Symbolic-Numeric Techniques for Cubic Surfaces

Dissertation

zur Erlangung des Akademischen Grades

Doktor der Naturwissenschaften

Eingereicht von

D.M. Ibolya Szilágyi

Juli 2005

Erster Begutachter: Prof. Dr. Josef Schicho
Zweiter Begutachter: Prof. Dr. Bert Jüttler

Angefertigt am: Forschungsinstitut für symbolisches Rechnen
Technisch Naturwissenschaftliche Fakultät
Johannes Kepler Universität Linz

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Dissertation selbständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht bennutzt bzw. die wörtlich oder sinngemäs entnommenen Stellen als solche kenntlich gemacht habe.

Linz, Juli 2005

Ibolya Szilágyi

Zusammenfassung

In der Geometrischen Modellierung und in verwandten Gebieten werden algebraische Kurven/Flächen gewöhnlich entweder als Menge der Nullstellen einer algebraischen Gleichung (*implizite Darstellung*) oder als Bild einer rationalen Abbildung (*parametrische Darstellung*) angegeben. Aus der Verfügbarkeit beider Darstellungen resultiert häufig eine effizientere Berechnung.

Theorien und Techniken der Algebraischen Geometrie im Umfeld von Berechnungen mit Gleitkommazahlen sind für die Forschungsgemeinde der Geometrischen Modellierung von höchstem Interesse. Deshalb wurde die Entwicklung von Algorithmen, die auf numerische Daten anwendbar sind, ein sehr aktives Forschungsgebiet. Diese Doktorarbeit konzentriert sich auf die beiden Problemstellungen der Konvertierung – Implizitisierung und Parametrisierung – aus Sicht der Numerik.

Eine sehr wichtiger Punkt bei der Implizitisierung ist das Verhalten parametrisch gegebener Objekte unter Störungen. Für eine numerisch gegebene Parametrisierung können wir keine exakte Darstellung der impliziten Gleichung berechnen, nur eine genäherte. Wir führen eine *Konditionszahl für Implizitisierungsprobleme* ein, um die schlimmst-mögliche Auswirkung einer kleinen Störung der Eingabedaten zu messen. Unter Verwendung dieser Konditionszahl untersuchen wir die algebraische und geometrische Robustheit des Implizitisierungsprozesses.

Es gibt verschiedene Techniken zur Parametrisierung rationaler algebraischer Flächen als Ganzes. In vielen Anwendungen jedoch ist es ausreichend, einen kleinen Teil der Fläche zu parametrisieren. Dies motiviert die Untersuchung *lokaler Parametrisierungen*, d.h. Parametrisierungen einer kleinen Umgebung eines gegebenen Punktes P auf der Fläche S. Wir geben mehrere Techniken zur Erzeugung solcher Parametrisierungen für glatte kubische Flächen an. Für diese Klasse von Flächen wird gezeigt, dass sich das lokale Parametrisierungsproblem für alle Punkte lösen läßt und dass jede solche Fläche vollständig überdeckt werden kann.

Abstract

In geometric modelling and related areas algebraic curves/surfaces typically are described either as the zero set of an algebraic equation (*implicit representation*), or as the image of a map given by rational functions (*parametric representation*). The availability of both representations often result in more efficient computations.

Computational theories and techniques of algebraic geometry in floating point environment are of high interest in geometric modelling related communities. Therefore, deriving approximate algorithms that can be applied to numeric data have become a very active research area. In this thesis we focus on the two conversion problems, called implicitization and parametrization, from the numeric point of view.

A very important issue in the implicitization problem is the perturbation behavior of parametric objects. For a numerically given parametrization we cannot compute an exact implicit equation, just an approximate one. We introduce a *condition number of the implicitization problem* to measure the worst effect on the solution, when the input data is perturbed by a small amount. Using this condition number we study the algebraic and geometric robustness of the implicitization process.

Several techniques for parameterizing a rational algebraic surface as a whole exist. However, in many applications, it suffices to parameterize a small portion of the surface. This motivates the analysis of *local parametrizations*, i.e. parametrizations of a small neighborhood of a given point P of the surface S. We introduce several techniques for generating such parameterizations for nonsingular cubic surfaces. For this class of surfaces, it is shown that the local parametrization problem can be solved for all points, and any such surface can be covered completely.

Acknowledgements

First of all, I would like to express my gratitude to my supervisor, Josef Schicho, who helped me so much during my Ph.D. studies in RISC. He is the person and teacher who influenced my scientific progress the most. I feel very fortunate to be one of his students. I thank for his creative influence, helpful suggestions, patient guidance and inspiration.

I would like to give my warm thanks to Bert Jüttler for the scientific cooperations. I am grateful to him for the ideas, scientific experiences and helpful suggestions.

I give my warm thanks to Karoly Erdei for all his support and encouragement. He is the person who draw my attention to the existence of a beautiful castle and the great institute, RISC, in it.

There are several people who contributed to my developments in one or another way. I would like to give my thanks to my colleagues in the algebraic geometry group. Thanks for all their support and help! I am especially grateful to Tobias Beck (also for translating the abstract of this thesis to german) and Janka Pilnikova (I really enjoyed those hiking trips). Special thanks to the TEXpert Manuel Kauers for helping me so much with LATEX. I thank Mohamed Shalaby for his helpful suggestions, remarks and discussions both on the scientific and personal level. Thanks to Martin Aigner for the joint work. I really enjoyed it! Warm thanks to Florina Piroi for all her support. It was really great to work in the same office, and share our experiences on research and life! I give my special thank to Sonia Perez, who kept her eyes on my scientific developments, and became a really good friend of mine during these years.

I am grateful to the RISC members who created a pleasant environment during the years I spent here.

It was not easy for me to leave my country and come to Austria. I would like to thank all of my friends in Hungary for the love and support they gave me through these years!

Last, but not least I want to express my gratitude to my family. I am grateful to my parents for their moral support, encouragement, and for all their help.

The work described in this thesis has been supported by the PhD scholarship program of the Research Institute for Symbolic Computation (RISC), by the Austrian Science Foundation (FWF) in the frame of the Special Research Area SFB 013 “Numerical and Symbolic Scientific Computing”, subproject

03, and by the Johann Radon Institute for Computational and Applied Mathematics (RICAM).

Contents

Chapter 1

Introduction

"Interest in algebraic techniques for CAGD is growing, and it is evident that algebraic geometry is a valuable resource for computer aided geometric design."

T.W. Sederberg, J. Zheng

Numerous concepts, methods and tools in algebraic geometry bear a particular value for, and can be applied on problems in computer aided geometric design (CAGD). Symbolic algorithms have been widely used in geometric modelling and related areas to provide exact answers for exactly given input. In general, procedures using algebraic geometry methods are carried out using exact real arithmetic (integer, rational). However, in geometric applications the input data is often given in terms of floating point numbers, and furthermore, CAGD systems are based on floating point arithmetic. This fact heavily hinders the application of algebraic geometry methods in CAGD, and symbolic approaches tend to be insufficient in numerous applications.

Therefore, computational theories and techniques of algebraic geometry in floating point environments are of high interest. Lately, deriving approximate algorithms that can be applied to numeric data, such as approximate implicitization and parametrization algorithms or singularity computation, have become a very active research area. A consequence of applying approximate algorithms is that we have to worry about error propagation. Therefore, the analysis of approximate algorithms is very important.

In this thesis we focus on two conversion problems called implicitization and parametrization. We develop and analyze algorithms that can be applied to numeric input data.

In Computer Aided Geometric Design (CAGD) and related areas algebraic curves, surfaces and, in general, hypersurfaces can be given in various forms.

Typically, they are described as the zero set of an algebraic equation (*implicit representation*), or as the image of a map given by rational functions (*parametric representation*).

Over a field $\mathbb{K}$ the implicit equation of a hypersurface is

$$F(x_1, \ldots, x_d) = 0,$$

where F is a polynomial in the variables $x_1, \ldots, x_d$ with coefficients in $\mathbb{K}$.

The parametric representation defined by rational functions is

$$x_1 = \frac{p_1(\bar{t})}{p_{d+1}(\bar{t})}, x_2 = \frac{p_2(\bar{t})}{p_{d+1}(\bar{t})}, \ldots, x_d = \frac{p_d(\bar{t})}{p_{d+1}(\bar{t})},$$

where $p_1, \ldots, p_{d+1}$ are polynomials in the parameters $\bar{t} = (t^1, \ldots, t^{d-1})$.

Both representations are appropriate for solving different types of problems, and they have their own advantages and disadvantages. The implicit equation of hypersurfaces is essentially unique, up to multiplication by a constant factor, and the degree is apparent from their implicit equation. We can also easily determine whether a point lies on a curve or on a surface using the implicit form. In general, a particular disadvantage of the implicit representation is the large number of terms. An implicit equation of degree n in d variables can have as much as $\binom{n+d}{d}$ terms. On the other hand, the parametric representation is more appropriate for generating points. It can be done by simply plugging in particular values for the parameters in the rational function.

Depending on the particular circumstances we want to use either the parametric or the implicit form. The availability of both representations often yield more efficient computations. For instance, surface/surface-intersections can be traced efficiently if one of the surfaces is given in implicit form, and the other in parametric one. Another example is the detection of self-intersections of a surface, which becomes much simpler if both representations are available.

To avoid the weaknesses of these representations and to exploit the strength of both of them, the automatic transition between these two representations is of fundamental importance.

The conversion from the implicit form to a parametric one is called *parametrization*. Unfortunately not every algebraic curve/surface admits a rational parametrization. (For more details on parametrization and rationality see Section 2.4.

The reverse problem, i.e. the conversion from the parametric form to the implicit one is called *implicitization*. In principle, rationally given curves and surfaces can always be implicitized, namely they have an algebraic representation. Several symbolic-computation-based methods have been introduced for implicitization, based on resultants, Gröbner bases, moving surfaces, or on residue calculus.

Both parametric and algebraic representations are extensively used for low degree curves and surfaces in CAGD. However, for higher degree parametric curves/surfaces most often only the parametric representation is used. One of the reasons is that the existing symbolic-computation-based implicitization methods assume exact arithmetic, and their implementation, using floating point arithmetic, can result in unwanted effects. Therefore, to bridge the gap between algebraic geometry techniques and CAGD is very important.

Another very important issue of the implicitization problem is the perturbation behavior of parametric objects. If the input is not given exactly, but it is contaminated by numerical errors, then using approximate techniques for implicitization may be more appropriate. In the first part of this thesis we study the effects caused by using approximate implicitization.

When we use approximate implicitization we are faced with questions of the following type: How precise can we say something about the output? How well does the computed algebraic hypersurface approximate the corresponding parametric one? More precisely, we take a closer look at the following problems:

(1) How stable is the computation of the coefficient vector of an approximate implicitization? (algebraic stability)

(2) How robust (geometrically) is the resulting implicit representation with respect to small perturbations of its coefficients? (geometric stability)

Typically, the output error can be upper estimated by the input error multiplied by a constant, called condition number, which measures the stability of the algorithm. Hence, to answer the first question we introduce the *condition number of the implicitization problem*. This condition number depends not only on the input, but also on the estimation of the degree of the implicit form. For hypersurfaces with a high condition number, the computation of the coefficients of an approximate implicitization is numerically unstable, no matter which numerical method for implicitization is chosen.

Using this condition number the distance between the two coefficient vectors of two approximate implicitizations can be estimated. It is shown that for two parametrizations close to each other, the difference between the computed implicit equations can be bounded by the condition number. The condition number allows us to give a stability test for various implicitization techniques. The test is able to reject unstable implicitization techniques, but it cannot prove that a certain technique is stable. Various other applications of the defined condition number are shown in this thesis.

Unfortunately, even if the coefficients can be computed in a numerically stable way, it is not guaranteed that the varieties defined by two approximate implicitizations are close in a geometric sense. Two polynomials that differ only a little in their coefficients may define hypersurfaces with completely different shapes.

However, one can estimate the Hausdorff distance between the zero sets of an exact and a perturbed equation using a result of Aigner et al. (2004). This leads to a constant expressing the robustness of an implicit representation.

Our approach to answer the second question is based on the introduced condition number. We derive an upper bound for the Hausdorff distance between the point set defined by a given parametric representation, and the zero set of an approximate implicitization of the given parametric form.

Combining these two robustness results allows us to examine the suitability of a given rational parametric hypersurface for approximate implicitization. In certain contexts, a hypersurface might be said to be "well behaved", if it is robust in algebraic as well as in the geometric sense.

The second part of this thesis is dedicated to the parametrization problem of cubic surfaces. While algebraic techniques often parameterize algebraic surfaces as a whole, the geometries addressed within CAGD are located in a small limited portion of the two or three dimensional real affine space. Therefore, in many applications (such as geometric modelling and related areas) it suffices to have a parametrization defined in some open subset in the parameter space that covers the intersection of the surface with a certain region of interest. We will refer to this as the problem of *local parametrization*: find a parametrization of a small neighborhood of a given point of the surface. We give a method for computing local parameterizations of non-singular cubic surfaces.

Cubic surfaces are the vanishing set of a degree 3 polynomial. This class of surfaces has both an implicit and a rational parametric representation (with the exception of cones over elliptic planar cubic curves). This property may make them particularly useful in a number of geometric modelling operations. On the other hand, these surfaces are sufficiently general, since any real-valued function on $\mathbb{R}^3$ can efficiently be approximated by a piecewise cubic function (Hoschek and Lasser, 1993).

The local parametrization method described in this thesis works without analyzing the system of lines on the cubic surface. It produces rational maps defined in some neighborhood of the origin of the parameter plane with the property that the image is an open subset of a given nonsingular cubic surface containing a given point.

We use three local parametrization techniques for cubic surfaces, which are called the 2-curve technique, the repeater technique, and the reflection technique. The first two techniques can be traced back to Manin (1986) and Abhyanker and Bajaj (1987). They are based on the classical theory of rational curves on cubic surfaces. Such curves may be generated as the intersection of a surface with the tangent plane at a generic surface point.

We give a complete geometrical analysis of the introduced techniques for nonsingular cubic surfaces and we show that each of the three algorithms computes a local parametrization for a given nonsingular cubic surface S, and a surface point P. As a last remark we mention that the computed parametrization is improper.

The structure of the thesis

The remainder of the thesis is organized as follows:

Chapter 2 gives a survey on cubic surfaces with a special emphasis on the conversion problems (parametrization via implicitization).

Chapter 3 is dedicated to the numerical stability of the implicitization process. A condition number is introduced to measure the worst effect on the solution when the input data is perturbed by a small amount. We show how the condition number can be used to give an upper bound on the error using approximate implicitization. It is also shown that for any approximate parametrization of a given curve/surfcace, the curve/surface obtained by an approximate implicitization with a given precision is contained within a certain perturbation region. The main results stated in this section are the outcome of the joint work with the auhoress' advisor J. Schicho.

Chapter 4 introduces several techniques for generating local parameterizations for nonsingular cubic surfaces. For this class of surfaces, it is shown that the local parametrization problem can be solved for all points, and any such surface can be covered completely. The main results stated in this section are the outcome of the auhoress' cooperation with J. Schicho and B. Jüttler.

Chapter 2

State of the Art

Cubic surfaces are algebraic surfaces of order 3. They have possessed a great fascination since the days of the discovery of the 27 lines upon them. In the heyday of cubic surfaces the configuration of the lines and classification problems were subjects of intense investigation.

Nowadays, cubic surfaces play an important role in geometric modelling and related areas. Conversion issues, such as parametrization and implicitization are essential in applications.

This chapter gives a survey on cubic surfaces with a special emphasis on conversion problems. As the literature concerning cubic surfaces is very wide, we present results that are relevant for the rest of this thesis. Therefore, no results concerning topics such as cubic Bernstein-Bézier patches or diophantine problems are treated.

2.1 Notation

Throughout this thesis, if otherwise is not stated, we work in the projective settings over the real numbers. Cubic surfaces will be denoted by the symbol S.

In this section we list those parametric/implicit formulations of surfaces that are used all through this thesis. The rest of the notation will be introduced at that point of the thesis, where it is first used.

There are several popular parametric/implicit representations of surfaces. The parametric form can be given by rational functions or in polynomial form, with homogenous or non-homogenous coordinates. The implicit representation is either given as a non-homogenous or a homogenous polynomial. For different purposes different representations are more appropriate.

The homogenous formulation of a parametric representation of a surface is given by

$$\begin{aligned} x_1 &= p_1(y_1, y_2, y_3) \\ x_2 &= p_2(y_1, y_2, y_3) \\ x_3 &= p_3(y_1, y_2, y_3) \\ x_4 &= p_4(y_1, y_2, y_3) \end{aligned} \tag{2.1}$$

where $p_1, \ldots, p_4$ are homogenous polynomials of the same degree. The corresponding implicit equation is of the form:

$$F(x_1, x_2, x_3, x_4) = 0 \tag{2.2}$$

Surfaces in CAGD are usually defined by rational functions:

$$\begin{aligned} x &= p_1(s,t)/p_4(s,t) \\ y &= p_2(s,t)/p_4(s,t) \\ z &= p_3(s,t)/p_4(s,t) \end{aligned} \tag{2.3}$$

where $p_1(s,t), \ldots, p_4(s,t)$ are polynomials. The corresponding implicit form is given by

$$F(x, y, z) = 0 \tag{2.4}$$

Another very popular form is the so-called "half" homogenous formulation:

$$\begin{aligned} x &= p_1(s,t) \\ y &= p_2(s,t) \\ z &= p_3(s,t) \\ w &= p_4(s,t) \end{aligned} \tag{2.5}$$

where $p_1(s,t), \ldots, p_4(s,t)$ are polynomials in the parameters s, t. The corresponding implicit form is given by

$$F(x, y, z, w) = 0 \tag{2.6}$$

2.2 Historical remarks

It was in 1849, when two British mathematicians Cayley and Salmon published their famous theorem about the 27 straight lines on a smooth cubic surface. First Cayley showed that there could be only a finite number of straight lines on a cubic surface. Later Salmon proved that there are exactly 27 lines in general. Figure 2.1 shows a cubic surfaces with 27 real lines[1].

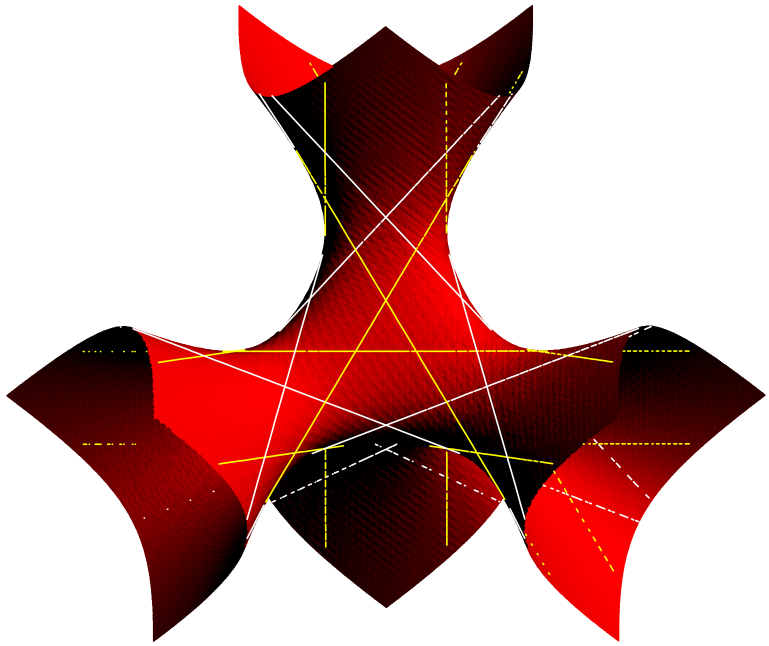


Figure 2.1: Cubic surface with 27 real lines.

With this discovery the intensive study of cubic surfaces started resulting in numerous books and publications. Steiner wrote a fruitful article containing results that serve as basis for a purely geometric treatment of cubic surfaces. He formulated several theorems, but for most of them he did not even give the idea of the proof. Many of these theorems were proved several years later by Cremona and Sturm, who were awarded the Steiner Prize.

Clebsch gave the first proof that any cubic surface can be written in the pentahedral form (the theorem was proposed by Sylvester). Among others one of his most important result is the following:

Theorem 1. *Every smooth cubic surface can be represented in the plane using four plane cubic curves through six points and vice versa.*

In 1858 Schläfli was the first to classify the cubic surfaces with respect to the number of real lines. Later he classified cubic surfaces into 23 species with

[1]The surface shown, with implicit equation $3x_4x_1^2 + 3x_4x_2^2 + 3x_4x_3^2 - 10x_1x_2x_3 - 3x_4^3$, is the favorite surface of my advisor Josef Schicho.

respect to the nature of singularities. Schläfli published an extensive memoir on the subject of cubic surfaces (Schläfli, 1858), which served as a basis for Cayley's work (Cayley, 1869).

The fact that the equation of the 27 lines can not be solved by an equation of degree less than 27 was proved by Jordan. (The reason for the above fact is that the simple subgroup of index two of the galois group of the equation does not contain any subgroup of index less than 27.) That time a lot of mathematician started to work on this subject. Klein and some of his students Maschke, Burkhardt worked on the possibility of solving the equation of the 27 lines by using theta functions.

In 1871 Klein published a series of zinc–models. One year later Clebsch presented a diagonal surface, also known as Clebsch's cubic surface, with the 27 real lines on it. Some of the models of cubic surfaces were even presented on the World Exhibition in Chicago in 1894.

In 1915 Clobe found an explicit equation of the cubic surface depending on six points on the plane.

Later mathematicians started an extensive research on the topic of finding rational points on cubic surface, as it is a key step in several parametrization techniques. In 1943 Segre showed that a nonsingular cubic surface either has no rational points or carries infinitely many. Later he extended this work to singular cubics and showed, that the existence of points with rational coordinates depends on the configuration of the singular points.

It had been conjectured by L. J. Mordell, that the Hasse principle holds for a cubic surface in 3-dimensional projective space which is not a cone; (i.e., if the cubic surface is defined over the rational field $\mathbb{Q}$ and has points defined over each p-adic field $\mathbb{Q}_p$, then it has a rational point). Swinnerton-Dyer showed the conjecture to be false for cubic surfaces in general. Later Bremner reported on a family of cubics with no rational points. Among the others Manin, Hooley, Valeiras, Siksek, Beukers, Heath-Brown contributed to the subject.

Nowadays cubic surfaces are interesting from geometric design point of view. Hence, parametrization, implicitization, intersection techniques of cubics are of great interest.

There were several books denoted for cubic surfaces such as Segre (1942), Henderson (1960), Manin (1986), moreover in several algebraic geometry books cubic surfaces reserve a whole chapter, see Reid (1988), Hunt (1996), Shafarevich (1999).

Cubic surfaces still possess the same fascination as they did in the days the 27 lines were discovered. They still inspire research, and the number of paper dealing with them keeps on growing day after day...

2.3 Geometry of cubic surfaces

There are exactly 27 distinct lines on a nonsingular cubic surface. One may conclude this theorem from the fact, that the number of lines on a nonsingular cubic surface is equal to the number of double tangent planes of an arbitrary tangent cone to the surface (Henderson, 1960). Due to Salmon, the number of double tangent planes drawn through a point to a surface of degree n is $\frac{n}{2}(n-1)(n-2)(n^3-n^2+n-12)$.

The classical reasoning, i.e. counting hyperplane sections through a line for which the residual intersection degenerates into two lines is described in Hunt (1996). A different approach using basic algebraic geometry tools can be found in Reid (1988).

2.3.1 Classification

Cubic surfaces can be classified in different manners. Taking into account the number of real lines or the nature of singularities are the most usual way for distribution.

Cubic surfaces can be partitioned into two classes in the following way:

- cubic surfaces containing finite number of lines
- cubic surfaces containing infinite number of lines

The objects of the second class can be characterized by the following theorem.

Theorem 2. *The class of cubic surfaces containing infinite number of lines consists of*

- *reducible cubic surfaces*
- *irreducible cones*
- *irreducible cubic surfaces having a double line, i.e. general ruled cubic surfaces.*

(For a proof of the theorem we refer to Brundu and Logar (1998).)

The class containing finite number of lines can be divided into numerous subclasses with respect to the number of lines, that can vary between 1 and 27.

A point $P \in S$ is called a nonsingular point of S if $\frac{\partial F}{\partial x_i}(P) \neq 0$ for some i (F is the implicit equation of S). Otherwise P is a singular point (singularity) of S. Due to the fact, that a surface contains singularities or not it is called singular or nonsingular . Concerning the type of the singularities or taking into account other geometric properties these two main classes might be divided further into different subclasses.

Classifying the nonsingular cubic surfaces with respect to the number of real lines lying upon them is due to Schläfli (Schläfli, 1858). The nonsingular cubic surfaces can be divided into 5 species $\mathbf{F}_1$, $\mathbf{F}_2$, ..., $\mathbf{F}_5$ with respect to the number of real lines (27, 15, 7, 3 and 3, respectively) and real components (1, 1, 1, 1 and 2, respectively); see Figure 2.2. (The pictures are courtesy of O. Labs.)

