



Technisch-Naturwissenschaftliche
Fakultät

Fat Arcs and Fat Spheres for Approximating Algebraic Curves and Solving Polynomial Systems

DISSERTATION

zur Erlangung des akademischen Grades

Doktorin

im Doktoratsstudium der

Naturwissenschaften

Eingereicht von:
Szilvia Béla

Angefertigt am:
Doctoral Program „Computational Mathematics“

Beurteilung:
Univ.- Prof. Bert Jüttler (Betreuung)

Assoc. Prof. Jens Gravesen

Linz, März 2011

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 benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Linz, März 2011

Szilvia Béla

Abstract

Studying objects defined by algebraic equations has been an active research area for a long time. The reason for the interest is the wide variety of applications, which appear in mathematical modeling and physics. Modeling algebraic objects is an essential ingredient of free-form surface visualization and numerical simulations. Thus modeling algorithms are frequently used in CAD-systems, manufacturing, robotics etc. Several problems in applications are described by multivariate polynomial systems with a low dimensional solution set. In the thesis we present a method to generate bounding regions for one- or zero-dimensional solution sets of multivariate polynomial systems.

The one-dimensional solution set of a multivariate polynomial system forms an algebraic curve. These curves are defined as the intersection curves of algebraic surfaces. Representing these algebraic curves is a fundamental problem of some geometric algorithms. For instance such algebraic curves appear as the boundary curves of surfaces created by Boolean operations or the self-intersection curves of surfaces. Due to the importance of these curves several algorithms have been introduced to approximate them, especially for curves embedded in lower dimensional spaces. We formulate in the thesis a new geometrical method, which approximates one-dimensional algebraic sets. The algorithm generates a set of quadratic regions, the so called “fat arcs” , which encloses the algebraic curve within a user specified tolerance. We describe different methods, how to generate these bounding regions, and we study their behavior. Then we combine the fat arc generation with the standard subdivision technique.

The computation of zero-dimensional solution sets of multivariate polynomial systems has also several applications in algebra and geometry. Therefore various methods exist to find or to isolate the roots of polynomial systems. They use symbolic, numeric or combined techniques in order to find the solutions. In the end of the thesis we generalize the definition of fat arcs to the concept of fat spheres. We introduce an iterative domain reduction method based on fat sphere generation. This method generates sequences of bounding regions, which converge with order three to the single roots of a multivariate polynomial system.

Zusammenfassung

Analyse und Bearbeitung von Objekten aus der reellen algebraischen Geometrie sind seit langem ein bedeutendes Forschungsfeld. Ein Grund dafür sind ihre vielfältigen Anwendungen, welche unter anderem in der mathematischen Modellierung und Physik auftreten. Methoden zur Modellierung algebraischer Objekte sind für die Darstellung von Freiformflächen und numerischen Simulationen von essentieller Bedeutung. Dementsprechend finden diese Methoden Anwendung in CAD-Systemen, in der industriellen Fertigung, der Robotik, etc. Viele Probleme werden dabei in Form multivariater polynomieller Systeme mit niedrigdimensionaler Lösungsmenge dargestellt. In dieser Arbeit präsentieren wir eine Methode zur Erstellung von „bounding regions“ für ein- und nulldimensionale Lösungsmengen multivariater polynomieller Systeme.

Die eindimensionale Lösungsmenge eines multivariaten polynomiellen Systems bildet eine algebraische Kurve. Diese Kurven können als Schnittkurven algebraischer Flächen betrachtet werden. Die Darstellung dieser algebraischen Kurven ist ein fundamentales Problem der algorithmischen Geometrie. Solche Kurven entstehen zum Beispiel als Randkurven von Flächen bei Anwendung boolescher Operationen oder als Selbstschnitte von Flächen. Aufgrund ihrer großen Bedeutung existieren bereits viele Algorithmen zur Approximation algebraischer Kurven, speziell Kurven eingebettet in niedrigdimensionale Räume. Wir formulieren in dieser Arbeit eine neue geometrische Methode, die eindimensionale, algebraische Mengen approximiert. Der Algorithmus erzeugt Regionen zweiten Grades, sogenannte „fat arcs“, die die algebraische Kurve unter Berücksichtigung einer vorgegebenen Toleranz abdecken. Wir beschreiben verschiedene Methoden diese „bounding regions“ zu erzeugen und analysieren deren Verhalten. Weiters vereinen wir die „fat arc“ Erzeugung mit der Subdivisionsmethode.

In der Algebra und in der Geometrie hat das Auffinden nulldimensionaler Lösungsmengen multivariater polynomieller Systeme zahlreiche Anwendungen. Daher existieren viele Methoden solche Lösungen polynomieller Systeme zu finden oder zu isolieren. Diese Methoden verwenden symbolische, numerische oder kombinierte Techniken zum Auffinden der Lösungen. Am Ende dieser Arbeit verallgemeinern wir die Definition der „fat arcs“ zum Konzept der „fat spheres“. Wir führen eine iterative Gebietunterteilungsmethode ein, die auf „fat spheres“ basiert. Diese Methode erzeugt „bounding regions“, die in dritter Ordnung gegen die einfachen Wurzeln multivariater polynomieller Systeme konvergieren.

Összefoglalás

Az algebrai felületek és görbék vizsgálata már husszú ideje igen aktív kutatási terület. Ennek oka, hogy a matematikai modellezés és a fizika területén számos alkalmazásuk ismert. Az algebrai objektumok modellezése fontos összetevője a felületek megjelenítésének és bizonyos numerikus szimulációknak. Ennek megfelelően gyakran találkozhatunk különböző modellező algoritmusokkal CAD-rendszerekben, gyártási folyamatok során, a robotikában stb. A gyakorlatban számos probléma írható le olyan többváltozós polinomrendszerek segítségével, melyek megoldástere alacsony dimenziós. A következőkben egy olyan módszert ismertetünk, amely többváltozós polinomrendszerek egy- vagy nulldimenziós megoldáshalmazát közelíti úgynevezett határoló területek (bounding regions) segítségével.

Egy többváltozós polinom-egyenletrendszer egydimenziós megoldáshalmazát algebrai görbét határoz meg. Az ilyen görbék mint algebrai felületek metszéspontjai állnak elő. Néhány geometriai algoritmusnak alapvető építőköve ezen görbék leírása. Ilyen görbék például a Bool-féle műveletek segítségével előállított felületek határgörbái vagy önátmetsző felületek metszéspontjai is. Fontosságuknak köszönhetően ilyen görbék közelítésére számos algoritmus ismert, különösen alacsony dimenziós terekbe ágyazott görbére. A disszertációban egy olyan új geometriai módszert mutatunk be, amely segítségével egydimenziós algebrai sokaságokat közelíthetünk. Az új algoritmus kvadratikus határoló területeket, úgynevezett „vastagított íveket” (fat arcs) számol, melyek magukba foglalják az algebrai görbét, mindamelllett átmérőjük nem halad meg egy előre megadott hibahatárt. A disszertációban több különböző módszert is ismertetünk a vastagított ívek számolására, és vizsgáljuk ezek különböző tulajdonságait is. Végül kombináljuk a határoló területek számolását az algebrai görbék felosztásával.

Többváltozós polinomrendszerek nulldimenziós megoldáshalmazának kiszámítása az algebra és a geometria számos alkalmazásában fontos szerepet játszik. Ezért több különböző módszer is ismert polinom-egyenletrendszerek gyökeinek kiszámítására és szétválasztására. Ezek az algoritmusok szimbólikus, numerikus vagy vegyes megoldási technikákat alkalmaznak a megoldások keresésekor. A disszertáció utolsó fejezetében általánosítjuk a vastagított ívek definícióját, és bevezetjük a vastagított gömb fogalmát. Bemutatunk egy olyan iteratív algoritmust, mely vastagított gömböket használ a gyököket közelítő határoló területek csökkentésére. Ez a módszer régiók olyan sorozatával közelíti az egyes megoldásokat, amely harmadrendben konvergál a többváltozós egyenletrendszer egyszeres gyökeihez.

Acknowledgments

I would like to thank my supervisor Bert Jüttler, for the opportunities he provided me and for the several helpful suggestions he improved my work. In addition, I express my gratitude to Jens Gravesen and Bernard Mourrain, who supported my visit and work at their institute. My special thanks also goes to Prof. Peter Paule, the head of the Doctoral Program and the Research Program SFB. This work has been supported by the Austrian Science Found (FWF) in the frame of the Special Research Program SFB F013 and the Doctoral Program “Computational Mathematics” .

I would like to give my thanks to all my working colleagues at the Institute of Applied Geometry, Doctoral Program, the project SFB and the institutes I have visited as a guest. I feel lucky to have met you! You have helped me a lot, and I spend very nice time with you during my PhD studies, I am grateful.

Last, but not least I say thanks to my family members for their support and encouragement.

Szilvia Béla
Linz, March 2011.

Contents

1	Introduction	1
1.1	Polynomial Solvers	1
1.2	Outline	2
2	Fat Arcs for Implicitly Defined Planar Curves	5
2.1	Fat Arcs in 2D	5
2.2	Fat Arc Generation with Parametric Arcs	8
2.3	Fat Arc Generation with Implicitly Defined Arcs	15
2.4	Comparison of the Methods	26
2.5	Examples and Application	29
3	Fat Arcs for 3D Implicit Algebraic Curves	35
3.1	Fat Arcs in 3D	35
3.2	Approximation of Regular Curve Segments	37
3.3	Median Arc Generation	40
3.4	Distance Estimate	49
3.5	Convergence	52
3.6	Fat Arc Generation for Algebraic Space Curves	58
4	Fat Arcs for Implicitly Defined Algebraic Curves	67
4.1	Generalized Fat Arcs	67
4.2	Approximation of Regular Curve Segments	69
4.3	Median Arc Generation	71
4.4	Distance Estimate	83
4.5	Convergence and Global Algorithm	86
5	Fat Spheres for Solving Multivariate Polynomial Systems	93
5.1	Fat Arcs and Fat Spheres	93
5.2	Bounding Region Generation	95
5.3	Convergence Rate for Single Roots	103
5.4	Iterative Domain Reduction Algorithm	108
6	Conclusion	119

Chapter 1

Introduction

1.1 Polynomial Solvers

Solving multivariate polynomial systems is a key problem in algebra and geometry. It has several applications, therefore various methods exist to compute the solution sets of polynomial systems. These methods are using symbolic, numeric or combined techniques in order to find the solutions.

Representing algebraic curves is a fundamental problem of geometric computing. Implicitly defined algebraic curves can be described as the intersection curves of algebraic surfaces. Computation of such a surface-surface intersection is a basic operation in geometric modeling.

Intersecting low degree implicitly defined surfaces has attracted a lot of interest in the literature. Quadratic surfaces are the simplest curved surfaces, therefore they are frequently used in computational geometry. The intersection computation of such surfaces has been discussed thoroughly in [8, 11, 40, 45, 46].

Several different methods have been developed for computing the intersection of algebraic surfaces (see [19, 31, 39]). Many of them are symbolic-numeric algorithms. The most widely used numeric methods are the lattice evaluation, tracing and subdivision-based methods.

Lattice evaluation techniques generate a set of low dimensional sub-problems. The solutions of these sub-problems are interpolated to approximate the general solution. Marching or tracing methods generate point sequences along the connected components of the curve. They necessarily use some topological information to find starting, turning and singular points [3, 10, 16, 22].

Subdivision algorithms decompose the problem into several sub-problems, and sort these problems according to the curve topology [2, 25]. The decomposition terminates if suitable approximating primitives can be generated for each sub-problem [29]. In order to construct these approximating primitives several local approximation techniques can be applied, such as interpolation, bounding region generation or least-squares approximation.

Real root finding is considered as a difficult task. It is an important problem, therefore several methods were developed to solve it. A general overview about the multivariate root finding algorithms is given in [13, 38]. The solvers described in the literature are using either algebraic or geometric tools.

Algebraic approaches, such as Gröbner-basis technique [5], resultant based methods or continuous fractions methods assure exact and efficient solution algorithms. These algorithms frequently provide more information about the solutions than it is needed in the applications. It is often unnecessary to compute all solutions. For instance, CAD-systems usually require information only about real solutions, which lie in a certain domain. Moreover these symbolic methods are not really suitable for numerical computations. An algebraic solver, which is using the Gröbner-basis technique, was developed for instance by Rouillier [33] for bi-variate polynomial systems. Busé et al. considered resultant based methods in [6, 7]. In [14] an algebraic method is described, which is using Sturm-Habicht sequences.

Homotopy solvers compute a family of root-finding problems. The method transforms a simple problem to the original one in several steps, and computes the roots of each intermediate problem. The computed sequence of roots converges to the solutions of the original root-finding problem. However such computations usually require inefficient memory and time. Polynomial solvers based on homotopy methods are described in [24, 28].

In order to develop robust approximation algorithms a great leap forward was to use Bernstein-Bézier polynomials. The stability of this representation form allows to develop algorithms for approximating algebraic sets given in higher dimensional space. The first general numerical algorithms using polynomials given in BB-form were developed by Sherbrooke and Patrikalakis [39]. These are subdivision methods for finding zero dimensional solution sets of multivariate polynomial equations.

Subdivision algorithms are based on the “divide and conquer” paradigm. They compute in a certain domain (usually in an axis-aligned box), and provide information only about real-roots. If we are interested in certain properties of a root, like multiplicity, then further computations are necessary. Subdivision algorithms decompose the problem into several sub-problems. The decomposition terminates if suitable bounding primitives can be generated in each sub-problem [29]. In order to construct these approximating primitives several local domain reduction strategies can be applied. The first subdivision solvers were developed by Sederberg et al. for bivariate Bézier-polynomials. They are using clipping and subdivision techniques [35, 36]. Later on a family of algorithms was invented, which is using projection techniques [39]. The most recently developed solvers are published by Mourrain et al. [13] and Elber et al. [12].

1.2 Outline

In the thesis we present a method to bound one- or zero-dimensional solution sets of multivariate polynomial systems. In order to approximate the solution of such polynomial systems we put the emphasis on the geometrical properties of them. We develop numeric algorithms based on a new bounding region generation method and the standard subdivision technique. We introduce an algorithm, which generates a set of quadratic bounding regions for implicitly defined algebraic curves. Later these regions are generalized to bound implicitly defined surfaces. Computing the intersection of these bounding regions leads to a technique, which generates bounding domains around the real roots of multivariate polynomial systems.

In the next chapter we describe a method, which generates bounding regions for implicitly defined planar curves. This method is using special bounding primitives, the so called “fat arcs”. The construction of fat arcs was introduced by Sederberg [37] to approximate planar parametric Bézier curves. We generalize this definition in order to bound implicitly defined

curves. We present several different techniques to generate fat arcs in \mathbb{R}^2 . After discussing the main steps and the properties of these techniques we compare them. Then we combine the local fat arc generation with iterative subdivision. In the end of the chapter we present several examples and show an application.

In Chapter 3 we present an algorithm, which generates bounding regions for implicitly defined algebraic space curves. The method is the generalization of the fat arc generation technique from Chapter 2. We analyze the properties of the curve approximation method in the three-dimensional case, and combine it with subdivision technique. Finally we present some examples and applications of the method.

The three-dimensional fat arc generation technique can also be generalized to implicitly defined algebraic curves represented in the n -dimensional space. In the fourth chapter we present an algorithm, which generates fat arcs to bound one-dimensional algebraic sets in \mathbb{R}^n . We describe also the general properties of the technique.

In Chapter 5 we introduce fat spheres as multidimensional bounding regions for implicitly defined algebraic objects. Then we describe a local domain reduction strategy to bound intersection points of algebraic objects. We combine this local algorithm with iterative subdivision. This hybrid algorithm can be applied for approximating the real roots of multivariate polynomial systems. In the end of the chapter we present several two- and three-dimensional examples.

Finally we summarize the results of the thesis in Chapter 6.

Chapter 2

Fat Arcs for Implicitly Defined Planar Curves

The approximation of implicitly defined curves is one of the fundamental problems in computational geometry. In this chapter we present two techniques, which generate bounding regions for implicitly defined curves. Both methods are the generalization of an approximation technique for planar parametric curves using special bounding primitives. After discussed the main steps and the properties of the algorithms, we compare them. In the end of the chapter we present several examples and show an application.

2.1 Fat Arcs in 2D

In this section first we give a short overview of the related work on planar curve approximation. Then we introduce fat arcs, which are special type of bounding primitive for planar parametric curves. In order to generalize the definition of these primitives to implicitly defined curves, we describe the planar algebraic curves as the zero level set of polynomials in Bernstein-Bézier tensor product form.

2.1.1 Bounding Region Generation

Bounding regions, which enclose segments of planar curves, are frequently needed for various geometric computations, e.g., for solving the intersection problem between two planar curves. Axis-aligned bounding boxes (min-max boxes), which can easily be generated both for planar parametric curves and for implicitly defined curves, are one of the simplest instances. Other useful primitives include fat lines (bounding strips, see e.g. [4]), the convex polygons obtained as a convex hull of the control polygons or fat arcs [37].

The performance of a bounding region depends on the approximation order. For a bounding primitive with approximation order k the number of primitives needed to bound a curve with a given tolerance ε grows like $\sqrt[k]{1/\varepsilon}$. Consequently, the use of geometric primitives with higher approximation order may provide computational advantages. Bounding boxes have only approximation order $k = 1$, while both the convex hull of control polygons and fat lines provide approximation order 2, and fat arcs even have approximation order 3.

Clearly, it is possible to define bounding regions with an even higher approximation order. Fat arcs seem to be particularly useful since they provide a reasonable trade-off between

geometric flexibility and the computational simplicity of elementary geometric operations. For instance, the computation of the intersection of two circular arcs requires solely the solution of quadratic equations, while this becomes far more complicated for higher order objects.

Various methods have been described in the literature for generating an arc spline curve which approximates a given parametric curve with a prescribed tolerance, see e.g. [48] for many related references. The use of arc splines for geometric design applications can be traced back to a classical VTO report of Sabin [34]. Marciniak and Putz dealt with the minimization of the number of arcs to approximate a curve under a give tolerance [27]. Later Qiu et al. improved their method [32]. In a number of papers, Meek and Walton applied arc splines to approximate parametric curves [42, 43, 44] Yong used arc splines for quadratic Bézier curve approximation [49]. Feichtinger et al. compared various biarc interpolation schemes [41]. Held and Eibl approximated with biarcs simple planar polygons either for symmetric and asymmetric tolerance bounds [18].

2.1.2 Fat Arcs for Planar Parametric Curves

The construction of fat arcs was introduced by Sederberg [37]. He developed a method to approximate planar parametric Bézier curves. His method generates a set of bounding regions, each consisting of an approximating circular arcs with some finite thickness.

The approximating arc – the so called median arc – is usually defined by three points from the parametric curve. These interpolation points can be chosen in various ways [26, 37], for instance as the two endpoints of the curve segment and the intersection point of the bisector of the endpoints and the curve segment. Of course, any other approximating arc generation technique can be used for median arc generation.

The next step of the method is to measure the distance between the curve and the median arc. Frequently an upper bound is used to estimate the distance. An offset of the median arc can be defined with this distance bound. The boundaries of the offset are concentric arcs, whose radii are the sum and the difference of the median arc radius and the distance bound. This offset is a part of an annulus, and it defines a bounding region for the original curve segment.

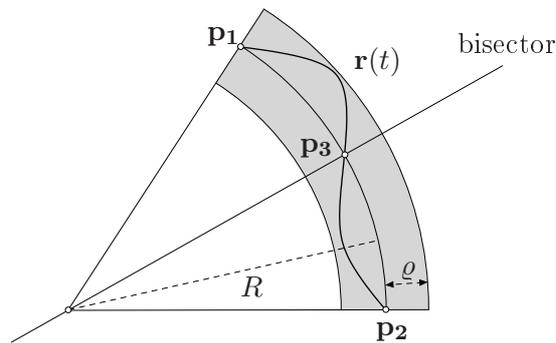


Figure 2.1: Fat arc generation for planar parametric curve.

Since the approximation order of circular arcs is equal to three, the offset distance behaves as $O(h^3)$, where h is the length of the given curve segment. So the method is simple, and it

has a relatively high convergence rate.

All existing algorithms for fat arc generation deal exclusively with parametric curves. Our aim is to apply this method to implicitly defined curves. Although the steps of the algorithm are more complicated in the implicit case, the expected convergence rate is the same as for the parametric curves. So we implemented different approximation and distance bounding techniques to get a fast and accurate computational method.

2.1.3 Regular Planar Algebraic Curves

In order to construct fat arcs for planar algebraic curves, we assume that the bivariate polynomial f defining the curve is given by its tensor-product Bernstein-Bézier (BB) representation with respect to the rectangular domain $\Omega_0 = [\alpha_1, \beta_1] \times [\alpha_2, \beta_2]$.

$$f(x, y) = \sum_{i=0}^m \sum_{j=0}^n d_{ij} B_{i,m}^1(x) B_{j,n}^2(y), \quad (2.1)$$

with certain coefficients $d_{ij} \in \mathbb{R}$, where

$$B_{i,n}^k(t) = \binom{n}{i} \left(\frac{t - \alpha_k}{\beta_k - \alpha_k} \right)^i \left(\frac{\beta_k - t}{\beta_k - \alpha_k} \right)^{n-i}, \quad t \in [\alpha_k, \beta_k]. \quad (2.2)$$

The curve is given as the zero set of the bivariate polynomial

$$\mathcal{C}(f, \Omega_0) = \{(x, y) : f(x, y) = 0\} \cap \Omega_0. \quad (2.3)$$

Clearly, the curve may be an empty point set, or it may consist of more than one curve segment. In order to control the behavior of the curve in the computational domain we use the following definition.

Definition 2.1. *A point \mathbf{p} of an algebraic curve $\mathcal{C}(f, \Omega)$ is called singular in the domain $\Omega \subseteq \Omega_0$, if the gradient vector $\nabla f(\mathbf{p})$ is zero (and called regular otherwise). A curve segment is regular, if any point of the segment is regular.*

A regular curve consists of one or more single branches of the curve without any self-intersection or loops.

Observation 2.2. A general lower bound can be given for the gradient length in any point (x, y) of a domain Ω with using the BB-representation of the polynomial f . The tensor-product BB-representation of the square of the gradient length is

$$\left(\frac{\partial f(x, y)}{\partial x} \right)^2 + \left(\frac{\partial f(x, y)}{\partial y} \right)^2 = \sum_{i=0}^{2m} \sum_{j=0}^{2n} h_{ij} B_i^{2m}(x) B_j^{2n}(y). \quad (2.4)$$

It can be found using the differentiation, product and degree elevation formulas of BB-polynomials (see [19]). This representation provides us a general lower bound for the gradient length

$$\|\nabla f(\mathbf{x})\| \geq \sqrt{\max\{0, \min_{i,j} h_{ij}\}} = G. \quad (2.5)$$

If G is non-zero, then the gradient does not vanish in the domain Ω . This implies also, that the curve $\mathcal{C}(f, \Omega)$ is regular in the domain Ω .

2.2 Fat Arc Generation with Parametric Arcs

In this section we describe a fat arc generation method for implicitly defined curves. The algorithm is based on a technique, which is similar to the original construction for planar parametric curves. It generates interpolating circular arcs as median arcs, and computes fat arc thickness with the help of one-sided Hausdorff distance.

