

Implicitization of Polynomial Surfaces

Ilias S. Kotsireas, Edmond Lau, and Richard Voino
{ikotsire@wlu.ca, edmond@maximalgorithms.com, rvoino@hal.wlu.ca}

Wilfrid Laurier University
Computer Algebra Research Group
Department of Physics and Computer Science
Waterloo, Ontario, Canada
<http://www.cargo.wlu.ca>

Abstract. In this paper we present an implementation of an implicitization algorithm for surfaces given by polynomial parametric equations. Several examples illustrate in detail the implementation and some appealing perspectives for further work are briefly touched upon.

1 Introduction

In [2] the authors introduced a new algorithm for implicitization of parametric curves, surfaces and hypersurfaces. The algorithm uses essentially Linear Algebra, works in both symbolic and numeric contexts and is applicable to a wide variety of types of parametric equations as well as families of parametric equations indexed by a parameter different than the parameterization parameter. This algorithm has been implemented in Maple in the `algcsv` package.

In [3, 4] the authors used various tools from algebraic geometry, in particular sparse elimination theory, in order to predict the support of the implicit equation of rational parametric hypersurfaces. These ideas reduce dramatically the size of the implicitization matrices and can also be applied to other implicitization methods based on resultants. The resulting IPSOS algorithm gives optimal results in all of the examples tested. IPSOS is implemented in Maple and requires interfacing several freely available C/C++ programs.

In this paper we study more closely the case of surfaces given by polynomial parametric equations. An efficient implementation of the algorithm using exclusively C and GMP¹ arithmetic allows us to treat relatively big examples. We also establish some structural properties of the implicitization matrices that could potentially lead to more efficient strategies to compute their nullspaces as well as other optimizations. We show that the implicitization matrices exhibit a Hankel-like structure when we consider blocks with respect to the degrees of the monomials. A similar Hankel-like manifests itself in the case of curves as described in [7].

2 Surfaces Given by Polynomial Parametric Equations

In this section we give an overview of the implicitization algorithm in [2] emphasizing the case of algebraic curves given by polynomial parametric equations.

Suppose that a surface is given by polynomial parametric equations of the form

$$x = P(s, t), \quad y = Q(s, t), \quad z = R(s, t)$$

where P, Q, R are bivariate polynomials with integer (or rational) coefficients and of total degrees p, q, r respectively. The variables s, t are called the **parameters** of the parameterization of the surface.

After denoting the (total) degree m of the sought implicit equation, we need to generate all the monomials in the three variables x, y, z up to total degree $2m$. These are:

$$\ell_{2m} = \left[\underbrace{1}_{\text{deg } 0}, \underbrace{x, y, z}_{\text{deg } 1}, \underbrace{x^2, xy, xz, y^2, yz, z^2}_{\text{deg } 2}, \dots, \right] \quad (1)$$

¹ See <http://www.swox.com/gmp/> for more details

$$\underbrace{x^{2m}, x^{2m-1}y, x^{2m-1}z, x^{2m-2}y^2, x^{2m-2}yz, x^{2m-2}z^2, \dots, yz^{2m-1}, z^{2m}}_{\text{deg } 2m}$$

A simple counting argument shows that the number of these monomials is equal to

$$|\ell_{2m}| = \binom{2m+3}{3}$$

For each of the $\binom{2m+3}{3}$ monomials of the form $x^i y^j z^k$ of the list ℓ_{2m} we need to compute integrals of the form

$$\int_{\alpha}^{\beta} \int_{\gamma}^{\delta} x^i y^j z^k ds dt = \int_{\alpha}^{\beta} \int_{\gamma}^{\delta} P(s, t)^i Q(s, t)^j R(s, t)^k ds dt \tag{2}$$

We can choose for example $\alpha = \gamma = 0, \beta = \delta = 1$, since we are dealing with polynomial functions only. The computation of integral (2) requires two steps:

- **Expand** the polynomial to be integrated, that is: $P(s, t)^i Q(s, t)^j R(s, t)^k$. The degree of this expanded bivariate polynomial in s, t will be equal to $p \times i + q \times j + r \times k$.
- **Integrate** the expanded polynomial by using the simple rule to integrate monomials.