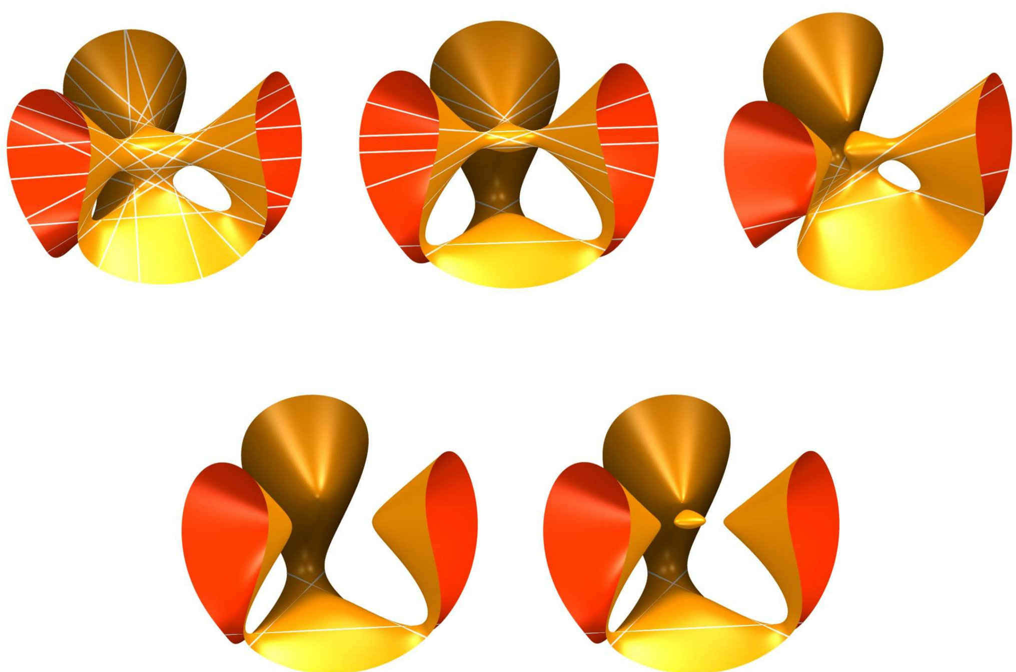


Figure 2.2: Nonsingular cubic surfaces from the classes $\mathbf{F}_1, \ldots, \mathbf{F}_5$.

Later, Schläfli classified the cubic surfaces (singular and non-singular ones) into 23 species with respect to the nature of the singularities on the surfaces. A complete classification with 21 classes over $\mathbb{C}$ using terminology

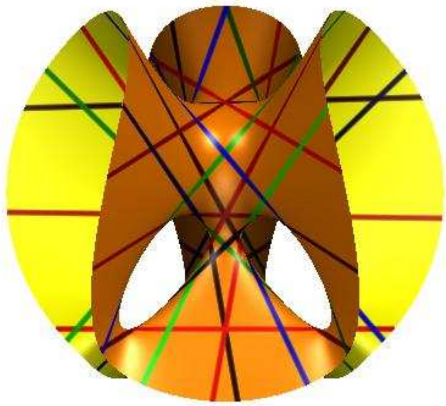


Figure 2.3: Clebsch diagonal cubic.

from modern singularity theory has been given in Bruce and Wall (1979). The distribution of cubic surfaces up to projective motions using computational methods is also available, see Brundu and Logar (1998).

2.3.2 Geometry of lines

The 27 lines on a nonsingular cubic surface has a fascinating geometry. There are several books denoted to the configuration of the lines, a detailed study can be found in Segre (1942).

Every line on a nonsingular cubic surface is met by ten others. A plane containing three of the lines is called a *tritangent* plane. There are 45 such planes on a nonsingular cubic surface ($45, 15, 5, 7, 13$ real ones respectively to the different classes). If three coplanar lines have a common point, this point is called an Eckardt point of the surface. The maximum number of Eckardt point is 10. Clebsch diagonal cubic is the unique cubic surface with the maximum number of Eckardt point.

The maximum number of non-intersecting lines of a nonsingular cubic surface is 6. We call 6 nonintersecting lines a sextuplet. Each sextuplet has a well-defined complementary sextuplet, and together they form a so-called *double-six*. The sextuplets are in a one-to-one correspondence; two lines correspond iff they are skew. There are 36 double-sixes on a nonsingular cubic surface.

A plane in general intersects a cubic surface in a cubic curve. A plane through a line of the surface intersects the surface besides this line in a residual conic. Taking a pencil of planes through a line of the surface, the residual conics generate an involution on the line. The two double points of the involution are called parabolic points of the line. A real line of a nonsingular cubic surface is called elliptic or hyperbolic according as it has two conjugate complex or real parabolic points.

It can be shown that a real nonsingular cubic surface of the type $\mathbf{F}_1, \ldots, \mathbf{F}_4$ consists of a single connected component, while a surface of type $\mathbf{F}_5$ consist of two different pieces (Segre, 1942).

A real cubic surface of type $\mathbf{F}_1, \ldots, \mathbf{F}_4$ and the nonconvex component of a surface $\mathbf{F}_5$ is divided by its lines into a certain number of cells constituting a so-called generalized polyhedron of the surface. This polyhedron contains triangular, quadrangular and pentagonal cells. For further details on the generalized polyhedron, see Segre (1942).

2.3.3 Base points and degree

Denoting the parametric representation of a surface as (2.1) the *base point* is any parametric triple (y_1, y_2, y_3) for which

$$p_1 = p_2 = p_3 = p_4 = 0.$$

The number of base points in the case of cubic surfaces is 6. Base points might be complex or infinite.

These base points determine cubic surfaces up to a linear change of coordinates. It is well–known that four cubics through six base points parameterize in general a cubic surface, see Sederberg (1990b). Given a set of six base points, the linear system of cubic curves passing through all of them form a vector space. The dimension of this vector space is 4. Any choice of basis of the linear system can be thought as a parametrization of a cubic surface. Defining base points and using undetermined coefficients, the solutions of the arising equation system give a parametric representation of a surface (containing the given base points).

If the base point are generic, i.e. no 3 of them are collinear and not all the 6 lie on a conic, then they determine a nonsingular cubic surface over the complex numbers. In case the base points are not generic, the corresponding surface will be singular.

The base points of a cubic surface can be used to show the existence of the lines. Each base point of the parameter space, any line connecting two base points and each conic containing five of the base points map to a line on the surface. Assume, the base points are generic. Then the 6 base points map to

type	# real b. p.	# real l.
$\mathbf{F}_1$	6	27
$\mathbf{F}_2$	4	15
$\mathbf{F}_3$	2	7
$\mathbf{F}_4, \mathbf{F}_5$	0	3

Table 2.1: The relation between the type of the base points and the number of real lines on a nonsingular cubic surface.

6 lines of the cubic surface. The possible 15 pairs of base points account for another 15 of the lines. As there are 6 ways a conic can pass through five of the six base points, we get additional 6 straight lines. These lines altogether give the 27 lines of a nonsingular cubic surface.

If all six base points are real, then all the resulting lines are also real. If some of the base points are complex, then some of the lines on the surface are also complex. Table 2.1 shows the relation between the type of base points and the number of real lines on a nonsingular cubic surface.

The *degree* of a surface is apparent from its implicit equation. It is important as it is a major indicator of surface characteristic. In general the degree of the implicit equation of a surface is higher than the degree of the parametric form. The degree of the implicit equation can be thought of as the number of times that the surface is intersected by a generic straight line. Assume we have given a surface in parametric form of parametric degree n. (On parametric degree we mean the maximum of the total degree of the defining polynomials) A generic straight line can be defined as the intersection of two distinct planes in general position. These planes intersect the parametric surface in two degree n curves. By Bezout's theorem it is known, that these two curves have n^2 intersection points. This is also the number of times that the a line (intersection of the two distinct planes) intersects the surface. Hence, n^2 is the *implicit degree* of the surface.

As a natural question it arises if there are parametric surfaces whose implicit degree is not a square number, and if yes, under what conditions the degree decreases. If there exist a base point, the intersection of any plane with the surface will contain this point. As the base point does not map to a unique point on the surface, a generic line will have $n^2 - 1$ intersection points with the surface. Therefore, the degree of the surface is $n^2 - 1$. Each additional simple base points decreases the degree of the surface by one. Hence, a parametrization of degree 3 with 6 base points lead to an algebraic surface of implicit degree $3^2 - 6 = 3$.

More complicated base points have different influence on the implicit degree. If one of the base points is a double point, it decreases the implicit degree by 2. The general degree formula is $n^2 - p$, where p is the number of base points counted with multiplicities. For more details we refer to Farin et al. (2002), Sederberg (1990a).

2.3.4 Computation of the 27 lines

Given the parametric representation of a cubic surface, it is not complicated to determine the lines lying on it using the relation described in Section 2.3.3. However, given the implicit form it can be cumbersome to compute all the existing lines. It was proved already by Jordan that the equation of the 27 lines (in the nonsingular case) can not be solved by an equation of degree less than 27.

Hunt (Hunt, 1996) sketched two methods for solving the equation of degree 27 determining the lines on a nonsingular cubic surface. One of the strategy was suggested by Klein, the other one is due to Coble, both consisting an algebraic and a transcendental part.

A general cubic form can be put in a four-dimensional family of forms $z_1^3. \dots, z_6^3 = 0$, where z_i are linear in the variables x_i. This special form is called the *hexahedral form*. Cremona noted, that one can explicitly write down the equations for the lines of a cubic surfaces using the hexahedral form. However, there is no purely algebraic method of passing from the form given by (2.2) to the hexahedral form (Hunt, 1996).

Here we present three possible ways of computing the equations of the lines of a cubic surface.

Construction 1

First we show how to compute one line of the surface. Using this line it is possible to compute the rest of them.

Assume that we have a cubic surface given by its implicit equation F as (2.4). The parametrization of a line with unknown coefficients is $l(t) = (t, y_0 + y_1 t, z_0 + z_1 t)$. Substituting the parametric equation of the line into the implicit equation of the surface yields an equation $F(l(t))$ of degree 3 in the parameter t. If this equation is identically zero, it guarantees that the line lies entirely on the surface. The condition for $F(l(t))$ to be identically zero is that all the coefficients vanish simultaneously.

In this way the problem of finding a line of a surface is transformed into the problem of solving a system of four nonlinear equations in four unknowns. Using elimination methods or some standard numeric technique we can find at least one solution of the system.

Assume, we have found one line on a cubic surface. We show how to use this line to find other lines of the surface.

First we use some suitable coordinate change, so that the given line becomes the z axis. Then we take a pencil of planes through the z axes and intersect it with the surface. The intersection consists of a line (the z axes) and a residual conic. The condition for this conic to degenerate into a pair of lines is that the determinant of its Hessian vanishes. The determinant of the Hessian of a conic is a degree 5 polynomial. After computing the roots of this polynomial, it is possible to get further lines of the cubic surface. For more details, see Sederberg (1990b).

Construction 2

The method proposed in Bajaj et al. (1998) is based on constructing and solving a polynomial of degree 27. The procedure assumes that we have found a rational/real point on the given cubic surface. Numerically it is not a problem to compute a real point. However, we have to point out that in the exact situation it is not trivial (Manin, 1986).

Assume we have computed a real point on the given surface. Then using an appropriate transformation we move this rational point to the origin, such that the tangent plane at the origin is the xy plane. We parameterize the curve $\mathcal{C}$ the tangent plane at the origin cuts out from the surface (rational cubic with a double point at the origin) using the parameter t. We transform the surface again such that a general point of $\mathcal{C}$, depending on the parameter value t is moved to the origin, and the tangent plane at the origin is $z = 0$. The intersection of the transformed surface with the tangent plane at the origin gives an expression in the parameter t. Values of t for which the obtained expression is reducible give rise to the lines of the surface.

Using this construction we obtain a polynomial of degree 27 in the parameter t. The solutions of the polynomial correspond to the lines of the (transformed) cubic surface.

Construction 3

If we are working with floating point coefficients, or the implicit equation is given exactly but we are satisfied with a numeric answer, concerning the lines, the following can be done.

As in Construction 1 we transform the problem of finding the lines of a surface into the problem of solving a system of four nonlinear equations in four unknowns. Our task is to find (possibly all) the solutions of the system. The solutions give rise to the equation of the lines.

After experimenting several methods and systems, we found the usage of PHCpack the most convenient to find the solutions of the system. PHCpack is a general-purpose solver for polynomial systems by homotopy continuation method. It computes numerically approximations to all isolated solutions of a system of n polynomial equations in n unknowns.

For more details on PHCpack and homotopy continuation method see Emiris and Canny (1995), Emiris and Verschelde (1999).

2.3.5 Rational curves on cubic surfaces

Here we consider lines, quadrics and cubic curves on cubic surfaces. As we have seen before, every cubic surface contains at least one real line. A plane through a straight line of the cubic intersects the surface besides the line in a conic, which may degenerate into two lines. If P is a non-parabolic point of a real line l of S not lying on any other line, then the tangent plane at P intersects the surface in l and an irreducible conic. The conic intersects l in P and in a further point P'. As every conic in a cubic lies in a plane that intersects the surface in this conic and a straight line, there is a bijective correspondence between lines and pencils of conics on the surface.

In principle a plane cuts a cubic surface in an irreducible plane cubic curve. In general this intersection curve is elliptic, i.e. it does not have a rational parametrization. However, if it contains a double point P it becomes rational. This double point P is either a node of the surface, or the plane is a tangent plane to the surface at P.

The tangent plane at an arbitrary point of a cubic surface cuts a degree 3 rational curve out of the surface. This curve can be either reducible or irreducible, depending on the point where the tangent plane was taken.

Depending on the local behavior of a surface with respect to its tangent plane at a point, one arrives at different types of surface points. A point is called *elliptic* if the tangent plane at the point intersects the surface (locally) in an isolated point, *hyperbolic* if the tangent plane intersects the surface (locally) in a pair of intersecting curves with two different tangents, and *parabolic* otherwise.

The parabolic points of any cubic surface lie on the so-called parabolic curve of the surface. In general the parabolic curve of any nonsingular cubic is nonsingular. If it is singular then it has at most finite number of double points (which are the real Eckardt points of the surface). The parabolic curve touches each of the lines of the cubic in its two parabolic points. The parabolic curve arise as the intersection of the cubic surface with its Hessian surface.

Given an arbitrary hyperbolic point P of a nonsingular cubic surface S not lying on any line of F, the intersection of S with the tangent plane at P results in an irreducible curve, which has a nodal point at P.

Considering the intersection of a cubic surface with the tangent plane at a point P we can distinguish six possible shapes of cubic curves, see Figure 2.4. The cubic curve can be irreducible with a node, cusp or with an isolated singularity at P, or reducible consisting a line and a conic, three lines not belonging to a pencil or three lines going through P (in this case P is an Eckardt point of the surface).

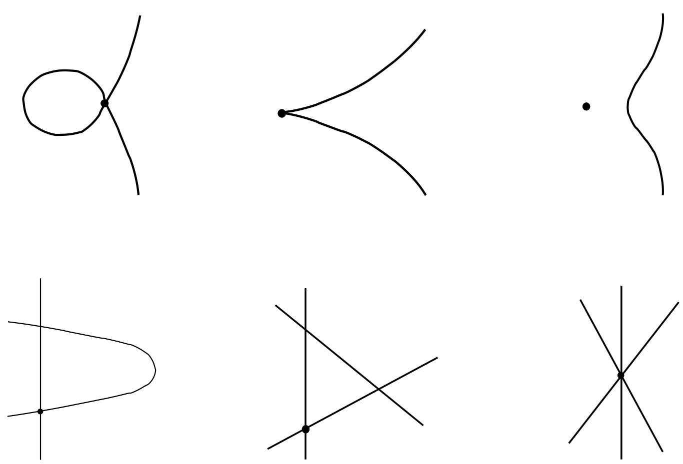


Figure 2.4: Cubic curves arising as the intersection of a cubic surface with a tangent plane at a surface point.

2.4 Rationality of cubic surfaces

While implicitization is always possible in general, not every algebraic surface admits a parametrization. Surfaces that can be parameterized are called *unirational*. If the parametrization is birational, i.e. the parameters can be expressed in terms of rational functions, it is called *proper*. Surfaces with a proper parametrization are called rational surfaces.

In the complex case Castelnuovo's criterion give a necessary and sufficient condition for unirationality. It is also known that for complex surfaces rationality and unirationality are equivalent.

Theorem 3 (Castenuovo). *Let S be a surface. Over the complex numbers the following are equivalent.*

1. *S is rational.*
2. *S is unirational.*
3. *$p_a(S) = P_2(S) = 0$,*

where $p_a(S), P_2(S)$ are the arithmetic genus and the second plurigenus respectively.

Remark 1. *For the definition of the arithmetic genus and the second plurigenus of an algebraic variety we refer to Shafarevich (1999). These numbers are birational invariants, their definition does not depend on the chosen field.*

In the real case Castelnuovo's criterion is still necessary. For deciding rationality we have the following theorem.

Theorem 4 (Comesatti). *Let S be a real surface. Over the real numbers rationality is equivalent to $p_a(S) = P_2(S) = h_0(S) - 1 = 0$, where h_0 is the number of connected components of S*

Remark 2. *The number of connected components of a surface is also a birational invariant in the real case. For more details see Canny et al. (1992). The number of connected components of a surface is counted in the projective setting after desingularization.*

A strategy for computing a parametrization is suggested by the following theorem.

Theorem 5 (Enriques, Manin). *A surface with $p_a(S) = P_2(S) = 0$ has a pencil of rational curves or it is birationally equivalent to a del Pezzo surface.*

Cubic surfaces have the valuable property that in general they can be parameterized. The only exception is the ruled cubic generated by a nonrational cubic curve.

Example 1. The surface with equation $x_2^2 x_3 - x_1^3 - x_1 x_3^2 = 0$ (see Figure 2.4) has $P_2 = 0$ and $p_a = -1$. Hence, it does not have a rational representation.

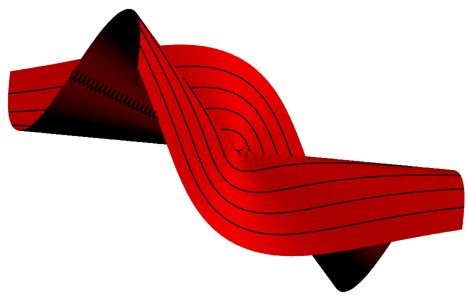


Figure 2.5: Nonrational cubic surface.

What is the necessary and sufficient condition for a nonsingular cubic surface to have a proper parametrization over a given field? This question was first raised by Segre and partial answers can be found in Segre (1942), Manin (1986). Later Swinnerton-Dyer (Swinnerton-Dyer, 1970) proved the following theorem:

Theorem 6. *A necessary and sufficient condition that a nonsingular cubic surface S should be birationally equivalent to the projective plane over a given field $\mathbb{K}$ is that S should contain a point defined over $\mathbb{K}$ and S should have at least one S_2, S_3 or S_6.*

Here S_n denotes any subset of the 27 lines on S which satisfies the following conditions:

- S_n consists of n nonintersecting lines
- If S_n consist a line, then it also consist all the conjugates of this line over the given field.

$\mathbf{F}_5$ surfaces have no real one-to-one parametrization, which is dues to the fact that they have two real components while the projective plane has only one. In this situation one either uses a two-to-one parametrization or parameterizes the components separately.

2.4.1 Existing parametrization techniques

Unfortunately, Castelnuovo's criterion does not provide a constructive approach to parametrization. Hence, various parametrization methods based on different approaches have been developed. Concerning the class of nonsingular cubic surfaces the lines lying upon them give the key to parametrization.

In the sequel we give an overview of existing parametrization methods that are applicable to cubic surfaces. Only the basic ideas are described, for full details we provide references.

Most of the parametrization methods, especially the ones for nonsingular cubic surfaces, are based on the existence of the real lines. However, other geometric properties that can be used for constructing a parametrization have not been fully explored.

The presented methods parameterize surfaces as a whole. However, within CAGD the geometries addressed are located in a small portion of 2 or 3 dimensional real affine space. Hence, parametrization techniques that deal only with a small piece of the surface could be of interest.

Skew line parametrization

As the name of this parametrization method already indicated, this technique requires the determination of two skew lines l_1, l_2 on the surface given by (2.4). Assume, we have found two such lines. First we parameterize each of them using parameter value s on one of them and t on the other line. Then we take a variable line through an arbitrary point on l_1 and an arbitrary point on t_2. This variable line has a parametric representation in terms of s and t. It will intersect the cubic surface in one additional point. We assign to this point the s, t parameters, which defined the variable line.

For each point on the cubic surface there exists a unique line that passes through the given point and a point on each of the given skew lines (Sederberg and Snively, 1987). Hence, this parametrization is one-to-one.

In the case of nonsingular cubics only surfaces in the classes $\mathbf{F}_1, \mathbf{F}_2$ and $\mathbf{F}_3$ contain real skew lines. As we have seen in Section 2.3.1 surfaces that belong to the $\mathbf{F}_4, \mathbf{F}_5$ classes contain only 3 real lines which are all coplanar. In the $\mathbf{F}_4$ case it is possible to use a complex conjugate pairs of skew lines for the parametrization (Bajaj et al., 1998). Unfortunately this is not the case for $\mathbf{F}_5$ surfaces. A cubic surface of class $\mathbf{F}_5$ does not have any complex conjugate skew lines.

Certain classes of singular cubics also contain two skew lines. Therefore, the described technique can be used to give a parametric description of several singular surfaces.

Single real line parametrization

The method is based on finding and parameterizing a real line on the cubic surface. At each point P of the line we take the tangent plane to the surface. It will intersect the surface in the given line and a conic. As P is a double point of the intersection, it will lie both on the line and the conic. The parametrization of the conic by a pencil of lines through P gives a parametrization of the surface (Sederberg and Snively, 1987).

The presented method provides a two-to-one parametrization, i.e. for each point on the surface there are two corresponding parameter pairs in the parameter space.

Every cubic surface contains at least one real line (Brundu and Logar, 1998), hence the single real line parametrization method can be used for any cubic surface. However, as it does not provide a one-to-one parametrization, other methods are preferable when possible.

Square root parametrization

This parametrization method is the most general, it can be applied to any cubic surface. First we locate a point P on the surface at which the surface intersects the plane at infinity. It can be done by solving the equation $F(x_1, 1, 0, 0) = 0$. It has three solutions, at least one of them which is real.

If we move the point P such that it lies along the x_3 axes, the equation of the surface takes the form

$$x_3^2 L(x_1, x_2, x_4) + x_3 D(x_1, x_2, x_4) + C(x_1, x_2, x_4) = 0, \tag{2.7}$$

where L, D and C are linear, quadratic and cubic homogeneous polynomials, respectively. As the above expression is quadratic in x_3 we get the following parametrization in the parameters y_1, y_2, y_3

$$x_1 = y_1, x_2 = y_2, x_3 = \frac{-D + \sqrt{D^2 - 4LC}}{2L}, x_4 = y_3.$$

Parametrization using the Hilbert-Burch theorem

In Berry and Patterson (2001) the authors described a sequence of steps that can be used to pass back and force between the parametric and the implicit description of a cubic surface. The method is based on the theorem of Hilbert-Burch (Hartshorne, 1977). The essence of the process is the construction of two matrices, which the authors call the Hilbert-Burch matrix M_{HB} and a related 3×3 matrix $M_{3\times3}$.

The Hilbert-Burch matrix is a 4×3 matrix containing linear form in the parameters, while $M_{3\times3}$ contains linear forms in $x_1, \ldots, x_4$. The signed determinants of the 3×3 submatrices of M_{HB} give the polynomials of the parametrization up to a constant multiple. For each M_{HB} a related $M_{3\times3}$ can be constructed, and vice versa. It can be shown that $det(M_{3\times3}) = 0$ is the implicit equation of the cubic surface.

In the parametrization process for a given implicit equation one builds up the matrix $M_{3\times3}$ and the related matrix M_{HB}, which provides a parametrization of the given surface. For further details on the construction of the mentioned matrices, see Berry and Patterson (2001).

The construction requires the computation of some of the 27 lines of the cubic surface. This parametrization method can be applied for nonsingular cubic surfaces of type $\mathbf{F}_1, \ldots, \mathbf{F}_4$. The resulting parametrization is in terms of degree 3 polynomials.

Parametrization of singular surfaces

A singular cubic surface can be parameterized only if it is not a cone over an elliptic cubic curve. (Cones are considered in the projective sense. In the affine situation neither cones nor cylinders over a nonsingular cubic curve posses rational parametrization.) A singular cubic surface can have maximum four discrete double point or a double line.

Assume we know one double point P of the surface. After moving the singularity to the origin, the equation of the cubic takes the form

$$wD(x, y, z) + C(x, y, z) = 0, \tag{2.8}$$

where D and C are quadratic and cubic homogeneous polynomials respectively.

As any line intersects a cubic surface in three points, lines through a double point have one further intersection point with the surface. Hence, for almost all points of the surface there is a unique line connecting this point with the double point. This one-to-one correspondence between the bundle of lines with center P and points of the cubic define a parametrization.

We can derive a parametric representation of the surface by expressing the lines through the origin as the intersection of two pencils of planes $y = sx$, $z = tx$, and making the necessary substitutions.

Schicho's parametrization method

For a very long time parametrization of rational surfaces was posed as an open problem in computational algebraic geometry. There were parametrization methods for particular classes of surfaces like quadric, cubic or canal surfaces. Schicho gave an algorithm that decides whether a parametric representation exists, and if the answer is yes, computes one using the concept of adjoints (Schicho, 1998b). The method is general, it can be used for cubic surfaces as well as for any other types.

After computing the adjoints p_a and P_2 can be determined to decide the existence of parametrization. If both numbers are zero a birational map from the given surface to another surface, which is easier to parameterize, can be computed. This other surface is one of the following:

- the projective plane
- a quadric surface in $\mathbb{P}^3$
- a rational scroll (s surface with a pencil of lines)
- a surface with a pencil of conics
- a Del Pezzo surface

The final step is the inversion of a birational map to the plane. For the computation of this inversion Gröbner bases or resultants can be applied.

For detail on the parametrization algorithm as well as on adjoint computation we refer to Schicho (1998b).

Parametrization of approximate surfaces

If the parametric surface is given approximately, the above symbolic approaches tend to be insufficient, hence approximate parametrization methods are needed.

In Pérez-Díaz et al. (2005) the authors describe a method that is applicable to ϵ- irreducible algebraic surfaces S of degree n having an ϵ-singularity of multiplicity $n-1$. Given a tolerance $\epsilon > 0$ and an ϵ-irreducible polynomial defining a surface S, the algorithm computes a rational algebraic surface $\bar{S}$ and a rational parametrization of it. The parametrization of $\bar{S}$ is done by taking a pencil of lines through its singular point. It can be shown that $\bar{S}$ lies in the offset region of S at distance at most $\mathcal{O}(\epsilon^{1/2n})$.