2.2.1 Topological Criterion

In order to generate fat arcs with interpolation technique, we need to detect domains containing only one segment of the implicitly defined curve. Various criteria have been discussed in the literature for isolating a single segment of an algebraic curve. For instance, different types of discriminating curve families have been used in [47]. These discriminating families are particularly useful in combination with algorithms that trace the algebraic curve segments.

We are interested in a criterion which guarantees that the sub-domain $\Omega \subseteq \Omega_0$ contains a regular single curve segment with exactly two transversal intersections with the boundaries. Empty domains, which do not contain any points of the curve, should be also detected. In order to find such domains we analyze the sign changes of the BB-coefficients in the representation.

Observation 2.3. If a polynomial is represented by only negative or only positive BB-coefficients in a sub-domain Ω , then none of the points in the domain belongs to the zero set of the polynomial.

This observation follows from the convex hull property of BB-polynomials. A function with only positive or negative coefficients can be bounded away from zero by the coefficient which has the smallest absolute value.

We would like to generate fat arcs for curve segments, which consist of a single arc, are regular and do not form loops in the domain. The following Lemma 2.4 gives sufficient conditions to detect domains with single segment of a regular algebraic curve.

Lemma 2.4. *Consider a regular algebraic curve segment defined by the polynomial in the form (2.1) over a domain Ω . We say that the coefficients exhibit a **corner event**, if*

- *the coefficient at one of the corners is equal to zero and*
- *the first non-zero coefficients along the two neighboring boundaries have a different sign.*

*We say that the the coefficients exhibit an **edge event**, if*

- *the control polygon along one of the domain boundaries has exactly one sign change from plus to minus or vice versa.*

If the number of the corner and edge events is equal to two in Ω , then the domain contains a single curve segment, which is connected, and which intersects the boundaries of the domain in exactly two points.

Proof. For the proof it suffices to observe that each event guarantees that the implicitly defined curve crosses the boundaries of the domain in exactly one point. Since we supposed that the curve is regular, no self-intersections or loops appears in the domain. \square

The conditions of Lemma 2.4 are sufficient, but not necessary. For example, the lemma excludes the case of a single arc, which crosses twice the same segment of the domain boundary. However, in some cases the conditions of Lemma 2.4 are also necessary in the asymptotic sense. The following lemma describes such a special case. It supposes that the coordinates of the gradient vector are not vanishing in any point of the algebraic curve in the computational domain (the curve is *regular in both coordinate directions*).

Lemma 2.5. *Consider an algebraic curve segment defined by the polynomial f over a domain Ω_0 . We suppose that there exists $G > 0$, such that if $(x, y) \in \Omega_0$ satisfies $f(x, y) = 0$, then the partial derivatives satisfy*

$$\min\{|f_x(x, y)|, |f_y(x, y)|\} \geq G. \quad (2.6)$$

Under these conditions after a certain number of successive subdivisions of Ω_0 each sub-domain satisfies either the condition of Lemma 2.4 or the conditions of Observation 2.3. This implies that all sub-domains are detected as a domain with single curve segment or an empty domain.

Proof. We supposed that there exists $G > 0$ which is a lower bound for the partial derivatives along the curve. Therefore the BB-coefficients in the representation of f_x and f_y can be bounded away from zero if we compute in a sufficiently small sub-domain of Ω close to the algebraic curve. Thus the restriction of f_x and f_y to a domain boundary ($x=\text{constant}$ or $y=\text{constant}$) has only positive or only negative control points in the BB-representation. It means that f has a sequence of control points restricted to each domain boundary, which is monotone increasing or decreasing.

If the first and last control points have the same sign along a certain domain boundary, then all control points have the same sign. In this case according to Lemma 2.4 no event occurs along the domain boundary. If the first and last control points have different sign, then exactly one sign change occurs along the control polygon. It means that an edge event occurs. If one of the end control point is zero, it gives a corner event.

Each control polygon is monotone increasing or decreasing along the domain boundaries, and they are connected in the corners of the domain. Therefore the sum of the number of corner and edge events has to be even. If this event number is zero, then Observation 2.3 is satisfied. It is because all coefficients are strictly positive or negative along the domain boundary, and the partial derivatives are also bounded away from zero. If the corner and edge event number is equal to two, then the conditions of Lemma 2.4 are satisfied. In the case when the number of corner and edge event is more than two the domain contains more than one curve segment. They are not intersecting each other since the gradient is not vanishing. Therefore these segments are separated via subdivision in to different domains. \square

Remark 2.6. The conditions of Lemma 2.5 are necessary in a sense, that after a certain number of subdivisions all domains satisfy either the condition of Lemma 2.4 or the conditions of Observation 2.3. The following example demonstrates the topology detection, if the conditions of Lemma 2.5 are not satisfied. Suppose that our algebraic curve is defined by the polynomial

$$f(x, y) = y - \left(x - \frac{1}{3}\right)^2, \quad (x, y) \in \Omega = [0, 1]^2.$$

Along the domain boundary, which is defined by $y = 0$, the point $x = \frac{1}{3}$ is a point of the algebraic curve. We study the domains around the point $(\frac{1}{3}, 0)$ generated by adaptive subdivision. Each such domain has a boundary along $y = 0$, which can be represented by the interval $x \in [a, b]$. The function restricted to this segment of the domain boundary is

$$f(x, 0) = g(x) = -\left(x - \frac{1}{3}\right)^2, \quad x \in [a, b].$$

The adaptive subdivision implies that a and b are rational numbers in the form $\frac{k}{2^n} \leq 1$, where $k, n \in \mathbb{Z}$. Thus $\frac{1}{3} \in (a, b)$ moreover

$$g(a) < 0, \quad g(b) < 0 \quad \text{and} \quad g\left(\frac{1}{3}\right) = 0.$$

Therefore the control polygon of g has always at least two sign changes, or it has a control point equal to zero, which is not the end point of the control sequence. This implies that neither the condition of Observation 2.3 nor the conditions of Lemma 2.4 can be fulfilled for any sub-domains generated around the curve point $(\frac{1}{3}, 0)$.

Remark 2.7. Suppose that the points of the algebraic curve, which have tangent vector parallel to a coordinate axes, are not on the grid lines of the adaptive subdivision. Then after a certain number of subdivisions all sub-domains are detected as a domain with single curve segment or as an empty domain.

2.2.2 Local Algorithm

We present here a local algorithm, which generates fat arc in domains consisting a single segment of the curve. It assumes that the conditions of Lemma 2.4 are satisfied. Later on we will describe a global algorithm, which detects the domains, where the local algorithm is applicable. The local algorithm –summarized in Algorithm 1 – is based on the corresponding techniques in the parametric case. It generates median arc in a parametric form with interpolation technique (see later in Section 2.2.3) and uses the estimated Hausdorff distance (in Section 2.2.4).

The algorithm is successful, if it finds the median arc, and the fat arc thickness is smaller than the prescribed tolerance ε . Then the algorithm returns with a fat arc, which bounds the curve segment.

It may happen, that there are no fat arc boundaries, or only one of the bounding arcs can be generated (e.g. when the distance bound of the median arc and the implicitly defined curve is greater than the radius of the meridian circle, or one of the bounding arcs does not intersect the computational domain). The local algorithm fails if no fat arc is generated and returns with the empty set.

Fig.2.2 presents three examples of fat arcs which have been generated with the help of Algorithm 1.

2.2.3 Median Arc Generation with Interpolation

This approximation technique is based on the corresponding techniques in the parametric case. Therefore we have to ensure that the algebraic curve has a single segment in the

Algorithm 1 FatArcLocal_2d1 (f, Ω, ε)

Require: The conditions of Lemma 2.4 are satisfied.

```

1:  $p_{end} = \{\mathbf{p}_1, \mathbf{p}_2\} \leftarrow$  approximate end points of the implicitly defined curve
2:  $p_{mid} = \{\mathbf{p}_3\} \leftarrow$  approximate inner point of the implicitly defined curve
3: if  $\#p_{end} = 2$  and  $\#p_{mid} = 1$  then
4:    $\mathcal{S} \leftarrow$  circle through  $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$  {median circle}
5:    $\varrho \leftarrow$  upper bound of  $\text{HD}_\Omega(\mathcal{S} \cap \Omega, \mathcal{C} \cup \partial\Omega)$  {see Lemma 2.8}
6:   if  $\varrho \leq \varepsilon$  and  $\varrho \leq$  radius of  $\mathcal{S}$  then
7:      $\mathcal{S}_\varrho \leftarrow$  offset ring of  $\mathcal{S}$  with distance  $\varrho$  {fat circle}
8:      $\mathcal{S}^+, \mathcal{S}^- \leftarrow$  inner and outer circle of  $\partial\mathcal{S}_\varrho$ 
9:     if there is no sign change of  $f$  along  $\mathcal{S}^+ \cap \Omega$  or  $\mathcal{S}^- \cap \Omega$  then
10:      return  $\Omega \cap \mathcal{S}_\varrho$  {fat arc has been found}
11:    end if
12:  end if
13: end if
14: return  $\emptyset$  {no fat arc has been found}
    
```

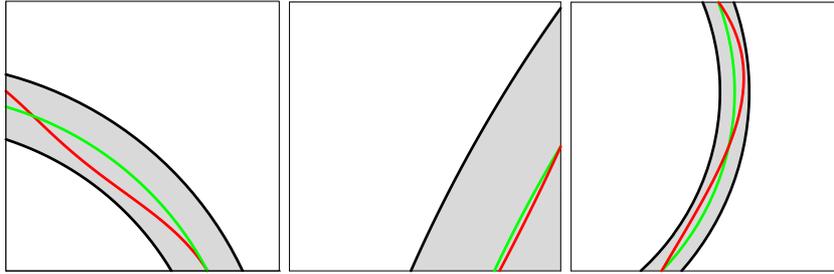


Figure 2.2: Examples for fat arc generation with the help of algorithm FatArcLocal_2d1. The red curves are the implicitly defined curves. The median circles are shown in green.

computational domain. We use Lemma 2.4 to detect such curve segments. In order to construct the median arc we approximate three points of the implicitly defined curve. Two of them are the intersection points of the curve with the domain boundary, while the third point is the intersection point of the bisector of the first two approximation points.

From Lemma 2.4 we know also, that in the case of a corner event the corner of the computational domain is a point of the curve. In the case of an edge event the corresponding edge contains an intersection of the curve with the boundary of the domain. It is approximated then, such that we consider the restriction of f to the edge, and generate its best L^2 approximation by a quadratic polynomial q^* . This polynomial additionally interpolates the values of f at the two end points of the edge. The root of q^* then defines the approximate intersection of the implicitly defined curve with the edge. If no simultaneous corner event occurs at the end points of the edge, then there is exactly one root of q^* , since the BB-coefficients of f possess exactly one sign change from plus to minus or vice versa.

After generating the first two points we restrict the function f to the intersection of their bisector with the domain. Again we generate its best L^2 approximation by a quadratic polynomial q^* , which additionally interpolates the values of f at the two end points. The

root of q^* then defines the approximate intersection of the curve with the bisector.

We use the linear parameterization of the line segments, the two edges and the bisector

$$\mathcal{L} = \{(l_x(t), l_y(t)), t \in [t_0, t_1]\}.$$

The general formulation of the quadratic approximation is

$$q^* = \min_{q \in Q} \left\| f|_{\mathcal{L}} - q \right\|_{L^2(\mathcal{L})}, \quad (2.7)$$

where Q denotes the set of the suitable quadratic polynomials along \mathcal{L} . The root of q^* is the approximate intersection point of the curve with the line segment \mathcal{L} . The median arc generation is successful if we find all three approximating points $\mathbf{p}_1, \mathbf{p}_2$ and \mathbf{p}_3 in Ω (see Fig.2.3).

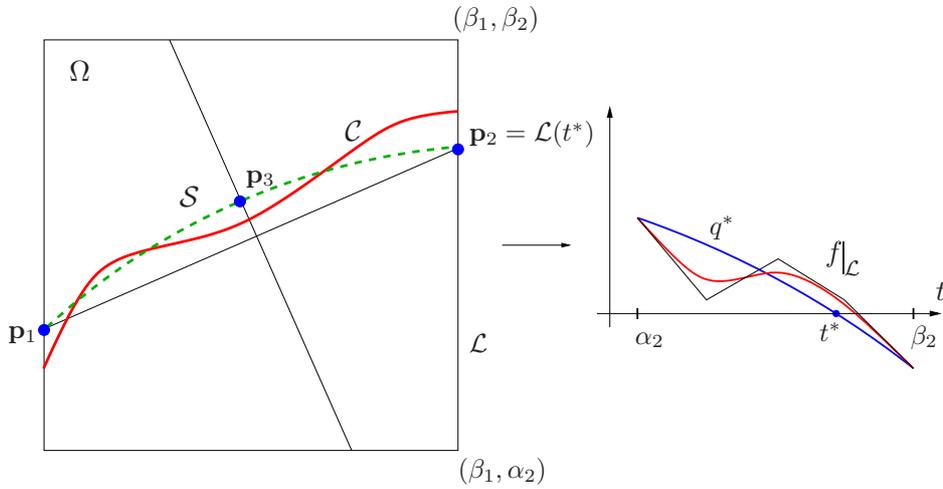


Figure 2.3: Median arc generation.

2.2.4 Distance of Parametric and Implicitly Defined Curves

We want to bound the distance between the median arc and the curve using a result from [1]. The implicitly defined curve is given as the zero set of the bivariate polynomial \mathcal{C} in the domain $\Omega = [0, 1]^2$. On one hand, we consider the median arc as a parametric curve $\mathbf{s} : t \mapsto \mathbf{s}(t)$ with parameter domain $t \in [0, 1]$, which traces the point set

$$\mathcal{S} = \{\mathbf{s}(t) : t \in [0, 1]\}, \quad (2.8)$$

where we assume that $\mathcal{S} \subset [0, 1]^2$. On the other hand, in order to avoid certain technical difficulties, we consider the set

$$\mathcal{C}^* = \mathcal{C} \cup \partial\Omega, \quad (2.9)$$

which is obtained by adding the boundary of the domain to the curve \mathcal{C} . The one-sided Hausdorff distance of \mathcal{C}^* and \mathcal{S} is defined as

$$\text{HD}_{\Omega}(\mathcal{S}, \mathcal{C}^*) = \sup_{t \in [0, 1]} \inf_{\mathbf{x} \in \mathcal{C}^*} \|\mathbf{x} - \mathbf{s}(t)\|. \quad (2.10)$$

We recall the following result from [1]

Theorem 2.8 (Aigner-Jüttler). *If there exist positive constants G, η such that*

$$\forall \mathbf{x} \in \Omega : \quad G \leq \|(\nabla f)(\mathbf{x})\| \quad \text{and} \quad \forall t \in [0, 1] : \quad |(f \circ \mathbf{s})(t)| \leq \eta \quad (2.11)$$

hold, then the one-sided Hausdorff distance is bounded by

$$HD_{\Omega}(\mathcal{S}, \mathcal{C}^*) \leq \frac{\eta}{G}. \quad (2.12)$$

Consequently, the parametric curve is contained in ϱ -neighborhood of \mathcal{C}^* , where $\varrho = \eta/G$. However, it should be noted that this distance bound does not guarantee that the implicitly defined curve is also contained in an ϱ -neighborhood of the parametric curve. The algorithm presented here uses an additional test to guarantee this property. Nevertheless, in all computed examples the above distance bound provided a safe and conservative estimate for the Hausdorff distance of the implicitly defined and the parametric curve.

Evaluation of the Constants. In order to find the constants G and η in Theorem 2.8, we represent the median arc as a quadratic rational Bézier curve,

$$\mathbf{s}(t) = \sum_{i=0}^2 \mathbf{s}_i \frac{\tilde{w}_i B_i^2(t)}{\sum_{j=0}^2 \tilde{w}_j B_j^2(t)}, \quad t \in [0, 1]. \quad (2.13)$$

Since it is a circular arc but not a whole circle, its weight satisfy

$$\tilde{w}_0 = \tilde{w}_2 = 1 \quad \text{and} \quad -1 < \tilde{w}_1 \leq 1. \quad (2.14)$$

The composition $f \circ \mathbf{s}$ is a rational function of degree $2(m+n)$ which can be represented in rational BB-form with certain coefficients d_i and weights w_i . The weights are computed with the evaluation of the $(m+n)$ th power of the denominator in (2.13).

$$|(f \circ \mathbf{s})(t)| = \frac{\sum_{i=0}^{2m+2n} d_i w_i B_i^{2m+2n}(t)}{\sum_{j=0}^{2m+2n} w_j B_j^{2m+2n}(t)} = \frac{s_n(t)}{s_d(t)}. \quad (2.15)$$

To find a suitable upper bound for the composition, first we consider with the denominator

$$s_d(t) = \sum_{j=0}^{2m+2n} w_j B_j^{2m+2n}(t) = \left(\sum_{i=0}^2 \tilde{w}_i B_i^2(t) \right)^{n+m}.$$

Because of (2.14) there exists λ , $0 < \lambda \leq 2$ such that $\tilde{w}_1 = \lambda - 1$. It means, that

$$\left(\sum_{i=0}^2 \tilde{w}_i B_i^2(t) \right)^{n+m} = ((2t-1)^2 + 2\lambda t(1-t))^{n+m}.$$

Since $t \in [0, 1]$ and $\lambda \in (0, 2]$

$$0 < \left(\frac{\lambda}{2} \right)^{n+m} \leq s_d(t) \leq 1, \quad \forall t \in [0, 1].$$

Therefore an upper bound η can be given as

$$|(f \circ \mathbf{s})(t)| \leq \frac{\|s_n\|_{\text{BB}}}{\left(\frac{\tilde{w}_1 + 1}{2} \right)^{n+m}} = \eta. \quad (2.16)$$

In order to find the second constant G , we use the same lower bound that we generated for certifying the regularity of the curve in the domain (see (2.5)).

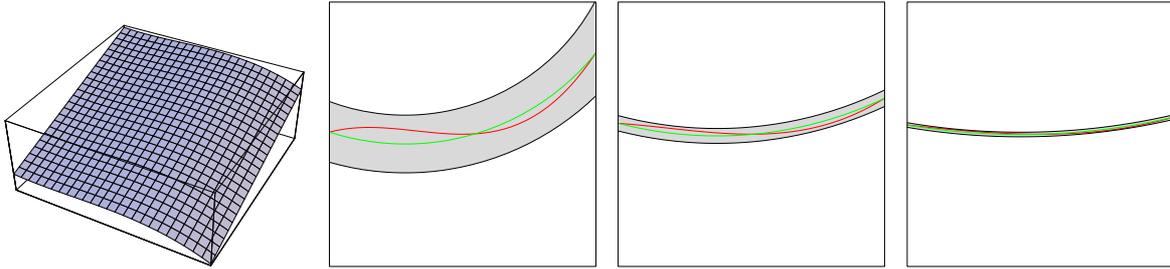


Figure 2.4: Left: The graph of f_1 . Right: Fat arcs for $k = 0.5, 0.75, 1.0$.

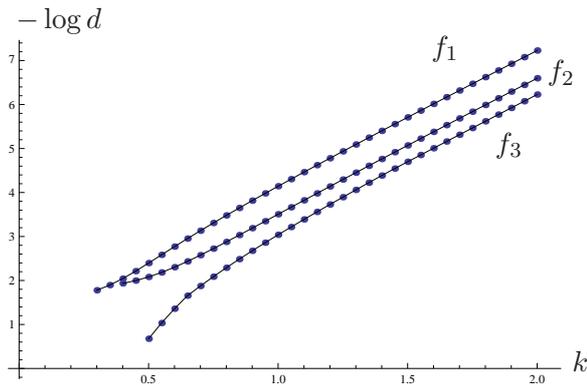


Figure 2.5: Dependency between the fat arc diameter and the domain diameter.

2.2.5 Convergence Rate

We confirm the approximation order of the fat arc generation algorithm (Algorithm 1) by numerical examples. It is also possible to prove the cubic approximation order of the method, but it is long and very technical.

Consider the three bivariate polynomials

$$\begin{aligned} f_1(\mathbf{x}) &= x_1^4 + x_1^3 x_2^2 + 2x_1^2 x_2 - 6x_1 x_2 + x_2^4 - 8x_2^2 - 12x_2 \\ f_2(\mathbf{x}) &= -x_1^3 - x_1^2 x_2 + x_1 x_2 - x_2^3 + x_2^2 - 2x_2 \\ f_3(\mathbf{x}) &= -4x_1^3 - 5x_1^2 + 2x_2 \end{aligned} \quad (2.17)$$

with the domains (in global coordinates)

$$\Omega_k = [-10^{-k}, 10^{-k}] \times [-10^{-k}, 10^{-k}], \quad k \in \mathbb{R}. \quad (2.18)$$

In the case of the first polynomial Fig.2.4 shows the result of the fat arc constructions for several values of k . The implicitly defined curve is the red one, the median arc denoted with green, and the fat arcs are represented with black.

Fig.2.5 visualizes the relation between the width of the fat arc and the size of the domain diameter for the three polynomials in (2.17). For sufficiently large values of k the slopes of the three curves in the doubly-logarithmic plot are all three, thus confirming the expected approximation order.

Algorithm 2 `GenerateFatArcs1`(f, Ω_0, ε)

```

1: if  $\min d_{ij} > 0$  or  $\max d_{ij} < 0$  then
2:   return  $\emptyset$                                      {the domain is empty}
3: end if
4: if  $f$  satisfies the conditions of Lemma 2.4 then
5:    $\mathcal{F} \leftarrow \text{FatArcLocal\_2d1}(f, \Omega, \varepsilon)$    {single fat arc generation}
6:   if  $\mathcal{F} \neq \emptyset$  then
7:     return  $\mathcal{F}$                                        {... has been successful}
8:   end if
9: end if
10: if diameter of  $\Omega > \varepsilon$  then
11:   subdivide the domain into 4 sub-domains  $\Omega_1, \dots, \Omega_4$    {quadsection}
12:   return  $\bigcup_{i=1}^4 \text{GenerateFatArcs1}(f, \Omega_i, \varepsilon)$    {recursive call}
13: end if
14: return  $\Omega$                                        {current domain is small enough}

```

2.2.6 Global Algorithm

The algorithm, `GenerateFatArcs1` (see Algorithm 2), combines the fat arc generation for single curve segments with recursive subdivision. First it analyzes the signs of the Bernstein–Bézier coefficients with respect to the current domain. If no sign change is present, then the current domain does not contain any components of the implicitly defined curve according to Observation 2.3. Otherwise it checks the conditions of Lemma 2.4, and tries to apply the local fat arc generation (Algorithm 1) for domains with single curve segments. If the local algorithm fails, then the algorithm either subdivides the current domain into four squares or returns the entire domain, if its diameter is already below the user-defined threshold ε .

Note that the algorithm may return domains which do not contain any segments of the implicitly defined curve ("false positive domains"). This can be avoided in the case when the partial derivatives are bounded away from zero and the user specified threshold ε is small enough (see in Lemma 2.5). However, it is always guaranteed, that the algorithm returns with a set of regions which contains the whole algebraic curve.