We construct a matrix M , starting with the list of monomials in x, y, z up to total degree m ,

$$\ell_m = [1, x, y, z, x^2, xy, xz, y^2, yz, z^2, \dots, x^m, x^{m-1}y, \dots, z^m] \tag{3}$$

Then we form the product $M = \ell_m^t \ell_m$. In detail we have:

$$M = \begin{pmatrix} 1 & x & y & z & \dots & x^m & \dots & z^m \\ x & x^2 & xy & xz & \dots & x^{m+1} & \dots & xz^m \\ y & xy & y^2 & yz & \dots & x^m y & \dots & yz^m \\ z & xz & yz & z^2 & \dots & x^m z & \dots & z^{m+1} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ x^m & x^{m+1} & x^m y & x^m z & \dots & x^{2m} & \dots & x^m z^m \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ z^m & xz^m & yz^m & z^{m+1} & \dots & x^m z^m & \dots & z^{2m} \end{pmatrix} \tag{4}$$

We call matrices of the form (4), **Implicitization Matrices**. Implicitization Matrices are symmetric (by construction) of dimension $d = \binom{m+3}{3}$. It contains only $\binom{2m+3}{3}$ different elements, which are exactly the elements of the list (1). Moreover, as m increases, $\binom{2m+3}{3}$ becomes much less than d^2 .

A matrix G is constructed by placing the results of the integrations of the elements of the list ℓ_{2m} into the matrix M . This raises some interesting combinatorial and programming problems. If an implicit equation of degree m exists, its coefficients will be given by a nullvector of the matrix G . For a more detailed presentation of the implicitization algorithm as well as many fully worked out examples, the reader can consult [2].

3 Implementation

The algorithm outlined above was implemented in C in a program called **IPSurfaces**. For most examples we tried, the Implicitization Matrices contain long rational entries with more than 100 digits in both numerators and denominators, we decided to use the GMP library to handle the base data type. The Nullspace computations have been performed with a C program by A. Storjohann.

The program **IPSurfaces** provides some extra functionality that makes it easy to interface with the Computer Algebra System Maple. In particular, **IPSurfaces** generates Implicitization Matrices in Maple format. This functionality is important for testing purposes. The main difference of **IPSurfaces** with the Maple implementation of [2] is the future perspective of the interfacing **IPSurfaces** with **IPSOS** and other C/C++ programs in order to produce a software that is suitable for Computer Aided Geometric Design(CAGD) applications.

One important point of efficiency improvement in `IPSurfaces` is the generation of the Implicitization Matrices not by performing the matrix multiplication on the left hand side of (4), but by using the vector ℓ_{2m} and the Hankel-like structural properties of the Implicitization Matrices. Thus `IPSurfaces` integrates each monomial only once and places it in the right position. Hankel-like structural properties of the Implicitization Matrices are studied in section 6 of this paper.

4 Examples

In this section we present four examples of implicitization of algebraic surfaces given by polynomial parametric equations using `IPSurfaces`. The results of these examples have been verified in different Computer Algebra Systems, using other available methods for implicitization. For surveys of available implicitization methods of algebraic curves and surfaces see [6] and [8]. All computations of the following examples were done on an Intel Pentium 4, 1.8GHz machine.

Example 1. Consider the polynomial parametric equations from [1]:

$$x = st, \quad y = st^2, \quad z = s^2$$

Degree arguments as detailed for instance in [3, 4] can be used to show that the total degree of the sought implicit equation is 4. We choose $m = 4$.