2.5 Implicitization methods

In the language of algebraic geometry the implicitization problem asks for the smallest variety containing the parametrization. As the implicitization problem can be reduced to eliminating the parameters (Cox et al., 1997), several techniques involve the use of elimination theory.

Having a determinantal representation of the implicit equation is useful for geometric modelling. Therefore, methods representing the implicit equation by a matrix, such as the ones based on resultants or moving surfaces, are of high interest.

The implicit surface often contains unwanted branches and singularities. For these reasons several approximate implicitization techniques have been proposed.

For an exactly given input, using symbolic-computation-based techniques, an exact implicit representation can be computed. However, the input data coming from applications is often contaminated by errors, it is given in terms of floating point numbers. In such a situation we can not hope for an exact answer, only an approximation can be computed. As a consequence of working with unprecise data we have to worry about error propagation. The implementation of exact methods using floating point arithmetic can give unwanted effects.

In order to guarantee that a particular method produces nearly the right answer for the given input, the numerical behavior of particular implicitization methods should be analyzed. More generally, one could analyze the numerical behavior of the implicitization process independently from any particular implicitization technique.

In this section we give an overview on existing implicitization methods based on symbolic and approximative techniques.

2.5.1 Symbolic techniques

The most common and well-known techniques mentioned throughout the literature are the ones based on Gröbner bases, resultants or moving surfaces. Despite these technique the implicitization problem has been addressed using a variety of mathematical methodologies and techniques such as characteristic sets, perturbation methods, multidimensional Newton formulae, elimination theories, symmetric functions and approximate complexes (Busé and Chardin, 2005). Some of the techniques work only for special categories of curves, surfaces, hypersurfaces, and many of them handle only special kinds of parametric representations.

In the sequel we sketch the idea od various symbolic implicitization techniques.

Gröbner bases

Given a surface in parametric form as in (2.1). We consider the ideal $I = \langle x_1 - p_1, \ldots, x_4 - p_4\rangle$ and compute the Gröebner bases with respect to the ordering $y_1 > \cdots > x_1 > \cdots > x_4$. By the elimination theorem (Cox et al., 1997) the elements of the Gröebner basis not involving $y_1, \ldots, y_3$ define the smallest variety containing the parametrization.

When the parametrization is given by rational functions (2.3) the considered ideal is $I = \langle xp_4 - p_1, \ldots, zp_4 - p_3, 1 - p_4\bar{t}\rangle$. Gröbner bases are computed with respect to the ordering $\bar{t} > s > t > x > y > z$. The additional condition $1 - p_4\bar{t} = 0$ guarantees that the computed implicit equation is the smallest variety containing the given parametrization. For further details see Buchberger (1988).

The described method fails to determine the implicit equation if the parametrization has base points. In such situations the implicit equation does not belong to the ideal I, but to the saturated ideal of I by the ideal of the base points. In Fix et al. (1996), Gao and Chou (1992), Manocha and Canny (1992) the authors describe some perturbation techniques which allow to compute the implicit representation in the presence of base points.

Resultants

If the parametrization does not have base points than multipolynomial resultants can be used to determine the implicit representation of the surface (Cox et al., 1998). If the system $p_1 = \cdots = p_4 = 0$ has only the trivial solution, then the resultant of $x_1 - p_1, \ldots, x_4 - p_4$ provides he implicit equation. The most often used resultant formulations are Dixon's (Sederberg et al., 1984), Bezouth's (Busé et al., 2000) an Macaulay's (C. Bajaj, 1988). The resulting expression might contain an extraneous factor, whose separation can be a time consuming task involving multivariate factorization.

However, in the event of base points the classical and toric resultants are degenerated. One solution to overcome this problem to use perturbation techniques (Manocha and Canny, 1992). In Busé (2001) the author describes a method based on residual resultant to compute the implicit equation of a rational surface with base points. An overview on resultant theory with a particular emphasis on the implicitization problem can be found in Busé et al. (2003b).

Moving surfaces method

This radically new approach has been proposed in Sederberg and Chen (1995). A *moving surface* is defined as $g(x, y, z, s, t) = \sum_{i=1}^{\sigma} h_i(x, y, z)\gamma_i(s, t) = 0$, where the equations $h_i(x, y, z) = 0$ define a collection of implicit surfaces, and $\gamma_i(s, t)$ are a collection of polynomials in s, t. The $\gamma_i(s, t)$, the blending functions of the moving surface, are required to be linearly independent and relatively prime. When h_i is linear or quadratic the moving surface is called a *moving plane* or *moving quadric* respectively. If A moving surface is said to follow a rational surface if

$$g\Big(\frac{p_1(s,t)}{p_4(s,t)}, \frac{p_2(s,t)}{p_4(s,t)}, \frac{p_3(s,t)}{p_4(s,t)}, s, t\Big) \equiv 0.$$

If a set of moving surfaces can be found, each of which follow a given rational surface, then a matrix can be defined. The implicit equation of the rational surface is expressed as the determinant of this matrix. The rows of the matrix correspond to moving planes or moving quadrics. Hence, the key to this method is finding a square matrix, which means that the number of blending functions is equal to the number of linearly independent moving surfaces.

In the presence of base points the rank of the matrix drops and the expressions simplify. The method of moving surfaces has been proved valid in the presence of base points under suitable algebraic conditions (Busé et al., 2003a). A more general version of the method presented in Zheng et al. (2003) is capable to deal with multiple base points. The method also applicable for tensor product surfaces and triangular patches.

A "simple" method

The procedure that was proposed in Wang (2004) starts with writing out the implicit polynomial F of estimated degree with undetermined coefficients a_i and substituting the parametric forms. This yields an expression in the parameters. Equating the coefficient of this expression to zero a linear equation system in a_i is generated. Hence, the implicitization problem is reduced to solving a sparse linear system for a_i. A nontrivial solution of the system yields to the implicit equation. If the system does not have nontrivial solution the same process can be repeated after increasing the degree of F.

The method works for curves as well as for surfaces and hypersurfaces; the presence of base points do not diminish its efficiency.

Multidimensional Newton formulae

In Gonzalez-Vega (1997) resultant and Gröbner basis computations are replaced by the computation of Newton Sums and by the application od multidimensional Newton formulae which relate Newton Sums with the coefficients of the polynomials in the considered polynomial system of equations. The method is applicable to surfaces whose parametrization has a particular structure.

2.5.2 Approximate techniques

An implicitized parametric surface might have unwanted components or self-intersection (singularities). Moreover, the degree of the implicit equation is usually high, which implies huge number of terms. The undesirable components and singularities often lead to topological inconsistency and computation instability in geometric modelling. To overcome these problems, recently several approximate implicitization techniques have been proposed.

A power series method to obtain local approximation about a regular point was suggested in de Montaudouin et al. (1986). In Chuang and Hoffmann (1989) this method was extended to compute a low-degree implicit equation to locally approximate a properly parameterized rational surface (curve). Dokken proposed a way to globally approximate the parametric surface (curve). Global means that the approximation is valid in the whole domain of the surface patch. Later algorithms based on monotonicity and least square approximation (Dahmen, 1989; Guo, 1991), certain constraints on the coefficients of the implicit representation (Bajaj et al., 1995; Sederberg, 1985) and monoid surfaces have been introduced (Sederberg et al., 1999).

Approximate implicitization

The first presentation of approximate implicitization techniques of Dokken was in his Ph.D. dissertation (Dokken, 1997).

Let $\mathbf{p}(y_1, \ldots, y_3)$ a parametric representation of a surface. The nontrivial algebraic surface $F(x_1, \ldots, x_4) = 0$ is an *approximate implicitization* of $\mathbf{p}$ within the tolerance $\epsilon \geq 0$, if we can find a continuous function $g(y_1, \ldots, y_3)$ describing the error direction for error measurement and an error function $\mu(y_1, \ldots, y_3)$, such that $F(\mathbf{p}+\mu g) = 0$ with the conditions $\|g(y_1, \ldots, y_3)\| = 1$ and $|\mu(y_1, \ldots, y_3)| \leq \epsilon$. If $\epsilon = 0$ then we have exact implicitization.

The combination of an algebraic surface of degree m and a parametric one can be expressed as a matrix vector product $F(\mathbf{p}) = (Mb)^t\alpha$, where M is a matrix, b contains the coefficients of F and α contains the basis functions related to the coordinate functions of $\mathbf{p}$. If b is in the null space of M, then $F(\mathbf{p}) = 0$. Thus the implicitization problem is reformulated to finding the null space of a matrix. Singular value decomposition can be used to find an approximate implicitization. It can be shown, that vectors belonging to the smallest singular value of M are the best candidates for approximate implicit representation of $\mathbf{p}$. For details see Dokken (2001), Dokken and Thomassen (2003).

The accuracy of the approximation is also addressed in the above papers. A weak formulation of the method can be found in Dokken et al. (2001). It constructs an implicit object that approximates the parameterized one on a bounded parameter interval. The technique of approximate implicitization works for curves, surfaces as well as for hypersurfaces.

Numerical implicitization with linear algebra

The implicitization problem can be considered as an eigenvalue problem (Corless et al., 2001). For a chosen degree m a vector v of all power products of total degree up to m in the variables of the implicit form is constructed. After computing the matrix $M = v^t \cdot v$ and substituting the variables by the parametric representations the elements of the matrix are integrated. After integration a null-vector nv can be computed. The implicit equation will be $M \cdot nv$.

The method works for polynomial, rational as well as for trigonometric parametric representations.

Approximation using monoid surfaces

A monoid surface of degree n has a multiple point of order $n-1$, hence they are rational surfaces. As monoids can closely approximate a parametric surface they can be use to determine approximate implicit representations for rational curves.

The procedure described in Sederberg et al. (1999) consists in four steps. Firs a reference tetrahedron for the monoid surface should be determined. After converting the parametric surface p into barycentric form $\bar{\mathbf{p}}$ (using the reference tetrahedron), it should be combined with the implicit equation F (with unknown coefficients). The composition can be written as $F(\bar{\mathbf{p}}) = (Mb)^t\alpha$, where M is a matrix, b contains the coefficients of F and α contains the basis functions. Performing least square approximation the unknown coefficient vector b can be determined.

Chapter 3

Implicitization and Numerical Stability

In geometric applications the input data is often given in terms of floating point numbers. However, for numerically given parametrization we can never compute the exact implicit equation, but an approximate one.

Given numeric input data the question is how precise we can say something about the output. Typically the output error can be upper estimated by the input error times a constant, which measures the stability of the algorithm.

Normally the stability depends on the algorithm. The stability constant for the best possible algorithm is the condition number of the problem itself: no matter which algorithm we use, we can not say anything more on the accuracy of the output than described by this condition number.

In this section we introduce the *condition number of the implicitization problem*. It depends not only on the input, but also on the estimation of the degree of the implicit form.

3.1 The problem

The process of computing the implicit form of a parametrically given object is called implicitization. In general, implicitization is always possible. There are several existing symbolic implicitization methods as it was described in Section 2.5. These techniques produce an exact implicit representation for the input parametric object.

However, in real life problems the input is often not given exactly but it is contaminated by numerical errors. Given numeric input data it might be better to use approximate techniques, like the methods described in Section 2.5. In such situation it is not possible to compute the exact implicit representation but an approximate one. Hereinafter, we study the effects caused by using approximate implicitization.

The question is how precise we can say something about the output. In other words we would like to estimate how far the computed implicit equation is from the nominal one (the implicit equation we would get without numeric errors), i.e. to estimate the error in the coefficient vector. Typically the output error can be upper estimated by the input error times a constant which measures the stability of the algorithm.

To measure the sensitivity of the solution to small changes in the input we introduce a constant that we call the *condition number of the implicitization problem*. It depends not only on the input, but also on the estimation of the degree of the implicit equation. Such an estimation must always be used, see Corless et al. (2001), Dokken (2001). If this condition number is "big" then even small changes in the input can result in a big change in the output, i.e. the implicitization problem is ill-conditioned.

Normally the stability depends on the algorithm. The stability constant for the best possible algorithm is the condition number of the problem itself: no matter which algorithm we use, we can not say anything more on the accuracy of the output than described by this condition number.

Using this constant we can answer the question how far the computed implicit equation is from the nominal one. We can give an upper bound for the error caused by using approximate implicitization.

We started this research with the study of a particular implicitization algorithms for cubic surfaces. Later it emerged that the results are more general. They hold not only for cubic surfaces but they can be generalized for curves, surfaces and in general for hypersurfaces.

3.2 Preliminaries

In this section we give a short overview on a powerful matrix factorization tool called singular value decomposition. We also intend to clarify the notions conditioning and stability following the concepts of Higham (2002).

3.2.1 Singular value decomposition

Let $m, n \in \mathbb{Z}$ arbitrary. A matrix $M \in \mathbb{R}^{m \times n}$ can be written as

$$U \cdot \Sigma \cdot V^t,$$

where $\Sigma \in \mathbb{R}^{m \times n}$ is diagonal, $U \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{n \times n}$ are orthogonal matrices. This factorization of M is called *singular value decomposition* (SVD).

The diagonal entries $\sigma_1, \ldots, \sigma_r$ of Σ are the *singular values*. They are nonnegative and in decreasing order $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0$, where $r = min(m, n)$. The columns of the matrices U and V are called *left* and *right singular vectors* respectively. The right (left) singular vectors $v_1, \ldots, v_n$ $(u_1, \ldots, u_m)$ are orthogonal and have unit length.

It can be proved that every matrix has a singular value decomposition. Furthermore, the singular values are uniquely determined. SVD is applicable for real as well as for complex matrices.

There is a strong relation between SVD and geometry. It is motivated by the geometric fact that the image of the unit sphere under any linear transformation given by $m \times n$ matrix is a hyperellipse. (A hyperellipse might be defined as a surface obtained by stretching the unit sphere in $\mathbb{R}^m$ by some factors $\sigma_1, \ldots, \sigma_m$ in some orthogonal directions.) Assume, the linear transformation, by which the hyperellipse is obtained, is given by the matrix M. Then the singular values of M are the length of the principal semiaxes of the hyperellipse, the left singular vectors $u_1, \ldots, u_m$ are oriented in the directions of the principal semiaxes, while the right singular vectors are the preimages of them $(M \cdot v_j = \sigma_j \cdot u_j)$.

We list some properties that show important connections between SVD and other parts of linear algebra.

Theorem 7. *Let M be an $m \times n$ matrix, $r = min(m, n)$ and $p \leq r$ denote the number of nonzero singular values.*

- *The number of nonzero singular values is equal to the rank of M.*
- *$range(M) = \langle u_1, \ldots, u_p \rangle$ and $null(M) = \langle u_{p+1}, \ldots, u_n \rangle$*
- *$\|M\|_2 = \sigma_1$ and $\|M\|_F = \sqrt{\sigma_1^2 + \cdots + \sigma_p^2}$*
- *If $M = M^t$, then the singular values of M are the absolute values of the eigenvalues of M.*

- *The nonzero singular values of M are the square roots of the nonzero eigenvalues of MM^t or M^tM.*

Proof. For the proof of the above statements we refer to Trefethen and Bau (1997). □

SVD is known as the most accurate method for computing an orthonormal basis of a range or a null space. The numerical stability of the singular value decomposition makes it an important ingredient in robust algorithms in numerous problems.

3.2.2 Conditioning via stability

Real problems many times are not exact, but the coefficients are given in floating point numbers. Working with floating point numbers or using numerical algorithms, methods for solving mathematical problems we have to be aware the fact, that we only can compute approximations to the true solution. Errors can be introduced into a problem in several ways. Besides mathematical truncation and rounding errors, which appear most often in numerical problems, errors can be introduced by using not sufficiently accurate mathematical models, working with physical data that contains error or uncertainty. Hence, it is important to bound or estimate the resulting error in the solution.

The the terms conditioning and stability refers to the perturbation behavior of a mathematical problem and an algorithm used to solve the problem respectively.

A mathematical problem can be viewed as a function from the vector space of (input) data to the vector space of solutions. Typically, the behavior of the problem at a particular input is considered.

A problem can be *well-conditioned* or *ill-conditioned*. We call the problem well-conditioned if all small perturbation of the input lead to only small changes in the solution. If there are some small perturbations of the input that lead to a big change in the solution, the problem is called ill-conditioned.

A constant called *condition number* can be introduced to measure the sensitivity of the solution to small changes in the input data. Small condition number indicates well-conditioned, while large condition number reflects ill-conditioned problems. The condition number is closely related to the maximum accuracy in the solution. It attempts to measure the worst effect on the solution when the input data is perturbed by a small amount.

An algorithm to solve a given mathematical problem can also be viewed as a map from the vector space of (input) data to the vector space of solutions. An implementation of the algorithm produces for a given input data an output that belong to the vector space of solutions of the considered problem.

If the algorithm is good it might be expected that for a given input it computes an output such that the error (in the relative sense) is on the order of machine epsilon. Then we would say that the algorithm for the given problem is *accurate*. However, rounding is unavoidable in the input data, and this perturbation might lead to significant change in the output. Hence, it is better to peg at stability instead of accuracy.

An algorithm can be *stable* or *unstable*. A stable algorithm gives nearly the right answer to nearly the right question. If we consider a mathematical problem and and algorithm that solves this problem, we can say the following. The algorithm is stable if the sensitivity of the output to the input data is not bigger than in the mathematical problem.

It is possible to introduce a measure of stability called stability constant to measure the maximum stability that can be attained when using floating point numbers and computer arithmetic. The stability constant for the best possible algorithm is the condition number of the problem itself: no matter which algorithm we use, we can not say anything more on the accuracy of the output than described by this condition number.

We would like to point out that for problems and related algorithms defined over finite dimensional vector spaces the properties of accuracy, stability are independent of the choice of norm (i.e. they all hold or fail to hold).

3.3 The condition number

Given a parametrization $\mathbf{p}$ and a constant m (the estimation of the implicit degree) we define a constant, called condition number, for the implicitization problem, and we show how to compute it. We work in the projective setting over the real numbers.

3.3.1 Settings

In the sequel we shall use three vector spaces $\mathscr{P}$, $\mathscr{I}$, and $\mathscr{R}$ of polynomials, where the letters stand for parametrization, implicitization, and residuals respectively.

Let $n, m \in \mathbb{Z}^+$. Let $\mathscr{P}$ be the set of d–tuples of homogenous polynomials of degree n in the variables $t^1, \ldots, t^{d-1}$ over $\mathbb{R}$. The elements of $\mathscr{P}$ define rational parametric hypersurfaces of degree n in homogenous coordinates.

Let $\mathscr{I}$ be the set of all homogenous polynomials of degree m in the variables $x_1, \ldots, x_d$ over $\mathbb{R}$. The elements of $\mathscr{I}$ serve to represent a (possibly approximate) implicitization of a given hypersurface.

We denote by $\mathscr{R}$ the set of homogenous polynomials of degree nm in the variables $t^1, \ldots, t^{d-1}$ over $\mathbb{R}$.

In order to simplify the notation, the elements of the above vector spaces will be identified with their coefficient vectors with respect to the chosen basis. (We use either monomial or Bernstein basis.)

The sets $\mathscr{P}, \mathscr{I}, \mathscr{R}$ are linear spaces with finite dimension and can be identified with $\mathbb{R}^k$, where k is the corresponding dimension. The usual inner product and the associated norm in $\mathbb{R}^k$ defines an inner product and a norm in $\mathscr{P}$, $\mathscr{I}$, and $\mathscr{R}$ respectively. The inner product and the induced norm are denoted by $\langle\cdot\rangle_{\mathscr{V}}$ and $\|\cdot\|_{\mathscr{V}}$ respectively, where $\mathscr{V}$ is one of $\mathscr{P}, \mathscr{I}, \mathscr{R}$. If no confusion arise we omit the subscript. Both the inner product and the norm depend on the choice of a basis.

Finally we define an evaluation map eval: $\mathscr{I} \times \mathscr{P} \to \mathscr{R}$ by

$$(F, \mathbf{p}) \mapsto \operatorname{eval}(F, \mathbf{p}) = F(\mathbf{p}).$$

Note that the evaluation map is linear in its first argument, but not linear in the second one.

3.3.2 Definition-Special case

Assume that $\mathbf{p} \in \mathscr{P}$, $\|\mathbf{p}\|_{\mathscr{P}} = 1$ is a parametrization and $F \in \mathscr{I}$, $\|F\|_{\mathscr{I}} = 1$ is the implicit equation of the same surface. In particular we assume that $\mathbf{p}$ has an exact implicitization of degree m. Then F is the normed solution H of $\operatorname{eval}(H, \mathbf{p}) = 0$, which is unique up to multiplication by -1.

Let

$$F^{\perp} := \{J \in \mathscr{I} \mid \langle F, J\rangle_{\mathscr{I}} = 0\}.$$

(Recall, that $F^{\perp}$ depends on the chosen basis.) Then $\operatorname{eval}(J, \mathbf{p})$ is a nonzero vector for all $J \in F^{\perp}$. The following amount:

$$\kappa := \min_{J \in F^{\perp}, \|J\|_{\mathscr{I}} = 1} \|\operatorname{eval}(J, \mathbf{p})\|_{\mathscr{V}}$$

is a numerical measurement of the uniqueness of the implicitization problem. If $\kappa = 0$, then there are several linearly independent equations H with $\mathrm{eval}(H, \mathbf{p}) = 0$. If κ is small, we are close to such a case. The condition number is defined as:

$$K := 1/\kappa,$$

in the case where we have a parametrization $\mathbf{p}$ and the implicit equation F of the same surface.

If $\|\mathbf{p}\| \neq 1$, then the condition number is always the condition number of the normed equation .

3.3.3 General definition and computation of the condition number

For any $F \in \mathscr{I}$, $\mathbf{p} \in \mathscr{P}$ we can write

$$\mathrm{eval}(F, \mathbf{p}) = M_{\mathbf{p}} \cdot F$$

where $M_{\mathbf{p}}$ is a matrix of size $\bar{m} \times \bar{n}$ depending on $\mathbf{p}$, where $\bar{m} = \binom{mn+d-2}{mn}$, $\bar{n} = \binom{m+d-1}{m}$. Using singular value decomposition $M_{\mathbf{p}}$ can be factorized as

$$U \cdot \Sigma \cdot V^t$$

where $\Sigma \in \mathbb{R}^{\bar{m} \times \bar{n}}$ is diagonal, $U \in \mathbb{R}^{\bar{m} \times \bar{m}}$, $V \in \mathbb{R}^{\bar{n} \times \bar{n}}$ are orthogonal matrices. We have the following:

Proposition 1. *If $\boldsymbol{p} \in \mathscr{P}$, $\|\boldsymbol{p}\| = 1$ is a parametrization and $F \in \mathscr{I}, \|F\| = 1$ is the implicit equation of the same surface, then the following are true:*

- *The smallest singular value of $M_{\boldsymbol{p}}$ is zero.*
- *The right singular vector belonging to the smallest singular value is F.*
- *$F^{\perp}$ is spanned by the first $\bar{n} - 1$ right singular vector.*
- *The right singular vector belonging to the second smallest singular value minimizes the function $H \mapsto \mathrm{eval}(H, \boldsymbol{p})$ in the unit sphere of $F^{\perp}$.*
- *The second smallest singular value is κ.*

For an arbitrary nonzero vector $\mathbf{p} \in \mathscr{P}$, $\|\mathbf{p}\| = 1$ we define the condition number K as the reciprocal of the *formally second smallest* singular value of $M_{\mathbf{p}}$. With "formally second smallest singular value," we mean that we take multiplicities into account. For instance, if 0 is a multiple singular value, then the condition number is infinity. Note, that the condition number K depends not just on $\mathbf{p}$, but also on the integer m.

Remark 3. To compute the condition number of an implicitization problem, the implicit equation of the parametrically given surface does not need to be computed. Computation of the formally second smallest singular value is easier than the computation of the implicit equation, at least numerically, because the singular value is numerically stable, whereas the implicitization problem can be very badly conditioned. The last step in the condition number computation, taking inverse, can also be very bad conditioned, when the singular value is small.

The condition number can be computed using the following algorithm:

Algorithm 1 (Condition Number).
Input: A d-tuple of polynomials $\mathbf{p} = (p_0, \ldots, p_{d-1})$ of total degree n in the parameters, such that $\|\mathbf{p}\| = 1$, and an $m \in \mathbb{Z}^+$.
Output: Condition number of the implicitization problem.

1. Initialize $M_{\mathbf{p}}$ by an empty matrix.
   For each b_i in the basis $B_{\mathscr{I}}$ of $\mathscr{I}$, $i = 1, \ldots, \bar{n}$
   (a) substitute $\mathbf{p}$ into b_i,
   (b) expand the result in the basis $B_{\mathscr{R}}$ of $\mathscr{R}$
   (c) append the column to $M_{\mathbf{p}}$

   (Now we constructed the matrix $M_{\mathbf{p}}$)
2. Compute the singular value decomposition of the matrix $M_{\mathbf{p}}$.
3. $1/\sigma_{\bar{n}-1}$ is the condition number, where $\bar{n} = \binom{m+d}{m}$

3.3.4 Examples

As we originally started this research with the error analysis of a particular implicitization method for cubic surfaces, and we already had experimented with some implicitization tools (Berry and Patterson, 2001; Dokken et al., 2001; Zheng et al., 2003) for this class of surfaces, we decided to take cubic surfaces as test examples.

In the following examples we compute the implicit condition number of a cubic surface given in parametric form. The computation is done using monomial basis.