2.3 Fat Arc Generation with Implicitly Defined Arcs

In this section we describe local fat arc generation technique using implicitly defined arcs. We show different techniques to generate approximating circular arcs in implicit form. Then we also describe how to estimate the distance of implicitly defined planar curves. In the end we present a global algorithm using hierarchical subdivision for generating bounding regions for planar algebraic curves.

2.3.1 Local Algorithm

As an alternative we consider a local algorithm for generating fat arcs using implicitly defined arcs. It is summarized in Algorithm 3. This algorithm generates an approximate quadratic polynomial s with circular zero contour, and uses the BB-norm to estimate the

Algorithm 3 FatArcLocal_2d2 (f, Ω, ε)**Require:** The gradient does not vanish in Ω .

```

1:  $\hat{f} = lf$  modified polynomial and its special quadratic approximation  $s$ 
2: if  $l$  exists then
3:    $\mathcal{S} \leftarrow$  zero contour of  $s$  {median circle}
4:    $d \leftarrow \|\hat{f} - s\|_{\text{BB}}^{\Omega}$ 
5:    $G \leftarrow$  lower bound for  $\|\nabla \hat{f}\|$  {see (2.5)}
6:   if  $G$  exists and  $\varrho = \frac{d}{G} \leq \varepsilon$  then
7:      $\mathcal{S}^+, \mathcal{S}^- \leftarrow$  zero contour of  $s + d$  and  $s - d$  {fat circle boundaries}
8:      $\mathcal{F}(s, \varrho, \Omega) = \{\mathbf{x} : \exists \mathbf{x}_0, |\mathbf{x} - \mathbf{x}_0| \leq \varrho, s(\mathbf{x}_0) = 0\} \cap \Omega$  {fat arc}
9:     return  $\mathcal{C}_d$  {fat arc has been found}
10:  end if
11: end if
12: return  $\emptyset$  {no fat arc has been found}

```

fat arc thickness (Algorithm 3). For generating the approximate quadratic polynomial we will present two different methods in the next sections. The first one is using least-squares technique. The second one is operating with the modified Taylor expansion of the original polynomial f . Both methods are computing quadratic approximating polynomial for a modified polynomial \hat{f} , which is the original polynomial f multiplied with a linear polynomial l . This additional linear term guarantees the existence of the approximating polynomial and its convergence rate.

This fat arc generation technique only assumes that we have a regular curve segment in the domain. Therefore no other topological information is necessary for the fat arc generation.

The algorithm succeeds if it finds the median arc, and the fat arc thickness is smaller than the prescribed tolerance ε . Then the output is the intersection of a fat arc and the domain Ω , which contains the curve. It can also happen like in Algorithm 1, that there are no fat arc boundaries, or only one of the bounding arcs can be generated. Then the local algorithm fails and returns the empty set.

Fig.2.6 presents five local fat arc generation examples with different median arc and distance estimation technique. In the first row the figures are generated with the help of Algorithm 1 using parametric median arcs and computing an upper bound for the one-sided Hausdorff distance. In the lower rows Algorithm 3 is used generating implicitly defined median arcs in four different ways and using the distance estimation for implicitly defined curves (see Section 2.3.5). The second, third and the fourth rows contain the results of the least-squares technique. In the second row we applied least square approximation with linear normalization, while in the next two rows quadratic normalization. The last row presents the outputs from the algorithm, which uses modified quadratic Taylor expansion to generate median arcs. However, even if the distance estimation technique seems to be weaker in the case of Algorithm 3, it provides the cubic approximation order as we will see later.

2.3.2 Median Arc Generation with Least-Squares Approximation

Least-squares approximation is a standard technique for finding an approximating polynomial. In order to generate a quadratic polynomial with circular zero level set we are searching

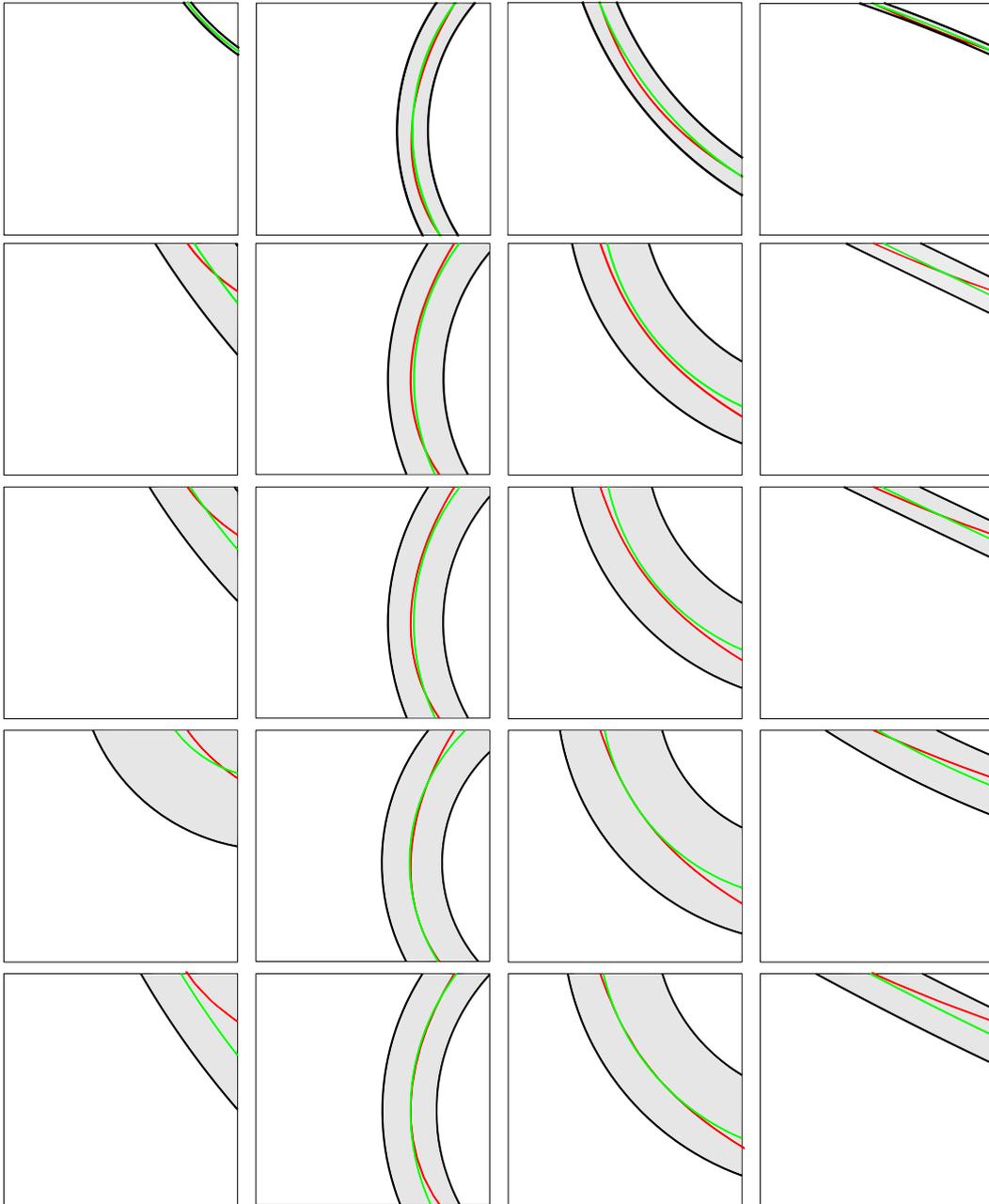


Figure 2.6: Comparison of the local fat arc generation techniques. In the first row the fat arcs are generated by `FatArcLocal_2d1`, in the other rows the algorithm `FatArcLocal_2d2` is used. Three rows in the middle are showing results of least-squares approximation with different normalization techniques. In the second row we used linear normalization condition (see in (2.20)). In the next two rows the quadratic conditions (2.21) and (2.22) are used. In the last row the outputs from Taylor expansion modification are shown (see in Section 2.3.3). The red curves are the implicitly defined curves. The median arcs are shown in green.

for a polynomial in the form

$$s(x, y) = a(x^2 + y^2) + bx + cy + d.$$

To provide the cubic convergence rate for the approximation, we modify the original function using a linear term

$$\hat{f}(x, y) = l(x, y)f(x, y) = (l_0 + l_1x + l_2y)f(x, y).$$

The optimization problem can be formed as follows

$$\min_{(a,b,c,d,l_0,l_1,l_2)} \iint_{\Omega} \|\hat{f} - s\|^2 dx dy. \quad (2.19)$$

In order to get a unique solution we have to normalize the minimization problem. Here we present three different strategies. The first normalization technique is using a linear condition. It is a natural condition in the sense that the modified polynomial \hat{f} keeps the original function value in the center (c^x, c^y) of the computational domain Ω

$$l_0 + l_1c^x + l_2c^y = 1. \quad (2.20)$$

Another possible choice for normalization is to control the gradient length of the approximating polynomial s . Such condition determines two possible solutions for $s(x, y)$. The one with smaller value in (2.19) can be used as an approximating polynomial. A natural choice of the quadratic normalization condition is

$$\|\nabla s\|^2 \Big|_{(c^x, c^y)} = 1. \quad (2.21)$$

Another possibility is to use a quadratic normalization condition which approximates better the secondary shape of the original implicitly defined curve. Namely we can suppose for instance

$$\|\nabla s\|^2 \Big|_{(c^x, c^y)} = \|\nabla f\|^2 \Big|_{(c^x, c^y)}. \quad (2.22)$$

A few examples are shown in Fig.2.6 for the result of the different least-square approximation methods. The three rows in the middle contain the result of the three different normalization techniques. They are shown in the same order as we described them here.

2.3.3 Median Arc Generation Using Taylor Expansion

In this technique we also reformulate the approximation problem in the computational domain Ω . The given polynomial f will be multiplied by a linear term l

$$\hat{f} = lf = (u + l_1(x - c^x) + l_2(y - c^y))f(x, y), \quad (2.23)$$

where $\mathbf{c} = (c^x, c^y)$ denotes the center point of Ω . Obviously the zero level set of the polynomial \hat{f} will contain each point of the zero level set of f . We choose the linear function l such that the Hessian of \hat{f} at the center of the domain is a scalar multiple of the identity matrix.

$$\mathcal{H}(\hat{f})(\mathbf{c}) = \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix}, \quad \lambda \in \mathbb{R}. \quad (2.24)$$

If such \hat{f} can be found, then the quadratic Taylor expansion of \hat{f} about \mathbf{c} has a special form. More precisely, this polynomial has always circular zero contour. In order to find \hat{f} we need

to solve a linear system for u, l_1 and l_2 . The condition (2.24) leads to the following system of equations in the center of the domain

$$\begin{aligned}\hat{f}_{xx}(\mathbf{c}) - \hat{f}_{yy}(\mathbf{c}) &= 0 \\ \hat{f}_{xy}(\mathbf{c}) &= 0.\end{aligned}\tag{2.25}$$

If the system has full rank, we have a solution set with one degree of freedom. Therefore we handle the constant term of l, u as a parameter of the solution set.

Lemma 2.9. *Given a bivariate polynomial f over the domain Ω . We suppose that the gradient of f does not vanish in the center \mathbf{c} of Ω*

$$\|\nabla f(\mathbf{c})\| \neq 0.$$

Then for any value of $u \neq 0$ and $u \in \mathbb{R}$ there exists a unique solution for l .

Proof. The Hessian matrix of \hat{f} can be expressed with the help of f and l

$$\begin{aligned}\mathcal{H}(\hat{f})(\mathbf{c}) &= \nabla l(\mathbf{c})\nabla f(\mathbf{c})^T + \nabla f(\mathbf{c})\nabla l(\mathbf{c})^T + l(\mathbf{c})\mathcal{H}(f)(\mathbf{c}) = \\ &= \nabla l(\mathbf{c})\nabla f(\mathbf{c})^T + \nabla f(\mathbf{c})\nabla l(\mathbf{c})^T + u\mathcal{H}(f)(\mathbf{c}).\end{aligned}\tag{2.26}$$

In the center of the domain the equation system (2.25) can be written as

$$\mathbf{A}\mathbf{l} = \begin{pmatrix} f_x(\mathbf{c}) & -f_y(\mathbf{c}) \\ f_y(\mathbf{c}) & f_x(\mathbf{c}) \end{pmatrix} \begin{pmatrix} l_1 \\ l_2 \end{pmatrix} = -u \begin{pmatrix} \frac{1}{2}(f_{xx}(\mathbf{c}) - f_{yy}(\mathbf{c})) \\ f_{xy}(\mathbf{c}) \end{pmatrix}.\tag{2.27}$$

We supposed that the gradient vector does not vanish in \mathbf{c} . Therefore the determinant of \mathbf{A} is not zero. Then there exist l_1 and l_2 , which satisfy (2.27). It implies, that l can be computed for any non-zero value of u uniquely. For an arbitrary $u \neq 0$ the solution is $(l_1, l_2) = (0, 0)$ if and only if the Hessian of f already fulfills the condition (2.24). In this case the polynomial l is the constant function

$$l \equiv u.\tag{2.28}$$

□

With the conditions (2.25) for a fixed value of $u \neq 0$ the polynomial l can be computed uniquely according to Lemma 2.9. Therefore we can compute then \hat{f} uniquely for any $u \neq 0$. We introduce the function \mathcal{G} , which assign to a function f , a value of u and the center point \mathbf{c} of a domain Ω the associated \hat{f} function according to the construction in Lemma 2.9

$$\mathcal{G}(f, u, \mathbf{c}) = \hat{f} = lf.\tag{2.29}$$

Observation 2.10. The choice of the parameter value u has no effect on the zero contour of the computed new polynomial \hat{f} . It is just a constant multiplier of the linear polynomial l in the solution. Therefore it can be chosen arbitrarily.

The quadratic Taylor expansion of \hat{f} about \mathbf{c} will have the following form, since the condition (2.24) is satisfied

$$s(x, y) = T_{\hat{f}(\mathbf{c})}^2(x, y) = \hat{f}(\mathbf{c}) + \hat{f}_x(\mathbf{c})(x - c^x) + \hat{f}_y(\mathbf{c})(y - c^y) + \frac{1}{2}\hat{f}_{xx}((x - c^x)^2 + (y - c^y)^2).\tag{2.30}$$

It is a bivariate quadratic polynomial with a circular zero contour. Therefore the algebraic curve $s = 0$ will be chosen as median arc to approximate the curve $f = 0$ in Ω . Later on the error of the approximation is estimated by a distance bound for the implicitly defined curves $s = 0$ and $\hat{f} = 0$.

2.3.4 Connection with the Osculating Circles

We analyze here the properties of the median arcs generated by the Taylor expansion modification technique.

Lemma 2.11. *Consider a function f , which defines an algebraic curve in $\Omega_0 \subset \mathbb{R}^2$*

$$\mathcal{C}(f, \Omega_0) = \{\mathbf{x} : f(\mathbf{x}) = 0\} \cap \Omega_0.$$

We assume that the point $\mathbf{p} \in \Omega_0$ is on the algebraic curve $\mathbf{p} \in \mathcal{C}$. Suppose that Ω is a sub-domain of Ω_0 and it has the center point \mathbf{p} . We compute $\hat{f} = \mathcal{G}(f, u, \mathbf{p})$. Then the arc, defined by the zero set of the quadratic Taylor expansion $s = T_{\mathbf{p}}^2(\hat{f})$, is the osculating circle of $\mathcal{C}(f, \Omega_0)$ in the point \mathbf{p} .

Proof. The function \hat{f} defines the following algebraic curve

$$\hat{\mathcal{C}}(\hat{f}, \Omega_0) = \{\mathbf{x} : \hat{f}(\mathbf{x}) = 0\} \cap \Omega_0.$$

We know from the computational method generates \hat{f} , that the algebraic curves \mathcal{C} and $\hat{\mathcal{C}}$ satisfy

$$\mathcal{C} \subseteq \hat{\mathcal{C}}.$$

If we consider only a small regular segment of \mathcal{C} and $\hat{\mathcal{C}}$ which is contained by $\Omega \subset \Omega_0$, then both of them represent the same single arc of the algebraic curve with the point \mathbf{p} on it.

The circle defined by the zero set of the quadratic Taylor expansion $s = T_{\hat{f}(\mathbf{p})}^2$ is

$$\mathcal{S}(s, \Omega) = \{\mathbf{x} : s(\mathbf{x}) = 0\} \cap \Omega.$$

We would like to show that \mathcal{S} has a second order contact with the algebraic curve \mathcal{C} in the point \mathbf{p} . This is sufficient in order to prove that \mathcal{S} is the osculating circle of \mathcal{C} , since the osculating circle is unique and \mathcal{S} is a circular arc. According to the definition of \mathcal{S} , it has a second order contact with $\hat{\mathcal{C}}$ in the point \mathbf{p} . As we already noticed $\hat{\mathcal{C}} = \mathcal{C}$ in the neighborhood of \mathbf{p} , therefore \mathcal{S} has a second order contact with \mathcal{C} in the point \mathbf{p} . \square

Remark 2.12. We can compute the curvature of \mathcal{C} in the point \mathbf{p} . For an implicitly defined curve it is computed from the first and second partial derivatives of the function with the help of the formula (see for instance in [15])

$$\kappa(f, \mathbf{p}) = \frac{\nabla f(\mathbf{p})^\perp \mathcal{H}(f)(\mathbf{p}) (\nabla f(\mathbf{p})^\perp)^\top}{\|\nabla f(\mathbf{p})\|^3},$$

where

$$\nabla f(\mathbf{p})^\perp = (-f_y(\mathbf{p}), f_x(\mathbf{p})).$$

We know, that

$$\kappa(f, \mathbf{p}) = \kappa(\hat{f}, \mathbf{p}) = \kappa(s, \mathbf{p}).$$

Thus

$$\kappa(f, \mathbf{p}) = \frac{\nabla f(\mathbf{p})^\perp \mathcal{H}(f)(\mathbf{p}) (\nabla f(\mathbf{p})^\perp)^\top}{\|\nabla f(\mathbf{p})\|^3} =$$

$$= \frac{\nabla \hat{f}(\mathbf{p})^\perp \mathcal{H}(f)(\mathbf{p}) \left(\nabla \hat{f}(\mathbf{p})^\perp \right)^\top}{\left\| \nabla \hat{f}(\mathbf{p}) \right\|^3} \stackrel{(2.24)}{=} \frac{\lambda}{\left\| \nabla \hat{f}(\mathbf{p}) \right\|^2} \stackrel{\mathbf{p} \in \mathcal{C}}{=} \frac{\lambda}{\left\| l(\mathbf{p}) \nabla f(\mathbf{p}) \right\|^2}.$$

Therefore $\lambda = \hat{f}_{xx}(\mathbf{p})$ satisfies

$$\lambda = u^2 \kappa(f, \mathbf{p}) \left\| \nabla f(\mathbf{p}) \right\|^2$$

for an arbitrary $u \neq 0$ in a point \mathbf{p} , where $f(\mathbf{p}) = 0$.

2.3.5 Distance of Implicitly Defined Curves

If we generate fat arcs for implicitly defined curve segments, the distance measuring becomes more complicated. The reason is the representation of the curve. Nevertheless, the approximating curve can be represented either in parametric or in implicit form. In order to measure the distance of two implicitly defined curves we consider the norm $\|\cdot\|_{\text{BB}}^\Omega$ over the domain Ω , which is the maximum absolute value of the coefficients in the BB-representation. We define a distance bound of the polynomial f and the approximating polynomial s for all points in the domain

$$\varepsilon = \|f - s\|_{\text{BB}}^\Omega. \quad (2.31)$$

Due to the convex hull property

$$|f(\mathbf{x}) - s(\mathbf{x})| \leq \varepsilon, \quad \forall \mathbf{x} \in \Omega.$$

This means that

$$s(\mathbf{x}) - \varepsilon \leq f(\mathbf{x}) \leq s(\mathbf{x}) + \varepsilon, \quad \forall \mathbf{x} \in \Omega.$$

A region can be defined in Ω by the approximating polynomial and the distance bound

$$\mathcal{D}(s, \varepsilon, \Omega) = \{\mathbf{x} : |s(\mathbf{x})| \leq \varepsilon\} \cap \Omega.$$

This is a bounding region for the zero level set of the polynomial f in Ω

$$\mathcal{Z}(f) \subseteq \mathcal{D}(s, \varepsilon) \subseteq \Omega.$$

It is a fat region defined by the median curve $s = 0$, which contains the implicitly defined curve $f = 0$.

In order to bound the thickness of this fat region $\mathcal{D}(s, \varepsilon)$ in the domain Ω we have to bound the gradient length of f from below. Suppose that G is a positive constant, which fulfills in any point \mathbf{x} of Ω , that

$$\left\| \nabla f(\mathbf{x}) \right\| \geq G.$$

Then the distance of the point sets $s = \pm\varepsilon$ from $s = 0$ is bounded by

$$\varrho = \frac{\varepsilon}{G}. \quad (2.32)$$

Thus the fat arc can be defined as the point set

$$\mathcal{F}(s, \varrho, \Omega) = \{\mathbf{x} : \exists \mathbf{x}_0, s(\mathbf{x}_0) = 0, |\mathbf{x} - \mathbf{x}_0| \leq \varrho\} \cap \Omega.$$

2.3.6 Convergence Rate

Since the approximation order of curves by segments of circular arcs is three (see [37]), the same result is anticipated for the results produced by local fat arc generation algorithm. Here we confirm the cubic convergence rate of the method, which generates fat arcs by modifying the Taylor expansion of the polynomial (see (2.3.3)). We defined $\hat{f} = lf$ for any point \mathbf{c} of the domain Ω , where \mathbf{c} is always the center of the corresponding sub-domain $\Omega \subseteq \Omega_0$. The approximating arc $s = 0$ is given by the quadratic Taylor expansion of \hat{f} about \mathbf{c} . The distance bound has been generated with the help of the BB-norm and a lower bound for the gradient length.

In order to prove the convergence rate of the method first we have to show, that \hat{f} depends continuously on the points of Ω_0 . It means, that the computed polynomial \hat{f} depends continuously on the sub-domain Ω .

Lemma 2.13. *If the gradient of f does not vanish in the domain Ω_0 , then \hat{f} depends continuously on the points of the domain.*

Proof. We have to show that the computed $l = u + l_1(x - c^x) + l_2(y - c^y)$ linear polynomial depends continuously on the point $\mathbf{c} = (c^x, c^y)$. We compute the coefficient vector (l_1, l_2) , such that it satisfies (2.27) for a fixed value of u . The entries of the matrix in (2.27) depends continuously on (c^x, c^y) since f is a polynomial. Therefore the direction of the solution vector depends also continuously on the point. Since we know that $u \neq 0$ is fixed, then also l depends continuously on the point (c^x, c^y) . \square

The next corollary follows from Lemma 2.11 and Lemma 2.13. If we use the Taylor expansion modification technique described in Section 2.3.3, then we can establish the following result about the behavior of the generated median circles.