We generate the 165 monomials in the three variables x, y, z up to total degree $2m$ and after performing the integrations and the substitutions we construct a 35×35 symmetric Implicitization Matrix, shown in the appendix. The generation of the matrix takes 0.747 second. By computing the Nullspace of this matrix, which takes 0.098 seconds, we find the (irreducible) implicit equation of total degree 4:

$$y^2z - x^4 = 0$$

Example 2. Consider the polynomial parametric equations:

$$x = s - t^3, \quad y = s + st, \quad z = t + s^2$$

We choose $m = 6$ and generate 455 monomials in three variables x, y, z up to total degree $2m$. After the integrations and substitutions we construct a 84×84 symmetric Implicitization Matrix. The generation of the matrix takes 7.424 seconds. By computing the Nullspace of this matrix, which takes 26.188 seconds to compute, we find the (irreducible) implicit equation of total degree 6:

$$\begin{aligned} &2x^2 - 7xy + xz + 5y^2 - yz - x^3 + 6xy^2 + xyz - 5y^3 - 4y^2z + yz^2 \\ &+ 3x^2y^2 - 9xy^2z - xy^2z^2 + 2xz^3 + 11y^3z + 3y^2z^2 - 6yz^3 + z^4 \\ &- x^2z^3 - 3xy^4 + 6xy^2z^2 - 3y^4z - 2y^3z^2 + y^6 = 0 \end{aligned}$$

Example 3. Consider the polynomial parametric equations from [5]:

$$x = s^2 + t^2, \quad y = s^3 + t^2, \quad z = s^2 + t^3$$

We choose $m = 9$ and generate 1330 monomials in the three variables x, y, z up to total degree $2m$. After the integrations and substitutions we construct a 220×220 symmetric Implicitization Matrix. The generation of the matrix takes 2 minutes and 6.743 seconds. By computing the Nullspace of this matrix, which takes 54 minutes and 8.038 seconds, we find the (irreducible) implicit equation of total degree $m = 9$

$$\begin{aligned} &2x^4 - 8x^3y - 8x^3z + 12x^2y^2 + 24x^2yz + 12x^2z^2 - 8xy^3 \\ &- 24xy^2z - 24xyz^2 - 8xz^3 + 2y^4 + 8y^3z + 12y^2z^2 + 8yz^3 + 2z^4 \\ &- 5x^5 + 22x^4y + 22x^4z - 38x^3y^2 - 76x^3yz - 38x^3z^2 + 28x^2y^3 \end{aligned}$$

$$\begin{aligned}
& +96x^2y^2z + 96x^2yz^2 + 28x^2z^3 - 5xy^4 - 44xy^3z - 78xy^2z^2 \\
& -44xy^2z^3 - 5xz^4 - 2y^5 + 2y^4z + 16y^3z^2 + 16y^2z^3 + 2yz^4 - 2z^5 \\
& -6x^5y - 6x^5z + 27x^4y^2 + 24x^4yz + 27x^4z^2 - 38x^3y^3 - 60x^3y^2z \\
& -60x^3yz^2 - 38x^3z^3 + 18x^2y^4 + 60x^2y^3z + 54x^2y^2z^2 \\
& +60x^2yz^3 + 18x^2z^4 - 18xy^4z - 18xy^3z^2 - 18xy^2z^3 - 18xyz^4 \\
& +y^6 + 3y^4z^2 + 3y^2z^4 + z^6 + 5x^7 - 14x^6y - 14x^6z + 9x^5y^2 \\
& +48x^5yz + 9x^5z^2 - 36x^4y^2z - 36x^4yz^2 - 3x^3y^4 + 21x^3y^2z^2 \\
& -3x^3z^4 + 3x^6y^2 + 3x^6z^2 - x^9 = 0
\end{aligned}$$

Example 4. Consider the polynomial parametric equations:

$$x = t^2 + s, \quad y = s^2 + t, \quad z = ts^7$$

We choose $m = 16$ and generate 6545 monomials in the three variables x, y, z up to total degree $2m$. After the integrations and substitutions we construct a 969×969 symmetric Implicitization Matrix. The generation of the matrix takes 4 hours 18 minutes and 19 seconds.

5 Experiments with SHARCNET and Comparisons

The implicitization algorithm in [2] experiences a phase of computation of definite integrals of monomials. Because these integrals are independent, this phase is quite naturally parallelizable. We run some preliminary tests with IPSurfaces in SHARCNET². SHARCNET is a project featuring a network of high-performance Beowulf computing clusters across several universities and other institutions in Ontario. SHARCNET is structured as a computational grid in order to provide supercomputing capabilities.