Example 2. Given a quadruple of cubics through the points as in the table below. These *base points* determine the cubics up to a linear change of coordinates. (We do not write out the cubic polynomials here because of space reason.) It is well-known that four cubics through six base points parameterize in general a cubic surface (Sederberg, 1990b). Hence, we have $n = m = 3$.

base points:	$(1, 0, 0), (0, 1, 0), (0, 0, 1), (1, 1, 1), (1, 2, 3), (2, -1, 1)$
$\kappa = \sigma_{19}$:	$0.29202 \cdot 10^{-1}$
condition number:	34.24411

Let $b_1, \ldots, b_{\bar{n}}$ denote the basis of $\mathscr{I}$, and $\bar{b}_1, \ldots, \bar{b}_{\bar{m}}$ the basis of $\mathscr{R}$, where $\bar{n} = 20, \bar{m} = 55$. Furthermore let $p_1, \ldots, p_4$ denote the cubics through the base points mentioned above. To compute the i -th element of the j -th column of $M_{\mathbf{p}}$, substitute $p_1, \ldots, p_4$ into b_j, and take the coefficient with respect to $\bar{b}_i$. In the singular value decomposition of $M_{\mathbf{p}}$ we get the following singular values:

$$
\begin{gathered}
0.33374, 0.30854, 0.29552, 0.25145, 0.23386, 0.21704, 0.17216 \\
0.16895, 0.14591, 0.12698, 0.11438, 0.10388, 0.091498, 0.077036 \\
0.055713, 0.046582, 0.039372, 0.035224, 0.028083, 0.21487 \cdot 10^{-14}
\end{gathered}
$$

The smallest but one singular value gives κ, its inverse 34.24411 provides the condition number, which is small for the class of cubic surfaces. (The last singular value is zero, the result above is due to numerical errors in the singular value decomposition.)

Example 3. Given again a quadruple of cubic polynomials through the base points given in the table below.

base points:	$(37, 25, 34), (60, -27, -47), (85, 54, 61),$ $(68, -40, 9), (-4, 51, 23), (-63, -91, -70)$
$\kappa = \sigma_{19}$:	$0.266702 \cdot 10^{-4}$
condition number:	37494.96829

In the singular value decomposition of $M_{\mathbf{p}}$ we get the following singular values:

$$0.40384, 0.38722, 0.32744, 0.294571, 0.25819, 0.24115, 0.20737$$
$$0.200387, 0.17805, 0.17468, 0.14380, 0.13577, 0.12065, 0.11013, 0.10673$$
$$0.0878083, 0.49430 \cdot 10^{-4}, 0.44513 \cdot 10^{-4}, 0.26670 \cdot 10^{-4}, 0.40140 \cdot 10^{-14}$$

The condition number 37494.96829 is the inverse of σ_{19}. This is quite high for the case of cubics, the implicitization problem is not well-conditioned for this surface. There are small perturbations in the input that result in a big change in the implicit equation.

3.4 Error analysis based on K

The condition number can be used to give an upper bound for the error in the computed implicit equation. We show, if there are two parametrizations close to each other, then the difference between the computed implicit equations can be estimated by the condition number. The following theorem deals with the surface case.

Theorem 8. *Let $\boldsymbol{p}_1$ be a quadruple of homogenous polynomials of parametric degree n in t^1, t^2, t^3, with $\|\boldsymbol{p}_1\| = 1$. Let $F_1 \in I$ be a homogenous polynomial of degree m in the variables $x_1, \ldots, x_4$ with $\|F_1\| = 1$, such that $\|\operatorname{eval}(F_1, \boldsymbol{p}_1)\| \leq \epsilon_1$. Then for any parametrization $\boldsymbol{p}_2$ and implicitization F_2, where $\|\boldsymbol{p}_2\| = 1$, $\|F_2\| = 1$ and $\|\operatorname{eval}(F_2, \boldsymbol{p}_2)\| \leq \epsilon_1$, with $\|\boldsymbol{p}_1 - \boldsymbol{p}_2\| \leq \epsilon_2$, we have one of the following*

$$\|F_1 - F_2\| \leq K \cdot c_{m,n} \cdot \max\{\epsilon_1, \epsilon_2\}$$
$$\|F_1 + F_2\| \leq K \cdot c_{m,n} \cdot \max\{\epsilon_1, \epsilon_2\}$$

where K is the condition number of $\boldsymbol{p}_1$ and $c_{m,n}$ is a constant.

Proof. Let F_3 be such that $\|F_3\| = 1$ and $\|\operatorname{eval}(F_3, \mathbf{p}_1)\|$ is minimal. It follows that $\|\operatorname{eval}(F_3, \mathbf{p}_1)\| \leq \|\operatorname{eval}(F_1, \mathbf{p}_1)\| \leq \epsilon_1$

Let

$$R_1 := F_1 - \lambda_1 F_3,$$
$$R_2 := \lambda_2 F_3 - F_2,$$

where $\lambda_1 := \langle F_3, F_1\rangle, \lambda_2 := \langle F_3, F_2\rangle$. Then $R_1, R_2 \in F_3^{\perp}$.

From the definition of κ we have

$$\|\operatorname{eval}(R_i, \mathbf{p}_1)\| \geq \kappa \cdot \|R_i\|$$

$$\|R_i\| \leq \| \operatorname{eval}(R_i, \mathbf{p}_1)\|/\kappa \tag{3.1}$$

for $i = 1, 2$.

To estimate $\|R_1\|$ we write:

$$\begin{aligned} \|R_1\| &= K \cdot \| \operatorname{eval}(R_1, \mathbf{p}_1)\| = K \cdot \| \operatorname{eval}(F_1 - \lambda_1 F_3, \mathbf{p}_1)\| \\ &\leq K \cdot (\| \operatorname{eval}(F_1, \mathbf{p}_1)\| + \| \operatorname{eval}(\lambda_1 F_3, \mathbf{p}_1)\|) \\ &\leq K \cdot (1 + \lambda_1)\epsilon_1, \end{aligned} \tag{3.2}$$

Let $\mu\colon \mathbb{R}^{4\frac{(n+1)(n+2)}{2}} \to \mathbb{R}^{\bar{n}\times\bar{m}}$, $\mathbf{p} \mapsto M_{\mathbf{p}}$. The map μ is differentiable.

$$\begin{aligned} \| \operatorname{eval}(F_2, \mathbf{p}_1) - \operatorname{eval}(F_2, \mathbf{p}_2)\| &\leq \|M_{\mathbf{p}_1} - M_{\mathbf{p}_2}\| \cdot \|F_2\| \\ &= \|M_{\mathbf{p}_1} - M_{\mathbf{p}_2}\| \\ &= \|\mu(\mathbf{p}_1) - \mu(\mathbf{p}_2)\| \\ &\leq \|\operatorname{Jac}(\mu)(\mathbf{p})\| \cdot \|\mathbf{p}_1 - \mathbf{p}_2\| \\ &\leq \bar{c}_{m,n} \cdot \|\mathbf{p}_1 - \mathbf{p}_2\|, \end{aligned}$$

Here $\bar{c}_{m,n} = \max\limits_{\|\mathbf{p}\|=1} \|\operatorname{Jac}(\mu)(\mathbf{p})\|$

To estimate $\|R_2\|$ we write:

$$\begin{aligned} \|R_2\| &= K \cdot \| \operatorname{eval}(R_2, \mathbf{p}_1)\| = K \cdot \| \operatorname{eval}(\lambda_2 F_3 - F_2, \mathbf{p}_1)\| \\ &\leq K \cdot (\| \operatorname{eval}(\lambda_2 F_3, \mathbf{p}_1)\| + \| \operatorname{eval}(F_2, \mathbf{p}_1)\|) \\ &\leq K \cdot (\lambda_2\epsilon_1 + \| \operatorname{eval}(F_2, \mathbf{p}_1) - \operatorname{eval}(F_2, \mathbf{p}_2)\| + \| \operatorname{eval}(F_2, \mathbf{p}_2)\|) \\ &\leq K \cdot (\lambda_2\epsilon_1 + \bar{c}_{m,n} \cdot \epsilon_2 + \epsilon_1), \end{aligned} \tag{3.3}$$

Let $c_{m,n} = 4(4 + \bar{c}_{m,n})$. We distinguish two cases.

Case1: $\|\lambda_1\|, \|\lambda_2\| \geq 1/2$

Combining (3.1), (3.2), (3.3) we get the following:

$$\begin{aligned} \|F_1 - F_2\| &\leq \|F_1 - F_3\| + \|F_3 - F_2\| \\ &\leq \|R_1\|/\lambda_1 + \|R_2\|/\lambda_2 \\ &\leq \| \operatorname{eval}(R_1, \mathbf{p}_1)\|/\lambda_1\kappa + \| \operatorname{eval}(R_2, \mathbf{p}_1)\|/\lambda_2\kappa \\ &\leq (\lambda_1 + 1)\epsilon_1/\lambda_1\kappa + ((1 + \lambda_2)\epsilon_1 + \bar{c}_{m,n} \cdot \epsilon_2)/\lambda_2\kappa \\ &\leq 2K \cdot ((\lambda_1 + 1)\epsilon_1 + (1 + \lambda_2)\epsilon_1 + \bar{c}_{m,n} \cdot \epsilon_2) \\ &\leq 2K \cdot ((2 + \lambda_1 + \lambda_2) \cdot \max\{\epsilon_1, \epsilon_2\} + \bar{c}_{m,n} \cdot \max\{\epsilon_1, \epsilon_2\}) \\ &= 2K \cdot (2 + \lambda_1 + \lambda_2 + \bar{c}_{m,n}) \cdot \max\{\epsilon_1, \epsilon_2\} \\ &\leq K \cdot 2(4 + \bar{c}_{m,n}) \cdot \max\{\epsilon_1, \epsilon_2\} \\ &\leq K \cdot c_{m,n} \cdot \max\{\epsilon_1, \epsilon_2\} \end{aligned}$$

Case2: One of the λ_i is less than 1/2.

Case2.1: Assume $\lambda_1 < 1/2$.

Then $R_1 > \sqrt{3}/2$. Combining it with (3.1), (3.2) we have:
$\sqrt{3} < \|R_1\| \le 2K(1+\lambda_1)\epsilon_1 \le 4K\epsilon_1$

$$\begin{aligned}\|F_1 - F_2\| &\le 2 \\ &< 2\sqrt{3} \\ &< 4\|R_1\| \\ &< 4K(1+\lambda_1)\epsilon_1 \\ &< 8K\epsilon_1 \\ &< 2K \cdot (4+\bar{c}_{m,n}) \cdot \epsilon_1 \\ &< 2K \cdot (4+\bar{c}_{m,n}) \cdot \max\{\epsilon_1, \epsilon_2\} \\ &< K \cdot c_{m,n} \cdot \max\{\epsilon_1, \epsilon_2\}\end{aligned}$$

Case2.2: Assume $\lambda_2 < 1/2$.

Then $R_2 > \sqrt{3}/2$. Combining it with (3.1), (3.3) we have:
$\sqrt{3} < 2K((1+\lambda_2)\epsilon_1 + \bar{c}_{m,n}\epsilon_2) < 2K(2\epsilon_1 + \bar{c}_{m,n}\epsilon_2)$

$$\begin{aligned}\|F_1 - F_2\| &\le 2 \\ &< 2\sqrt{3} \\ &< 4K(2\epsilon_1 + \bar{c}_{m,n}\epsilon_2) \\ &< 4K(2\epsilon_1 + \bar{c}_{m,n}\epsilon_2) + 8K\epsilon_1 \\ &< 4K \cdot (4+\bar{c}_{m,n}) \cdot \max\{\epsilon_1, \epsilon_2\} \\ &< K \cdot c_{m,n} \cdot \max\{\epsilon_1, \epsilon_2\}\end{aligned}$$

□

Remark 4. The generalization of the theorem for the hypersurface case is straightforward.

Remark 5. The computation of the constant $c_{n,m}$ is cumbersome and very technical, but it can be computed for each parametric degree n and implicit degree m. A rough upper estimate gives $c_{n,m} \le n^2 \cdot (m!)^3$. If $\max\{\epsilon_1, \epsilon_2\}$ is small enough we can use first order approximation and the constant $c_{m,n}$ becomes smaller by a factor of 4.

The theorem above allows to give a stability test of various implicitization techniques. The output error can be computed by applying the technique to a slightly perturbed input. If it is bigger than the upper bound in Theorem 8, then the use of the technique is responsible for the output error, and therefore the stability test rejects the technique for the given input. If it is smaller, than we cannot say anything (hence, this test is only able to reject unstable techniques, but it cannot prove that a certain technique is stable). We should point out that the test does not allow to rank the methods, because the examples below are not statistically significant.

Example 4. We continue Example 2 from the previous section. For $n = m = 3$ we have $c_{3,3} = 1.8$. We compare the numerical stability of the following methods:

Method 1: Implicitization technique due to Berry and Patterson (2001)

Method 2: Moving planes method using Gauss elimination (Zheng et al., 2003)

Method 3: Dokken's method using SVD (Dokken et al., 2001)

The following table shows the introduced input error, the computed error bound, and output error using algorithms based on the listed methods.

input error:	$0.80000 \cdot 10^{-9}$
error bound:	$0.10255 \cdot 10^{-6}$
Method 1:	$0.11371 \cdot 10^{-8}$
Method 2:	$0.42242 \cdot 10^{-8}$
Method 3:	$0.14239 \cdot 10^{-8}$

After introducing some error in the coefficients of the parametric form, the output error using all three methods is smaller than the worst case bound in Theorem 8. Therefore, in this example all three methods are accepted.

Example 5. Given a surface by the following base points:

$$(1, 0, 0), (5, 4, 1), (9, -1, 1), (12, 2, 1), (-4, 5, 1), (-8, -4, 1).$$

Using Theorem 8 we compare the numerical stability of the methods listed in Example 4 for the given surface.

input error:	$0.7071025530 \cdot 10^{-8}$
error bound:	$0.1226753025 \cdot 10^{-3}$
Method 1:	1.9999999999
Method 2:	$0.7770229524 \cdot 10^{-8}$
Method 3:	$0.7418004884 \cdot 10^{-7}$

After introducing some error in the coefficients of the parametric form, the output error using methods Method 2, Method 3 is smaller than the worst case bound in Theorem 8. However, using method Method 1 the output error is above the computed bound. Therefore, in this example, the first method is not accepted.

In the introduction we claimed that the condition number K is not a stability constant of a particular algorithm, but a condition number of the implicitization problem. To justify this statement, we need to show that big output errors do arise when the condition number is big. Here is the precise statement.

Theorem 9. *Let $\boldsymbol{p}_1$ be as in the previous theorem, and $F_1 \in \mathscr{I}$ with $\|F_1\| = 1$, such that $\|\operatorname{eval}(F_1, \boldsymbol{p}_1)\| \leq \epsilon_1$. Then there exists a parametrization $\boldsymbol{p}_2$, $\|\boldsymbol{p}_1 - \boldsymbol{p}_2\| \leq \epsilon_2$, and $F_2 \in \mathscr{I}$ with $\|F_2\| = 1$, such that $\|\operatorname{eval}(F_2, \boldsymbol{p}_2)\| \leq \epsilon_1$, and*

$$\|F_1 - F_2\| \geq \tfrac{1}{2} \cdot \epsilon_1 \cdot K$$

Proof. Let $M_{\mathbf{p}_1}$ the matrix belonging to $\mathbf{p}_1$ in the matrix- vector decomposition. We choose F_1 as the right singular vector belonging to the smallest singular value of the matrix $M_{\mathbf{p}_1}$. Then $\|F_1\| = 1$, and $\|\operatorname{eval}(F_1, \mathbf{p}_1)\| = \sigma_{\bar{m}}$, where $\sigma_{\bar{m}}$ is the smallest singular value of $M_{\mathbf{p}_1}$.

Let $\mathbf{p}_2 := \mathbf{p}_1$. Then we have $\|\mathbf{p}_1 - \mathbf{p}_2\| \leq \epsilon_2$ for any $\epsilon_2 > 0$. (This is the reason why ϵ_2 does not appear in the bound for the output error.) Let $F_2 := F_1 + \delta \cdot v$, where v is the right singular vector corresponding to the smallest but one singular value of the matrix $M_{\mathbf{p}_1}$. We assume that $\epsilon_1 \geq 2 \cdot \sigma_r$, and choose $\delta = \frac{1}{2} \cdot \epsilon_1 \cdot K$.

Then

$$\begin{aligned}\|\operatorname{eval}(F_2,\mathbf{p}_2)\| = \|\operatorname{eval}(F_2,\mathbf{p}_1)\| &= \sqrt{\|\operatorname{eval}(F_1,\mathbf{p}_1)\|^2 + \delta^2\cdot\|\operatorname{eval}(v,\mathbf{p}_1)\|^2} \\ &\leq \sqrt{\tfrac{1}{4}\epsilon_1^2 + \tfrac{1}{4}\cdot\epsilon_1^2\cdot K^2\cdot\kappa^2} \\ &= \sqrt{\epsilon_1^2(\tfrac{1}{4}+\tfrac{1}{4})} \\ &= \epsilon_1\cdot\tfrac{1}{2}\cdot\sqrt{2}\end{aligned}$$

The $\|\operatorname{eval}(F_2,\mathbf{p}_2)\| \leq \epsilon_1$ requirement is fulfilled.

By the choice of δ above we have $F_2 = F_1 + \frac{1}{2}\cdot\epsilon_1\cdot K\cdot v$. From this it follows that

$$\|F_1 - F_2\| \geq \tfrac{1}{2}\cdot\epsilon_1\cdot K$$

□

3.5 Observations

In most of our testing examples the condition number was between 1 and 100, in less examples between 100 and 500, and in some cases over 500. The best conditioned example we have found had condition number 18.137085.

In our test examples the cubic surfaces were given by six base points. Choosing different basis for the linear system passing through the given base points we got different condition numbers. If two basis differed only by an orthogonal linear transformation, the condition numbers only slightly changed.

Changing the basis by a nonorthogonal transformation resulted in a noticeable change in the condition number. It seems that our condition number gets multiplied by a factor which is proportional to the square of the condition number of the nonorthogonal transformation. We demonstrate this behavior in the following example.

Example 6. We use the surface given in Example 2. The computed condition number for the given parametrization is 34.24411.

After applying (random) nonorthogonal transformation to the given parametrization, we compute the condition number of the new parametrization, and the condition number of the transformation. The condition number of the transformation is the condition number of the nonorthogonal matrix used as transformation matrix. Performing SVD the last but one singular value serves as the condition number of the matrix.

Figure 3.1 shows the relation between the ratio of the new and the original condition number and the square of the condition number of the nonorthogonal matrix in 80 test cases. On the figure the logarithm of the computed values are displayed. As we can see, in most of the cases the magnitude of the ratio of the new and the original condition number is of the same order as the square of the condition number of the nonorthogonal transformation.

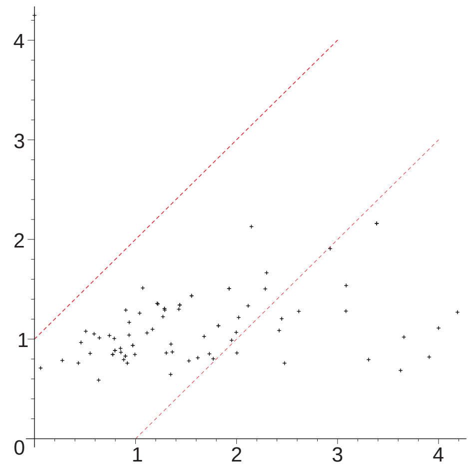


Figure 3.1: Relation between the ratio of the original and the new condition number of the parametrization, and the square of the condition number of the nonorthogonal transformation.

The question how the geometry of the base points effect the condition number seems more difficult. This is a topic of future research. Maybe the methods in Castro et al. (2002) are useful for this investigation.

Due to our observations singularities do not effect the stability of the implicitization, i.e. surfaces with singularities do not have big condition number. To illustrate this behavior, we show an example.

Example 7. In Table 3.1 we show two surfaces. The first one is a singular one, as three of the base points are on a line, the second one is a modified example. One of the base points is slightly moved so that no three points are on a line. As the condition number shows, this problem has nearly the same stability as the previous one. The singular point does not destroy the stability of the problem.

Example 8. In Table 3.2 we show again a singular and a nonsingular surface with the corresponding condition numbers. The first surface is singular as

	singular case	nonsingular case
base points	$(1,1,1),(2,2,1),$ $(3,3,1),(-1,1,1),$ $(-1,-2,1),(2,-2,1)$	$(1,1,1),(2,2.01,1),$ $(3,3,1),(-1,1,1),$ $(-1,-2,1),(2,-2,1)$
$\kappa=\sigma_{19}$:	$0.28113\cdot 10^{-1}$	$0.30226\cdot 10^{-1}$
K:	35.57099	33.08406

Table 3.1: Comparing the condition number of singular and nearly singular surfaces 1.

five of its base points lie on a conic. The two problems have nearly the same condition number. It presents once more that singularities do not badly influence the stability of the implicitization process.

	singular case	nonsingular case
base points	$(0,1,1),(0,-1,1),$ $(2,0,1),(-2,0,1),$ $(1,\sqrt{3}/2,1),(2,3,1)$	$(0,1,1),(0,-1,1),$ $(2.001,0,1),(-2,0,1),$ $(1,\sqrt{3}/2,1),(2,3,1)$
$\kappa=\sigma_{19}$:	$0.25334\cdot 10^{-1}$	$0.25459\cdot 10^{-1}$
K:	39.47275	39.27864

Table 3.2: Comparing the condition number of singular and nearly singular surfaces 2.

The following example shows that being nonsingular is not a sufficient criterion to have a well-conditioned problem.

Example 9. As the base points are generic, i.e. no three are on a line, not all of them lie on a conic,

$$(1,0,0),(5,4,1),(9,-1,1),(12,2,1),(-4,5,1),(-8,-4,1),$$

the surface is nonsingular. One would expect numerical stability, however we get quite big condition number, $K=9638.33951$, i.e. the implicitization problem is not stable in this case. There are small perturbations of the input which lead to big changes in the output. (We have not found any explanation for this behavior yet.)

3.6 Numerical degree guessing

In approximate implicitization methods the desired degree of the implicit equation has to be known in advance. Usually approximate algorithms start with the given parametric representation and as the first step an integer m should be chosen, see Corless et al. (2001), Dokken (2001). Here m is the estimated degree of the implicit equation. It is possible to use some known upper bounds on the predicted degree of the implicit equation.

The condition number K depends on the integer m, which is the estimate of the implicit degree of the surface. In this section we describe a way of obtaining information that allows more accurate guessing of the degree.

Let $\mathbf{p}$ be an element of $\mathscr{P}$. We compute the singular values of the matrix $M_{\mathbf{p},m}$ for $m = 1, 2, \ldots$ until the last singular value is sufficiently small. Then we know that for this value of m there is an equation F such that $\mathrm{eval}(F, \mathbf{p})$ is small (namely the right singular vector belonging to the smallest singular value). We distinguish two cases.

First case: The second smallest singular value is big. In this case the implicitization problem is well-conditioned, and we can say that the equation F above is the solution of the implicitization problem.

Second case: The second smallest singular value is small. The implicitization problem is ill conditioned. There are at least two equations F_1, F_2 for which the residual is small. There are two possible explanations. Either $\mathbf{p}$ is numerically close to a parametrization of a curve (namely the intersection of F_1 and F_2). Or some small perturbations of F_1 and F_2 have a common factor. This common factor F_3 has degree smaller than m, and $\mathrm{eval}(F_3, \mathbf{p})$ is small, therefore this case should have been noticed before.

(a) first case

degree	$\sigma_{\bar{n}}$	$\sigma_{\bar{n}-1}$
1	big	big
2	big	big
$\vdots$	$\vdots$	$\vdots$
$m-1$	big	big
m	small	big
$\vdots$	$\vdots$	$\vdots$

(b) second case

degree	$\sigma_{\bar{n}}$	$\sigma_{\bar{n}-1}$
1	big	big
2	big	big
$\vdots$	$\vdots$	$\vdots$
$m-1$	big	big
m	small	small
$\vdots$	$\vdots$	$\vdots$

Example 10. Let $\mathbf{p}$ be the following quadruple:

$$\begin{aligned}
p_1 =\ & 0.33014\, t^{12} t^2 + 0.11889\, t^{12} t^3 - 0.62851\, t^1 t^2 t^3 + 0.23483\, t^1 t^{22} \\
& - 0.53898\, t^{32} t^1 + 0.33994\, t^{22} t^3 + 0.14368\, t^2 t^{32} \\
p_2 =\ & 0.0091976\, t^{12} t^2 + 0.17598\, t^{12} t^3 + 0.17918\, t^1 t^2 t^3 - 0.58353\, t^1 t^{22} \\
& - 0.090543\, t^{32} t^1 + 0.67456\, t^{22} t^3 - 0.36484\, t^2 t^{32} \\
p_3 =\ & - 0.071139\, t^{12} t^2 + 0.61051\, t^{12} t^3 + 0.22714\, t^1 t^2 t^3 - 0.32573\, t^1 t^{22} \\
& - 0.44603\, t^{32} t^1 - 0.36177\, t^{22} t^3 + 0.36702\, t^2 t^{32} \\
p_4 =\ & 0.39218\, t^{12} t^2 + 0.18953\, t^{12} t^3 - 0.49534\, t^1 t^2 t^3 - 0.47488\, t^1 t^{22} \\
& + 0.53353\, t^{32} t^1 - 0.22121\, t^{22} t^3 + 0.076182\, t^2 t^{32}
\end{aligned}$$

The table below shows the last two singular values computed for the corresponding implicit degree m.

m	$\sigma_{\bar{n}}$		$\sigma_{\bar{n}-1}$	
1	0.5	(big)	0.5	(big)
2	$0.87627 \cdot 10^{-1}$	(big)	0.12182	(big)
3	$0.21898 \cdot 10^{-14}$	(small)	$0.29202 \cdot 10^{-1}$	(big)

Hence, $m = 3$ is the correct degree.

The right singular vector belonging to the smallest singular value gives the implicit surface.

$$\begin{aligned}
F =\ & 0.12037\, x_1^3 + 0.234111\, x_1^2 x_2 - 0.28728\, x_1^2 x_3 + 0.21993\, x_1^2 x_4 - 0.33179\, x_1 x_2^2 \\
& - 0.287875\, x_1 x_2 x_3 + 0.26653\, x_1 x_2 x_4 + 0.16530\, x_1 x_3^2 - 0.04950\, x_1 x_3 x_4 \\
& - 0.12789\, x_1 x_4^2 - 0.028536\, x_2^3 - 0.37570\, x_2^2 x_3 + 0.12459\, x_2^2 x_4 \\
& - 0.20386\, x_2 x_3^2 - 0.36427\, x_2 x_3 x_4 + 0.32183\, x_2 x_4^2 + 0.086380\, x_3^3 \\
& + 0.036834\, x_3^2 x_4 + 0.13637\, x_3 x_4^2 - 0.16410\, x_4^3
\end{aligned}$$

3.7 Implicitization and distance bound

In many situations, especially if the input curve/surface is contaminated by numerical errors, using approximate implicitization may be more appropriate. As a consequence of the use of floating point arithmetic the resulting implicit equation is inflicted with rounding errors.