Corollary 2.14. *Suppose we have a nested sequence of sub-domains $(\Omega_i)_{i=1,2,3,\dots} \subset \Omega_0$*

$$\Omega_{i+1} \subset \Omega_i,$$

which have decreasing diameters δ_i , such that

$$\lim_{i \rightarrow \infty} \delta_i = 0,$$

and \mathbf{c}_i denotes the center point of Ω_i . Consider a function f , which defines an algebraic curve in $\Omega_0 \subset \mathbb{R}^2$

$$\mathcal{C}(f, \Omega_0) = \{\mathbf{x} : f(\mathbf{x}) = 0\} \cap \Omega_0.$$

Suppose that there exists a point \mathbf{p} , which satisfies $f(\mathbf{p}) = 0$ and for all $i : \mathbf{p} \in \Omega_i$. We compute $\hat{f}_i = \mathcal{G}(f, u, \mathbf{c}_i)$. The median arc is defined by the zero set of the quadratic Taylor expansion $s_i = T_{\mathbf{c}_i}^2(\hat{f}_i)$ about \mathbf{c}_i . Then the sequence of computed median circles $s_i(x, y) = 0$ converges to a limit circle, which is the osculating circle of \mathcal{C} in the point \mathbf{p} .

In order to certify the convergence rate of the fat arc generation method using Taylor expansion modification we consider the behavior of the gradient of the new polynomials $\hat{f} = \mathcal{G}(f, u, \mathbf{c})$. The following lemma (Lemma 2.15) ensures, that any \hat{f} has also a non-vanishing gradient, if we are computing close to the algebraic curve in a sufficiently small sub-domain of the original computational domain Ω_0 .

Lemma 2.15. *Suppose that there exists G in Ω_0 for the polynomial f such that*

$$\forall \mathbf{x} \in \Omega_0, \quad \|\nabla f(\mathbf{x})\| \geq G > 0. \quad (2.33)$$

We choose an arbitrary but fixed value of $u \neq 0$. Then there exists ε such that, if $\hat{f} = \mathcal{G}(f, u, \mathbf{c})$ is computed in a domain $\Omega \subset \Omega_0$ which has a diameter $\delta_\Omega < \varepsilon$, and there is a point $\mathbf{p} \in \Omega$ which fulfills $f(\mathbf{p}) = 0$, then for any $\mathbf{x} \in \Omega$ a general positive bound \hat{G} can be given as

$$\|\nabla \hat{f}(\mathbf{x})\| \geq \hat{G} > 0.$$

Proof. If $\mathbf{x} \in \Omega \subset \Omega_0$ then

$$\nabla \hat{f}(\mathbf{x}) = f(\mathbf{x})\nabla l(\mathbf{x}) + l(\mathbf{x})\nabla f(\mathbf{x}).$$

According to the triangular inequality

$$\|\nabla \hat{f}(\mathbf{x})\| \geq \|l(\mathbf{x})\nabla f(\mathbf{x})\| - \|f(\mathbf{x})\nabla l(\mathbf{x})\| = |l(\mathbf{x})| \|\nabla f(\mathbf{x})\| - |f(\mathbf{x})| \|\nabla l(\mathbf{x})\|. \quad (2.34)$$

Since we know that there exists a point $\mathbf{p} \in \Omega$ such that $f(\mathbf{p}) = 0$, and (2.33) is satisfied, then

$$|f(\mathbf{x})| \leq \frac{\varepsilon}{G}, \quad (2.35)$$

where ε is an upper bound of the diameter of Ω .

We can suppose that there exists $H \neq 0$ such that

$$\forall \mathbf{x} \in \Omega_0, \quad \sqrt{\frac{1}{4}(f_{xx}(\mathbf{x}) - f_{yy}(\mathbf{x}))^2 + f_{xy}(\mathbf{x})^2} \leq H,$$

since f is a polynomial. If the linear system formulated as

$$\mathbf{A}\mathbf{l} = \begin{pmatrix} f_x(\mathbf{c}) & -f_y(\mathbf{c}) \\ f_y(\mathbf{c}) & f_x(\mathbf{c}) \end{pmatrix} \begin{pmatrix} l_1 \\ l_2 \end{pmatrix} = -u \begin{pmatrix} \frac{1}{2}(f_{xx}(\mathbf{c}) - f_{yy}(\mathbf{c})) \\ f_{xy}(\mathbf{c}) \end{pmatrix},$$

then

$$\|\mathbf{A}\mathbf{l}\| \leq |u|H.$$

Since

$$\|\mathbf{A}\mathbf{l}\| = \sqrt{(f_x^2 + f_y^2)(l_1^2 + l_2^2)} \geq G \|\nabla l(x)\|$$

we obtain that

$$\|\nabla l(\mathbf{x})\| \leq \frac{|u|H}{G}. \quad (2.36)$$

From (2.35) and (2.36) it follows that

$$|f(\mathbf{x})| \|\nabla l(\mathbf{x})\| \leq \frac{\varepsilon |u| H}{G^2}. \quad (2.37)$$

Suppose that

$$\varepsilon < \frac{\sqrt{2}G}{H}, \quad (2.38)$$

then a positive lower bound can be given for $|l(\mathbf{x})|$

$$|l(\mathbf{x})| \geq |u| - \frac{|l_1| + |l_2|}{2} \varepsilon \geq |u| - \frac{\|\nabla l(\mathbf{x})\|}{\sqrt{2}} \varepsilon \geq |u| - \frac{\varepsilon |u| H}{\sqrt{2} G} > 0.$$

Therefore it implies that

$$|l(\mathbf{x})| \|\nabla f(\mathbf{x})\| \geq |u| G - \frac{\varepsilon |u| H}{\sqrt{2}} > 0. \quad (2.39)$$

So from (2.37) and (2.39) follows that

$$\|\nabla \hat{f}(\mathbf{x})\| \geq |u| G - \frac{\varepsilon |u| H}{\sqrt{2}} - \frac{\varepsilon |u| H}{G^2}.$$

If we would like to choose ε , such that $\|\nabla \hat{f}(\mathbf{x})\|$ has a positive lower bound, then

$$|u| G - \frac{\varepsilon |u| H}{\sqrt{2}} - \frac{\varepsilon |u| H}{G^2} > 0,$$

which means that

$$\frac{\sqrt{2} G^3}{H (\sqrt{2} + G^2)} > \varepsilon. \quad (2.40)$$

In this case also (2.38) is satisfied

$$\varepsilon < \frac{\sqrt{2} G}{H} \frac{G^2}{(\sqrt{2} + G^2)} < \frac{\sqrt{2} G}{H}$$

Therefore for any domain Ω with the diameter $\delta_\Omega < \varepsilon$ fulfills (2.40)

$$\|\nabla \hat{f}(\mathbf{x})\| > 0.$$

□

Corollary 2.16. *According to Observation 2.10 we can choose $u = 1$ and we suppose that*

$$\varepsilon = \frac{1}{2} \frac{\sqrt{2} G^3}{H (\sqrt{2} + G^2)}.$$

Thus for any domain Ω , which has the diameter

$$\delta_\Omega < \frac{1}{2} \frac{\sqrt{2} G^3}{H (\sqrt{2} + G^2)},$$

and contains a point of the curve $f = 0$, it is true that

$$\|\nabla \hat{f}(\mathbf{x})\| \geq G - \frac{G}{2} = \hat{G} > 0.$$

Both bounds, ε and \hat{G} , are independent of the choice of Ω (if Ω is small and contains some points of the curve). They only depend on f , Ω_0 and u .

Corollary 2.17. *We consider a polynomial f , which fulfills the condition (2.33) on a domain $\Omega \subset \Omega_0$. We compute $\hat{f} = \mathcal{G}(f, u, \Omega)$, and the median arc is defined by the zero set of the quadratic Taylor expansion $s = T_{\mathbf{c}}^2(\hat{f})$ about the center point \mathbf{c} of Ω . If Ω has a sufficiently small diameter and contains a point \mathbf{p} satisfying $f(\mathbf{p}) = 0$, then s is non-constant.*

Now we will show that the fat arc thickness is sufficiently small compared with the diameter of the computational domain. The following lemma shows, how the computed fat arc thickness behaves for a given function f in a certain domain.

Lemma 2.18. *Given a bivariate polynomial f defined over the domain $\Omega_0 = [\alpha_1, \beta_1] \times [\alpha_2, \beta_2]$. We suppose that there exists a non-negative lower bound G such that*

$$\|\nabla f\| \geq G > 0.$$

For any sub-domain $\Omega \subset \Omega_0$, which has a sufficiently small diameter and contains a segment of the algebraic curve $f = 0$, there exists a constant $C \in \mathbb{R}$ not depending on Ω such that

$$\varrho < C\delta_{\Omega}^3, \quad (2.41)$$

where ϱ is the corresponding fat arc thickness computed like in (2.32).

Proof. We denote by s_{Ω} the quadratic Taylor expansion of \hat{f}_{Ω} about the center \mathbf{c} of the domain Ω , then

$$\|\hat{f}_{\Omega} - s_{\Omega}\|_{\infty} < \frac{1}{6} \underbrace{\max_{\mathbf{v} \in S^1, \mathbf{x} \in \Omega} \left| \frac{d^3 \hat{f}_{\Omega}}{d\mathbf{v}^3}(\mathbf{x}) \right|}_{*} \delta_{\Omega}^3.$$

Recall from Lemma 2.13 that \hat{f}_{Ω} depends continuously on the points of the computational domain Ω_0 . Therefore for all \hat{f}_{Ω} a general upper bound C_1 can be given for (*). The fat arc thickness is defined by

$$\varrho = \frac{\|\hat{f}_{\Omega} - s_{\Omega}\|_{\text{BB}}}{\|\nabla \hat{f}_{\Omega}\|}.$$

We know from Corollary 2.16 that for a certain u there exists a general lower bound

$$0 < \hat{G} \leq \|\nabla \hat{f}_{\Omega}\|$$

for any sub-domain $\Omega \subset \Omega_0$, which has a sufficiently small diameter. Because of the norm equivalences there exists a constant C_2 such that

$$\varrho \leq \frac{C_2 \|\hat{f}_{\Omega} - s_{\Omega}\|_{\infty}}{\hat{G}} \leq \frac{1}{6} \frac{C_1 C_2 \delta^3}{\hat{G}}.$$

In order to bound ϱ , we choose

$$C = \frac{1}{6} \frac{C_1 C_2}{\hat{G}},$$

and arrive at

$$\varrho \leq C\delta_{\Omega}^3.$$

□

Algorithm 4 `GenerateFatArcs2`(f, Ω, ε)

```

1: if  $\min d_{ij} > 0$  or  $\max d_{ij} < 0$  then
2:   return  $\emptyset$                                      {the domain is empty}
3: end if
4: if  $\forall \mathbf{x} \in \Omega \quad \|\nabla f(\mathbf{x})\| > 0$  then
5:    $\mathcal{A} \leftarrow \text{FatArcLocal\_2d2}(f, \Omega, \varepsilon)$    {single fat arc generation}
6:   if  $\mathcal{A} \neq \emptyset$  then
7:     return  $\mathcal{A}$                                        {... has been successful}
8:   end if
9: end if
10: if diameter of  $\Omega > \varepsilon$  then
11:   subdivide the domain into 4 sub-domains  $\Omega_1, \dots, \Omega_4$    {quadsection}
12:   return  $\bigcup_{i=1}^4 \text{GenerateFatArcs2}(f, \Omega_i, \varepsilon)$    {recursive call}
13: end if
14: return  $\Omega$                                        {current domain is small enough}

```

Similarly the cubic convergence can be proven for any fat region generated as the thickened neighborhood of the zero locus of the quadratic Taylor expansion of a polynomial. Even if this technique is more general, we should not forget, that the fat regions are bounded by the offset curves of the median curve. These boundary curves should not have cusp or self-intersections. This is not guaranteed if we use a general bi-quadratic algebraic curve as the median curve. To avoid such critical cases and also to simplify the computations we restricted ourselves to use circular arcs as median curves.

2.3.7 Global Algorithm

The algorithm `GenerateFatArcs2` (see Algorithm 4) combines the fat arc generation for single curve segments with recursive subdivision. First it analyzes the signs of the Bernstein-Bézier coefficients with respect to the current domain. If no sign changes are present, then the current domain does not contain any components of the implicitly defined curve. Otherwise the algorithm tries to apply the fat arc generation for a single curve segment. If this is not successful, then the algorithm either subdivides the current domain into four squares, or returns the entire domain if its diameter is already below the user-defined threshold ε .

2.4 Comparison of the Methods

In the former sections (Section 2.2 and Section 2.3) we described and analyzed various algorithms to generate fat arcs for implicitly defined curves. These techniques are using different approximating arc generation and distance estimation methods. Here we compare these fat arc generation techniques.

2.4.1 Comparison of Fat Arc Generation

Median Arc Generation. The first step of the fat arc generation method is to generate an approximating arc, the median arc. This arc can be represented either in parametric

(rational BB) or in implicit form. We described an approximate interpolation method for generating a parametric approximating arc $\mathbf{s}(t)$ for a polynomial f in Section 2.2.3. It is clear that a list of median circles generated for a nested list of computational domains, which converge to a point of the implicitly defined curve \mathbf{p} , converges to the osculating circle of the implicitly defined curve in the point \mathbf{p} . The same was proven for the median circles in the case of the Taylor expansion modification technique (Corollary 2.14). It is a very important property of both methods, if we would like to develop techniques with cubic convergence. The osculating circle is the only circle with second order contact in a certain point of a planar curve. Therefore we can establish in advance for instance, that the fat arc generation technique using least-squares approximation with the quadratic normalization condition (2.21) cannot have cubic convergence rate. It is because the radius of the median arc is fixed via the normalization condition, a list of median arcs generated under the conditions of Corollary 2.14 not necessarily converge to an osculating circle of the algebraic curve.

Distance Bounding. In order to compute the fat arc thickness for parametric median arc, we use a bound given by Theorem 2.8. It is an upper bound of the one-sided Hausdorff distance. This bound is given by the ratio of an upper bound of the function value along the approximating arc and a lower bound of the gradient length in the computational domain Ω . These bounds can be computed with the help of the convex hull property of the BB-polynomials (described in Section 2.2.4). On the other hand if we generate the median arc in implicit form $s(x, y) = 0$ (see various methods in Section 2.3.2 and Section 2.3.3), then we have to measure the distance of implicitly defined curves. We described how to bound this distance in Section 2.3.5. This bounding technique is also using the convex hull property of the BB-polynomials. The bound is the ratio of the BB-norm of the polynomial $(f - s)$ in the computational domain Ω and the lower bound of the gradient length of f in Ω .

The lower bound of the gradient length is computed with the same method by both techniques, so it is sufficient to compare the nominator of the distance bounds. We observed that both methods generate median arcs which converge to the osculating circle under the conditions of Corollary 2.14. It means that for the polynomial f on a sufficiently small computational domain $\Omega \subset \Omega_0$ the median arcs generated by the two different techniques are close to each other. Therefore we compare here the fat arc thickness generated for the same median arc \mathcal{S} . Suppose that $s(x, y)$ represents the median arc in implicit form

$$\mathcal{S}(s, \Omega) = \{(x, y) : s(x, y) = 0\} \cap \Omega.$$

We also suppose that we know the parametric representation of the arc \mathcal{S} and it is denoted by $\mathbf{s}(t)$, $t \in [0, 1]$. Then we can established that

$$(f \circ \mathbf{s})(t) = (f - s)(x(t), y(t)),$$

and it implies that

$$\max_{t \in [0, 1]} |(f \circ \mathbf{s})(t)| \leq \max_{(x, y) \in \Omega} |(f - s)(x, y)|.$$

So the parametric fat arc generation estimates the function value of f along a curve on the computational domain, while the implicitly defined fat arc generation computes a bound depending on the whole domain Ω . Therefore usually the distance estimated by the parametric representation is smaller than in the case of the implicitly defined median arcs. This heuristic result will be confirmed also in the next section, where we analyze the convergence rate of all the described methods via an example.

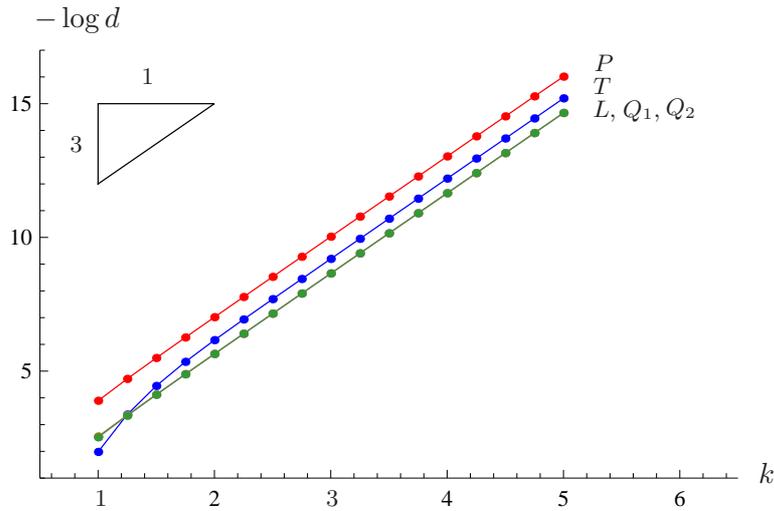


Figure 2.7: Comparison of relation between the fat arc diameter and the domain diameter for five different fat arc generation methods. The red line P shows the results from the parametric approximation. The results of least-square approximations are shown by the green line noted by L , Q_1 and Q_2 . The result of the Taylor expansion modification is represented by the line T (blue).

2.4.2 Comparison of Convergence Rate

We compare here the convergence rate of all described fat arc generation methods via an example. However, we just proved the rate of convergence of the Taylor expansion modification, we would like to show through this numerical example the behavior of all formerly described techniques.

We consider the polynomial

$$f(x, y) = -3x + 6x^2 - 2x^3 + y + y^2,$$

on the domains (in global coordinates)

$$\Omega_k = [-10^{-k}, 10^{-k}] \times [-10^{-k}, 10^{-k}], \quad k \in \mathbb{R}. \quad (2.42)$$

We visualize for the five different fat arc generation strategies the relation between the width of the generated fat arcs and the size of the domain diameter in Fig.2.7. For the values of $k = 1, 1.25, \dots, 5$ we show the negative logarithm of the associated fat arc diameter in a doubly-logarithmic plot. The expected approximation order is three. In this example it is confirmed for all of the strategies. The line denoted by P shows the results from the parametric approximation, it pretends to have a better approximating constant than the other techniques with cubic convergence rate. However, the least-square approximation with linear condition (L), the least-square approximation with quadratic normalization condition, denoted in the picture by Q_1 and Q_2 and the Taylor expansion modification T also shows cubic convergence rate. The least-squares approximation with the linear and quadratic normalization conditions show only a slight difference in the output.

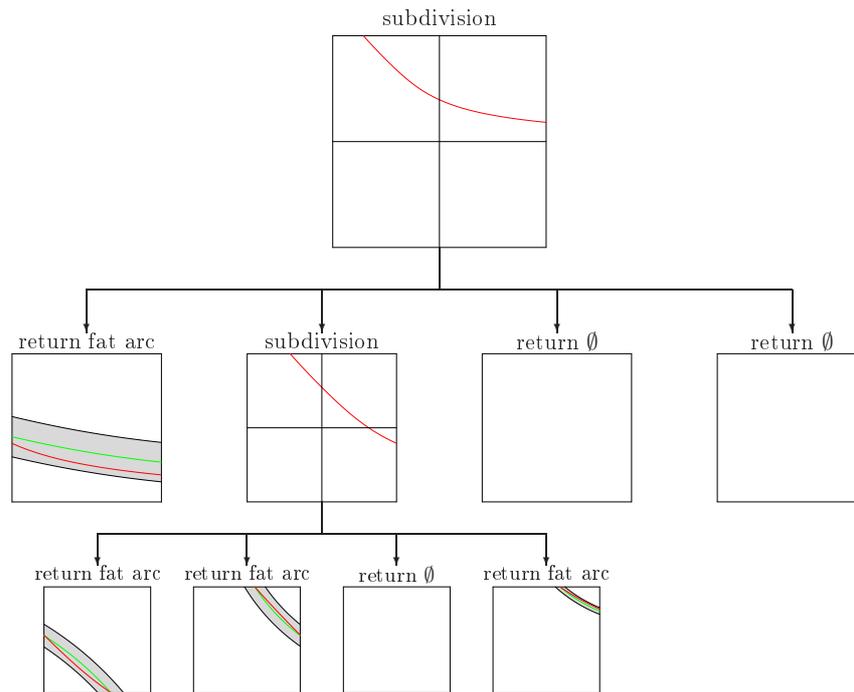


Figure 2.8: Example 2.19: The decision tree of algorithm `GenerateFatArcs`.

2.5 Examples and Application

2.5.1 Examples

We illustrate the performance of both algorithms Algorithm 2 and Algorithm 4 by examples.

Example 2.19. The first example (see Fig.2.8) visualizes the entire algorithm. We apply the algorithm to a bivariate polynomial of degree $(1, 4)$, which has only one arc in the region of interest, and choose a relatively large tolerance ε . The first call of the algorithm produces four sub-domains which are then analyzed independently. The first domain contains an arc which can be approximated by a single fat arc. The second domain produces other four sub-domains, while the third and the fourth domains do not contain any points of the implicitly defined curve. Finally, analyzing the four second-generation sub-domains leads to three additional fat arcs and one empty domain. The output is generated by collecting all sub-domains in the leaves of the subdivision tree.

Example 2.20. We consider a polynomial f of degree $(6, 9)$ with randomly generated BB coefficients in $[-1, 1]$. Fig.2.9 (a) shows the surface and the implicitly defined curve segments in the unit square. Fig.2.9 (b) and (c) demonstrate the behavior of the algorithm for different tolerances ε . The upper row shows the entire domain, while the lower row shows a zoomed view of the lower left corner of the domain. In the case of $\varepsilon = 0.1$, which is shown in (b), some domains are returned as bounding regions, since `FatArcLocal_2d1` fails and the diameter of the sub-domains are smaller than ε . For the smaller value of $\varepsilon = 0.01$, the fat arc generation succeeded in all generated sub-domains.

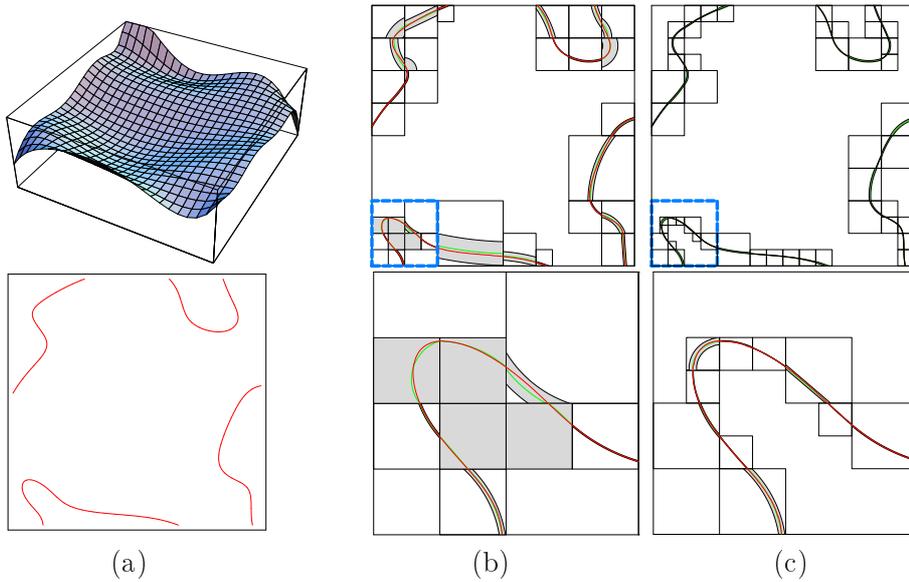


Figure 2.9: Example 2.20: Fat arc generation for different tolerances. The graph of f and the implicitly defined curve (a), and The fat arcs (top) and a zoomed view (bottom) for $\varepsilon = 0.1$ (b) and for $\varepsilon = 0.01$ (c).