The following random example of a surface given by polynomial parametric equations was tested on the SHARCNET cluster at Wilfrid Laurier University.

$$x = -35 + 97s + 50s^2t, \quad y = 49t^2 + 63s^3t^2, \quad z = 45s^2 - 8s^4.$$

The implicit equation is of degree 33. The size of the generated file containing the implicitization matrix is well over 12M. The time it took to finish the computation in IPSurfaces was 49271 minutes. This and other examples that we tested on the SHARCNET cluster show that a parallel version of IPSurfaces will indeed be worthwhile to develop and will be able to treat much larger examples.

A theoretical comparison of the implicitization method that we implemented in IPSurfaces with other available methods of implicitization as described for example in [6], would take us too far away from the purpose of this paper. An experimental comparison is equally problematic, because at this point IPSurfaces is not optimized, i.e. it doesn't incorporate the results of recent theoretical progress in our implicitization method. In addition, comparisons between implementations written by different programmers in different languages are of questionable value regarding the efficiency of the underlying algorithms. However, we would like to offer a brief comment on the efficiency of IPSurfaces compared with Gröbner-based implicitization for instance. Examples 1 and 2 are easily done using Maple's implementation of Gröbner bases. However, example 3 runs out of memory in Maple, as pointed out by the anonymous referee. Moreover, example 4 leads to a Gröbner bases computation which does not terminate neither in Maple 8 nor in Magma V2.5-1. Here is the Magma code that we used:

```

Q:=RationalField();
P<s,t,x,y,z>:=PolynomialRing(Q,5);
p1:=x-s-t^2; p2:=y-s^2-t; p3:=z-t*s^7;
L:=[p1,p2,p3];
GroebnerBasis(L);

```

² Shared Hierarchical Research Canadian Network, <http://www.sharcnet.ca/>

6 Hankel-Like Structural Properties of Implicitization Matrices

In this section we establish some interesting properties pertaining to the structure of the Implicitization Matrices. In particular we show that if one uses the degree ordering to write the vector of monomials ℓ_m as defined in (3), then the associated Implicitization Matrix is revealed to have a type of Hankel-like³ structure. It is interesting to note that the Hankel structure is of a different type if we use the lexicographical ordering to write the vector of monomials ℓ_m . In general, the Hankel structure for the degree ordering will be maintained if we group together the monomials of same degree in the vector ℓ_m . In the sections below, we illustrate the Hankel structure by examining in detail the case of the degree ordering. Similar results hold for the case of the lexicographical ordering.

6.1 Hankel-Like Structure for the Degree Ordering

We illustrate the Hankel-like structural properties in degree 2. The corresponding general result is easy to state and prove. We start by defining the vector $u = [1, x, y, z, x^2, xy, xz, y^2, yz, z^2]$ and computing $p = u^t \cdot u$:

$$p = \begin{bmatrix} 1 & x & y & z & x^2 & xy & xz & y^2 & yz & z^2 \\ x & x^2 & xy & xz & x^3 & x^2y & x^2z & xy^2 & xyz & xz^2 \\ y & xy & y^2 & yz & x^2y & xy^2 & xyz & y^3 & y^2z & yz^2 \\ z & xz & yz & z^2 & x^2z & xyz & xz^2 & y^2z & yz^2 & z^3 \\ x^2 & x^3 & x^2y & x^2z & x^4 & x^3y & x^3z & x^2y^2 & x^2yz & x^2z^2 \\ xy & x^2y & xy^2 & xyz & x^3y & x^2y^2 & x^2yz & xy^3 & xy^2z & xyz^2 \\ xz & x^2z & xyz & xz^2 & x^3z & x^2yz & x^2z^2 & xy^2z & xyz^2 & xz^3 \\ y^2 & xy^2 & y^3 & y^2z & x^2y^2 & xy^3 & xy^2z & y^4 & y^3z & y^2z^2 \\ yz & xyz & y^2z & yz^2 & x^2yz & xy^2z & xyz^2 & y^3z & y^2z^2 & yz^3 \\ z^2 & xz^2 & yz^2 & z^3 & x^2z^2 & xyz^2 & xz^3 & y^2z^2 & yz^3 & z^4 \end{bmatrix}$$