In the previous sections we studied the effect caused by using an approximate implicitization in the algebraic sense. We gave an upper bound on the distance between two coefficient vectors of two approximate implicitization using the introduced condition number.

However, even if the coefficients can be computed in a numerically stable way, it is not guaranteed that the zero set of the approximate curve/surface lies "near" to the parametrically given one.

The question is: How robust (geometrically) is the resulting implicit representation with respect to small perturbations of its coefficients?

To answer this question, we combine results concerning the numerical stability of the implicitization process with results on the stability of the resulting implicit representation. More precisely, it is shown that for any approximate parametrization of the given curve/surface, the curve/surface obtained by an approximate implicitization with a given precision is contained within a certain perturbation region.

The main result presented in this section is the outcome of the cooperation with M. Aigner and B. Jüttler.

3.7.1 The problem

We address the following problem: Given a parametric representation $\mathbf{p} = \mathbf{p}(u)$, $u \in [0,1]$, of a planar curve segment with domain $[0,1]$. Let $\mathcal{V}(\mathbf{p}) = \mathbf{p}([0,1])$ be the point set defined by the curve. Consider the zero set $\mathcal{C}_f$ of an approximate implicitization f of the parametric curve, where the coefficients of the residuum $f \circ \mathbf{p}$ are bounded by a positive constant ϵ. How close is it to the given curve? As an answer, we derive an upper bound for the Hausdorff distance between $\mathcal{V}(\mathbf{p})$ and $\mathcal{C}_f$. The bound is valid for all approximate implicitizations f, where the residuum can be bounded by ϵ.

Our approach is based on the condition number introduced in Section 3.3.2. It allows to estimate the distance between the two coefficient vectors of two approximate implicitizations. For curves/surfaces with a high condition number, the computation of the coefficients of an approximate implicitization is not numerically stable, no matter which numerical method for implicitization is chosen.

Unfortunately, even if the coefficients can be computed in a numerically stable way, it is not guaranteed that the zero sets of two approximate implicitizations are close in a geometric sense. However, one can estimate the Hausdorff distance between zero sets of an exact and a perturbed equation, using a result by Aigner et al. (2004). This leads to a constant expressing the robustness of an implicit representation.

Combining these two robustness results allows to examine the suitability of a given rational parametric curve/surface for approximate implicitization. A curve could be said to be "*well behaved*", if

(1) the computation of the coefficients of an approximate implicitization is numerically stable, and

(2) the resulting implicit representation is geometrically robust with respect to small perturbations of its coefficients.

3.7.2 Preliminaries

In this section to represent curve segments we use Bézier and triangular Bernstein-Bézier representations. Following the book Prautzsch et al. (2002) we recall the basic notions and definitions. We also give a short motivation for the use of interval arithmetic.

Bernstein-Bézier-representation

The polynomials

$$B_i^n(u) = \binom{n}{i} u^i (1-u)^{n-i}, \quad i = 0, \ldots, n$$

are called *Bernstein polynomials* of degree n.

It can be shown that $n+1$ linearly independent Bernstein polynomials B_i^n form a basis for all polynomials of degree $\leq n$. Therefore, every polynomial curve $\mathbf{p}(u)$ of degree $\leq n$ has a unique *Bézier representation* of degree n

$$\mathbf{p}(u) = \sum_{i=0}^{n} b_i B_i^n(u).$$

The coefficients b_i are called Bézier points or control points. They are the vertices of the Bézier polygon or control polygon of $\mathbf{p}$ over the interval $[0, 1]$.

Let $\Delta := \Delta ABC$ be a triangle in $\mathbb{R}^2$. Then each point $P \in \mathbb{R}^2$ can be uniquely written as $P = uA+vB+wC$, where $u+v+w = 1$. The coefficients u, v, w are called the *barycentric coordinates* off P with respect to the triangle Δ.

Bernstein polynomials of degree n can be described by barycentric coordinates as

$$B^n_{ijk}(u,v,w) = \frac{n!}{i!j!k!}u^i v^j w^k,$$

where $i,j,k \geq 0$ and $i+j+k=n$. Any polynomial F of degree n can be expressed in the Bernstein-Bézier form over Δ as

$$F(x_0,x_1,x_2) = \sum_{i+j+k=n} b_{ijk} B^n_{ijk}(u,v,w),$$

where b_{ijk} are the control points.

Interval arithmetic

Using computers to make computations involving real numbers, we are restricted to use a finite set of floating point numbers, i.e. we approximate real numbers by the floating point number system. In principle we have two choices: either we use one floating point number for approximation or we use two. In the later case the approximated real number belongs to the set of real numbers which is bounded by the two floating point numbers.

Interval arithmetic is an arithmetic defined on sets of intervals. In interval arithmetic whenever an operation on reals is specified, the corresponding operation on their intervals is executed. The results are intervals in which the exact result must lie.

For a detailed introduction and explanation of interval arithmetic see Alefeld and Herzberger (1983).

3.7.3 Settings

Throughout this section, we shall use three spaces $\mathscr{P}$, $\mathscr{I}$, and $\mathscr{R}$ of polynomials, a slightly modified version of the vector spaces that were introduced in Section 3.3.1.

Consider two positive integers n, m, and let $\mathscr{P}$ be the set of *triples of polynomials* of degree less or equal than n in the variable t over $\mathbb{R}$. These polynomials will be described by their coefficient vectors with respect to the Bernstein basis with respect to the parameter domain $[0,1]$.

The elements of $\mathscr{P}$ define rational parametric curves of parametric degree less or equal than n in homogeneous coordinates,

$$\mathbf{p}(t) := (p_0(t), p_1(t), p_2(t)) = \sum_{i=0}^{n} \mathbf{b}_i \binom{n}{i} t^i (1-t)^{n-i}, \quad t \in [0,1], \qquad (3.4)$$

where the corresponding Cartesian coordinates are $(\frac{p_1}{p_0}, \frac{p_2}{p_0})$.

Let $\mathscr{I}$ be the set of all *homogeneous polynomials* of degree m in the variables x_0, x_1, x_2 over $\mathbb{R}$, represented with respect to the usual power basis.

These functions serve to represent a (possibly approximate) implicitization of a given curve $\mathbf{p}$, by an algebraic curve segment $\mathcal{C}$ of order m

$$\mathcal{C} = \{ (x_1, x_2) \,|\, (x_1, x_2) \in \Omega \subset \mathbb{R}^2 \wedge F(1, x_1, x_2) = 0\}. \tag{3.5}$$

Such an algebraic curve is defined in a certain planar domain $\Omega \subset \mathbb{R}^2$ by the zero contour of a bivariate polynomial $F \in \mathscr{I}$ of degree m. This polynomial is given by its homogeneous monomial representation

$$F(x_0, x_1, x_2) := \sum_{\substack{i,j,k \in \mathbb{N} \\ i+j+k=m}} c_{ijk}\, x_0^i x_1^j x_2^k \tag{3.6}$$

with certain coefficients c_{ijk}. Sometimes we will also use the inhomogeneous representation $F(1, x_1, x_2)$. If no confusion can arise, we shall write $F(x_1, x_2)$ instead.

Finally, we denote by $\mathscr{R}$ the set of *polynomials* of degree less or equal than nm in the variable t over $\mathbb{R}$. Again, these polynomials will be described by their coefficient vectors with respect to the Bernstein basis with respect to the parameter domain $[0, 1]$.

Clearly, the sets $\mathscr{P}, \mathscr{I}, \mathscr{R}$ are linear spaces with a finite dimension, which can be identified with $\mathbb{R}^m$, where m is the corresponding dimension. Then, the usual inner product and the associated norm in $\mathbb{R}^m$ defines an inner product and a norm in $\mathscr{P}$, $\mathscr{I}$, and $\mathscr{R}$ respectively. In order to simplify the notation, any element of $\mathscr{R}$ and $\mathscr{I}$ will be identified with its coefficient vector with respect to the corresponding basis.

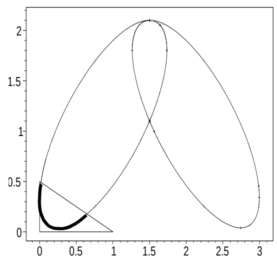
Finally, we define the *evaluation map*

$$\mathrm{eval}\colon \mathscr{I} \times \mathscr{P} \to \mathscr{R} \ \text{ by } \ (f, \mathbf{p}) \mapsto \mathrm{eval}(f, \mathbf{p}) = f \circ \mathbf{p}.$$

3.7.4 Examples

The planar algebraic curves shown below will serve as test examples. We considered segments of four well known curves. First we computed an approximate parametric representation and then an approximative implicitization. Both representation are presented. In addition, the domain triangle has been specified, and it is also shown in the figures. Although the computations

The Tacnode

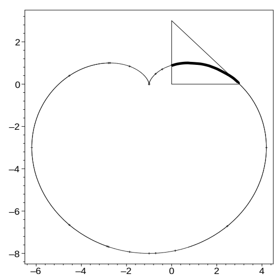


$$\begin{pmatrix} 0.95233\cdot 10^{-2}t^3 + 0.43723t^2 - 0.5478t + 0.89 \\ 0.15\cdot 10^{-1} - 0.1848t + 0.1132\cdot 10^{-1}t^3 + 0.6836t^2 + 0.8523\cdot 10^{-4}t^4 \\ 0.11043\cdot 10^{-3}t^4 + 0.20358\cdot 10^{-1}t^3 + 0.93016t^2 - 1.2295t + 0.429 \end{pmatrix} 10^{-2}$$

$$0.5449 - 0.64792x_1 - 0.25435x_2 + 0.3853x_1^2 - 0.26083x_1x_2 + 0.04559x_2^2$$

$$(0,0), (1,0), (0,0.5)$$

The Cardioid

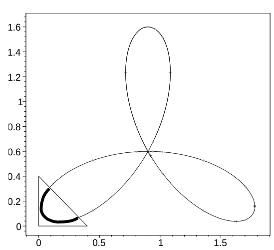


$$\begin{pmatrix} -\frac{23134208}{5}t + 4194304 + \frac{63799808}{25}t^2 - \frac{87973954}{125}t^3 + \frac{15527402881}{160000}t^4 \\ -23134208t + 12582912 + \frac{63799808}{5}t^2 - \frac{263921862}{125}t^3 - \frac{15527402881}{160000}t^4 \\ \frac{46268416}{5}t - \frac{63799808}{5}t^2 + \frac{703791632}{125}t^3 - \frac{15527402881}{20000}t^4 \end{pmatrix}$$

$$\begin{gathered} 0.00775 + 0.37537x_1 + 0.19263x_2 + 0.00762x_1^2 + 0.061820x_1x_2 - 0.41813x_2^2 \\ +0.03663x_1^3 + 0.35689x_1^2x_2 + 0.42381x_1x_2^2 - 0.68875x_2^3 \end{gathered}$$

$$(0,0), (3,0), (0,3)$$

The Trifolium

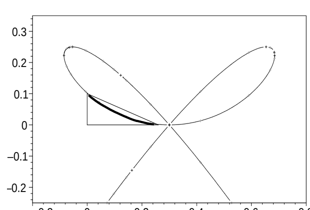


$$\begin{pmatrix} 1.86624t^4 + 3.59424t^3 + 4.58784t^2 - 1.86336t + 0.19629 \\ -1.34784t^2 - 1.08864t + 0.45096 + 3.31776t^4 + 3.31776t^3 \\ 1.1881 + 3.6576t^2 + 1.5696t + 2.0736t^4 + 2.0736t^3; \end{pmatrix}$$

$$\begin{gathered} -0.1034x_1^4 + 0.37226x_1^3 - 0.020681x_1^2x_2^2 - 0.62044x_1^2x_2 \\ -0.39088x_1^2 - 0.37226x_1x_2^2 + 0.11168x_1x_2 + 0.10051x_1 - 0.1034x_2^4 \\ +0.35158x_2^3 - 0.57701x_2^2 + 0.15076x_2 - 0.01312 \end{gathered}$$

$$(0,0), (0.4,0), (0,0.4)$$

The Bicorn



$$\begin{pmatrix} 1 \\ 0.399 + 0.5335t - 0.16637t^3 - 0.0.9075t^2 \\ 0.28435t^2 + 0.1078t - 0.0915t^4 - .0.06655t^3 + 0.0099 \end{pmatrix}$$

$$\begin{gathered} -0.35901 - 0.34195x_1 + 0.33064x_2 - 0.22577x_1^2 + 0.57652x_1x_2 - \\ 0.10127x_2^2 - 0.11083x_1^3 + 0.46583x_1^2x_2 - 0.14776x_1x_2^2 + 0.01032x_2^3 \end{gathered}$$

$$(0,0), (0.25,0), (0,0.1)$$

Table 3.3: Tacnode, cardioid, trifolium, bicorn with their approximative parametric representation, approximative implicit representation and domain triangle.

curve	degree	$\sigma_{\bar{n}-1}$	K
trifolium	4	$0.40345 \cdot 10^{-3}$	2478.62
tacnode	2	0.07907	12.687
cardioid	3	$0.22976 \cdot 10^{-5}$	$0.43523 \cdot 10^{6}$
bicorn	4	$0.50253 \cdot 10^{-8}$	$0.19899 \cdot 10^{9}$

Table 3.4: Condition numbers of the implicitization.

are done in Bernstein-Bézier-representation, the parametric representation is given with respect to the monomial basis, since many coefficients vanish and a basis transformation is relatively simple (Farouki, 1991).

Using Algorithm 1 we compute the condition number of the four curves. Remember, the condition number of the implicitization problem depends not only on the parametric form, but also on the estimate of the degree of the implicit form. From classical algebraic geometry it is known that any degree n polynomial or rational parametric curve can be represented using a degree n algebraic equation. However, if approximate implicitization is used the degree may be lower than n.

Table 3.4 contains the singular value $\sigma_{\bar{n}-1}$, the condition number K and the degree of the implicit representation of each curve. In the case of the tacnode and the trifolium we can say that the implicitization problem is well-conditioned.

3.7.5 Distance between implicit and parametric curves

This Section is dedicated to the computation of the distance between a parametric and an implicitly given curve. For quantifying the distance between the two curves we use the concept of the Hausdorff distance.

Since we use a triangular Bernstein-Bézier representation, the implicit curve is tied to its domain triangle. In the remainder of this paper we assume that this triangle is chosen in such way, that the whole parametric curve is contained in it.

In order to avoid some technical difficulties which may arise if the parametric curve hits the boundary of the triangular domain Δ, we consider the distance between $\mathcal{V}(\mathbf{p})$ and

$$\mathcal{C}^* := \mathcal{C} \cup \partial\Delta. \tag{3.7}$$

More precisely, we consider the distance

$$\mathrm{HD}_\Delta(\mathcal{V}(\mathbf{p}),\mathcal{C}^*) = \sup_{y\in\mathcal{V}(\mathbf{p})\cap\Delta}\ \inf_{x\in\mathcal{C}^*\cap\Delta} \|x-y\|. \tag{3.8}$$

We call this distance the *one-sided Hausdorff distance*[1] of $\mathcal{V}(\mathbf{p})$ and $\mathcal{C}$ with respect to the domain triangle Δ.

Lemma 1 (Aigner, Jüttler). *Let $\mathcal{C} \subset \triangle \subset \mathbb{R}^2$ be an algebraic curve which is defined by the homogeneous polynomial F of degree m and $\mathcal{C}^* := \mathcal{C} \cup \partial\triangle$. We assume that the gradient field $\nabla F(1,.,.)$ of the inhomogeneous polynomial does not vanish in $\triangle$. Furthermore a curve $\mathcal{V}(\mathbf{p}) \subset \triangle$ is given by its parametric representation $\mathbf{p} = \mathbf{p}(t)$, $t \in [0,1]$. If*

$$c \leq ||\ \nabla F(1,.,.)|_{(x_1,x_2)}\ ||_{\mathbb{R}^2}\ \forall\ (x_1,x_2) \in \triangle \quad \textit{and}\ \ ||\,\mathrm{eval}(F,\mathbf{p})|| \leq \epsilon$$

and

$$p_0(t) \geq N \quad \forall t \in [0,1]$$

hold, where c, ϵ and N are certain positive constants, then the one-sided Hausdorff distance can be bounded by

$$\mathrm{HD}_\triangle(\mathcal{V}(\mathbf{p}),\mathcal{C}^*) \leq \frac{\epsilon}{c\,N^m}. \tag{3.9}$$

Proof. The proof is a consequence of the mean value theorem, which is applied to the integral curves of the gradient field emanating from the parametric curve and hitting the implicit curve or the boundary of the triangle. More precisely, we consider the integral curve $\gamma(t) := (x_1(t), x_2(t))$ of the normalized gradient field $V = \nabla\bar{F}/||\nabla\bar{F}||$ of $\bar{F}(x_1,x_2) = F(1,x_1,x_2)$. We choose the starting point $\gamma(0)$ of the integral curve $\gamma(t)$ to lie on the parametric curve $\mathbf{p}$. In the absence of points with a vanishing gradient in the domain of interest, $\gamma(t)$ hits the implicit curve or the boundary of the triangle in $\gamma(s)$ for some $s \in \mathbb{R}$. Since the curve is parameterized by its arc length,

$$\|\gamma(s) - \gamma(0)\| \leq s.$$

Let $f(t)$ be the restriction of $\bar{F}$ to $\gamma(t)$:

$$F(t) := \bar{F}(x_1(t), x_2(t)).$$

Applying the mean value theorem we obtain

$$\exists \xi \in]0,s[: \qquad \frac{f(s)-f(0)}{s-0} = f'(\xi)$$

[1] The symmetric version is $\max(\mathrm{HD}_\Delta(\mathcal{V}(\mathbf{p}),\mathcal{C}^*), \mathrm{HD}_\Delta(\mathcal{C}^*,\mathcal{V}(\mathbf{p})))$.

Due to $f'(t) = \dot\gamma(t) \cdot \nabla \bar F\big|_{\gamma(t)} = \|\nabla \bar F\big|_{\gamma(t)}\|$ we obtain

$$\frac{|f(s) - f(0)|}{|s-0|} = |f'(\xi)| = \|\nabla \bar f\big|_{\gamma(\xi)}\|,$$

hence

$$s = \frac{|f(s)|}{\|\nabla \bar F(\gamma(\xi))\|} = \frac{|\bar F(\gamma(s))|}{\|\nabla \bar F(\gamma(\xi))\|}$$

Finally, we observe that the values

$$F\left(1, \frac{p_1(t)}{p_0(t)}, \frac{p_2(t)}{p_0(t)}\right), \quad t \in [0,1], \tag{3.10}$$

of the values of the inhomogeneous implicit representation along the parametric curve are bounded by ϵ / N^m, which completes the proof. □

The next step is the computation of the constants c and N which are needed in Lemma 1. One can see that for curves that have singular points in the domain of interest, (3.9) is not defined, since the gradient vanishes. Consequently, such cases have to be excluded. More precisely, vanishing gradients correspond to singular points (including isolated points) of the original curve or of other iso-value curves ("algebraic offsets").

In the regular case, a lower bound for the minimal gradient can be computed. The essential ingredient of this algorithm is the convex hull property of Bernstein-Bézier-representations. Clearly, this algorithm gives only a moderate lower bound on $||\nabla F(1,.,.)||$. The result can be made more accurate by splitting the domain triangle into smaller ones.

Algorithm 2 (Minimal Gradient reg).
Input: control points c_{ijk} of a bivariate polynomial.
Output: lower bound for the minimal gradient

1. Compute the partial derivatives of $F(1, x_1, x_2)$ with respect to x_1 and x_2.
2. Describe them in Bernstein-Bézier form with respect to the domain triangle.
3. Combine the corresponding coefficients of the derivative patches together to vector-valued control points $d_{ijk} \in \mathbb{R}^2$
4. Compute the minimal distance from the origin to the convex hull of the d_{ijk}, see Figure 3.2.

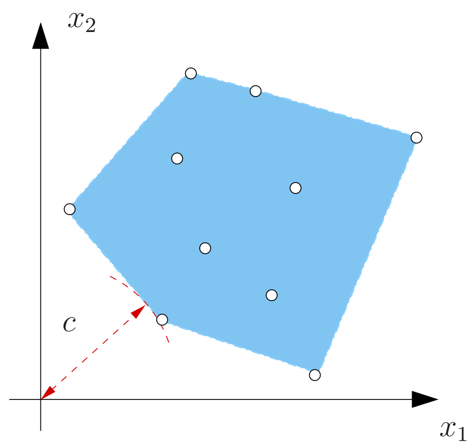


Figure 3.2: Bounding the minimal norm of the gradient.

5. This distance serves as constant c in Lemma 1.

Remark 6. The procedure can be generalized to the surface case. While the algorithms for the three-dimensional convex hull computations are more involved and need special data structures for the storage of the data points (Goodman and O'Rourke, 2004), the time complexity is still the same as in the planar case.

Remark 7. The constant N can be chosen as the minimum value of the 0th components of the control points in (3.4), provided that all of them are positive. More precisely, the so-called "weights" of the rational curve have to be positive, and N can be chosen as the minimum weight.

3.7.6 Distance bound for approximate implicitization

Based on the previous result, we derive an upper bound on the one-sided Hausdorff distance between the parametric curve and any approximate implicitization which has a certain accuracy.

Estimating the algebraic error

In order to estimate the Hausdorff distance between a parametric and an approximate implicit curve we use Theorem 8 that allows to estimate the error in the coefficient vector in terms of the condition number K. The following Lemma is a special case of Theorem 8.

Lemma 2. *Let* $\mathbf{p}$ *be a triple of polynomials of parametric degree* n*, with* $\|\mathbf{p}\| = 1$*. Furthermore, let* $F_1, F_2 \in I$ *be polynomials of degree* m *with* $\|F_1\| = \|F_2\| = 1$ *such that* $\|\operatorname{eval}(F_1, \mathbf{p})\| \leq \epsilon$ *and* $\|\operatorname{eval}(F_2, \mathbf{p})\| \leq \epsilon$*. Then we have one of the following:*

$$\|F_1 - F_2\| \leq K \cdot 4 \cdot \epsilon$$
$$\|F_1 + F_2\| \leq K \cdot 4 \cdot \epsilon,$$

where K *is the condition number of* $\mathbf{p}$*.*

Proof. The complete proof (of a more general result) is given in Section 3.4. Here we restrict ourselves to a sketch of the proof.

Let F_3 be such that $\|F_3\| = 1$ and $\|\operatorname{eval}(F_3, \mathbf{p})\|$ is minimal. It follows that $\|\operatorname{eval}(F_3, \mathbf{p})\| \leq \|\operatorname{eval}(F_1, \mathbf{p})\| \leq \epsilon$

In first order approximation we have, that

$$r_1 := F_1 - F_3, \quad \text{and} \quad r_2 := F_3 - F_2,$$

are in $F_3^{\perp}$. In order to estimate $\|r_i\|$, $i = 1, 2$, we get

$$\begin{aligned}\|r_1\| &= K \cdot \|\operatorname{eval}(r_1, \mathbf{p})\| = K \cdot \|\operatorname{eval}(F_1 - F_3, \mathbf{p})\| \\ &\leq K \cdot (\|\operatorname{eval}(F_1, \mathbf{p})\| + \|\operatorname{eval}(F_3, \mathbf{p})\|) \\ &\leq K \cdot 2 \cdot \epsilon,\end{aligned}$$

Similarly,

$$\|r_2\| = K \cdot \|\operatorname{eval}(r_2, \mathbf{p})\| \leq K \cdot 2 \cdot \epsilon.$$

It follows, that

$$\|F_1 - F_2\| \leq \|F_1 - F_3\| + \|F_3 - F_2\| \leq 4 \cdot K \cdot \epsilon.$$

□

Bounding the minimal gradient

From the previous Section we can conclude that the output of the implicitization process is no longer an exact polynomial, but that its coefficients can only be specified up to a certain tolerance. This means that the possible outputs of the implicitization process form a whole set of curves.

We denote the set of defining polynomials of the implicitized curves by

$$F_\epsilon := \{F \mid \|\operatorname{eval}(F, (\mathbf{p})\| \leq \epsilon, \|F\| = 1\}.$$

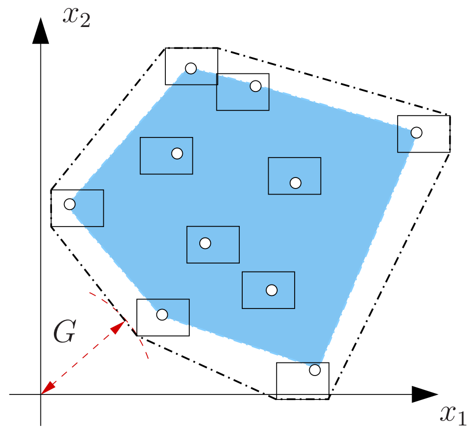


Figure 3.3: Bounding the minimal norm of the gradient of a perturbed polynomial. The rectangles represent the areas where all possible control points lie in.

The corresponding coefficients of all $F \in F_\epsilon$ lie in certain intervals. The length of these intervals can be bounded using Lemma 2.

On the other hand, Lemma 1 allows us to bound the Hausdorff distance between a parametric and an algebraic curve. In order to get a distance bound between a parametric and an arbitrary $F \in F_\epsilon$ we need to compute a lower bound for the minimal gradient for all possible $F \in F_\epsilon$. This bound is given by

$$G := \min_{F \in F_\epsilon} \min_{(x_1,x_2)\in\Omega} \| \nabla F(1,.,.)|_{(x_1,x_2)} \|.$$

In order to compute this bound the same technique as in Section 3.7.5 can be applied. One has to replace the exact control points by intervals. Hence, standard techniques interval arithmetics have to be applied for computing the d_{ijk}. These are no longer points in $\mathbb{R}^2$, but rectangles containing all possible positions of the control points. Consequently, one has to compute the convex hull of these rectangles in order to gain a lower bound for the minimal gradient, cf. Figure 3.3. For further information on interval arithmetics and related techniques we refer to Section 3.7.2 and Shou et al. (2003).