In the next three examples we compare fat arcs with (recursively generated) bounding boxes. In the latter case we also accepted sub-domains as bounding primitives in the fat arc generation, if their diameter were less than the prescribed tolerance.

Example 2.21. We approximate an implicitly defined curve, see Fig.2.10, by fat arcs (a) and by bounding domains (b). Clearly, the use of fat arcs leads to a much smaller number of bounding geometric primitives. This becomes even more dramatic for smaller tolerances. Figure (c) shows the relation between the number of generated primitives (fat arcs or boxes) and the tolerance $\varepsilon = \sqrt{2}/2^k$.

Example 2.22. This example is based on an implicitly defined curve which possesses a singular point (see Fig.2.11). In this situation, the fat arc generation will fail for any domain which contains the singular point, since no positive lower bound on $\|\nabla f\|$ exists. Consequently, the algorithm always returns a domain containing this point. Still, the results generated by our method (left) compare favorably with the use of bounding boxes (right).

Example 2.23. Here we approximated an implicitly defined curve $f = 0$, where f has the polynomial order (9, 8). Our domain of interest is the unit square $\Omega_0 = [0, 1] \times [0, 1]$. The figures (a) and (b) in Fig.2.12 are generated with the two different fat arc constructions. The first one with the usage of the local Algorithm `FatArcLocal_2d1`, the second with the local Algorithm `FatArcLocal_2d2`. The tolerance bound is 10^{-2} , which is relatively small compared with the size of the starting domain. In order to reach this precision the number of the bounding domains is 36 in the first case and 46 in the second one. It is much fewer than in the case of bounding boxes, where we need 685 boxes to give a sufficient approximation.

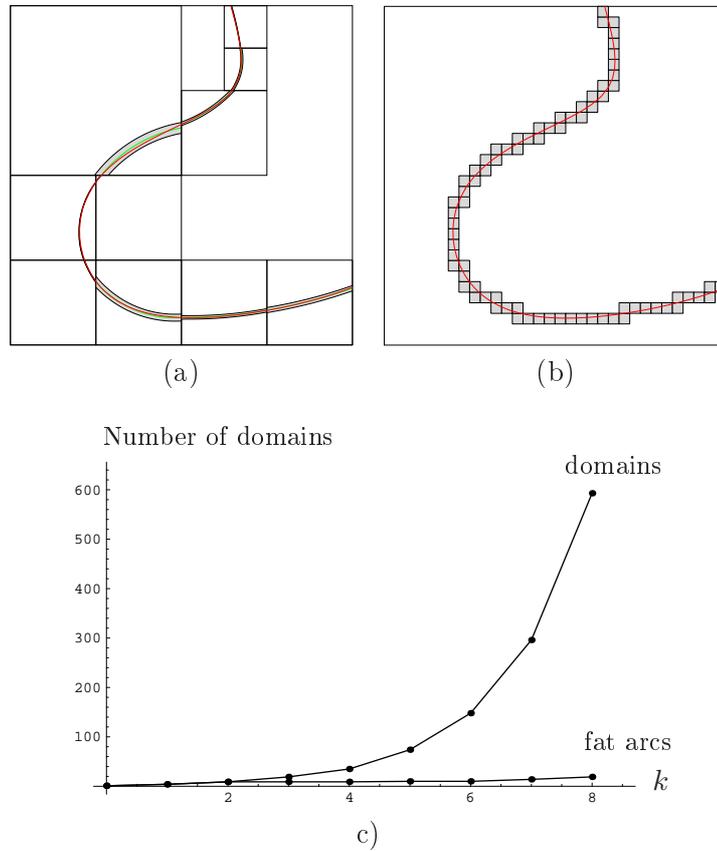


Figure 2.10: Example 2.21: Comparison of fat arcs a) and bounding domains b). The relation between tolerance and number of bounding primitives c).

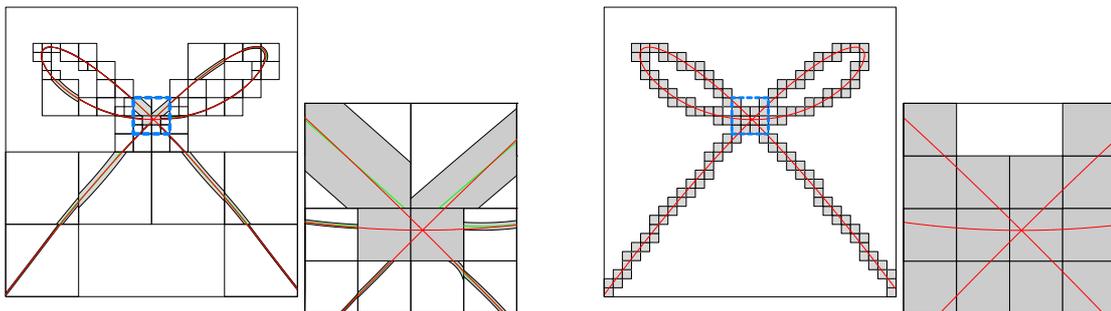


Figure 2.11: Example 2.22: Fat arcs (left) and bounding boxes (right) for an implicitly defined curve with a singular point, where $\varepsilon = \sqrt{2}/2^5$.

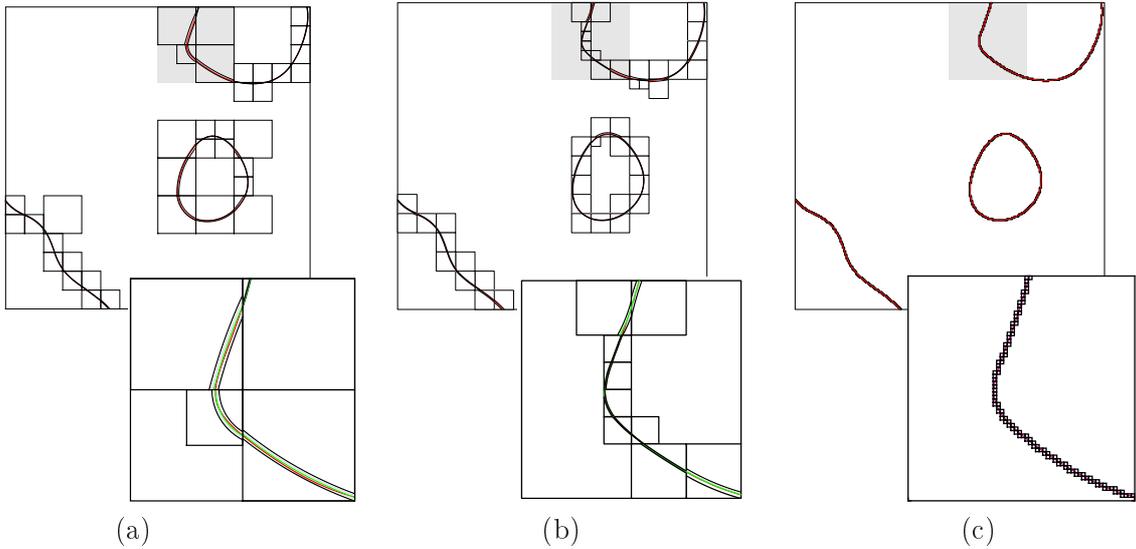


Figure 2.12: Example 2.23: Comparison of `FatArcLocal_2d1` (a) `FatArcLocal_2d2` (b) and bounding box generation (c). In the lower right corner of each output the generated bounding primitives are shown from the gray region of the computational domain.

2.5.2 Application: Surface-Surface Intersections

The computation of surface-surface intersections is a potential application of bounding region generation methods. Here we generate fat arcs to bound the intersection curve of an implicitly defined and a parametric surface. In practice this is the so called “mixed” intersection problem. It is one of the most frequently encountered cases [23]. A good survey on this topic is given in [30, 31].

Consider an implicitly defined surface $h(x, y, z) = 0$ and a parametric surface patch $\mathbf{r}(\xi_1, \xi_2)$ with domain $\Omega = [0, 1]^2$. Then the implicitly defined curve $f = h \circ r = 0$ describes the intersection curve in the domain of the parametric surface patch.

Using Algorithm `GenerateFatArcs`, one can construct a collection of fat arcs with maximum width ε in Ω . The region described by them corresponds to a certain subset (a strip) on the parametric surface patch.

Recall that the coefficients of the first fundamental form are defined as

$$g_{ij}(\xi_1, \xi_2) = \frac{\partial}{\partial \xi_i} \mathbf{r}(\xi_1, \xi_2) \cdot \frac{\partial}{\partial \xi_j} \mathbf{r}(\xi_1, \xi_2). \quad (2.43)$$

In order to relate the thickness of the bounding fat arcs to the thickness of the corresponding strip on the parametric surface, we present the following observation.

Lemma 2.24. *Consider a single fat arc with width 2ρ in the parameter domain of a parametric surface Ω . Then there exists a constant C depending only on the parametric surface, such that the width of the corresponding fat region on the parametric surface patch is bounded by*

$$2\rho\sqrt{C}.$$

Proof. We denote the matrix of first fundamental form corresponding to a point (ξ_1, ξ_2) of the parametric surface with

$$G(\xi_1, \xi_2) = \begin{pmatrix} g_{11}(\xi_1, \xi_2) & g_{12}(\xi_1, \xi_2) \\ g_{12}(\xi_1, \xi_2) & g_{22}(\xi_1, \xi_2) \end{pmatrix}.$$

The length L of a curve on the surface which corresponds to any straight line segment in the parameter domain Ω

$$(\xi_1(t), \xi_2(t)) = (\xi_1^0, \xi_2^0) + t(\eta_1, \eta_2), \quad t \in [a, b],$$

where

$$\eta_1^2 + \eta_2^2 = 1$$

is

$$L = \int_a^b \sqrt{g_{11}(\xi_1, \xi_2)\eta_1^2 + 2g_{12}(\xi_1, \xi_2)\eta_1\eta_2 + g_{22}(\xi_1, \xi_2)\eta_2^2} dt. \quad (2.44)$$

In order to find an upper bound for L we are looking for the extremal values of the quadratic form

$$g_{11}(\xi_1, \xi_2)\eta_1^2 + 2g_{12}(\xi_1, \xi_2)\eta_1\eta_2 + g_{22}(\xi_1, \xi_2)\eta_2^2 = (\eta_1, \eta_2)G(\xi_1, \xi_2)(\eta_1, \eta_2)^T$$

with the assumption

$$\eta_1^2 + \eta_2^2 = 1.$$

Using the method of Lagrange multipliers it is easy to show, that for any pair of (ξ_1, ξ_2) the eigenvalues of $G(\xi_1, \xi_2)$ are real. They can be computed as

$$\lambda_{1,2}(\xi_1, \xi_2) = \frac{g_{11}(\xi_1, \xi_2) + g_{22}(\xi_1, \xi_2) \pm \sqrt{(g_{11}(\xi_1, \xi_2) - g_{22}(\xi_1, \xi_2))^2 + 4g_{12}(\xi_1, \xi_2)^2}}{2}. \quad (2.45)$$

If $\lambda(\xi_1, \xi_2) = \max\{|\lambda_1(\xi_1, \xi_2)|, |\lambda_2(\xi_1, \xi_2)|\}$ then

$$|(\eta_1, \eta_2) \cdot G(\xi_1, \xi_2) \cdot (\eta_1, \eta_2)^T| \leq \lambda(\xi_1, \xi_2)$$

for any vector, which fulfills $\eta_1^2 + \eta_2^2 = 1$.

This observation can now be applied to the lines which pass through the center of the fat arc (o_1, o_2)

$$(\xi_1(t), \xi_2(t)) = (o_1, o_2) + t(\eta_1, \eta_2), \quad t \in [r - \varrho, r + \varrho], \quad (2.46)$$

where r is the radius of the median arc and ϱ is the fat arc radius. Since we assume, that the parametric function continuously differentiable, then $\lambda(\xi_1, \xi_2)$ is also continuous in Ω , which is a compact domain. Therefore there exists a constant such that

$$0 \leq \lambda(\xi_1, \xi_2) \leq C.$$

Thus for any line segment (2.46) the integral in (2.44) can be bounded by the general bound

$$L \leq 2\varrho\sqrt{C}.$$

□

Example 2.25. We consider the intersection of a cubic implicitly defined surface with a biquadratic surface patch. Fig.2.13, upper row, shows the intersecting surfaces and the implicitly defined intersection curve in the parameter domain. The lower row shows the regions on the surface which correspond to fat arcs in the parameter domain for three different values of the tolerance ε .

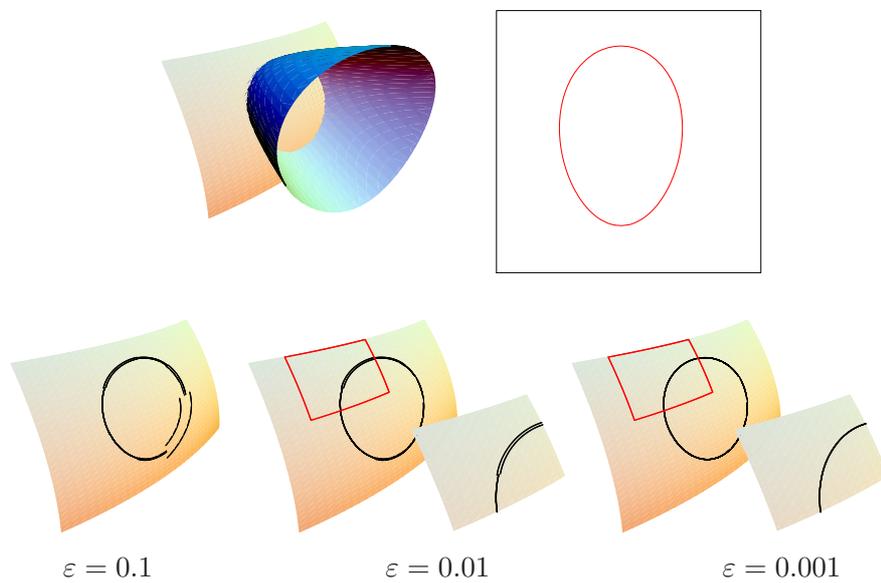


Figure 2.13: Example 2.25: Intersection of a cubic implicit and a biquadratic parametric surface, represented by fat arcs in the parameter domain. The number of fat arcs grows from 10 for $\varepsilon = 0.1$ to 25 for $\varepsilon = 0.01$. For the smaller two tolerances, we also zoomed into a segment of the surface patch.

Chapter 3

Fat Arcs for 3D Implicit Algebraic Curves

Implicitly defined algebraic space curves are defined by the intersection curve of algebraic surfaces. Such curves frequently arise in geometric modeling. Various methods have been developed for approximating or parameterizing them, and for analyzing their topology. In this chapter we present an algorithm, which generates bounding regions for algebraic space curves. The method is the generalization of the fat arc generation method for planar algebraic curves.

3.1 Fat Arcs in 3D

Here we summarize first the related results in algebraic curve approximation. Then we introduce the definition of fat arcs in \mathbb{R}^3 . In the end of the section we will state the approximation problem of algebraic space curves given by Bernstein–Bézier polynomials.

3.1.1 Approximating 3D Algebraic Curves

Computation of surface-surface intersections is a fundamental operation in geometric modeling. It is important for evaluating set operations, for computing boundary curves and closely related to self-intersection problems. A survey of the topic is given by Patrikalakis and Maekawa [30].

Intersecting low degree algebraic surfaces has attracted a lot of interest in the literature. Quadratic surfaces are the simplest curved surfaces, therefore they are frequently used in computational geometry. The intersection computation of such surfaces has been discussed thoroughly in [8, 11, 40, 45, 46].

Several different methods have been developed for computing the intersection of algebraic surfaces (see [19, 31, 39]). Many of them are symbolic-numeric algorithms. The most widely used numeric methods are the lattice evaluation, tracing and subdivision-based methods. The lattice evaluation techniques compute a set of low dimensional sub-problems. Then the solution of these sub-problems is interpolated to approximate the general solution. Marching or tracing methods generate point sequences along the connected components of the curve. They necessarily use some topological information to find starting, turning and singular points [3, 22]. Subdivision algorithms are based on the "divide and conquer" paradigm. They

decompose the problem into several sub-problems, and sort these problems according to the curve topology [2, 25]. The decomposition terminates if suitable approximating primitives can be generated in each sub-problems [29]. In order to construct these approximating primitives several local approximation techniques can be applied, such as interpolation, bounding region generation, least-squares approximation or Newton-type methods [12].

Several different methods have been developed for computing the intersection of general algebraic surfaces. Many of them are symbolic-numeric algorithms. For instance, tracing methods and subdivision-based methods are widely used in practice. These algorithms identify first the topology of the curve [2, 25]. Then they generate parametric space curves, which approximate the implicitly defined space curve [20].

3.1.2 Definition of 3D Fat Arcs

We present in this chapter an algorithm, which approximates algebraic space curves with a set of bounding regions. The bounding primitives are the generalization of the planar fat arcs (see in Section 2.1.2) in 3D space. The algorithm detects regular algebraic curve segments, and approximates them with circular arcs. Then the method bounds the distance of the approximating arc and the algebraic curve segment. Each bounding region is a tubular neighborhood of the approximating arcs with a certain radius, which is the appropriate error bound. Such a bounding primitive is bounded by a segment of a torus and two spherical caps (see Fig.3.1 (b)).

Definition 3.1. *A fat arc is defined \mathbb{R}^3 by*

- *a segment of a circular arc (median arc) $\mathcal{S} \subset \Omega \subset \mathbb{R}^3$.*
- *and a distance $\varrho \in \mathbb{R}$.*

Then the fat arc is the point set

$$\mathcal{F}(\mathcal{S}, \varrho) = \{(x, y, z) : \exists(x_0, y_0, z_0) \in \mathcal{S}, \sqrt{(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2} \leq \varrho\}.$$

The median arc can be represented in two different ways. We can use the parametric form, since circular arcs can be parametrized exactly by rational Bernstein-Bézier-polynomials. It provides the computational advantages of BB-representation form, such as the convex hull property. It is also possible, to define the median arc in an algebraic form. A circular arc always can be given as the zero set of two spherical equations. Representing it with these special quadratic equations is advantageous because of the simple intersection and offset computations.

3.1.3 3d Algebraic Curves

In order to construct fat arcs for algebraic space curves, we shall use the properties of the defining polynomials. We assume that these defining polynomials are given in the Bernstein-Bézier tensor product form with respect to an axis aligned box $\Omega_0 = [\alpha_1, \beta_1] \times [\alpha_2, \beta_2] \times [\alpha_3, \beta_3]$

$$f(x, y, z) = \sum_{i=0}^l \sum_{j=0}^m \sum_{k=0}^n d_{ijk} B_{i,l}^1(x) B_{j,m}^2(y) B_{k,n}^3(z), \quad (3.1)$$

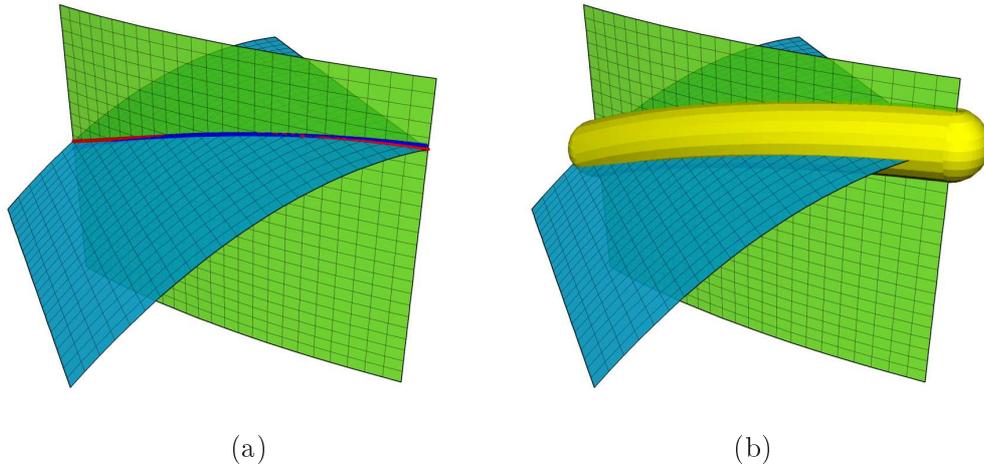


Figure 3.1: Fat arc in \mathbb{R}^3 . The intersection curve (red) approximated by a circular arc (blue) (see figure (a)). Figure (b) shows the δ neighborhood of the median arc, which is the fat arc (yellow).

with certain coefficients $d_{ijk} \in \mathbb{R}$. The basis functions $B_{i,n}^j$ are defined like in Chapter 2 (2.2). For such functions $f : \mathbb{R}^3 \rightarrow \mathbb{R}$, let us denote the defined algebraic surface in the domain Ω_0 with

$$\mathcal{Z}(f, \Omega_0) = \{(x, y, z) : f(x, y, z) = 0\} \cap \Omega_0.$$

The algebraic curve is given as the intersection of the zero sets of two polynomials f and g

$$\mathcal{C}(f, g, \Omega_0) = \mathcal{Z}(f, \Omega_0) \cap \mathcal{Z}(g, \Omega_0) = \{\mathbf{x} : f(\mathbf{x}) = 0, g(\mathbf{x}) = 0\} \cap \Omega_0 \quad (3.2)$$

with respect to the domain Ω_0 . Clearly, the curve may be an empty point set, or it may consist of more than one segment. In order to generate fat arcs, later we consider different segments of the curve $\mathcal{C}(f, g, \Omega)$ in different sub-domains of the original domain $\Omega \subseteq \Omega_0$. All these sub-domains are axis-aligned boxes as well.

3.2 Approximation of Regular Curve Segments

In order to generate fat arcs for 3d algebraic curves we present first a local approximation algorithm, which generates fat arcs only for regular segments of a space curve. Later on we will combine this local bounding region generation with subdivision technique.

3.2.1 Regularity Criterion

As in Section 2.2.1 first we identify the empty sub-domains of the computational domain. It is obvious, that Observation 2.3 in Chapter 2 is true in general for multivariate BB-polynomials.

Observation 3.2. Suppose, that an algebraic curve is represented by two BB-polynomial in the domain $\Omega \subset \mathbb{R}^3$. If one of the polynomials has only negative or only positive BB-coefficients over the domain, then none of the points in Ω belongs to the algebraic curve.