If we replace the elements in p with the degree of each monomial terms, then we form a new matrix:

$$p = \begin{bmatrix} 0 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 1 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ 1 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ 1 & 2 & 2 & 2 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ 2 & 3 & 3 & 3 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ 2 & 3 & 3 & 3 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ 2 & 3 & 3 & 3 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ 2 & 3 & 3 & 3 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ 2 & 3 & 3 & 3 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\ 2 & 3 & 3 & 3 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \end{bmatrix}$$

Furthermore if we group each term into submatrices and denote by i a block of monomials of total degree i , then the above matrix can be represented as follows:

³ The term Hankel-like here is used to describe a Hankel structure with respect to degrees of blocks of monomials.

$$p = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 2 & 3 \\ 2 & 3 & 4 \end{pmatrix}$$

This representation of p shows clearly its Hankel structure with respect to the degrees of blocks of monomials. Moreover, if we examine the structure of each degree block individually, we see that p can be rewritten as follows:

$$p = \begin{pmatrix} H^0 & C_1 & C_2 & C_3 \\ C_1^t & H^2 & C_3 & C_4 \\ C_2^t & C_3^t & H^4 & C_5 \\ C_3^t & C_4^t & C_5^t & H^6 \end{pmatrix}$$

where the t superscript denotes matrix transposition, the C_i are rectangular **banded** blocks formed by monomials of total degree i and the main diagonal contains square blocks H^i which are **Hankel** matrices of monomials of total degree i . Thus the Implicitization Matrices, aside from being symmetric and usually singular, demonstrate a much richer structure.

Currently, it is not clear to us how to take advantage of the Hankel-like structure exhibited by the Implicitization Matrices to improve the algorithm. However, since there is a vast literature on algorithms for structured matrices and in particular for Hankel-like matrices, we believe that this issue deserves further investigation.

Conclusions and Future Work

We presented an efficient implementation of the implicitization algorithm in [2] for surfaces given by polynomial parametric equations. We also showed that the Implicitization Matrices used in this algorithm exhibit different types of Hankel-like structure according to the orderings employed to write the monomials.

Future research directions that will result in significant speed-ups in the algorithm are the application of modulo p techniques as well as interfacing `IPSurfaces` with the implementation of the `IPSOs` algorithm described in [3, 4]. Another direction is to capitalize on the Hankel-like structure of the Implicitization Matrices. It might provide a useful approach in speeding up the algorithm, via a more efficient nullspace computation.

In addition, certain phases of the eigenvalue method for implicitization are naturally parallelizable. By applying `IPSurfaces` to clusters, we will be able to solve difficult benchmark problems.

It is clear that numerical techniques can be applied for performing the integrations and computing the nullspace. This is related to the approximate implicitization problem whose study is outside the scope of this paper.

References

1. Bruno Buchberger. Applications of Gröbner Bases in Non-Linear Computational Geometry. In J. R. Rice, editor, *Mathematical Aspects of Scientific Software*, volume 14 of *IMA Volumes in Mathematics and its applications*, pages 59–87. Springer-Verlag, 1988.
2. Robert M. Corless, Mark W. Giesbrecht, Ilias S. Kotsireas, and Stephen M. Watt. Numerical implicitization of parametric hypersurfaces with linear algebra. In *AISC'2000 Proceedings, Madrid*, volume 1930 of *LNAI*, pages 174–183. Springer, 2000.
3. Ioannis Z. Emiris and Ilias S. Kotsireas. On the support of the implicit equation of rational parametric hypersurfaces. Technical report, 2002. ORCCA TR-02-01, available on-line from <http://www.orcca.on.ca>.
4. Ioannis Z. Emiris and Ilias S. Kotsireas. Implicit polynomial support optimized for sparseness. In V. Kumar et al, editor, *Proceedings of ICCSA'2003*, LNCS 2669, pages 397–406, Montreal, Canada, 2003.
5. Laureano González-Vega. Implicitization of parametric curves and surfaces by using multidimensional newton formulae. *Journal of Symbolic Computation*, 23(2/3):137–151, 1997.
6. Ilias S. Kotsireas. Panorama of methods for exact implicitization of algebraic curves and surfaces, 2003. preprint.