Here is an algorithm to compute a lower bound for the minimal gradient for all possible $F \in F_\epsilon$:

Algorithm 3 (Minimal Gradient).
Input: parametric representation $\mathbf{p}$ of a curve, $\epsilon > 0$
Output: G

1. Compute K using Algorithm "Condition Number".
2. Compute the bound given in Lemma 2, and an approximate implicitization F for $\mathbf{p}$.
3. Generate intervals using the coefficients of F as center and adding/ subtracting the bound derived in the previous step.
4. Determine the derivative patches of $F(1, x_1, x_2)$ in x_1 and x_2 direction and describe them in Bernstein-Bézier (with interval coefficients !) form with respect to the domain triangle.
5. For all pairs of corresponding coefficients of the derivative patches, generate the Cartesian product of the intervals.
6. Collect the vertices of all these rectangles and determine their convex hull.
7. The shortest distance from the origin to this convex hull serves as G. (If the convex hull contains the origin, then $G = 0$)

Perturbation regions of parametric curves

In this Section we combine the results of the previous parts and determine an upper bound for the Hausdorff distance between an exact parametric and approximatively computed implicit curve.

Theorem 10. *Let $\mathbf{p}$ with $\|\mathbf{p}\|$ be a triple of polynomials of degree less or equal than n, and let $\mathcal{V}(\mathbf{p}) \subset \Delta$ be the curve defined by $\mathbf{p}$. If the bound G computed by Algorithm 3 is nonzero, then for any algebraic curve segment $\mathcal{C} \subset \Delta$ defined by an $F \in F_\epsilon$ of degree m,*

$$\mathrm{HD}_\Delta(\mathcal{V}(\mathbf{p}), \mathcal{C}^*) \leq \frac{\epsilon}{GN^m},$$

where the constant N is defined as in Lemma 1.

Proof. The proof is an immediate consequence of the previous two lemmata. □

The bound given in Theorem 10 defines two offset curves to $\mathbf{p}$ that enclose a perturbation region within the triangle. For any approximate parametrization $\mathbf{p}$ of a curve and for any approximate implicitization with a given precision ϵ the obtained curve $\mathcal{C}$ lies within this perturbation region.

Clearly, the result of this Theorem is only meaningful for points that are further away from the boundary than $\frac{\epsilon}{GN^m}$. This is due to the fact, that we do not only measure the distance to the implicitly defined curve, but also to the boundary of the triangle, see Figure 3.4.

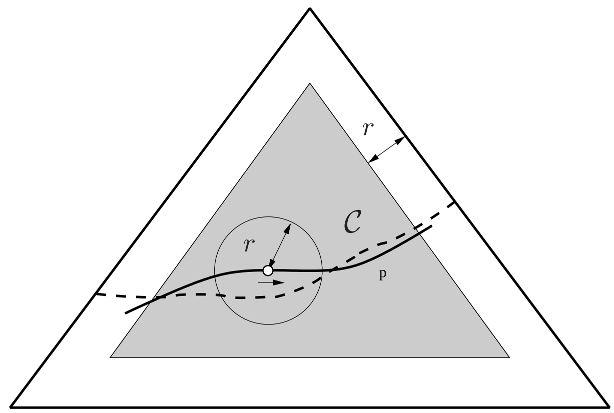


Figure 3.4: For each point that lies on **p** and in the grey shaded triangle exists within a certain bound $r := \epsilon/(GN^m)$ a point on the implicit curve $\mathcal{C}$.

Example 11. In the Table 3.5 we determine for each of the four examples the bound provided in Theorem 10. Using Algorithm 3 we compute a lower bound G for the minimal gradient. In the examples we set $\epsilon = 10^{-6}$.

curve	K	G	position err.
cardioid	$0.43523 \cdot 10^6$	0	∞
bicorn	$0.19899 \cdot 10^9$	0	∞
trifolium	2478.62	0	∞
tacnode	12.687	0.17537	$0.98718 \cdot 10^{-5}$

Table 3.5: Geometric robustness and position bound.

In the first two cases the high condition number K reflects the fact that the implicitization process is very unstable. Consequently, the coefficient bounds are very poor and the bound for the minimal gradient yields zero. No prediction for the position error of the obtained implicit curve can be made.

For the trifolium the implicitization is robust but the obtained implicit representation is unstable under some error in the coefficients. Again the geometric robustness is poor and the position bound is infinity.

In the last example the implicitization as well as the obtained implicit representation are stable under numerical perturbations; the geometrical robustness is good. Knowing the precision of the implicitization process we are able to predict the maximal displacement of the implicit curve.

As we can see, combining results on the robustness of approximate implicitizations (with respect to the resulting coefficients) with results on geometric robustness of implicit representations, the geometric stability of approximate implicitization can be predicted. For a given parametric representation we can compute a bound, such that any approximate implicitization lies within a certain neighborhood of the original parametric curve. The derived bound determines the width of this vicinity.

Note, that the accuracy of the bound given by Theorem 10 mainly depends on the computed minimal gradient. Algorithm 3 gives only a lower bound for the minimal gradient. Hence, it may happen that even if the gradient does not vanish in the region of interest the computed lower bound is zero. The result can be made more accurate by splitting the domain triangle into smaller ones.

Chapter 4

Local Parametrization

Several algebraic techniques for parameterizing a rational algebraic surface as a whole exist. However, in many applications in geometric modelling and related areas, it suffices to parameterize a small portion of the surface. In contrast to the classical problem, we will refer to this as the problem of *local parametrization*: find a parametrization of a small neighborhood of a given point of the surface.

In the sequel we introduce several techniques for generating such parameterizations for nonsingular cubic surfaces. The method works without analyzing the system of lines on the cubic surface. It produces rational maps defined in some neighborhood of the origin in the plane with the property, that the image is an open subset of a given nonsingular cubic surface containing a given point. It is also shown that for nonsingular cubic surfaces the local parametrization problem can be solved for all points, and any such surface can be covered completely.

4.1 The problem

Generating a rational parametric representation of rational algebraic surfaces is called *parametrization*. The parametrization problem is not always solvable (see Section 2.4). However, in general cubic surfaces admit both parametric and implicit representations. There are numerous available parametrization techniques as it was described in Section 2.4.1. In most cases, the existing parametrization methods produce a birational map. Many methods use the 27 lines on a nonsingular cubic surface for parametrization.

In certain applications it suffices to have a parametrization defined in some open subset in the parameter space that covers the intersection of the surface with a certain region of interest. In contrast to the classical problem, we will refer to this as the problem of *local parametrization*: find a parametrization of a small neighborhood of a given point of the surface. For a precise definition of local parametrization see Section 4.3.1.

We use three local parametrization techniques for cubic surfaces, which are called the 2-curve technique, the repeater technique, and the reflection technique. The first two techniques can be traced back to Manin (1986) and Abhyanker and Bajaj (1987). They are based on the classical theory of rational curves on cubic surfaces. Such curves may be generated as the intersection of the surface with the tangent plane at a generic surface point.

We give a complete geometrical analysis of the introduced techniques for nonsingular cubic surfaces, and we show that each of the three algorithms computes a local parametrization for a given nonsingular cubic surface S, and a surface point P. The computed parametrization is improper. Clearly, properness cannot be expected, since the so-called F_5 surface has no proper parametrization (Schicho, 1998a). As a potential advantage, this parametrization is found without analyzing the type of the cubic surface, i.e., without discussing the system of the 27 lines.

4.2 Exact via numeric computation

If the coefficients of the given surface are rational numbers, and all the computation is performed exactly, then the coefficients of the computed local parametrization are real algebraic numbers. These have to be represented somehow, for instance using Thom's code (Coste and Roy, 1988). Although exact geometric computation produces exact results, it has its limitations. For computations, where the result of an arithmetic operation is used in another arithmetic operation as an operand many times in a row, the use of exact geometric computation is less appropriate.

In CAGD applications one prefers to work with floating point numbers, taking into account the existence of small numerical errors. In particular the reference point P on the cubic surface S, which is part of the input, is probably not know exactly as a real algebraic number, but approximately as a finite floating point expansion. The output is also required to be in term of floating point numbers.

This imposes a question: *Is there a justification to apply an exact algorithm* *to floating point data?* In general, there is not, and such applications may lead to all kinds of failures, i.e. incorrect results, program crashing (see Schirra (2000) for an extended discussion).

Methodologically, we translate a mathematical/exact solution to an engineering/approximate solution, and something can get wrong in the process of the translation. A geometric problem defines a mapping between models, which are real mathematical models. On the other hand a computer program gives a mapping between representations of mathematical objects. In the ideal case there is a one-to-one correspondence between representations and models. In practise, i.e. using real algebraic numbers as models and floating point numbers on the computers, the correspondence is not one-to-one. Therefore, instead of correctness of the algorithm robustness can be expected. We call an algorithm for a geometric problem robust if for every computer representation $x_{rep} \in \mathcal{I}_{rep}$ there is a corresponding model $x \in \mathcal{I}$ such that the solution for x is in $\mathcal{O}$ corresponding to the computed output in $\mathcal{O}_{rep}$, see Figure 4.1.

$$\begin{tikzcd}
\mathcal{I} \arrow[r, "geom.prob."] & \mathcal{O} \arrow[d] \\
\mathcal{I}_{rep} \arrow[u] \arrow[r, "algorithm"] & \mathcal{O}_{rep}
\end{tikzcd}$$

Figure 4.1: Correspondence between models and representations. The symbols $\mathcal{I}, \mathcal{O}$ denote the set of inputs and outputs on the model level, $\mathcal{I}_{rep}, \mathcal{O}_{rep}$ denote the same on the representation level.

Often enough, the translation is sufficiently accurate to produce useful results (and we will give some evidence that this is particularly the case for our algorithm). But the question for a mathematical justification remains.

In general an algorithm is considered to be robust if it produces the correct result for some perturbation of the input. Hence, one way to prove the robustness of an algorithm is to show that there is always an input for which the algorithm gives the correct results. However, this is very often a complicated task. A complete error analysis of an algorithm even in more simple problems can be cumbersome, see Pérez-Díaz et al. (2004). As the presented algorithms are much too complicated, a detailed error analysis is beyond of the scope of the present paper.

A weaker justification would be to observe, that all operations and manipulations on the level of coefficients in our algorithms are continuous. This means that small numerical errors in the input and in the computations lead to small numerical errors in the output. The resulting local parametrization and the intermediate results are not exact, but they are close to an exact result with real algebraic coefficients. In the case of a continuous algorithm if we increase the precision, the computed result converges to the exact result. However, we have to admit that we failed to make all operations continuous in our algorithms. For further details on the problem see Section 4.6.

Our justification that the provided algorithms work on floating point data is empirical. Experiences show, that increasing the precision in the computation lead to more and more accurate results. For further details, see Section 4.5.2 and Section 4.5.4.

4.3 Preliminaries

Throughout this chapter we work in the real projective space. We will consider a nonsingular cubic surface S given by its implicit form F. A point of the surface will be called *generic*, if it does not belong to one of the lines lying completely on the surface.

For future reference we recall that each non-singular cubic surface has at least one real line, and that surfaces consist of one ($\mathbf{F}_1, \ldots, \mathbf{F}_4$) or two ($\mathbf{F}_5$) real components. One of the two components of the $\mathbf{F}_5$ surface is convex in the following sense:

Definition 1. *A connected component of a surface is said to be convex, if there exists an auxiliary plane, such that for any tangent plane of the component, the component is fully contained in one of the two cells defined by the planes.*

The auxiliary plane acts as the plane at infinity.

4.3.1 Local parametrization

Given the surface S and a point $P = (p_1 : \cdots : p_4)$ on it, we are interested in finding a rational map defined in a certain neighborhood of the origin, which is “well-behaved” at P, and covers a certain neighborhood of the given point.

Definition 2. *A quadruple of polynomials* $(\pi_1(u, v), \ldots, \pi_4(u, v))$ *is called a local parametrization of the surface* S *at the point* P*, if the image of the* *origin is* P*,*

$$(\pi_1(0,0) : \pi_2(0,0) : \pi_3(0,0) : \pi_4(0,0)) = (p_1 : p_2 : p_3 : p_4), \tag{4.1}$$

and the image of the rational map defined by the four polynomials is fully contained in the surface. The local parametrization is said to be regular, if the Jacobian matrix of the mapping

$$(u, v, \rho) \mapsto (\rho\, \pi_1(u, v), \ldots, \rho\, \pi_4(u, v)) \tag{4.2}$$

has full rank (i.e., 3) at $(0 : 0 : 1)$*.*

Remark 8. Note, that the regularity condition is equivalent with the fact that the tangent vectors at P are linearly independent, i.e. they are not coplanar.

Remark 9. Clearly, all proper parametrizations give a local parametrization.

The following result is an immediate consequence of the implicit mapping theorem (Kendig, 1977).

Proposition 2. *For any given regular local parametrization* G *there exists a neighborhood of the origin in the parameter space, such that the restriction of* G *to this neighborhood is faithful.*

4.4 Analyzing the system of all tangent planes

We analyze the location of points with respect to the system of all tangent planes of the given cubic surface.

4.4.1 The t–property

We introduce the following auxiliary notion.

Definition 3. *Let* S *be a cubic surface and* $P \in S$ *a generic point on the surface.* P *is said to have the t–property if* P *is contained in the tangent plane at another surface point.*

Clearly, this second tangent plane is different from the tangent plane at P.

We recall the definition of *contour generator* and *apparent contour* following Cipolla and Giblin (2000). Using a perspective projection, we project a given surface S from a given point P into a plane Π, $P \notin \Pi$. The point P is called the *center of the projection*, while Π acts as the *image plane*.

We consider the cone of lines through P which are tangent to the surface S. This cone is called the *tangent cone* to S with apex P. The curve $\mathcal{C}_P^c$ on S where this cone is tangent to S is called the *contour generator*, and the curve $\mathcal{C}_P^a$ where the cone intersects the image plane is the *apparent contour*.

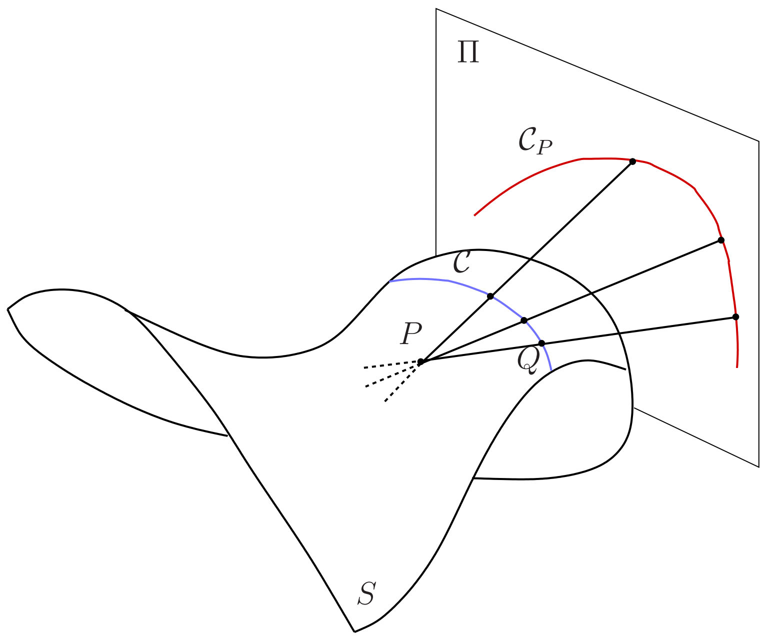


Figure 4.2: The contour generator and the apparent contour of a surface. On the figure $\mathcal{C}$ is the contour generator, $\mathcal{C}_P$ is the apparent contour of S with respect to the point P.

In the case of a cubic surface S, the contour generator is a space curve of degree 6, and the apparent contour is a planar quartic curve. In fact, if we move the point P (i.e., the center of the projection) to the origin $(0:0:0:1)$, the equation of the surface takes the form

$$x_4^2 L(x_1, x_2, x_3) + x_4 D(x_1, x_2, x_3) + C(x_1, x_2, x_3) = 0, \tag{4.3}$$

where L, D and C are linear, quadratic and cubic homogeneous polynomials, respectively. After a short computation one arrives at the the equation

$$[D(x_1, x_2, x_3)]^2 - 4L(x_1, x_2, x_3)\, C(x_1, x_2, x_3) = 0 \tag{4.4}$$

of the apparent contour $\mathcal{C}_P^a$. The linear form L is the equation of the line arising from the intersection of the tangent plane $T_P S$ with the image plane Π.

First we analyze the singularities which may be present in the apparent contour.

Lemma 3. *The apparent contour associated with a point P on a non-singular cubic surface has a singular point if and only if the point P lies on one of the lines on the surface.*

Proof. Let $P = (0 : 0 : 0 : 1)$, and assume that equation of the surface has the form (4.3). We may assume that $L = x_3$, i.e., that the tangent plane at P is $x_3 = 0$.

First case: The apparent contour has a singular point in the tangent plane at P. Without loss of generality we assume that it is located at $(1 : 0 : 0)$. A short computation reveals that this implies $q_{2,0,0} = k_{3,0,0} = 0$, where q_{ijk} and k_{ijk} are the coefficients of $x_1^i x_2^j x_3^k$ in D and C, respectively. Consequently, the line $(s : 0 : 0 : t)$ $(s, t \in \mathbb{R})$ is fully contained in the surface.

Second case: The apparent contour has a singular point which is not in the tangent plane at P. Without loss of generality we assume that it is located at $(0 : 0 : 1)$. A short computation reveals that this implies $k_{0,1,2} = \frac{1}{2} q_{0,0,2} q_{0,1,1}$, $k_{1,0,2} = \frac{1}{2} q_{0,0,2} q_{1,0,1}$ and $k_{0,0,3} = \frac{1}{2} q_{0,0,2}^2$. The surface is singular, since it has the singular point $(0 : 0 : -2 : q_{0,0,2})$.

Finally, it can be shown that any line through P generates a singular point of the apparent contour. $\square$

The t–property can now be characterized by using the apparent contour.

Proposition 3. *A generic point P of the cubic surface S has the t–property if and only if the apparent contour of the surface with center P has real points.*

Proof. The point P has the t–property, if and only if there exists a point $R \in S$, $R \neq P$, such that $P \in T_R S$, where $T_R S$ is the tangent plane to the surface S at R. This is equivalent to the fact that the line connecting P and R is a real line of the tangent cone with apex P. This line corresponds to a regular point of the apparent contour. Note that the apparent contour cannot have singularities, since P is assumed to be a generic point (cf. Lemma 3). $\square$

The following algorithm, which is based on Proposition 3, is needed for computing the local parametrization, as described in the next sections.

Algorithm 4 (Point on Contour).
Input: An implicit equation F of a cubic surface S and a general point $P \in S$.
Output: Decide the t–property for P. If P has the t–property find $R \in S$, such that $P \in T_R S$.

1. We move P to the origin by a linear transformation of the homogeneous coordinates such that the tangent plane at P becomes $x_3 = 0$, and compute the equation of the apparent contour (4.4).
2. Check whether the apparent contour has real points using methods similar to Gonzalez-Vega and Necula (2002).
   (a) If the apparent contour does not have real points, then the algorithm stops. The point P does not have the t–property.
   (b) If the apparent contour has non-singular real points, then P has the t–property. Go to the next step.
3. Find a real point $\bar{R}_a = (\bar{x}_1 : \bar{x}_2 : \bar{x}_3)$ on the apparent contour such that $\bar{x}_3 \neq 0$.
4. Compute the corresponding point R on the contour curve. If P is at $(0:0:0:1)$, then
$$\bar{R} = (\bar{x}_1 : \bar{x}_2 : \bar{x}_3 : \frac{-D(\bar{x}_1, \bar{x}_2, \bar{x}_3)}{2L(\bar{x}_1, \bar{x}_2, \bar{x}_3)}). \tag{4.5}$$
   (Note, $L(\bar{x}_1, \bar{x}_2, \bar{x}_3) = \bar{x}_3 \neq 0$.)
5. Using the inverse of the linear transformation in (1), we transform $\bar{R}$ to get $R \in S$.

Step 1 Assume the point P is not at infinity and the tangent plane at P has an equation $a_1x_1 + \cdots + a_4x_4 = 0$ with $a_3 \neq 0$. Indeed, in this situation we can use the projective transformation given by the matrix
$$\begin{pmatrix} 1 & 0 & 0 & -p_1/p_4 \\ 0 & 1 & 0 & -p_2/p_4 \\ a_1/a_3 & a_2/a_3 & 1 & a_4/a_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{4.6}$$
In the case one of these conditions does not hold, we relabel coordinates. This should also be done if p_4 or a_3 is close to zero (i.e. if its order of magnitude is the same as the precision).

Step 3 Finding a real point on a nonsingular real algebraic curve can be done by computing a real solution between the real zeros of its discriminant.

Step 4 If the point on the apparent contour is $\bar{R}_a = (\bar{x}_1 : \bar{x}_2 : \bar{x}_3)$, then the corresponding point on the contour generator is $\bar{R} = (\bar{x}_1 : \bar{x}_2 : \bar{x}_3 : \bar{x}_4)$. To compute $\bar{x}_4$ note, that the coordinates of $\bar{R}$ satisfy Equation 4.4 and Equation 4.3. A short computations reveals the following:

$$D(\bar{x}_1, \bar{x}_2, \bar{x}_3)^2 + 4\bar{x}_4 L(\bar{x}_1, \bar{x}_2, \bar{x}_3)^2 + 4\bar{x}_4^2 L(\bar{x}_1, \bar{x}_2, \bar{x}_3) D(\bar{x}_1, \bar{x}_2, \bar{x}_3) = 0$$

$$[D(\bar{x}_1, \bar{x}_2, \bar{x}_3) + 2\bar{x}_4 L(\bar{x}_1, \bar{x}_2, \bar{x}_3)]^2 = 0$$

Hence,

$$\bar{x}_4 = \frac{-D(\bar{x}_1, \bar{x}_2, \bar{x}_3)}{2L(\bar{x}_1, \bar{x}_2, \bar{x}_3)}.$$

4.4.2 Locating the points with the t–property

Given a non-singular cubic surface, we identify the regions of points with and without the t–property on a nonsingular cubic surface.

Lemma 4. *The regions on a cubic surface S containing points with and without the t–property are bounded by the real lines of S.*

Proof. Consider the system of apparent contours associated with all points on the surface. Clearly, the coefficients of these planar curves depend continuously on the location of the points.

If one moves along a curve from a point P with the t–property to a point Q without it, the apparent contour, which has at least one real component at P, has first to degenerate to a singular point, before disappearing eventually. Due to Lemma 3, this takes place exactly when one crosses one of the lines lying on the surface. □

We distinguish three types of surface points concerning the local behavior of a surface with respect to its tangent plane at a point (see Section 2.3.5). One may detect the three types of points by the Gaussian curvature $\mathbf{k}$. A point of a surface is called *elliptic* if $\mathbf{k} > 0$, *parabolic* if $\mathbf{k} = 0$ and *hyperbolic* if $\mathbf{k} < 0$.

The non-convex component of a cubic surface may consist of hyperbolic, elliptic, and parabolic points, while the convex component of the $\mathbf{F}_5$ surface has elliptic points only.

Lemma 5. *Generic hyperbolic points of a cubic surface have the t–property.*

Proof. A short computation reveals that the two asymptotic directions at a generic hyperbolic point (i.e., the tangent directions of the two branches of the intersection curve with the tangent plane at the point) correspond to real points of the apparent contour.

Let S be a cubic surface given in the form 4.3. Furthermore, let the apparent contour with respect to $P(0:0:0:1)$ given by 4.4. (Note, P is a hyperbolic point of the surface.) Assume, that $L(x_1, x_2, x_3) = x_3$. Then the equation of the surface takes the form $x_4 D(x_1, x_2, 0) + C(x_1, x_2, 0)$. As P is a hyperbolic point, $D(x_1, x_2, 0)$ splits into two different linear factors. Up to coordinate transformation $D(x_1, x_2, 0) = x_1 x_2$. Therefore, the points $(0:1:0)$, $(1:0:0)$ are points of the apparent contour. □

Theorem 11. *A generic point of a nonsingular cubic surface S has the t–property if and only if it lies on the non-convex component of the surface.*

Proof. Any non-singular cubic surface contains at least 3 real lines. Lines are always on the non-convex component of S, as the convex component does not contain any line. These lines define a partition of the component into several cells.

Any line of a nonsingular cubic surface contains only hyperbolic points, with the exception of the two parabolic points (Segre, 1942). Consequently, the neighborhood of any line contains hyperbolic points which have the t–property (Lemma 5). According to Lemma 4, if a point has the t–property, then this property is shared by all points in the cell.

It remains to be shown that the convex component of the $\mathbf{F}_5$ surface does not contain points with the t–property. Consider any point $R \in S$ on the convex component. Assume, that there is a point $Q \in S$ such that $R \in T_Q S$. As the tangent plane cannot intersect the convex component in a different point than Q, the point Q cannot lie on the convex component. The line in $T_Q S$ connecting Q and R has four intersections with the surface S, since Q has to be counted twice. This is a contradiction, since any line has at most three real intersections with a cubic surface. □

4.5 Three techniques for generating local parametrization

We describe three approaches to the solution of the local parametrization problem. The three techniques are based on the theory of rational curves on cubic surfaces. For the convenience of the reader, we summarize it in the next section.

4.5.1 Rational cubics on cubic surfaces

The intersection of a cubic surface with the tangent plane at a generic surface point P always gives a rational planar cubic, where the point will be the singular point of the curve. A rational cubic can be parameterized by a pencil of lines through the singularity of the curve, which intersect the cubic at exactly one other point. The coordinates of the latter point give parametric functions for the cubic curve.