In order to approximate algebraic space curves, we shall detect domains, which contain only regular segments of the curve. Such domains do not contain loops or self-intersections of the curve. Therefore we state following the definitions similarly like in the case of planar algebraic curves (see in Definition 2.1).

Definition 3.3. A point \mathbf{p} of an intersection curve $\mathcal{C}(f, g, \Omega) \subset \mathbb{R}^3$ of two algebraic surfaces $f = 0$ and $g = 0$ is called regular, if the vectors $\nabla f(\mathbf{p})$ and $\nabla g(\mathbf{p})$ are linearly independent (and called singular otherwise). An algebraic curve segment is regular on $\Omega \subset \mathbb{R}^3$, if each point of the segment is regular in the domain.

Definition 3.4. A point \mathbf{p} of an intersection curve $\mathcal{C}(f, g, \Omega) \subset \mathbb{R}^3$ of two algebraic surfaces is called u -regular (u can be equal x, y or z), if the u coordinate of the tangent vector $\nabla f(\mathbf{p}) \times \nabla g(\mathbf{p})$ is not equal to zero. An algebraic curve segment is u -regular in the domain $\Omega \subset \mathbb{R}^3$, if each point of the segment is u -regular in Ω .

The relation in between these definitions is formulated in the following lemma.

Lemma 3.5. If an algebraic curve segment in $\Omega \subset \mathbb{R}^3$ is x, y or z -regular (regular at least in one coordinate), then the curve segment is regular in the domain. Moreover it is not a loop and no self-intersection occurs in the domain.

Proof. If we know, that one of the coordinates of the tangent vector does not vanish in Ω , then the tangent vector does not vanish either in the domain. So the curve is regular. It means, that no self-intersection occurs in the domain. The regularity in one coordinate also excludes the situation, that the tangent vector of the algebraic curve returns to the same position if we trace the curve. The curve can not form loops in the domain. \square

Control of coordinate regularity. In order to certify domains containing regular algebraic curve segments, we use the convex hull property of the Bernstein polynomials. We give here a sufficient condition for detecting such domains. Namely it is sufficient to show, that there exists a bound for one of the coordinate of the vector $\mathbf{t} = \nabla f \times \nabla g$ in the domain, which bounds the coordinate function away from zero. Since we compute with BB-polynomials we can represent each coordinate of vector $\mathbf{t} = (t^1, t^2, t^3)$ in a BB tensor product form as

$$t^u = \sum_{i=0}^l \sum_{j=0}^m \sum_{k=0}^n t_{ijk}^u B_{i,l}^1(x) B_{j,m}^2(y) B_{k,n}^3(z),$$

where the coefficients t_{ijk}^u can be computed from the coefficient of f and g , and $u = 1, 2$ or 3 . For a certain u we denote with m^u the minimum and with M^u the maximum of t_{ijk}^u . If $m^u M^u > 0$ then

$$|t^u| \geq \min\{|m^u|, |M^u|\} = T^u > 0. \tag{3.3}$$

If such T^u exists for $u = 1, 2$ or 3 , it ensures us that the curve is regular at least in one coordinate in Ω .

3.2.2 Local Algorithm

We describe here an algorithm (Algorithm 5) to approximate regular algebraic curve segments. It is a local fat arc generation method in certain sub-domains of the original domain

Algorithm 5 FatArcLocal_3d($f, g, \Omega, \varepsilon$)**Require:** The curve is regular at least in one coordinate in Ω .

```

1:  $\hat{f}, \hat{g}$  modified and orthogonalized polynomials           {see (3.18)}
2:  $p, q \leftarrow T_c^2(\hat{f}), T_c^2(\hat{g})$  spherical approximations
3: if  $p, q$  exist then
4:    $\mathcal{P} \leftarrow$  zero contour of  $p$ 
5:    $\mathcal{Q} \leftarrow$  zero contour of  $q$ 
6:    $\mathcal{S} \leftarrow \mathcal{P} \cap \mathcal{Q}$                                    {median circle}
7:   if  $\mathcal{S} \neq \emptyset$  then
8:      $G \leftarrow$  lower bound for  $\|\nabla \hat{f}\|$  and  $\|\nabla \hat{g}\|$ 
9:      $K \leftarrow$  upper bound for  $|\nabla \hat{f} \cdot \nabla \hat{g}|$ 
10:    if  $0 < G$  and  $0 < G^2 - K$  then
11:       $\varrho \leftarrow$  upper bound of  $\text{HD}_\Omega(\mathcal{S}, \mathcal{C}(\hat{f}, \hat{g}, \Omega))$    {see Lemma 3.15}
12:      if  $\varrho \leq \varepsilon$  then
13:         $\mathcal{F} = \{\mathbf{x} : \exists \mathbf{y} \in \mathcal{S}, |\mathbf{x} - \mathbf{y}| \leq \varrho\} \cap \Omega$    {fat arc}
14:        return  $\mathcal{F}$                                            {fat arc has been found}
15:      end if
16:    end if
17:  end if
18: end if
19: return  $\emptyset$                                            {no fat arc has been found}

```

Ω_0 . Later on we will describe a global algorithm, which detects the domains, where the local algorithm is applicable. This algorithm is based on the fat arc generation technique, what we introduced in Section 2.3 for the planar algebraic curves.

The local algorithm assumes that the curve is regular at least in one coordinate in order to approximate a regular algebraic segment which is not a loop.

We have generalized the median arc generation techniques from Section 2.3.2 and Section 2.3.3. These algorithms generate the median arc in algebraic form, as the intersection of two implicitly defined spheres. The intersecting spheres are chosen from a family of spheres, which approximate certain combinations of the original polynomials. Later on, if we would like to represent the output in parametric form, it is easy to describe the circular arcs as rational quadratic curves.

The distance estimation method generalizes and combines the approaches in Section 2.3.5 and Section 2.2.4. It bounds the BB-distance in between each polynomials and the associated quadratic approximations. Then an upper bound is generated for the one sided Hausdorff-distance of the median arc and the algebraic space curve. This bound is used then as the thickness of the fat arc.

The algorithm is successful, if the median arc is found and the fat arc thickness can be computed and it is smaller than the prescribed tolerance ε . Then the algorithm returns with a fat arc, which bounds the curve segment in the appropriate domain. If the local algorithm fails then the algorithm returns the empty set.

Fig.3.2 presents three examples of fat arcs which have been generated with the help of Algorithm 5. According to the median arc generation technique we can see different results for the same algebraic curve segment in each column of the figure. The used median arc

generation techniques are described in Section 3.3. The distance estimation technique is described in Section 4.4.

3.3 Median Arc Generation

Implicitly defined spaces curves are given as the intersection curve of two implicitly defined surfaces. In order to approximate algebraic curve segments, we generate implicitly defined arcs in \mathbb{R}^3 . These algebraic arcs the so called median arcs can be given as the intersection curve of two implicitly defined spheres. In order to generate the defining spheres of the median arc, we can choose different strategies. Several fitting techniques, for instance least-squares fitting, are used in geometric computing. In this section we generalize the approximation techniques from Section 2.3.

3.3.1 Median Arc Generation with Least-Squares Approximation

Similarly to Section 2.3.2 we can use least-squares approximation to find approximating polynomials. In order to generate a quadratic polynomial with spherical zero level set, we are searching for a polynomial in the form

$$s_i(x, y, z) = a_i(x^2 + y^2 + z^2) + b_ix + c_iy + d_iz + e_i.$$

We modify the original functions using a linear term

$$\begin{aligned} \hat{f}(x, y, z) &= l(x, y, z) f(x, y, z) = (l_0 + l_1x + l_2y + l_3z) f(x, y, z), \\ \hat{g}(x, y, z) &= k(x, y, z) g(x, y, z) = (k_0 + k_1x + k_2y + k_3z) g(x, y, z). \end{aligned}$$

The approximation problems can be formed as the following optimization problems

$$\min_{\mathbf{v}_f \in \mathbb{R}^9} \iint_{\Omega} \|\hat{f} - s_1\|^2 dx dy dz, \quad (3.4)$$

$$\min_{\mathbf{v}_g \in \mathbb{R}^9} \iint_{\Omega} \|\hat{g} - s_2\|^2 dx dy dz. \quad (3.5)$$

where

$$\begin{aligned} \mathbf{v}_f &= (a_1, b_1, c_1, d_1, e_1, l_0, l_1, l_2, l_3), \\ \mathbf{v}_g &= (a_2, b_2, c_2, d_2, e_2, k_0, k_1, k_2, k_3). \end{aligned}$$

In order get a unique solution, we have to normalize both minimization problems. Here we present three different strategies. The first normalization technique is using a linear condition. It is a natural condition in the sense that the modified polynomial \hat{f} or \hat{g} keeps the original function value in the center of the computational domain Ω . For instance for the first problem (3.4) the condition can be formulated as

$$l_0 + l_1c^x + l_2c^y + l_3c^z = 1, \quad (3.6)$$

where $\mathbf{c} = (c^x, c^y, c^z)$ denotes the center of the domain. Another possible choice for normalization is to control the gradient length of the approximating polynomials s_i . Such a condition determine two possible solutions for $s_i(x, y, z)$. The one with smaller value in (3.4)

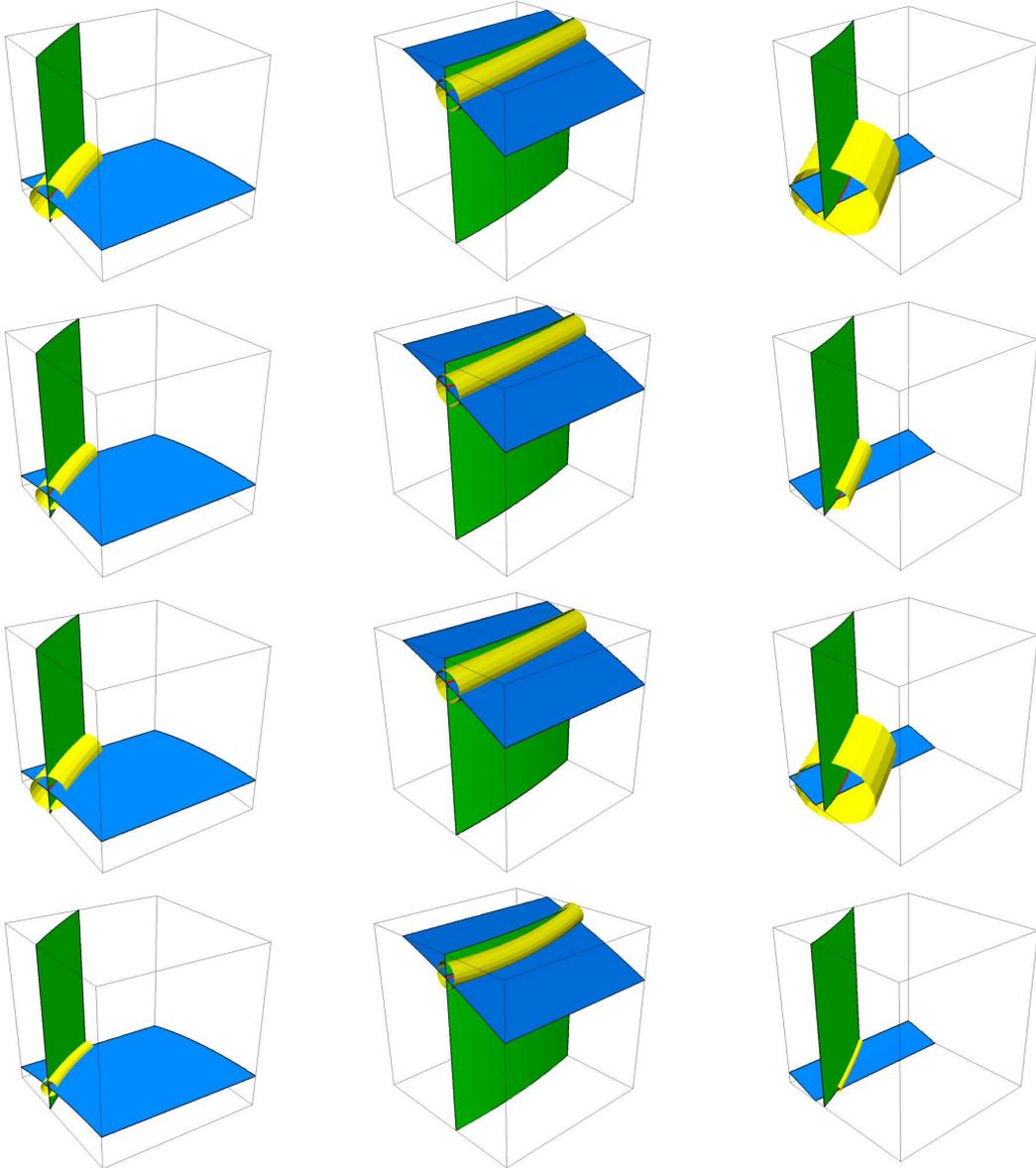


Figure 3.2: Examples for fat arc generation with the help of algorithm `FatArcLocal_3d`. We used four different median arc generation methods on three examples. In the first three rows we show fat arcs generated by least squares approximation with linear normalization and least squares approximation with two different quadratic normalization ((3.7) and (3.8)). In the last row we can see the fat arcs generated by Taylor expansion modification.

or (3.5) can be used as an approximating polynomial. A natural choice of the quadratic normalization condition is

$$\|\nabla s_i(\mathbf{c})\| = 1. \quad (3.7)$$

Another possibility is to use a quadratic normalization condition which approximates better the secondary shape of the original implicitly defined curve. Namely we can suppose for instance

$$\begin{aligned} \|\nabla s_1(\mathbf{c})\| &= \|\nabla f(\mathbf{c})\|, \\ \|\nabla s_2(\mathbf{c})\| &= \|\nabla g(\mathbf{c})\|. \end{aligned} \quad (3.8)$$

We compare the different least-square approximation methods via an example in the convergence rate analysis in Section 3.5.4.

3.3.2 Median Arc Generation Using Taylor Expansion

The algebraic space curve is given by the zero sets of two polynomials f and g in the domain $\Omega \subset \mathbb{R}^3$. In order to generate the median curve, we reformulate the approximation problem. More precisely, we try to find a certain combination of the given polynomials f and g , that possesses a special Hessian matrix in the center point $\mathbf{c} = (c^x, c^y, c^z)$ of the sub-domain Ω . Such a new polynomial h can be defined as the combination

$$h = kf + lg, \quad (3.9)$$

where k and l are linear polynomials and $(x, y, z) \in \Omega$

$$\begin{aligned} k(x, y, z) &= u + k_1(x - c^x) + k_2(y - c^y) + k_3(z - c^z) \\ l(x, y, z) &= v + l_1(x - c^x) + l_2(y - c^y) + l_3(z - c^z), \end{aligned} \quad (x, y, z) \in \Omega.$$

The zero level set of the polynomial h

$$\mathcal{Z}(h, \Omega) = \{\mathbf{x} : h(\mathbf{x}) = 0\} \cap \Omega$$

is a surface, which contains the algebraic curve defined by f and g

$$\mathcal{C}(f, g, \Omega) \subseteq \mathcal{Z}(h, \Omega).$$

We choose the coefficients of k and l such that the Hessian of h is a scalar multiple of the identity matrix in the center of the domain \mathbf{c} .

$$\mathcal{H}(h)(\mathbf{c}) = \begin{pmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ 0 & 0 & \lambda \end{pmatrix}, \quad \lambda \in \mathbb{R}. \quad (3.10)$$

If such an h can be computed, then the zero level set of the quadratic Taylor expansion of h about \mathbf{c} is a sphere. In order to find h , we solve a linear system with eight variables (the coefficients of k and l) and five equations, that can be deduced from (3.10)

$$\begin{aligned} h_{xx}(\mathbf{c}) - h_{yy}(\mathbf{c}) &= 0 \\ h_{yy}(\mathbf{c}) - h_{zz}(\mathbf{c}) &= 0 \\ h_{xy}(\mathbf{c}) &= 0 \\ h_{yz}(\mathbf{c}) &= 0 \\ h_{xz}(\mathbf{c}) &= 0. \end{aligned} \quad (3.11)$$

If the system has full rank, then the solution set in the space of coefficients of k and l is three-dimensional. Therefore we choose two coefficients as parameters in advance. More precisely, we suppose that the values of the constant terms of the polynomials k and l are arbitrary but fixed $(u, v) \in \mathbb{R}^2$ and different from zero ($u \neq 0$ and $v \neq 0$).

Lemma 3.6. *Given two polynomials f and g over the domain $\Omega \subset \mathbb{R}^3$. We suppose that in the center of the domain*

$$\|\nabla f(\mathbf{c}) \times \nabla g(\mathbf{c})\| \neq 0. \quad (3.12)$$

Then for any pair of $(u, v) \in \mathbb{R}^2$, where $u \neq 0$ and $v \neq 0$, there exists an exactly one-dimensional family of non-trivial polynomials, k and l , such that $h = kf + lg$ satisfies (3.11).

Proof. The Hessian matrix of h can be expressed with the help of f, g, k and l as

$$\begin{aligned} \mathcal{H}(h)(\mathbf{c}) &= \nabla k(\mathbf{c})\nabla f(\mathbf{c})^\top + \nabla f(\mathbf{c})\nabla k(\mathbf{c})^\top + u\mathcal{H}(f)(\mathbf{c}) \\ &\quad + \nabla l(\mathbf{c})\nabla g(\mathbf{c})^\top + \nabla g(\mathbf{c})\nabla l(\mathbf{c})^\top + v\mathcal{H}(g)(\mathbf{c}). \end{aligned} \quad (3.13)$$

For any values of the parameters u and v , the system (3.11) can be reformulated as

$$\mathbf{A}\mathbf{k} = \begin{pmatrix} f_x(\mathbf{c}) & -f_y(\mathbf{c}) & 0 & g_x(\mathbf{c}) & -g_y(\mathbf{c}) & 0 \\ 0 & f_y(\mathbf{c}) & -f_z(\mathbf{c}) & 0 & g_y(\mathbf{c}) & -g_z(\mathbf{c}) \\ f_y(\mathbf{c}) & f_x(\mathbf{c}) & 0 & g_y(\mathbf{c}) & g_x(\mathbf{c}) & 0 \\ 0 & f_z(\mathbf{c}) & f_y(\mathbf{c}) & 0 & g_z(\mathbf{c}) & g_y(\mathbf{c}) \\ f_z(\mathbf{c}) & 0 & f_x(\mathbf{c}) & g_z(\mathbf{c}) & 0 & g_x(\mathbf{c}) \end{pmatrix} \begin{pmatrix} k_1 \\ k_2 \\ k_3 \\ l_1 \\ l_2 \\ l_3 \end{pmatrix} = \mathbf{b}, \quad (3.14)$$

where the vector of constants is

$$\mathbf{b} = -u \begin{pmatrix} \frac{1}{2}(f_{xx}(\mathbf{c}) - f_{yy}(\mathbf{c})) \\ \frac{1}{2}(f_{yy}(\mathbf{c}) - f_{zz}(\mathbf{c})) \\ f_{xy}(\mathbf{c}) \\ f_{yz}(\mathbf{c}) \\ f_{xz}(\mathbf{c}) \end{pmatrix} - v \begin{pmatrix} \frac{1}{2}(g_{xx}(\mathbf{c}) - g_{yy}(\mathbf{c})) \\ \frac{1}{2}(g_{yy}(\mathbf{c}) - g_{zz}(\mathbf{c})) \\ g_{xy}(\mathbf{c}) \\ g_{yz}(\mathbf{c}) \\ g_{xz}(\mathbf{c}) \end{pmatrix}.$$

In order to be certain that the system (3.14) has a one-parameter family solution system, we have to show, that the matrix \mathbf{A} has rank 5. Therefore we analyze the 5×5 sub-matrices of \mathbf{A} . We denote with \mathbf{A}_i the matrix, which we get from \mathbf{A} by deleting i th column. The determinants of the matrices $\mathbf{A}_{4,5,6}$ are

$$\begin{aligned} \det(\mathbf{A}_4) &= -f_x(\mathbf{c}) \|\nabla f(\mathbf{c}) \times \nabla g(\mathbf{c})\|^2, \\ \det(\mathbf{A}_5) &= f_y(\mathbf{c}) \|\nabla f(\mathbf{c}) \times \nabla g(\mathbf{c})\|^2, \\ \det(\mathbf{A}_6) &= -f_z(\mathbf{c}) \|\nabla f(\mathbf{c}) \times \nabla g(\mathbf{c})\|^2. \end{aligned}$$

We know that $\|\nabla f(\mathbf{c}) \times \nabla g(\mathbf{c})\| \neq 0$. This observation also implies, that one of the coordinates of $\nabla f(\mathbf{c})$: $f_x(\mathbf{c})$, $f_y(\mathbf{c})$ or $f_z(\mathbf{c})$ is non-zero. It means, that one of the determinants of \mathbf{A}_4 , \mathbf{A}_5 or \mathbf{A}_6 is not zero. So \mathbf{A} always has full rank 5. Thus the solution of the system $\mathbf{A}\mathbf{k} = \mathbf{b}$ exists, and it is a one dimensional subspace in \mathbb{R}^6 . \square

According to Lemma 3.6, for any pair of (u, v) where $u \neq 0$ and $v \neq 0$, there exists a one-parameter family of polynomials k and l , such that $kf + lg$ satisfies (3.11). From this family of polynomials we always choose the one, which minimizes the l_2 norm

$$\|\mathbf{k}\|_2 \rightarrow \min \text{ subject to } \mathbf{A}\mathbf{k} = \mathbf{b}, \quad (3.15)$$

where $\mathbf{k} = (k_1, k_2, k_3, l_1, l_2, l_3)$ is the common coefficient vector of k and l . This guarantees that the solution behaves numerically well during the computations. With the additional condition (3.15) the polynomials k and l can be computed uniquely for each pair of (u, v) . Moreover the result depends continuously on the points of the domain. We introduce the function \mathcal{G} , which assigns to a function f and g , a value of (u, v) and the center point \mathbf{c} of a domain Ω the associated function according to the construction in Lemma 3.6 and the former assumption (3.15)

$$\mathcal{G}(f, g, (u, v), \mathbf{c}) = h = kf + lg. \quad (3.16)$$

Remark 3.7. Suppose that the right hand side of the system (3.14) i.e. the vector \mathbf{b} , vanishes for a certain pair of (u, v) . In this case the solution set of (3.14) is a line, which passes through the origin. Then the linear combination $uf + vg$ fulfills the condition (3.11). According to (3.15) we always choose the solution of the system (3.14), which has the smallest length. In this special case both k and l are constants.

