily have integer coefficients. They discuss post-processing to yield integer relations among the coefficients, but only in small examples.

Approximate implicitization over floating-point numbers was introduced in a series of papers. Today, there are direct [6,7] and iterative techniques [8]. An idea used in approximate implicitization is to use successively larger supports, starting with a quite small set and extending it so as to reach the exact implicit support. Existing approaches have used upper bounds on the total implicit degree, thus ignoring any sparseness structure. Our methods provide a formal manner to examine different supports, in addition to exploiting sparseness, based on the implicit polytope. When the kernel dimension is higher than one, one may downscale the polytope so as to obtain a smaller implicit support.

Sparse interpolation is the problem of interpolating a multivariate polynomial when information of its support is given [9, Ch.14]. This may simply be a bound $\sigma = |S|$ on support cardinality; then complexity is $O(m^3 \delta n \log n + \sigma^3)$, where δ bounds the output degree per variable, m is the actual support cardinality, and n the number of variables. A probabilistic approach in $O(m^2 \delta n)$ requires as input only δ.

2. Implicitization by support prediction

This section describes how the implicitization problem can be reduced to computing the sparse resultant of a polynomial system, how we can compute the implicit polytope as a projection of the

Download English Version:

https://daneshyari.com/en/article/440150

Download Persian Version:

https://daneshyari.com/article/440150

Daneshyari.com