More precisely, if we assume that $P = (0 : 0 : 0 : 1)$ and that the tangent plane at the origin equals $x_3 = 0$, the equation of the surface takes the form

$$x_4^2 x_3 + x_4 D(x_1, x_2, x_3) + C(x_1, x_2, x_3) = 0.$$

The cubic curve $\mathcal{C}_P$ cut by the tangent plane at the origin is

$$D(x_1, x_2, 0)x_4 + C(x_1, x_2, 0) = 0.$$

It has the rational parametrization $(D(1, t, 0) : t \cdot D(1, t, 0) : 0 : -C(1, t, 0))$. For further details we refer to Abhyankar and Bajaj (1988) .

4.5.2 The 2-curve technique

This technique has been described by Manin (1986). Let Q_1 and Q_2 be two real points on the cubic surface S as in Figure 4.3. We denote by $\mathcal{C}_{Q_i}$ the curves cut by the tangent plane $T_{Q_i}S$, $i = 1, 2$, from the surface S. The cubic curves $\mathcal{C}_{Q_i}$ have a double point at Q_i, therefore they can be parameterized by rational functions.

Let $\pi_i\colon \mathbb{R} \to \mathcal{C}_{Q_i}$ be the parametrization of the i–th curve. Then $\pi\colon \mathbb{R}^2 \to S$, $(t_1, t_2) \mapsto P$ gives a parametrization of a neighborhood of P, where P is the third point of the surface obtained by intersecting with the line $\pi_1(t_1), \pi_2(t_2)$.

This idea is formalized in the following algorithm.

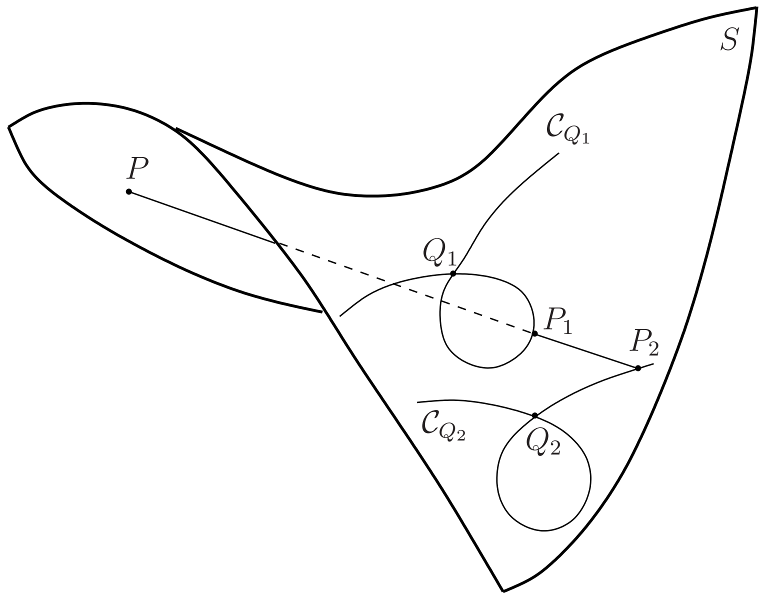


Figure 4.3: The 2-curve technique.

Algorithm 5 (2-curve Technique).
Input: An implicit equation F of a cubic surface S and a point $P \in S$.
Synopsis: Find 4 polynomials depending on 2 parameters, which define a local parametrization of S around P.

1. Check the t–property for P
   (a) If P does not have the t–property the algorithm stops.
   (b) If P has the t–property go to the next step.
2. Choose a line through P which has 2 further intersections P_1, P_2 with S enjoying the t–property.
3. Compute the intersection points P_1, P_2.
4. (a) Choose a point Q_1 on the contour of P_1 using Algorithm 4.
   (b) Choose a point Q_2 on the contour of P_2 such that the tangents at P_1 and P_2 to the curves $\mathcal{C}_{Q_1}$ and $\mathcal{C}_{Q_2}$ are not coplanar.
5. Parameterize the cubics $\mathcal{C}_{Q_i} = S \cap T_{Q_i}S$, such that the parameter 0 corresponds to P_i, $i = 1, 2$.
6. Let the parametrization of $\mathcal{C}_{Q_i}$ be $(x_i(t_i) : y_i(t_i) : z_i(t_i) : w_i(t_i))$. Intersect the line $(x_1(t_1) + \lambda x_2(t_2) : y_1(t_1) + \lambda y_2(t_2) : z_1(t_1) + \lambda z_2(t_2) :$

$w_1(t_1) + \lambda w_2(t_2))$ with S; this leads to a quadratic equation with one root at 0. Compute the remaining root $\lambda(t_1, t_2)$ and substitute it back into the equation of the line. This gives the parametrization of the surface S around the point P.

We give a more detailed description for some steps of Algorithm 5.

(Step 2) Points on Π can be distributed into 3 types with respect to the projection of S from a point P. Points that are images of points on the contour generator of the surface (apparent contour), points corresponding to two real surface points, and the ones corresponding to complex conjugate surface points. Those points on Π for which the equation of $\mathcal{C}_P$ has negative sign are images of two real points on the surface S, see Figure 4.4.

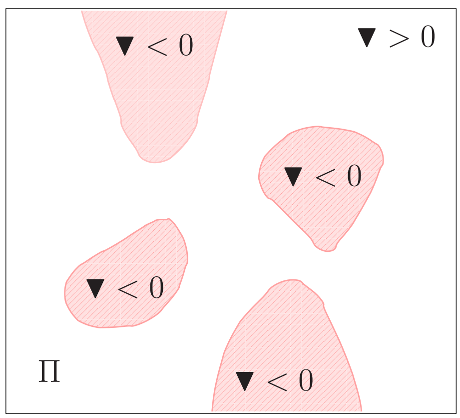


Figure 4.4: Types of points on the projection plane. The symbol ▼ stands for the evaluation of the equation of $\mathcal{C}_P$ at a point. Points in the shaded area correspond to two real points on the surface.

This fact can be used to to find a line through P which has 2 further intersections P_1, P_2 with S. We determine a point $\tilde{P}$ on Π such that the desired line is the one connecting P and $\tilde{P}$.

We compute the discriminant of the curve $\mathcal{C}_P$ and its real zeroes. Using the computed real roots we determine intervals where the discriminant has negative sign. We substitute a point of the first interval (the first coordinate of $\tilde{P}$) into the equation of the curve, and compute the real zeroes of the arising univariate polynomial. Then we determine the intervals where the equation of $\mathcal{C}_P^a$ has negative sign, and set the second coordinate of $\tilde{P}$ to the coordinate of a point of one such interval.

If one of the arising intersection points P_1, P_2 does not have the t–property we choose a different point of the second interval in the previous step.

(Step 3) Let the line l_P^r through P is given by $(p_1+\mu\cdot r_1 : \cdots : p_4+\mu\cdot r_4)$. To compute the intersection of S and l_P^r we substitute the equation of the line into the equation of the surface $F(l_P^r)$, and compute the solutions for μ. Since one of the solutions is equal to zero, we divide by μ and obtain a quadratic equation. Solving it we get the further intersection points of S and l_P^r.

(Step 4) Intersecting the tangent planes $T_{Q_1}S$ and $T_{P_1}S$ gives the tangent l_{P_1} at P_1 to the curve $\mathcal{C}_{Q_1}$. To get the forbidden tangent line $\bar{l}_{P_2}$ at P_2, we intersect the tangent plane at the point by l_{P_1}, see Figure 4.5.

We determine the tangents from the point $\bar{l}_{P_2} \cap \Pi$ to the curve $\mathcal{C}_{P_2}$. The points on the apparent contour determined by these tangent directions are the points, that should be avoided in the process of determining Q_2 in Step 4(b) of the algorithm. (The maximum number of tangents to a degree n curve can be determined by the Plückler formula $n(n-1)$. Hence, the maximum number of points to be avoided is 12.)

(Step 5) The curves $\mathcal{C}_{Q_i}, i = 1, 2$ can be parameterized using the method described in Section 4.5.1. However, in order to guarantee that the zero parameter value corresponds to the point P_i, reparametrization might be necessary.

(Step 6) As the last step of the algorithm, we have to intersect the line $(x_1(t_1) + \lambda x_2(t_2) : y_1(t_1) + \lambda y_2(t_2) : z_1(t_1) + \lambda z_2(t_2) : w_1(t_1) + \lambda w_2(t_2))$ with S, and compute λ. After substituting the equation of the line into F we get $B_1(t_1, t_2)\lambda + B_2(t_1, t_2)\lambda^2$. Hence, $\lambda = -B_1/B_2$.

We apply the algorithm to an example.

Example 12. Consider the surface S defined by

$$F = 3x_4x_1^2 + 3x_4x_2^2 + 3x_4x_3^2 - 10x_1x_2x_3 - 3x_4^3,$$

and a point $P = (1 : 3 : 27 : 27)$ on S. Using Algorithm 4 we check that the point P has the t–property. We take a line through P and intersect it with the surface S, giving 2 additional points on S:

$$P_1(-5.98687 : -3.36937 : -1.87809 : -5.09084),$$
$$P_2(-1.07744 : -0.44309 : 1.44983 : 0.83569)$$

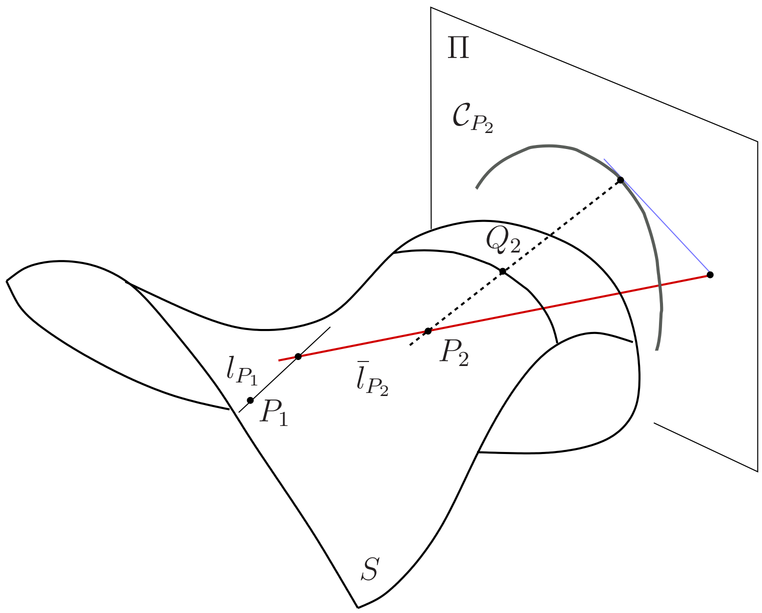


Figure 4.5: Forbidden tangent direction. The point Q_2 should be determined such that the tangents at P_1 and P_2 to the curves $\mathcal{C}_{Q_1}$ and $\mathcal{C}_{Q_2}$ are not coplanar (i.e. the direction indicated by the red line should we avoided).

We want to compute two points Q_1, Q_2, such that P_i is on the tangent plane $T_{Q_i}S$. For this we have to compute a point on the contour curve of S with respect to the projection from P_i.

The apparent contour of S with respect to the projection from P_1 has equation:

$$\begin{aligned} f_1 = (&-15.27251x_1^2 - 15.27251x_2^2 - 15.27251x_3^2 + 18.78088x_1x_2 + \\ &33.69367x_1x_3 + 59.86867x_2x_3)^2 + 40(119.58934x_1 - \\ &9.52124x_2 - 144.35332x_3)x_1x_2x_3. \end{aligned}$$

$\bar{Q}_{a,1}(6.53995 : 5.94157 : -2.97076)$ is a point on $\mathcal{C}^a_{P_1}$, which corresponds to the point

$$Q_1(0.55308 : 2.57216 : -4.84885 : -5.09084)$$

on S. The intersection of S with the tangent plane at Q_1 gives a curve $\mathcal{C}_{Q_1}$.

The parametrization of $\mathcal{C}_{Q_1}$ is:

$$\begin{pmatrix} 2.15038t_1^3 - 44.54884t_1^2 + 216.56989t_1 - 263.60396 \\ 4.48066t_1^3 - 74.41867t_1^2 + 205.08880t_1 - 148.34822 \\ -18.79020t_1^2 + 75.48726t_1 - 82.69451 \\ -24.43222t_1^2 + 160.93907t_1 - 224.14926 \end{pmatrix}$$

The apparent contour of S with respect to the projection from P_2 is defined by:

$$\begin{aligned} f_2 = {} & (1.25354x_1^2 + 1.25354x_2^2 + 1.25354x_3^2 - 7.24916x_1x_2 + 2.21548x_1x_3 \\ & + 5.38720x_2x_3)^2 + 40(0.25543x_1 + 3.34983x_2 + 0.62389x_3)x_1x_2x_3. \end{aligned}$$

Similarly, $\bar{Q}_{a,2}(0.77339 : 0.19981 : -0.39962)$ is a point on $\mathcal{C}^a_{P_2}$, which corresponds to the point

$$Q_2(0.23467 : -0.021736 : 0.32529 : 0.41785)$$

on S. The parametrization of $\mathcal{C}_{Q_2}$ is

$$\begin{pmatrix} -1.05829t_2^3 + 3.27466t_2^2 + 0.80973t_2 - 5.21309 \\ -0.85279t_2^3 - 0.88123t_2^2 + 4.94149t_2 - 2.14387 \\ 3.18407t_2^2 - 9.35177t_2 + 7.01474 \\ 5.18538t_2^2 - 10.67940t_2 + 4.04335 \end{pmatrix}$$

Let the equation of the line connecting $\mathcal{C}_{Q_1}$ and $\mathcal{C}_{Q_2}$ be $(x_1(t_1) + \lambda x_2(t_2) : y_1(t_1) + \lambda y_2(t_2) : z_1(t_1) + \lambda z_2(t_2) : w_1(t_1) + \lambda w_2(t_2))$. After substituting this equation into F we compute λ. Substituting λ back to the equation of the line gives a parametrization of a neighborhood of the point P.

Figure 4.6 shows several parameterized patches on a given cubic surface.

Table 4.5.2 shows the numerical behavior of the 2-curve parametrization technique in the case of Example 12. The letters P, P_0, P_g denote the given point on the surface S, the point generated by the parametrization for $(0,0)$ parameter values, and a point generated by the parametrization for $(2,-5)$ parameter values. We see if the error goes to zero, then P_0 converges to P and $F(P_g)$ converges to zero.

Remark 10. The skew line parametrization technique (see Section 2.4.1) is a special case of the 2-curve technique. If P_1, P_2 are on two (skew) lines, and the chosen Q_1, Q_2 are on the lines of P_1, P_2 respectively, then we arrive at the skew line parametrization technique.

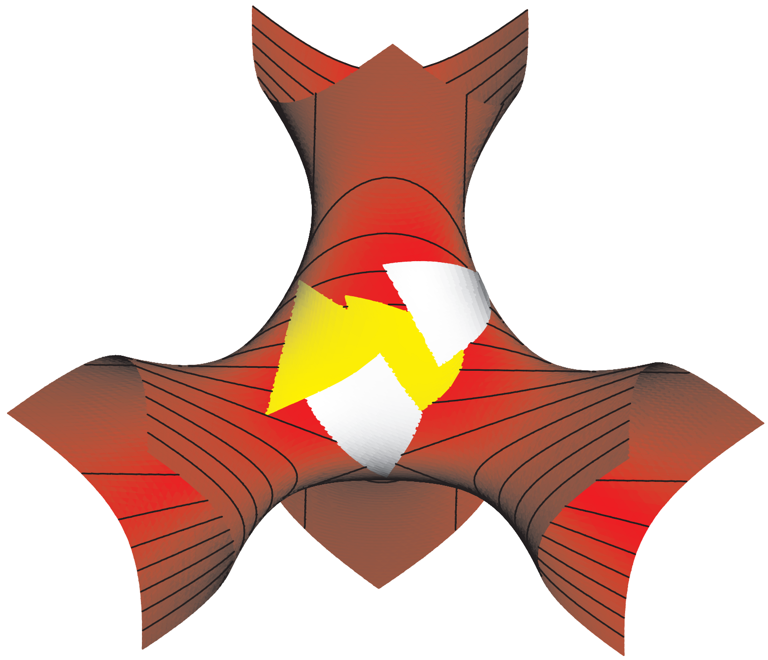


Figure 4.6: Implicit surface with parameterized patches.

error level	distance between P, P_0	$F(P_g)$
10^{-10}	10^{-9}	10^{-8}
10^{-15}	10^{-14}	10^{-13}
10^{-20}	10^{-18}	10^{-18}
10^{-30}	10^{-28}	10^{-28}
10^{-40}	10^{-37}	10^{-37}
10^{-50}	10^{-47}	10^{-48}

Table 4.1: Numerical experiment using the 2-curve technique.

Now we prove the correctness of Algorithm 5.

Theorem 12. *For a nonsingular cubic surface S and generic point $P \in S$, Algorithm 5 produces a regular local parametrization if and only if P is a point with the t–property.*

Proof. If P has the t–property, then the whole component containing P has only points with this property. Thus we can always find lines through P which intersect the surface S in 2 additional real intersections with the non-convex component of the surface, i.e., in points with the t–property.

Due to the construction of the algorithm, the image of the origin is P and the image of the map is contained in the surface.

It remains to be shown that – in Step 4 – it is always possible to choose a point Q_2 on the contour curve such that the tangent lines to the curves $\mathcal{C}_{Q_1}$ and $\mathcal{C}_{Q_2}$ are not coplanar.

Let l_{P_1} denote the tangent line at P_1 to the curve $\mathcal{C}_{Q_1}$. (l_{P_1} is the intersection of the tangent planes $T_{Q_1}S$ and $T_{P_1}S$.) Furthermore denote by $l^j_{P_2}$ the tangent line at P_2 to the curve $\mathcal{C}_{Q^j_2}$. (It is the intersection of the tangent planes $T_{Q^j_2}S$ and $T_{P_2}S$.) The line connecting P_2 with the intersection of l_{P_1} and $T_{P_2}S$ gives the tangent direction at P_2 which is forbidden. We have to show, that it is always possible to choose a point Q^j_2 on the contour with respect to P_2 such that $l^j_{P_2}$ is not the forbidden direction.

We show that it is not possible to have the same tangent direction for all points on the apparent contour with respect to P_2. For each point on the apparent contour we get a corresponding point on the contour generator Q^j_2 and a tangent direction $l^j_{P_2}$. If all points gave the same tangent direction l_{P_2}, then all tangent planes $T_{Q^j_2}S$ would go through this line, i.e. we would get a pencil of planes. Thus the envelope surface of these tangent planes would degenerate into a line, which is not possible.

As one can verify by direct computation, if the tangents to the curves $\mathcal{C}_{Q_1}$ and $\mathcal{C}_{Q_2}$ are not coplanar, then the Jacobian of the parametrization has full rank at P.

On the other hand, if P does not have the t–property, then Algorithm 5 stops in Step 1. In this situation it is clear, that any line through P would intersect S in another point which does not have the t–property. □

Note, if S is a nonsingular cubic with two components, and P in on the convex component of S, then Algorithm 5 does not produce a local parametrization. Any line through P intersecting S in two additional real points has another intersection with the convex component of S. This point does not bear the t–property, see Theorem 11.

Remark 11. It can be shown that the parametrization computed by Algorithm 5 has bidegree $(6, 6)$ and total degree 12.

We call a number $k \in \mathbb{N}$ the *index* of the parametrization if all points outside a Zariski closed subset are generated by k complex parameter pairs (Sendra and Winkler, 2001). A proper parametrization has index 1.

Proposition 4. *The index of the parametrization obtained by the 2-curve technique equals 6.*

Idea of the proof. If P is a point on S that can be parameterized using the points Q_1, Q_2, we have to compute how many lines through P exist, which intersects both curves $\mathcal{C}_{Q_1}$, $\mathcal{C}_{Q_2}$. As the planes of the two curves $\mathcal{C}_{Q_1}$, $\mathcal{C}_{Q_2}$ intersect in a line, the curves have three intersections on a line. If we project the two curves $\mathcal{C}_{Q_1}, \mathcal{C}_{Q_2}$ from P on an arbitrary plane we get nine intersection points from which three are on a line (Bezout's theorem). Hence, we can reach P six times using this parametrization method. See Manin (1986) for a complete proof. □

Remark 12. Methods for reducing the index of the parametrization of curves exist (Sederberg, 1984, 1986). Unfortunately, currently no methods for reducing the index in the surface case are available. Systematic techniques for reducing the index of a parametrization could be of some interest and should be explored.

4.5.3 The repeater technique

Here is an alternative idea for computing a local parametrization. We do not describe it in full details because it is inferior to the 2-curve technique in two ways. First, it is applicable in less situations, second it is more complicated to analyze the degeneracy conditions expressing the vanishing of the jacobian.

Let Q_0 be a real point on S as in Figure 4.7. The rational cubic $\mathcal{C}_{Q_0} = T_{Q_0}S \cap S$ has a rational parametrization $\pi_{Q_0} : \mathbb{R} \to \mathcal{C}_{Q_0}$. Let $\mathcal{C}_{Q(t)}$ be a curve cut by the tangent plane at the point $Q(t) := \pi_{Q_0}(t)$. Then, the parametrization of the curve $\mathcal{C}_{Q(t)}$, $\pi_{Q(t)} : \mathbb{R} \to \mathcal{C}_{Q(t)}, s \mapsto \pi_{Q(t)}(s)$ gives a parametrization of a neighborhood of the point $P := \pi_{Q(t)}(s)$.

The above technique leads to the following algorithm.

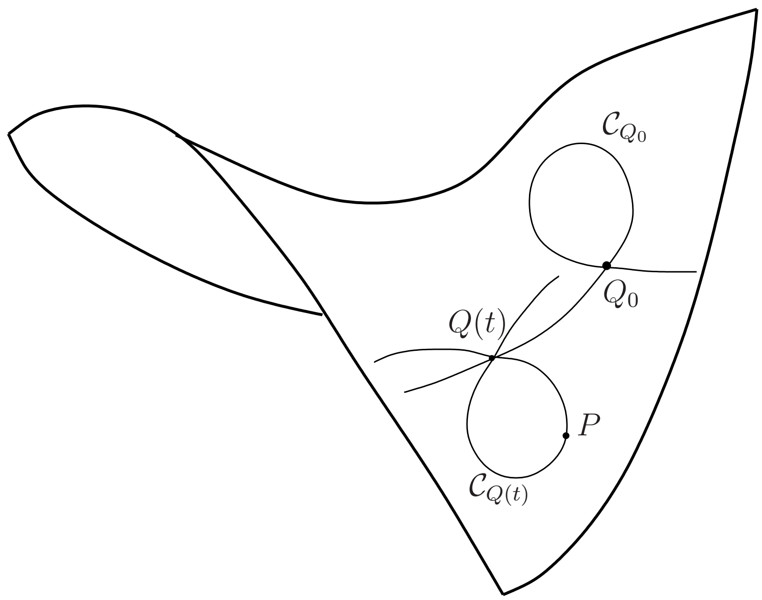


Figure 4.7: The repeater technique.

Algorithm 6 (Repeater Technique).
Input: An implicit equation F of a cubic surface S and a point $P \in S$.
Synopsis: Find 4 polynomials depending on 2 parameters, which give a local parametrization of S around P.

1. Check the t–property for P

   (a) If P does not have the t–property the algorithm stops.

   (b) If P has the t–property go to the next step.

2. Compute the contour generator $\mathcal{C}_P^c$ with respect to P and choose a point Q on it with the t–property. (see Algorithm 4)

   (a) If there is no such point the algorithm stops.

   (b) If there is such a point go to the next step.

3. Compute the contour generator with respect to Q and choose a point Q_0 on it such that $T_{Q_0}S$ does not contain the tangent at Q to $\mathcal{C}_P^c$.

4. Compute the intersection of S with tangent plane $T_{Q_0}S$: $\mathcal{C}_{Q_0}$.

5. Parameterize $\mathcal{C}_{Q_0}$, such that $Q = \pi_{Q_0}(0)$.

6. Parameterize the curve $\mathcal{C}_{Q(t)}$, which is the intersection of S with the tangent plane at the point $Q(t) := \pi_{Q_0}(t)$, such that $P = \pi_{Q(0)}(0)$.

Remark 13. The tangent plane $T_{Q(t)}S$ and hence the curve $\mathcal{C}_{Q(t)}$ depends on the parameter t. To parameterize $\mathcal{C}_{Q(t)}$ we need a pencil of lines through $Q(t)$ in the plane $T_{Q(t)}S$. We take a pencil of planes such that the common line of the planes in the pencil pass through $Q(t)$. (The pencil of planes depends on the parameter s.) Computing the intersection of the planes with $T_{Q(t)}S$ we get a pencil of lines through $Q(t)$ depending on the parameters t, s.

Remark 14. It may happen that the contour generator with respect to the point P lies on the convex component of S. In this case the algorithm stops at Step 2. In other words the t–property for P is not sufficient for Algorithm 6 to produce a result.

Remark 15. It is not difficult to prove that Algorithm 6 always produces a local parametrization, if the contour generator with respect to P has points with the t–property. In order to obtain a *regular* local parametrization we need an additional condition analogous to the condition in Algorithm 5 that the tangents at P_1, P_2 to $\mathcal{C}_{Q_1}$ and $\mathcal{C}_{Q_2}$ are not coplanar. More precisely, the tangent plane at $T_{Q_0}S$ must not contain the tangent at Q to the contour generator with respect to P. We do not give a detailed proof, as the repeater technique is less useful than the 2-curve technique for our purposes.

Remark 16. The total degree of the parametrization using Algorithm 6 is 12.