The polynomial $h = \mathcal{G}(f, g, (u, v), \mathbf{c})$ fulfills the special condition for the Hessian (3.11). Thus the quadratic Taylor expansion of h about \mathbf{c} has a spherical zero level set.

$$\begin{aligned} s(\mathbf{x}) &= T_{\mathbf{c}}^2(h^*)(\mathbf{x}) = \\ &= h(\mathbf{c}) + \nabla h(\mathbf{c})^T(\mathbf{x} - \mathbf{c}) + \frac{1}{2}h_{xx}(\mathbf{c})(\mathbf{x} - \mathbf{c})^T(\mathbf{x} - \mathbf{c}) \quad \forall \mathbf{x} \in \Omega. \end{aligned} \quad (3.17)$$

If we compute two polynomials for two different pairs of parameter $(u, v) \neq (u', v')$

$$\hat{f} = \mathcal{G}(f, g, (u, v), \mathbf{c}) \quad \text{and} \quad \hat{g} = \mathcal{G}(f, g, (u', v'), \mathbf{c}), \quad \text{such that} \quad u, v, u', v' \neq 0, \quad (3.18)$$

then their quadratic Taylor expansions about \mathbf{c} can be denoted by

$$p = T_{\mathbf{c}}^2 \hat{f} \quad \text{and} \quad q = T_{\mathbf{c}}^2 \hat{g}.$$

These two polynomials define the algebraic set

$$\mathcal{S}(p, q, \Omega) = \{\mathbf{x} : p(\mathbf{x}) = 0, q(\mathbf{x}) = 0\} \cap \Omega.$$

If this algebraic set is not empty, then it forms a circular arc. This arc can be used as an approximating circular arc of the curve $\mathcal{C}(f, g, \Omega)$. Later on the error of the approximation is estimated by a distance bound of the algebraic curves $\mathcal{C}(\hat{f}, \hat{g}, \Omega)$ and $\mathcal{S}(p, q, \Omega)$.

Orthogonal pair of functions We compute a pair of polynomials for two different pairs of parameter $(u, v) \neq (u', v')$

$$\hat{f} = \mathcal{G}(f, g, (u, v), \mathbf{c}) \quad \text{and} \quad \hat{g} = \mathcal{G}(f, g, (u', v'), \mathbf{c}).$$

In order to get efficient distance bound for the algebraic curve and the median arc, we prefer to generate such a pair of functions f^*, g^* , which has the property

$$\nabla f^*(\mathbf{c}) \perp \nabla g^*(\mathbf{c}). \quad (3.19)$$

in the center of the domain. If F and G are defined as

$$\begin{aligned} F &= \hat{f} \|\nabla \hat{g}(\mathbf{c})\| + \hat{g} \|\nabla \hat{f}(\mathbf{c})\| \\ G &= \hat{f} \|\nabla \hat{g}(\mathbf{c})\| - \hat{g} \|\nabla \hat{f}(\mathbf{c})\|, \end{aligned} \quad (3.20)$$

then the following polynomials satisfy (3.19)

$$f^* = \frac{F}{\|\nabla F(\mathbf{c})\|}, \quad g^* = \frac{G}{\|\nabla G(\mathbf{c})\|}. \quad (3.21)$$

Thus we introduce the function \mathcal{O} , which assigns to the polynomials \hat{f} and \hat{g} and the center point \mathbf{c} of a domain Ω . It generates a pair of functions

$$(f^*, g^*) = \mathcal{O}(\hat{f}, \hat{g}, \mathbf{c}), \quad (3.22)$$

which is computed with applying the orthogonalization steps (3.20) and (3.21).

The functions \hat{f} and \hat{g} are linearly independent since we computed them as the sum and the difference of two linearly independent non-zero functions.

Remark 3.8. Any linear combination of $h_i = T(f, g, (u_i, v_i), \mathbf{c})$, computed for the parameter values (u_i, v_i) , fulfills the property of functions with special Hessians (3.11). So if

$$\hat{h} = \sum_{i=1}^n c_i h_i, \quad c_i \in \mathbb{R},$$

then \hat{h} also fulfills the condition of special Hessian (see (3.11)). Thus $T_{\mathbf{c}}^2(\hat{h})(\mathbf{x}) = 0$ defines a sphere in \mathbb{R}^3 .

According to this remark, also the condition of special Hessians (3.11) is satisfied by f^* and g^* . So we define the following approximating polynomials as

$$\begin{aligned} p &= T_{\mathbf{c}}^2(f^*) \\ q &= T_{\mathbf{c}}^2(g^*). \end{aligned}$$

From the construction of f^* and g^* it is clear, that the vectors $\nabla p(\mathbf{c})$ and $\nabla q(\mathbf{c})$ are also independent and perpendicular to each other. Since \hat{f} and \hat{g} satisfy (3.11), the equations $p = 0$ and $q = 0$ are equations of a spheres. The median arc \mathcal{S} is defined by the intersection curve of the zero set of the polynomials p and q in the domain.

$$\mathcal{S}(p, q, \Omega) = \{\mathbf{x} : p(\mathbf{x}) = 0, q(\mathbf{x}) = 0\} \cap \Omega.$$

If it is the empty set, then no median arc is generated.

3.3.3 Connection with the Osculating Circle

In this section we suppose, that the center of the computational domain Ω is a point of the algebraic curve \mathcal{C} defined by the polynomials f and g . If the center point is denoted by \mathbf{c} , then

$$f(\mathbf{c}) = g(\mathbf{c}) = 0. \quad (3.23)$$

This special case plays an important role during the computations, since later we would like to approximate the curve in such sub-domains of the original domain, which tightly enclose the algebraic curve.

For an arbitrary pair of parameters we compute a new polynomial as the combination of f and g as defined in Section 3.3.2

$$h = \mathcal{G}(f, g, (u, v), \mathbf{c}).$$

Consider the quadratic polynomial

$$s = T_{\mathbf{c}}^2(h).$$

According to the assumption (3.23) the center of the domain is a point of the zero set of h and s .

$$h(\mathbf{c}) = s(\mathbf{c}) = uf(\mathbf{c}) + vg(\mathbf{c}) = 0. \quad (3.24)$$

Then the quadratic approximating polynomial s has the following form

$$s(\mathbf{x}) = \nabla h(\mathbf{c})^T(\mathbf{x} - \mathbf{c}) + \lambda(\mathbf{x} - \mathbf{c})^T(\mathbf{x} - \mathbf{c}), \quad (3.25)$$

where the gradient is

$$\nabla h(\mathbf{c}) = u\nabla f(\mathbf{c}) + v\nabla g(\mathbf{c}), \quad (3.26)$$

and the Hessian-matrix has the form

$$\mathcal{H}(h)(\mathbf{c}) = \lambda\mathbf{I}^3,$$

as in (3.10).

Observation 3.9. Suppose, that $\lambda \neq 0$, then the zero set of s can be written in the form

$$\left\langle \mathbf{x} - \left(\mathbf{c} + \frac{1}{\lambda} \nabla h(\mathbf{c}) \right), \mathbf{x} - \left(\mathbf{c} + \frac{1}{\lambda} \nabla h(\mathbf{c}) \right) \right\rangle = \frac{\|\nabla h(\mathbf{c})\|^2}{\lambda^2}.$$

Therefore the radius of the sphere $s = 0$ can be computed as

$$r = \frac{\|\nabla h(\mathbf{c})\|}{\lambda}.$$

Observation 3.10. The zero set of s defined in (3.25) depends only on the ratio of the chosen parameters u and v . Therefore the sphere family, computed for different values of (u, v) , is a one-parametric surface family. It can be parametrized by the ratio of u and v . This follows from the computational method of k and l and from the special form of the sphere equations (see in (3.25)). Fig.3.3 (a) visualizes several members of such a sphere family for different values of the ratio u/v .

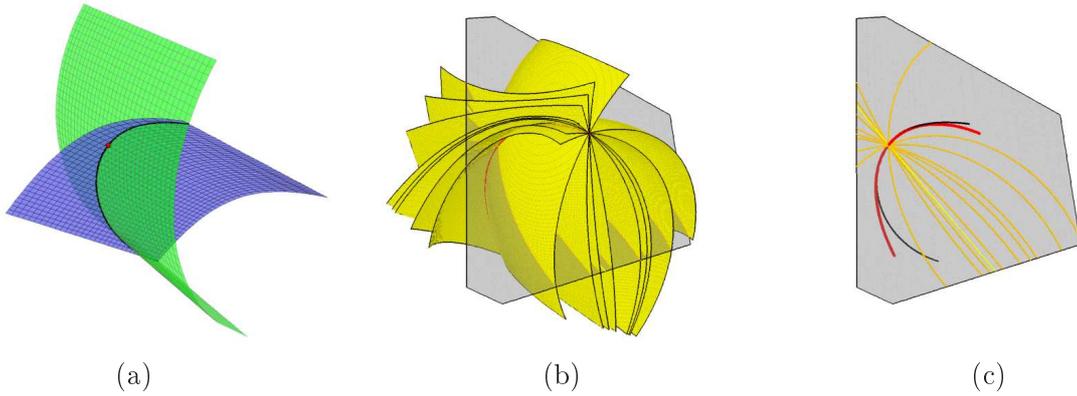


Figure 3.3: Sphere family computed with Taylor expansion modification (b) about a point on the algebraic curve (a) and its intersection with the normal plane of the curve (c). The thin, black curve is the algebraic curve. The red circle is the osculating circle.

Remark 3.11. In the computations we chose the two parameter pairs (u, v) and (u', v') usually as $(1, 2)$ and $(2, 1)$ or $(1, 5)$ and $(5, 1)$. Both choices generated similar results in our examples, since the generated median arcs converge to the same limit circle, the osculating circle. Therefore if $u, v, u', v' \neq 0$ and $u/v \neq u'/v'$, then (u, v) and (u', v') are good initial choices for parameters. It is not possible to improve the general behavior of the algorithm by the choice of these parameters.

Lemma 3.12. *We assume that (3.23) is satisfied in the point \mathbf{c} . Then for each sphere equation, computed for certain $(u, v) \in \mathbb{R}^2$, $u, v \neq 0$, the center of the sphere $s = 0$ lies in the normal plane of the algebraic curve in the point \mathbf{c} . Moreover the inverse of the radius of the sphere is exactly the normal curvature κ_n of the tangent direction $\nabla f(\mathbf{c}) \times \nabla g(\mathbf{c})$ of the surface $\mathcal{G}(f, g, (u, v), \mathbf{c})$ in the point \mathbf{c} .*

Proof. Suppose that in a certain neighborhood of the point \mathbf{c} the algebraic curve can be parametrized with arc length parametrization. It is not a restriction, since we are computing only with regular segment of the algebraic curve. The parametrization is denoted by

$$\mathbf{p}(t), \quad \text{where } \mathbf{p}(t_0) = \mathbf{c}.$$

This curve is a curve on the surface $h = 0$ according to the definition, therefore it satisfies

$$\frac{d^i h(\mathbf{p}(t))}{dt^i} = 0,$$

for any i .

If we compute the first derivative in the point \mathbf{c} :

$$\left. \frac{dh(\mathbf{p}(t))}{dt} \right|_{t=t_0} = \langle \nabla h(\mathbf{c}), \mathbf{p}'(t_0) \rangle = 0.$$

Thus the tangent vector of the algebraic curve is parallel with the cross product of the gradients $\nabla f(\mathbf{c})$ and $\nabla g(\mathbf{c})$. In (3.26) we observed, that

$$\nabla h(\mathbf{c}) = u \nabla f(\mathbf{c}) + v \nabla g(\mathbf{c}).$$

Since s is the quadratic Taylor expansion of h about \mathbf{c} , we obtain that

$$\langle \nabla s(\mathbf{c}), \mathbf{p}'(t_0) \rangle = 0.$$

This implies, that for any value of the parameters (u, v) the gradient of the associated sphere is in the normal plane of the algebraic curve in the point \mathbf{c} .

The second derivative in the point \mathbf{c} is

$$\begin{aligned} \left. \frac{d^2 h(\mathbf{p}(t))}{dt^2} \right|_{t=t_0} &= \langle \nabla h(\mathbf{c}), \mathbf{p}''(t_0) \rangle + \mathbf{p}'(t_0) \mathcal{H}(h)(\mathbf{c}) \mathbf{p}'(t_0) = \\ &= \langle \nabla h(\mathbf{c}), \mathbf{p}''(t_0) \rangle + \lambda \langle \mathbf{p}'(t_0), \mathbf{p}'(t_0) \rangle = 0. \end{aligned}$$

Since we used the arc length parametrization

$$\langle \nabla h(\mathbf{c}), \mathbf{p}''(t_0) \rangle - \lambda = 0.$$

The polynomial s is the quadratic Taylor expansion of h about \mathbf{c} , therefore also

$$\langle \nabla s(\mathbf{c}), \mathbf{p}''(t_0) \rangle = \lambda.$$

If we expand the scalar product:

$$\|\nabla s(\mathbf{c})\| \|\mathbf{p}''(t_0)\| \cos \varphi = -\lambda,$$

where φ denotes the angle of the surface normal $\nabla h(\mathbf{c})$ and the normal direction of the algebraic curve in \mathbf{c} . According to the Theorem of Meusnier and Observation 3.9 we finally arrive at

$$\kappa \cos \varphi = \kappa_n = \frac{\lambda}{\|\nabla s(\mathbf{c})\|} = \frac{1}{r},$$

which proves the lemma. □

As an example Fig.3.3 (b) shows the intersection of the sphere family and the normal plane of the algebraic curve. Each sphere of the family intersects this plane in a great circle. These circles intersect each other in two points on the normal of the algebraic space curve.

Corollary 3.13. *The functions f and g define an algebraic curve $\mathcal{C}(f, g, \Omega)$ in $\Omega \subset \mathbb{R}^3$. We assume that the point $\mathbf{c} \in \Omega$ lies on the algebraic curve $\mathbf{c} \in \mathcal{C}(f, g, \Omega)$. We compute the function family $h(u, v) = \mathcal{G}(f, g, (u, v), \mathbf{c})$ with special Hessian for f and g in the point \mathbf{c} . The quadratic Taylor expansion for any (u, v) pair $u, v \neq 0$ has a spherical zero level set. The intersection of this sphere family is a circle, which is the osculating circle of $\mathcal{C}(f, g, \Omega)$ in the point \mathbf{c} .*

Proof. In each point of a curve on a surface the osculating circle is the normal section of the curvature sphere of the surface [21]. In Lemma 3.12 we observed that this curvature sphere for any $h(u, v) = 0$ surface is the zero set of the quadratic Taylor expansion. These spheres have the same intersection curve with the osculating plane of $\mathcal{C}(f, g, \Omega)$ in the point \mathbf{c} , which is exactly the osculating circle. □

3.4 Distance Estimate

In this section we describe a method to estimate the distance of two algebraic space curves. In order to get a distance bound, we combine a distance bound of parametric and algebraic curves and a distance estimation strategy between algebraic surfaces.

3.4.1 Distance of Implicitly Defined Surfaces

First we generalize the distance estimation technique from Chapter 2.3.5 for algebraic surfaces. In order to measure the distance of algebraic surfaces, we consider the BB-norm over the domain $\Omega \subset \mathbb{R}^3$: $\|\cdot\|_{\text{BB}}^{\Omega}$, which is the maximum absolute value of the coefficients in the BB-representation. With the help of the norm, a distance bound can be defined between an arbitrary polynomial f and an approximating polynomial p for all point in the domain

$$\varepsilon = \|f - p\|_{\text{BB}}^{\Omega}. \quad (3.27)$$

Due to the convex hull property

$$|f(\mathbf{x}) - p(\mathbf{x})| \leq \varepsilon, \quad \forall \mathbf{x} \in \Omega.$$

This implies that

$$p(\mathbf{x}) - \varepsilon \leq f(\mathbf{x}) \leq p(\mathbf{x}) + \varepsilon, \quad \forall \mathbf{x} \in \Omega. \quad (3.28)$$

A region can be defined in Ω by the approximating polynomial and the distance bound

$$\mathcal{D}(p, \varepsilon) = \{\mathbf{x} : |p(\mathbf{x})| \leq \varepsilon\} \cap \Omega.$$

The algebraic surface defined by f is the point set

$$\mathcal{Z}(f, \Omega) = \{\mathbf{x} : f(\mathbf{x}) = 0\} \cap \Omega.$$

The region \mathcal{D} is a bounding region for the zero level set of the polynomial f in Ω

$$\mathcal{Z}(f, \Omega) \subseteq \mathcal{D}(p, \varepsilon) \subseteq \Omega.$$

3.4.2 Distance of Algebraic and Parametric Space Curves

In order to bound the distance of algebraic space curves, we recall a former result from [20]. In the paper a technique is described to bound the distance of parametric and algebraic space curves.

We assume that the a curve segment $\mathbf{r}(t)$ is defined with the parameter domain $t \in [0, 1]$ in $\Omega \subset \mathbb{R}^3$. The curve traces the point set

$$\mathcal{R} = \{\mathbf{r}(t) : t \in [0, 1]\}.$$

The algebraic curve $\mathcal{C}(f, g, \Omega)$ is defined by the intersection curve of f and g on the sub-domain Ω . In order to avoid certain technical difficulties, we bound the distance between the point set \mathcal{R} and

$$\mathcal{C}^* = \mathcal{C} \cup \partial\Omega,$$

where $\partial\Omega$ denotes the boundary of the domain. The one-sided Hausdorff-distance is defined as

$$\text{HD}_\Omega(\mathcal{R}, \mathcal{C}^*) = \sup_{t \in [0,1]} \inf_{\mathbf{x} \in \mathcal{C}^*} \|\mathbf{x} - \mathbf{r}(t)\|. \quad (3.29)$$

In order to bound this distance we use the following result from [20].

Theorem 3.14 (Jüttler–Chalmovianský). *Consider a curve segment $\mathbf{r}(t) : t \rightarrow \Omega$, which traces the point set \mathcal{R} . The functions f and g define the algebraic curve $\mathcal{C}(f, g, \Omega)$ in Ω . We assume that positive constants G and K exist, such that*

$$G \leq \|\nabla f\| \quad \text{and} \quad G \leq \|\nabla g\|,$$

and

$$|\nabla f \cdot \nabla g| \leq K.$$

If h denotes the function $\sqrt{f^2 + g^2}$, then

$$\|\nabla h\| \leq \sqrt{G^2 - K} \quad \forall \mathbf{x} \in \Omega.$$

Moreover if there exists a positive constant M , that $f(\mathbf{r}(t))^2 + g(\mathbf{r}(t))^2 \leq M^2$, then the one-sided Hausdorff-distance is bounded by

$$\text{HD}_\Omega(\mathcal{R}, \mathcal{C}^*) \leq \frac{M}{\sqrt{G^2 - K}}. \quad (3.30)$$

3.4.3 Distance of Algebraic Space Curves

If we would like to estimate the distance of algebraic space curve we can measure first the distance of the defining algebraic surfaces. Suppose that an algebraic curve $\mathcal{C}(f, g, \Omega)$ is defined by the polynomials f and g in the domain Ω . An approximating space curve \mathcal{S} is given by two approximating algebraic surfaces $p = 0$ and $q = 0$ as

$$\mathcal{S}(p, q, \Omega) = \{\mathbf{x} : p(\mathbf{x}) = 0, q(\mathbf{x}) = 0\} \cap \Omega.$$

The polynomial p approximates f , as q is an approximating polynomial of g . We estimate the distance between the algebraic surfaces and the approximating surfaces pairwise using the technique from Section 3.4.1.

$$\varepsilon_1 = \|f - p\|_{\text{BB}}^\Omega, \quad \varepsilon_2 = \|g - q\|_{\text{BB}}^\Omega.$$

For all points $\mathbf{x} \in \mathcal{S}(p, q, \Omega)$ it is satisfied that

$$|f(\mathbf{x})| \leq \varepsilon_1 \quad \text{and} \quad |g(\mathbf{x})| \leq \varepsilon_2 \quad (3.31)$$

according to the definition $p(\mathbf{x}) = 0$ and $q(\mathbf{x}) = 0$.

Corollary 3.15. *Consider two algebraic curves $\mathcal{C}(f, g, \Omega)$ and $\mathcal{S}(p, q, \Omega)$, defined by the polynomials f, g and p, q in the domain $\Omega \subset \mathbb{R}^3$. We denote by ε_1 and ε_2 the norms*

$$\varepsilon_1 = \|f - p\|_{\text{BB}}^\Omega \quad \text{and} \quad \varepsilon_2 = \|g - q\|_{\text{BB}}^\Omega.$$

We assume that the positive constants G and K exist, such that

$$G \leq \|\nabla f\| \quad \text{and} \quad G \leq \|\nabla g\|,$$

and

$$|\nabla f \cdot \nabla g| \leq K.$$

If $G > 0$ and $G^2 - K > 0$, then for all points $\mathbf{x} \in \mathcal{S}$ exists a point $\mathbf{y} \in \mathcal{C}$ such that

$$\|\mathbf{x} - \mathbf{y}\| \leq \sqrt{\frac{\varepsilon_1^2 + \varepsilon_2^2}{G^2 - K}}. \quad (3.32)$$

Proof. According to (3.31) for all $\mathbf{x} \in \mathcal{S}(p, q, \Omega)$

$$|f(\mathbf{x})| \leq \varepsilon_1 \quad \text{and} \quad |g(\mathbf{x})| \leq \varepsilon_2.$$

Therefore

$$\sqrt{f(\mathbf{x})^2 + g(\mathbf{x})^2} \leq \sqrt{\varepsilon_1^2 + \varepsilon_2^2}.$$

Then Theorem 3.14 can be applied to bound the distance of \mathcal{C} and \mathcal{S} with the help of the constants G, K and

$$M = \sqrt{\varepsilon_1^2 + \varepsilon_2^2}.$$

□

Corollary 3.15 gives us an upper bound of the distance between two algebraic space curves:

$$\varrho = \sqrt{\frac{\varepsilon_1^2 + \varepsilon_2^2}{G^2 - K}}. \quad (3.33)$$

So the bounding fat region can be defined as the point set

$$\mathcal{F}(p, q, \varrho, \Omega) = \{\mathbf{x} : \exists \mathbf{x}_0 : p(\mathbf{x}_0) = 0, q(\mathbf{x}_0) = 0, |\mathbf{x} - \mathbf{x}_0| \leq \varrho\} \cap \Omega.$$

Evaluation of the Constants The defining polynomials of the algebraic curves f, g and p, q are given in the BB-tensor product form. In order to find the constants in Corollary 3.15, we use the convex hull property of these polynomials.

The BB-norm of a polynomial is the maximum of the absolute values of the coefficients. Suppose that the difference polynomials are represented in the form

$$f - p = \sum_{i=0}^l \sum_{j=0}^m \sum_{k=0}^n a_{ijk} B_i^l(x) B_j^m(y) B_k^n(z)$$

and

$$g - q = \sum_{i=0}^{\hat{l}} \sum_{j=0}^{\hat{m}} \sum_{k=0}^{\hat{n}} b_{ijk} B_i^{\hat{l}}(x) B_j^{\hat{m}}(y) B_k^{\hat{n}}(z).$$

Then the norms can be evaluated as

$$\varepsilon_1 = \|f - p\|_{\text{BB}}^{\Omega} = \max_{i,j,k} |a_{ijk}|$$

and

$$\varepsilon_2 = \|g - q\|_{\text{BB}}^{\Omega} = \max_{i,j,k} |b_{ijk}|.$$

The constant G is a lower bound of the gradient length of both functions f and g . We can represent the square of the gradient length in a BB-tensor product form

$$f_x^2 + f_y^2 + f_z^2 = \sum_{i=0}^{2l} \sum_{j=0}^{2m} \sum_{k=0}^{2n} c_{ijk}^f B_i^l(x) B_j^m(y) B_k^n(z)$$

$$g_x^2 + g_y^2 + g_z^2 = \sum_{i=0}^{2\hat{l}} \sum_{j=0}^{2\hat{m}} \sum_{k=0}^{2\hat{n}} c_{ijk}^g B_i^{\hat{l}}(x) B_j^{\hat{m}}(y) B_k^{\hat{n}}(z).$$

Then a suitable lower bound is

$$G = \sqrt{\min\{\min_{i,j,k} c_{ijk}^f, \min_{i,j,k} c_{ijk}^g\}},$$

if the minimum is positive.