Proposition 5. *The index of the parametrization obtained by the repeater technique is 6.*

Idea of the proof: We want to compute how many times a point P is covered by the parametrization using the repeater technique and a point Q_0, i.e. how many (t, s) parameter pairs exist such that $\pi_{Q(t)}(s) = P$. Let $\mathscr{U}$ be defined as follows: $\mathscr{U} := \{Q \mid \exists s : \pi_Q(s) = P\}$. Then $\mathscr{U} = \{Q \mid P \in \mathcal{C}_Q\}$ is the contour curve of S projected from P. The contour curve has degree 6, so the number of intersection points of $\mathscr{U}$ and $\mathcal{C}_{Q_0}$, which is equal to the number of intersection points of $\mathscr{U}$ and $T_{Q_0}S$, is also 6. Hence, the index using the repeater technique is 6. $\square$

4.5.4 The reflection technique

Algorithm 5 and Algorithm 6 fail if the surface has two components, and the point around which we want to compute a local parametrization is on the

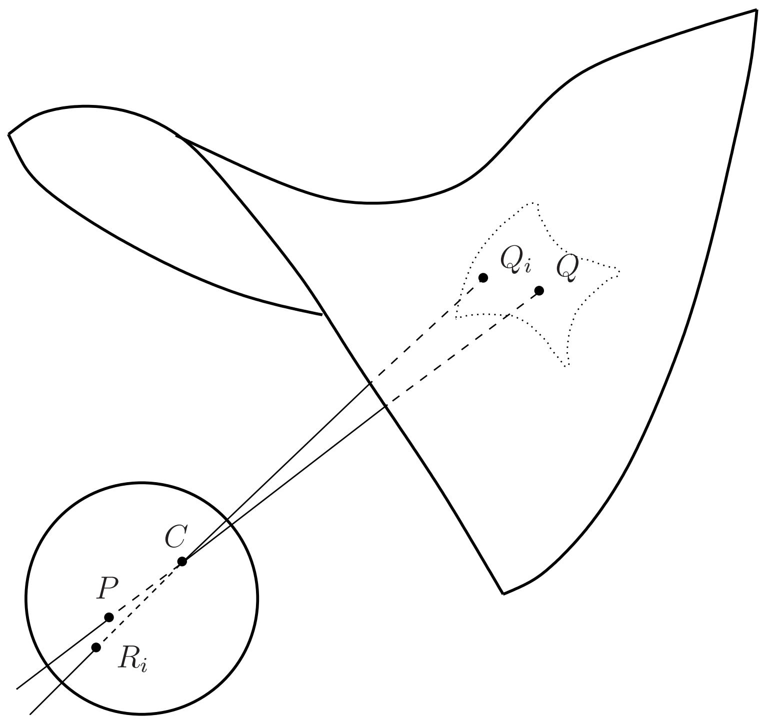


Figure 4.8: The reflection technique.

convex piece. We can detect this case simply by checking if the point has the t–property or not. If it does not have this property, then it is located on the convex component. In such situations, we can use the following technique.

Let P be the point on the surface S; see Figure 4.8. Using Algorithm 5 we can parameterize some region of S. Connect the point P with any point Q from the parameterized region and denote by C the further intersection point with S. From C reflect the points of the parameterized region. This gives a parametrization of the neighborhood of the point P.

We summarize this idea in the following algorithm.

Algorithm 7 (Reflection Technique).
Input: An implicit equation F of a cubic surface S and a point $P \in S$.
Synopsis: Find 4 polynomials depending on 2 parameters, which define a parametrization of a neighborhood of P.

1. Using Algorithm 5 compute a local parametrization for a point $Q \in S$ with the t–property, where $Q \notin T_P S$. Let the parametrization be $P_{t_1,t_2} := (X(t_1,t_2) : Y(t_1,t_2) : Z(t_1,t_2) : W(t_1,t_2))$

2. Connect P with the point Q, and intersect this line with S. Let $C(c_1 : c_2 : c_3 : c_4)$ be the further intersection point.

3. Intersect the line CP_{t_1,t_2} with S. Compute the third point of intersection in terms of t_1, t_2.

Remark 17. In Step 3 in Algorithm 7 we need to compute the third point of intersection of S with a line through two points of S. This can be done in the same way as in Step 6 of Algorithm 5.

Example 13. We use the same surface as in Example 12. We want to construct a local parametrization around $P = (1.53295 : 53.20912 : 10.85109 : 1)$ Using Algorithm 5 we compute a local parametrization around the point $Q = (1 : 3 : 27 : 27)$ as in Example 12. Let denote the computed parametrization by $(X(t_1, t_2) : Y(t_1, t_2) : Z(t_1, t_2) : W(t_1, t_2))$.
The parametric equation of the line connecting P and Q is

$$(s + 1.53295 : 3s + 53.20912 : 27s + 10.85109 : 27s + 1).$$

Substituting it into the equation of S, and solving the resulting equation for s we get the further intersection of the line with S:

$$C(-2.46836 : 41.20518 : -97.18435 : -107.03544).$$

The line l_{CQ_i} connecting C with the points of the parameterized region has the form

$$(\lambda X - 2.46836 :, \lambda Y + 41.20518 : \lambda Z - 97.18435 : \lambda W - 107.03544).$$

Substituting it into F we get an equation of the form $B_1(t_1, t_2)\lambda + B_2(t_1, t_2)\lambda^2$. Computing λ and substituting it back into the equation of l_{CQ_i}, we get a local parametrization around P.

Table 4.2 shows the numerical behavior of the reflection technique in the case of Example 13. The letters P, P_0, P_g again stand for the given point on the surface S, the point generated by the parametrization for $(0, 0)$ parameter values, and a point corresponding to the $(2, -5)$ parameter values. We can observe if the error goes to zero, then P_0 converges to P and $F(P_g)$ converges to zero.

Theorem 13. *For a nonsingular cubic surface S and point $P \in S$, Algorithm 7 always gives a regular local parametrization.*

error level	distance between P, P_0	$F(P_g)$
10^{-10}	10^{-8}	10^{-7}
10^{-15}	10^{-13}	10^{-12}
10^{-20}	10^{-18}	10^{-17}
10^{-30}	10^{-27}	10^{-27}
10^{-40}	10^{-37}	10^{-37}
10^{-50}	10^{-47}	10^{-47}

Table 4.2: Numerical experiment using the reflection technique.

Proof. We have to show that it is always possible to find $R \in S$ with the t–property, where $R \notin T_P S$. The intersection of S and $T_P S$ is a degree 3 curve. The non-convex component of S contains other points, and these points have the t–property. Let R be one of these points. As it was shown before, Algorithm 5 produces a regular local parametrization around the point R.

As $R \notin T_P S$, the line l_{PR} connecting P and R is not tangent at P. Let C be the third point of intersection of S with the line l_{PR}. Then the line l_{PR} intersects the two tangent planes at P and R transversally. This implies that the reflection of the cubic surface S at C restricts to a local isomorphism of sufficiently small regions of S around P and R. Therefore, the composition of this map with the regular local parametrization around R is regular. □

The reflection technique works for any type of cubic surfaces. It can be applied arbitrarily, but it is particularly interesting in the situation, when the given surface has two components, and the point around which we want to compute a local parametrization lies on the convex part.

Remark 18. As one may verify by a straightforward computation, the parametrization computed by Algorithm 7 has bidegree $(12, 12)$ and total degree 24.

Proposition 6. *The index of the parametrization obtained by the reflection technique is 6.*

Proof. Reflection at a point is birational and does not change the index. As the index of the 2-curve technique is 6, the index of the reflection technique is also 6. □

4.5.5 Covering a surface by local parameterizations

Theorem 14. *Given a nonsingular cubic surface. It can be covered by finite number of local parameterizations.*

Proof. For each point P on the surface we can compute a local parametrization π_P, which covers some open neighborhood U_P of P. Obviously $S = \cup_{P \in S} \mathrm{U}_P$. Since S is compact there exist a finite subcover. $\square$

Remark 19. Figure 4.6 shows a surface with several parameterized patches. To cover a surface with parameterized patches we need a systematic approach to choose the points $P_i, i = 1, 2, 3, \ldots$ around which we compute local parametrization, and to determine how to extend the parameter space for each local parametrization.

4.6 Continuity failures

The presented algorithms branch due to the sign of the value of an arithmetic expression. As in practise instead of exact values only approximations are computed, a real valued expression might be evaluated incorrectly. This might lead to incorrect sign, and incorrect decision in the diverging step. Incorrect decisions finally result in incorrect outputs. Showing continuity we could guarantee that in each branching step of the algorithm the same decision is made.

Most of the operations of the presented algorithms can be made continuous, but there are some operations where we ran into troubles. We demonstrate the difficulties with an example: choose a point on the apparent contour, i.e. determine a point on a planar degree four curve.

Our idea to solve this problem was the following: compute the discriminant of the curve and the real zeros of the discriminant. Using the computed real roots we determine intervals where the discriminant has negative sign. Substituting a point (chosen in a continuous way) of the first interval into the equation of the curve, and taking the smallest solution of the arising univariate polynomial gives a point on the curve. However, this operation is not continuous as even for small perturbations in the coefficients of the equation of the apparent contour the number of intervals (where the discriminant has negative sign) changes. The reason for this change is not topological!

4.7 Refinement of the algorithm

The described algorithms are implemented in Maple. As observed in our experiments, the quality of the resulting parametrization can greatly be enhanced by using some heuristic ideas for optimizing certain steps in the algorithms.

In Algorithm 5 a line through the given point P has to be chosen, such that the line intersects the surface in two further real intersection points with t–property. In principle we can choose any random line, but it could often result in intersection points that lie near to the tangent plane of P or to the tangent planes of each other.

One possibility to avoid such situations to take n random lines through P and compute the further intersections with the surface. For those lines which have further real intersections with S we compute the distance between the points to the tangent planes at the other intersection points. We get 6 values for each line. After normalizing these values (the average of the 6 values should be 1), we take the ratio of the lowest and the highest one. To proceed further the line for which this ratio is the highest might be chosen.

Another delicate step, where we have a big freedom of choice, is to pick up a point from the contour curve of P_i. The points Q_i should be chosen such that the angle between the tangent line at P_i to the curve $\mathcal{C}_{Q_i}$ and the line connecting P_i and Q_i is not too small. We can choose several sample points from the contour curve $\mathcal{C}^c_{P_i}$, compute the mentioned angle, and choose the point for which the computed angle is the biggest.

Bibliography

Abhyankar, S. S., Bajaj, C. L., 1988. Automatic parameterization of rational curves and surfaces. III. Algebraic plane curves. Comput. Aided Geom. Design 5 (4), 309–321.

Abhyanker, S. S., Bajaj, C., 1987. Automatic parametrization of rational curves and surfaces. II. Cubics and cubicoids. Comput. Aided Des. 19 (9), 499–502.

Aigner, M., Jüttler, B., Kim, M.-S., 2004. Analyzing and enhancing the robustness of implicit representations. In: Geometric modelling and Processing. IEEE Press, pp. 131–142.

Alefeld, G., Herzberger, J., 1983. Introduction to interval computations. Computer Science and Applied Mathematics. Academic Press Inc. [Harcourt Brace Jovanovich Publishers], New York, translated from the German by Jon Rokne.

Bajaj, C. L., Chen, J., Xu, G., 1995. Modeling with cubic a-patches. ACM Trans. Graph. 14 (2), 103–133.

Bajaj, C. L., Holt, R. J., Netravali, A. R., 1998. Rational parametrization of non-singular real cubic surfaces. ACM Transactions on Graphics 17, 1–31.

Berry, T. G., Patterson, R. R., 2001. Implicitization and parametrization of nonsingular cubic surfaces. Comput. Aided Geom. Design 18 (8), 723–738.

Bruce, J. W., Wall, C. T. C., 1979. On the classification of cubic surfaces. J. London Math. Soc. (2) 19 (2), 245–256.

Brundu, M., Logar, A., 1998. Parametrization of the orbits of cubic surfaces. Transform. Groups 3 (3), 209–239.

Buchberger, B., 1988. Applications of Gröbner bases in nonlinear computational geometry. In: Trends in computer algebra. Springer, Berlin, pp. 52–80.

Busé, L., 2001. Residual resultant over the projective plane and the implicitization problem. In: Proc. ISSAC. ACM, New York, pp. 48–55.

Busé, L., Chardin, M., 2005. Implicitizing rational hypersurfaces using approximation complexes. To appear in the J. of Symbolic Computation.

Busé, L., Cox, D., D'Andrea, C., 2003a. Implicitization of surfaces in $\mathbb{P}^3$ in the presence of base points. J. Algebra Appl. 2 (2), 189–214.

Busé, L., Elkadi, M., Mourrain, B., 2000. Generalized resultants over unirational algebraic varieties. J. Symbolic Comput. 29 (4-5), 515–526, symbolic computation in algebra, analysis, and geometry (Berkeley, CA, 1998).

Busé, L., Elkadi, M., Mourrain, B., 2003b. Using projection operators in computer aided geometric design. In: Topics in algebraic geometry and geometric modeling. Vol. 334 of Contemp. Math. Amer. Math. Soc., Providence, RI, pp. 321–342.

C. Bajaj, T. Garrity, J. W., 1988. On the Applications of Multi-Equational Resultants. Technical Report 826 11/21/88, Purdue University, Computer Sciences.

Canny, J., Grigorev, D. Y., Vorobjov, J. N. N., 1992. Finding connected components of a semialgebraic set in subexponential time. Appl. Algebra Engrg. Comm. Comput. 2 (4), 217–238.

Castro, D., Montaña, J. L., Pardo, L. M., San Martín, J., 2002. The distribution of condition numbers of rational data of bounded bit length. Found. Comput. Math. 2 (1), 1–52.

Cayley, A., 1869. A memoir on cubic surfaces. Phil. Trans. Roy. Soc., 159, 231–326.

Chuang, J. H., Hoffmann, C. M., 1989. On local implicit approximation and its applications. ACM Trans. Graph. 8 (4), 298–324.

Cipolla, R., Giblin, P., 2000. Visual motion of curves and surfaces. Cambridge University Press, Cambridge.

Corless, R. M., Giesbrecht, M. W., Kotsireas, I. S., Watt, S. M., 2001. Numerical implicitization of parametric hypersurfaces with linear algebra. In: AISC 2000. LNCS. Springer, Berlin, pp. 174–183.

Coste, M., Roy, M.-F., 1988. Thom's lemma, the coding of real algebraic numbers and the computation of the topology of semi-algebraic sets. J. Symbolic Comput. 5 (1-2), 121–129.

Cox, D., Little, J., O'Shea, D., 1997. Ideals, varieties, and algorithms, 2nd Edition. Undergraduate Texts in Mathematics. Springer-Verlag, New York.

Cox, D., Little, J., O'Shea, D., 1998. Using algebraic geometry. Vol. 185 of Graduate Texts in Mathematics. Springer-Verlag, New York.

Dahmen, W., 1989. Smooth piecewise quadric surfaces. In: Mathematical methods in computer aided geometric design (Oslo, 1988). Academic Press, Boston, MA, pp. 181–193.

de Montaudouin, Y., Tiller, W., Vold, H., 1986. Applications of power series in computational geometry. Comput. Aided Design 18 (10), 514–524.

Dokken, T., 1997. Aspects of intersection algorithms and approximation. Ph.D. thesis, University of Oslo, Norway.

Dokken, T., 2001. Approximate implicitization. In: Mathematical methods for curves and surfaces. Vanderbilt Univ. Press, Nashville, TN, pp. 81–102.

Dokken, T., Kellermann, H. K., Tegnander, C., 2001. An approach to weak approximate implicitization. In: Mathematical methods for curves and surfaces. Vanderbilt Univ. Press, Nashville, TN, pp. 103–112.

Dokken, T., Thomassen, J. B., 2003. Overview of approximate implicitization. In: Topics in algebraic geometry and geometric modeling. Vol. 334 of Contemp. Math. Amer. Math. Soc., Providence, RI, pp. 169–184.

Emiris, I. Z., Canny, J. F., 1995. Efficient incremental algorithms for the sparse resultant and the mixed volume. J. Symbolic Comput. 20 (2), 117–149.

Emiris, I. Z., Verschelde, J., 1999. How to count efficiently all affine roots of a polynomial system. Discrete Appl. Math. 93 (1), 21–32.

Farin, G., Hoschek, J., Kim, M.-S. (Eds.), 2002. Handbook of computer aided geometric design. North-Holland, Amsterdam.

Farouki, R. T., 1991. On the stability of transformations between power and Bernstein polynomial forms. Comput. Aided Geom. Design 8 (1), 29–36.

Fix, G., Hsu, C.-P., Luo, T., 1996. Implicitization of rational parametric surfaces. J. Symbolic Comput. 21 (3), 329–336.

Gao, X.-S., Chou, S.-C., 1992. Implicitization of rational parametric equations. J. Symbolic Comput. 14, 459–470.

Gonzalez-Vega, L., 1997. Implicitization of parametric curves and surfaces by using multidimensional Newton formulae. J. Symbolic Comput. 23 (2-3), 137–151.

Gonzalez-Vega, L., Necula, I., 2002. Efficient topology determination of implicitly defined algebraic plane curves. Comput. Aided Geom. Design 19 (9), 719–743.

Goodman, J. E., O'Rourke, J. (Eds.), 2004. Handbook of discrete and computational geometry, 2nd Edition. Discrete Mathematics and its Applications (Boca Raton). Chapman & Hall/CRC, Boca Raton, FL.

Guo, B., 1991. Surface generation using implicit cubics, 485–503.

Hartshorne, R., 1977. Algebraic geometry. Springer-Verlag, New York, graduate Texts in Mathematics, No. 52.

Henderson, A., 1960. The twenty-seven lines upon the cubic surface. Hafner, New York.

Higham, N. J., 2002. Accuracy and Stability of Numerical Algorithms, 2nd Edition. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Hoschek, J., Lasser, D., 1993. Fundamentals of computer aided geometric design. AK Peters, Wellesley, MA.

Hunt, B., 1996. The geometry of some special arithmetic quotients. Vol. 1637 of Lecture Notes in Mathematics. Springer-Verlag, Berlin.

Kendig, K., 1977. Elementary algebraic geometry. Springer, New York.

Manin, Y. I., 1986. Cubic forms. North-Holland, Amsterdam.

Manocha, D., Canny, J. F., 1992. Implicit representation of rational parametric surfaces. J. Symbolic Comput. 13 (5), 485–510.

Pérez-Díaz, S., Sendra, J., Sendra, J. R., 2004. Parametrization of approximate algebraic curves by lines. Theoret. Comput. Sci. 315 (2-3), 627–650.

Pérez-Díaz, S., Sendra, J., Sendra, J. R., 2005. Parametrization of approximate algebraic surfaces by lines. Comput. Aided Geom. DesignArticle in press.

Prautzsch, H., Boehm, W., Paluszny, M., 2002. Bézier and B-spline techniques. Mathematics and Visualization. Springer-Verlag, Berlin.

Reid, M., 1988. Undergraduate algebraic geometry. Vol. 12 of London Mathematical Society Student Texts. Cambridge University Press, Cambridge.

Schicho, J., 1998a. Rational parameterization of real algebraic surfaces. In: Proc. ISSAC. ACM, New York, pp. 302–308.

Schicho, J., 1998b. Rational parametrization of surfaces. J. Symbolic Comput. 26 (1), 1–29.

Schirra, S., 2000. Robustness and precision issues in geometric computation. In: Handbook of computational geometry. North-Holland, Amsterdam, pp. 597–632.

Schläfli, L., 1858. An attempt to determine the twenty-seven lines upon a surface of the third order and to divide such surfaces into species in reference to the reality of the lines upon the surface. Quarterly Journal for Pure and Applied Mathematics II, 55–66.

Sederberg, T., Anderson, D., Goldman, R., 1984. Implicit representation of parametric curves and surfaces. Computer Vision, Graphics, and Image Processing 28, 72–84.

Sederberg, T. W., 1984. Degenerate parametric curves. Comput. Aided Geom. Design 1 (4), 301–307.

Sederberg, T. W., 1985. Piecewise algebraic surface patches. Comput. Aided Geom. Design 2 (1-3), 53–59.

Sederberg, T. W., 1986. Improperly parameterized rational curves. Comput. Aided Geom. Design 3 (1), 67–75.

Sederberg, T. W., 1990a. Techniques for cubic algebraic surfaces. I. IEEE Computer Graphics and Applications 10 (4), 14–25.

Sederberg, T. W., 1990b. Techniques for cubic algebraic surfaces. II. IEEE Computer Graphics and Applications 10 (5), 12–21.

Sederberg, T. W., Chen, F., 1995. Implicitization using moving curves and surfaces. In: SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques. pp. 301–308.

Sederberg, T. W., Snively, J. P., 1987. Parametrization of cubic algebraic surfaces. In: The mathematics of surfaces, II. Oxford Univ. Press, pp. 299–319.

Sederberg, T. W., Zheng, J., Klimaszewski, K., Dokken, T., 1999. Approximate implicitization using monoid curves and surfaces. Graph. Models Image Process. 61 (4), 177–198.

Segre, B., 1942. The Non-singular Cubic Surfaces. Oxford University Press.

Sendra, J. R., Winkler, F., 2001. Tracing index of rational curve parametrizations. Comput. Aided Geom. Design 18 (8), 771–795.

Shafarevich, I. R. (Ed.), 1999. Algebraic geometry. Encyclopaedia of Mathematical Sciences. Springer-Verlag.

Shou, H., Lin, H., Martin, R., Wang, G., 2003. Modified affine arithmetic is more accurate than centred interval arithmetic or affine arithmetic. In: The Mathematics of Surfaces X. Vol. 2768 of LNCS. Springer, Heidelberg, pp. 355–365.

Swinnerton-Dyer, H. P. F., 1970. The birationality of cubic surfaces over a given field. Michigan Math. J. 17, 289–295.

Trefethen, L. N., Bau, D. I., 1997. Numerical linear algebra. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA.

Wang, D., 2004. A simple method for implicitizing rational curves and surfaces. J. Symbolic Comput. 38 (1), 899–914.

Zheng, J., Sederberg, T. W., Chionh, E.-W., Cox, D. A., 2003. Implicitizing rational surfaces with base points using the method of moving surfaces. In: Topics in algebraic geometry and geometric modeling. AMS, Providence, RI, pp. 151–168.

Index

Curriculum Vitae

Ibolya Szilágyi

Personal Data

Date of Birth:	17.05.1977.
Place of Birth:	Debrecen, Hungary
Nationality:	Hungarian
Office:	Castle of Hagenberg, A-4232 Hagenberg i.M., Austria
Phone:	+43 732 2468 9950
Fax:	+43 732 2468 9930
E-mail:	szibolya@risc.uni-linz.ac.at
WWW:	http://www.risc.uni-linz.ac.at

Education

1991 – 1995:	High School, Debrecen, Hungary
1995 – 2000:	Studies in Mathematics, University of Debrecen
2001 – 2005:	Ph.D. studies at Research Institute for Symbolic Computation, University of Linz, Austria. Dissertation: "Symbolic-Numeric Techniques for Cubic Surfaces", advisor: Dr. Josef Schicho.

Career History

1999 – 2000:	Junior teaching assistant, University of Debrecen.
2000 – 2001:	Teaching Assistant, University of Debrecen.
2002 – 2004:	Research Assistant at RISC-Linz. Member of the project SFB013 "Proving and Solving over the Reals".
2004 – 2005:	Member of the group 'Symbolic Computation" at Johann Radon Institute of Computational and Applied Mathematics, Austrian Academy of Sciences.

Research Interests

Algebraic Geometry, Geometric Modelling, Computer Algebra.

Career Related Activities

2003 – 2004:	Member of the publicity committee of "International Symposium for Symbolic and Algebraic Computation" (ISSAC) 2004.

Publications

1. J. Schicho and I. Szilágyi. Numerical Stability of Surface Implicitization. To appear in Journal of Symbolic Computation, 2005.

2. I. Szilágyi, B. Jüttler and J. Schicho. Local Parametrization of Cubic Surfaces. To appear in Journal of Symbolic Computation, 2005.

3. J. Schicho and I. Szilágyi. Numerical Stability of Surface Implicitization. SFB-Report 2004-27, J. Kepler University, Linz, 2004.

4. I. Szilágyi, B. Jüttler and J. Schicho. Local Parametrization of Cubic Surfaces. SFB-Report 2004-31, J. Kepler University, Linz, 2004.

5. M. Aigner, I. Szilágyi, B. Jüttler and J. Schicho. Implicitization and Distance Bounds. To appear in Algebraic Geometry and Geometric Modelling (Mathematics and Visualization, Springer-Verlag), 2005.

Conference talks

1. Implicitization and Parametrization of nonsingular cubic surfaces. SFB Statusseminar, Strobl, Austria, April 24–26, 2003.

2. Symbolic-Numeric Techniques for Cubic Surfaces. International Workshop on "Numerical and Symbolic Scientific Computing", Strobl, Austria, June 16–19, 2003.

3. Symbolic-Numeric Techniques for Cubic Surfaces. Computational Methods for Algebraic Spline Surfaces (COMPASS), Kefermarkt, Austria, October 2, 2003.

4. Implicitization and Parametrization Using Symbolic-Numeric Techniques. International Conference in Applied Informatics, Eger, Hungary, January 28–30, 2004.

5. Workshop on algebraic geometry and singularities, Schwaz, Marc 10–13, 2004.

6. Local parametrization of cubic surfaces. SFB Statusseminar, Strobl, Austria, April 5–7, 2004.

7. Local parametrization of cubic surfaces. Geometri Tagung, Vorau, Austria, June 7–11, 2004.

8. J. Schicho, I. Szilágyi, “Numerical Stability of Surface Implicitization”-poster presentation. ISSAC-International Symposium on Symbolic and Algebraic Computation, Santander, Spain, July 4–7, 2004.
9. Workshop on algebraic geometry and singularities, Gmunden, September 16–18, 2004.
10. Local parametrization of cubic surfaces. Algebraic Geometry and Geometric Modelling, Nice, France, September 27–29, 2004.
11. I. Szilágyi, M. Aigner, “Implicitization and Distance Bounds”. SFB Statusseminar, Strobl, Austria, April, 2005.

Invited talks

1. Local parametrization of cubic surfaces. University of Debrecen, Hungary, April 16, 2004.
2. Numerical Stability of Surface Implicitization. University of Debrecen, Hungary, December 17, 2004.