The value of K can also be given with the help of the convex hull property of BB-polynomials. The representation of $\nabla f \cdot \nabla g$ can be computed as

$$f_x g_x + f_y g_y + f_z g_z = \sum_{i=0}^{l+\hat{l}} \sum_{j=0}^{m+\hat{m}} \sum_{k=0}^{n+\hat{n}} d_{ijk} B_i^{l+\hat{l}}(x) B_j^{m+\hat{m}}(y) B_k^{n+\hat{n}}(z).$$

Then a suitable upper bound is

$$K = \max_{i,j,k} |d_{ijk}|.$$

3.5 Convergence

Since we generate quadratic approximating curves, we expect that the fat arc generation algorithm has cubic convergence rate. We analyze in this section the convergence rate of the method and certify the third order convergence of the fat arcs in Lemma 3.21. Then we demonstrate the behavior of the bounding regions through some examples.

3.5.1 Continuity of Taylor Expansion Modification

The local fat arc generation technique first approximates the intersection curve of two algebraic surfaces defined by the polynomials f and g by a circular arc. This arc is defined as the intersection curve of two spheres. These spheres are given as the zero level set of the polynomials p and q . The polynomials are the quadratic Taylor expansion of certain polynomials with a special Hessian (see Section 3.3.2) about the center point \mathbf{c} of the sub-domain $\Omega \subseteq \Omega_0$. The polynomials with special Hessian are computed as the combination of the polynomials f and g in the form $h = kf + lg = \mathcal{G}(f, g, (u, v), \mathbf{c})$ for certain pair $(u, v) \neq (0, 0)$, where the terms k and l are linear polynomials.

In order to prove the convergence of the generated arcs, we have to show, that the computed polynomials depend continuously on the points of Ω_0 for a fixed choice of (u, v) . It means, that the polynomial $\mathcal{G}(f, g, (u, v), \mathbf{c})$ depends continuously on the choice of the point \mathbf{c} .

Lemma 3.16. *Given two polynomials f, g over the domain $\Omega \subseteq \Omega_0$. We suppose that for any point $\mathbf{c} \in \Omega_0$*

$$\|\nabla f(\mathbf{c}) \times \nabla g(\mathbf{c})\| \neq 0. \quad (3.34)$$

For an arbitrary but fixed pair of u and $v \in \mathbb{R} \setminus \{0\}$ we compute the polynomial

$$h = \mathcal{G}(f, g, (u, v), \mathbf{c})$$

with a special Hessian (see Lemma 3.6) under the condition (3.15). Then h depends continuously on the points of the domain Ω_0 .

Proof. We have to show that the computed linear factors k and l depend continuously on the point \mathbf{c} . We computed the coefficient vector $\mathbf{k} = (k_1, k_2, k_3, l_1, l_2, l_3)$, such that it satisfies the linear system $\mathbf{A}\mathbf{k} = \mathbf{b}$ in (3.14) and minimizes the l_2 -norm of the vector \mathbf{k} (see (3.15)). If (3.34) is true, then \mathbf{A} has full rank in any point $\mathbf{c} \in \Omega_0$ (see proof of Lemma 3.6). For a full rank matrix the vector, which satisfies (3.14) and (3.15), can be computed as

$$\mathbf{k} = \underbrace{\mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}}_{\mathbf{A}^\dagger} \mathbf{b}.$$

The matrix \mathbf{A}^\dagger is the so called Moore-Penrose generalized inverse of \mathbf{A} (see [9]). Since f and g are polynomials, the entries of the matrix \mathbf{A} and the vector \mathbf{b} depend continuously on the point \mathbf{c} . Therefore the vector \mathbf{k} also depends continuously on the point \mathbf{c} . The values of $u \neq 0$ and $v \neq 0$ are fixed real numbers. So all coefficients $u, v, k_i, i = 1 \dots 3$ and $l_i, i = 1 \dots 3$ depend continuously on \mathbf{c} . Therefore also $kf + lg$ depends continuously on the point \mathbf{c} . \square

The next corollary follows from Lemma 3.13 and Lemma 3.16. If we modify the Taylor expansion as it is described in Section 3.3.2, then we can establish the result considering the behavior of a sequence of the generated median circles.

Corollary 3.17. *Suppose we have a nested sequence of sub-domains $(\Omega_i)_{i=1,2,3\dots} \subset \Omega_0$*

$$\Omega_{i+1} \subset \Omega_i,$$

which have decreasing diameters δ_i , such that

$$\lim_{i \rightarrow \infty} \delta_i = 0,$$

and \mathbf{c}_i denotes the center point of Ω_i . Consider a pair of functions f and g , which defines an algebraic curve in $\Omega_0 \subset \mathbb{R}^3$

$$\mathcal{C}(f, g, \Omega_0) = \{\mathbf{x} : f(\mathbf{x}) = 0, g(\mathbf{x}) = 0\} \cap \Omega_0.$$

Suppose that there exists a point \mathbf{p} , which satisfies $f(\mathbf{p}) = g(\mathbf{p}) = 0$, not an inflection point of $\mathcal{C}(f, g, \Omega_0)$ and $\mathbf{p} \in \Omega_i$ for all i . We compute

$$\hat{f}_i = \mathcal{G}(f, g, (u, v), \mathbf{c}_i) \quad \text{and} \quad \hat{g}_i = \mathcal{G}(f, g, (u', v'), \mathbf{c}_i)$$

for fixed values of $u, v, u', v' \neq 0$. We consider the circles defined by the zero set of the quadratic Taylor expansions

$$p_i = T_{\mathbf{c}_i}^2 \hat{f}_i \quad \text{and} \quad q_i = T_{\mathbf{c}_i}^2 \hat{g}_i.$$

Then the sequence of these circles converges to a limit circle, which is the osculating circle of $\mathcal{C}(f, g, \Omega_0)$ in the point \mathbf{p} .

Corollary 3.18. *For all $\mathbf{c} \in \Omega_0$ if we compute $\hat{f} = \mathcal{G}(f, g, (u, v), \mathbf{c}) = kf + lg$, then the norm of the common coefficient vector \mathbf{k} can be bounded by a constant*

$$\|\mathbf{k}\| < L,$$

which depends only on f, g, Ω_0 and on the choice of (u, v) .

3.5.2 General Lower Bound for the Gradient Length

The following lemma (Lemma 3.19) ensures, that $\mathcal{G}(f, g, (u, v), \mathbf{c})$ has also a non-vanishing gradient if we compute fat arcs in sufficiently small sub-domains, which enclose the algebraic curve.

Lemma 3.19. *Suppose that there exists $G > 0$ in Ω_0 for the polynomials f and g such that*

$$\forall \mathbf{x} \in \Omega_0, \quad \|\nabla f(\mathbf{x})\| \geq G \quad \text{and} \quad \|\nabla g(\mathbf{x})\| \geq G. \quad (3.35)$$

Consider a domain $\Omega \subset \Omega_0$, which has a diameter $\delta_\Omega < \varepsilon$, and there is a point $\mathbf{p} \in \Omega$ such that for all $i = 1, \dots, n-1$,

$$f(\mathbf{p}) = g(\mathbf{p}) = 0.$$

The parameters $u \neq 0$ and $v \neq 0$ are arbitrary but fixed. We compute $h = \mathcal{G}(f, g, (u, v), \mathbf{c})$. If ε is sufficiently small, then there exists $\hat{G} > 0$ constant, such that for any $\mathbf{x} \in \Omega$

$$\|\nabla h(\mathbf{x})\| \geq \hat{G} > 0.$$

Proof. If $\mathbf{x} \in \Omega \subseteq \Omega_0$ then

$$\nabla h(\mathbf{x}) = f(\mathbf{x})\nabla k(\mathbf{x}) + k(\mathbf{x})\nabla f(\mathbf{x}) + g(\mathbf{x})\nabla l(\mathbf{x}) + l(\mathbf{x})\nabla g(\mathbf{x}),$$

where k and l are computed as described in Section 3.3.2. According to the triangle inequality

$$\begin{aligned} \|\nabla h(\mathbf{x})\| &\geq \|k(\mathbf{x})\nabla f(\mathbf{x}) + l(\mathbf{x})\nabla g(\mathbf{x})\| - \|f(\mathbf{x})\nabla k(\mathbf{x}) + g(\mathbf{x})\nabla l(\mathbf{x})\| \geq \\ &\geq \|k(\mathbf{x})\nabla f(\mathbf{x}) + l(\mathbf{x})\nabla g(\mathbf{x})\| - |f(\mathbf{x})| \|\nabla k(\mathbf{x})\| - |g(\mathbf{x})| \|\nabla l(\mathbf{x})\|. \end{aligned} \quad (3.36)$$

Since we know that there exists a point $\mathbf{p} \in \Omega$ such that $f(\mathbf{p}) = g(\mathbf{p}) = 0$, we obtain

$$|f(\mathbf{x})| \leq \frac{\varepsilon}{G} \quad \text{and} \quad |g(\mathbf{x})| \leq \frac{\varepsilon}{G}, \quad (3.37)$$

where ε is an upper bound of the diameter of Ω . In Corollary 3.18 we also observed, that there exists $L > 0$ such that

$$\|\mathbf{k}\| \leq L.$$

Since $\mathbf{k}^\top = (\nabla k^\top, \nabla l^\top)$,

$$\|\nabla k(\mathbf{x})\| \leq L \quad \text{and} \quad \|\nabla l(\mathbf{x})\| \leq L.$$

We can also bound the value of the linear polynomials k and l on a sufficiently small sub-domain Ω . Suppose that the diameter of Ω is smaller than ε . If $\mathbf{x} \in \Omega$, then

$$\begin{aligned} |k(\mathbf{x})| &= |u + k_1(x - c^x) + k_2(y - c^y) + k_3(z - c^z)| > |u| - \frac{\varepsilon}{2}\sqrt{3}L, \\ |l(\mathbf{x})| &= |v + l_1(x - c^x) + l_2(y - c^y) + l_3(z - c^z)| > |v| - \frac{\varepsilon}{2}\sqrt{3}L, \end{aligned} \quad (3.38)$$

where $\mathbf{c} = (c^x, c^y, c^z)$ denotes the center of Ω . Since u and v non-zero, if

$$\varepsilon < \frac{\min\{u, v\}}{\sqrt{3}L}, \quad (3.39)$$

then $|k(x)| \geq u/2$ and $|l(x)| \geq v/2$.

We supposed that $\nabla f(\mathbf{x})$ and $\nabla g(\mathbf{x})$ are linearly independent in any point of Ω_0 . If (3.39) is satisfied for an $\Omega \subseteq \Omega_0$, then there exists a general bound \tilde{G} depending on u, v and G , such that

$$\|k(\mathbf{x})\nabla f(\mathbf{x}) + l(\mathbf{x})\nabla g(\mathbf{x})\| \geq \tilde{G} > 0, \quad \forall \mathbf{x} \in \Omega.$$

Therefore for all $\mathbf{x} \in \Omega$

$$\begin{aligned} \|\nabla h(\mathbf{x})\| &\geq \tilde{G} - \|f(\mathbf{x})\nabla k(\mathbf{x})\| - \|g(\mathbf{x})\nabla l(\mathbf{x})\| \stackrel{(3.15)}{\geq} \\ &\geq \tilde{G} - L|f(\mathbf{x})| - L|g(\mathbf{x})|. \end{aligned}$$

Since we know that there exists a point $\mathbf{p} \in \Omega$ such that $f(\mathbf{p}) = g(\mathbf{p}) = 0$,

$$\|\nabla h(\mathbf{x})\| \geq \tilde{G} - \frac{2\varepsilon L}{G}.$$

Suppose that

$$\varepsilon = \min \left\{ \frac{\tilde{G}G}{2L}, \frac{u}{\sqrt{3}L}, \frac{v}{\sqrt{3}L} \right\}. \quad (3.40)$$

If the diameter of Ω , denoted by δ_Ω , satisfies

$$\delta_\Omega < \frac{\varepsilon}{2},$$

then

$$\|\nabla h(\mathbf{x})\| \geq \frac{\tilde{G}}{2} = \hat{G} > 0.$$

□

Corollary 3.20. *Suppose that the conditions of Lemma 3.19 are fulfilled for a certain pair of polynomials f and g on the domain Ω_0 . If $h = \mathcal{G}(f, g, (u, v), \mathbf{c})$ is computed in a sufficiently small sub-domain $\Omega \subset \Omega_0$ for an arbitrary pair of parameters $u, v \neq 0$, then $s = T_{\mathbf{c}}^2(h) \neq 0$.*

3.5.3 Convergence of Taylor Expansion Modification

Now we have to show that the fat arc thickness is sufficiently small compared with the diameter of the computational domain. The following lemma shows, how the computed fat arc thickness behaves as the size of the domain tends to zero.

Lemma 3.21. *Given two polynomials f, g defined over the domain $\Omega_0 = [\alpha_1, \beta_1] \times [\alpha_2, \beta_2] \times [\alpha_3, \beta_3] \subset \mathbb{R}^3$. We suppose that the conditions of Lemma 3.19 are satisfied. We compute a pair polynomials with special Hessian and apply the orthogonalization function (see (3.22))*

$$(\hat{f}, \hat{g}) = \mathcal{O}(\mathcal{G}(f, g, (u, v), \mathbf{c}), \mathcal{G}(f, g, (u', v'), \mathbf{c}), \mathbf{c})$$

in the center point \mathbf{c} of a sub-domain $\Omega \subset \Omega_0$. If the diameter δ_Ω of the sub-domain Ω is sufficiently small, then there exists a constant $C \in \mathbb{R}$ such that

$$\varrho \leq C\delta_\Omega^3, \quad (3.41)$$

where ϱ is the corresponding fat arc thickness computed as in (3.33).

Proof. Since the conditions of Lemma 3.19 are satisfied, we know that there exists \hat{G} such that

$$\min\{\|\nabla\hat{f}\|, \|\nabla\hat{g}\|\} \geq \hat{G},$$

for any sufficiently small sub-domain Ω , which encloses the curve. We denote by p and q the quadratic Taylor expansion of \hat{f} and \hat{g} about the center \mathbf{c} of the domain Ω . Then

$$\|\hat{f} - p\|_\infty = \|\hat{f} - T_{\mathbf{c}}^2(\hat{f})\|_\infty < \frac{1}{6} \underbrace{\max_{\mathbf{v} \in S^1, \mathbf{x} \in \Omega} \left| \frac{d^3\hat{f}}{d\mathbf{v}^3}(\mathbf{x}) \right|}_{*} \delta_\Omega^3$$

and

$$\|\hat{g} - q\|_\infty = \|\hat{g} - T_{\mathbf{c}}^2(\hat{g})\|_\infty < \frac{1}{6} \underbrace{\max_{\mathbf{v} \in S^1, \mathbf{x} \in \Omega} \left| \frac{d^3\hat{g}}{d\mathbf{v}^3}(\mathbf{x}) \right|}_{**} \delta_\Omega^3.$$

Recall from Lemma 3.16 that \mathcal{G} depends continuously on the points of the computational domain Ω_0 for each pair of (u, v) , where $u \neq 0$ and $v \neq 0$. Therefore also \hat{f} and \hat{g} depend continuously on the points of the computational domain Ω_0 . Thus for all \hat{f} a general upper bound C_1 can be given for (*) and for all \hat{g} a general upper bound C_2 can be given for (**).

The fat arc thickness is defined by

$$\varrho_\Omega = \sqrt{\frac{\varepsilon_1^2 + \varepsilon_2^2}{G_\Omega^2 - K_\Omega}},$$

where

$$\varepsilon_1 = \|\hat{f} - p\|_{\text{BB}} \quad \text{and} \quad \varepsilon_2 = \|\hat{g} - q\|_{\text{BB}}$$

Because of the norm equivalences there exist C_3 and C_4 , such that

$$\varepsilon_1 \leq C_3 \|\hat{f} - p\|_\infty \quad \text{and} \quad \varepsilon_2 \leq C_4 \|\hat{g} - q\|_\infty.$$

So we observe, that

$$\sqrt{\varepsilon_1^2 + \varepsilon_2^2} \leq \frac{1}{6} \underbrace{\sqrt{(C_1 C_3)^2 + (C_2 C_4)^2}}_M \delta_\Omega^3.$$

We assumed that $\hat{G} < G_\Omega$ is a general lower bound for $\|\nabla\hat{f}\|$ and $\|\nabla\hat{g}\|$ independent of the choice of the sub-domain Ω , if it is sufficiently small. Since we also applied the orthogonalization step to the polynomials \hat{f} and \hat{g} ,

$$\left| \nabla\hat{f}(\mathbf{c}) \cdot \nabla\hat{g}(\mathbf{c}) \right| = 0 \quad (3.42)$$

in the center point \mathbf{c} of a domain Ω . If the diameter of the sub-domain Ω is sufficiently small, then there exists $K > 0$, which does not depend on Ω and each $\mathbf{x} \in \Omega$ satisfies

$$\left| \nabla \hat{f}(\mathbf{x}) \cdot \nabla \hat{g}(\mathbf{x}) \right| \leq K.$$

According to (3.42) if the diameter of the sub-domain is sufficiently small, then the general bound K satisfies $K < \hat{G}^2$. Thus this implies, that

$$\varrho_\Omega \leq \frac{M \delta_\Omega^3}{\sqrt{\hat{G}_\Omega^2 - K_\Omega}} \leq \frac{M \delta_\Omega^3}{\sqrt{\hat{G}^2 - K}} \leq C \delta_\Omega^3.$$

□

3.5.4 Comparison of Convergence Rate

We confirm here the approximation order of the local fat arc generation algorithm (Algorithm 5) by numerical examples. We generate fat arcs for single algebraic space curve segments in different domains. We show, how the fat arc diameter behaves, if we reduce the size of the computational domain.

The domains are the axis aligned boxes in the global coordinate system:

$$\Omega_k = [-10^{-k}, 10^{-k}]^3, \quad k \in \mathbb{R}. \quad (3.43)$$

Fig.3.4 shows the result of the fat arc constructions with using the Taylor expansion modification in three different Ω_k for the pair of polynomials

$$\begin{aligned} f_1(x, y, z) &= x - yz \\ g_1(x, y, z) &= x^2 + y^2 + (z - 1)^2 - 1. \end{aligned}$$

In the first figure on the top is the implicitly defined curve shown in red. The other figures visualize the generated fat arcs for $k = 1, 1.5$, and 2 .

We consider two pairs of polynomials in the domains Ω_k , $1 \leq k \leq 8.25$:

$$\begin{aligned} f_2(x, y, z) &= x + z^2 - yz^2, \\ g_2(x, y, z) &= x^2 - 4y + y^2 - z + 0.5z^2. \end{aligned}$$

and

$$\begin{aligned} f_3(x, y, z) &= 0.04x - x^2 + x^3 + 0.44y - 0.4xy + x^2y - 1.4y^2 + \\ &\quad xy^2 + y^3 + 0.04z + x^2z - 0.4yz + y^2z - z^2 + xz^2 + yz^2 + z^3, \\ g_3(x, y, z) &= x - x^2 + xy + y^2 + yz + 0.25z^2. \end{aligned}$$

In Fig.3.5 we visualize the relation between the width of the generated fat arcs and the size of the domain diameter for the fat arc generation strategies with different median arc generation techniques. In the left the results for the polynomials f_2 and g_2 are shown for the values of $k = 1, 1.25 \dots 6$. In the right side the test polynomials were f_3 and g_3 , and the outputs are

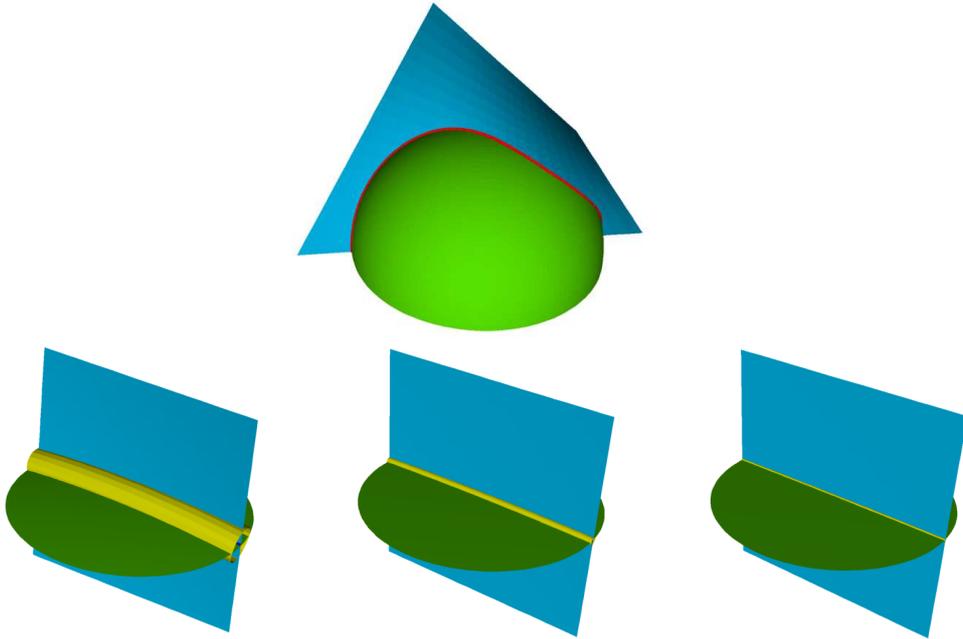


Figure 3.4: Above: The zero set of f_1 and g_1 with the intersection curve for $k = 0$. Below: Fat arcs for $k = 1, 1.5, 2$

computed for the values $k = 2, 2.25 \dots 7$. We show the negative logarithm of the associated fat arc diameter in a doubly-logarithmic plot. The expected approximation order is three. In the examples it is confirmed for the Taylor expansion modification. The lines denoted by L show the result of least-squares approximation with linear condition (see condition in (3.6)). The least-square approximations with quadratic normalization conditions (3.7) and (3.8), denoted in the figures by Q_1 and Q_2 . However, these least-squares approximation strategies gave nearly the same results. By the Taylor expansion modifications we used two different (u, v) parameter pairs. The line denoted by T_1 shows the result by the choice

$$(u, v) = (1, 2), \quad (u', v') = (2, 1),$$

and T_2 shows the output by the parameter pair

$$(u, v) = (1, 5), \quad (u', v') = (5, 1).$$

However, here the outputs for the two strategies show nearly the same results again.

3.6 Fat Arc Generation for Algebraic Space Curves

Subdivision is a frequently used technique and it is often combined with local approximation methods. Such hybrid algorithms subdivide the computational domain in order to separate regions where the topology of the curve can be described easily. The local curve approximation techniques can be applied in the sub-domains, where the topology of the curve has been successfully analyzed. The regions with unknown curve behavior can be made smaller and smaller with subdivision.

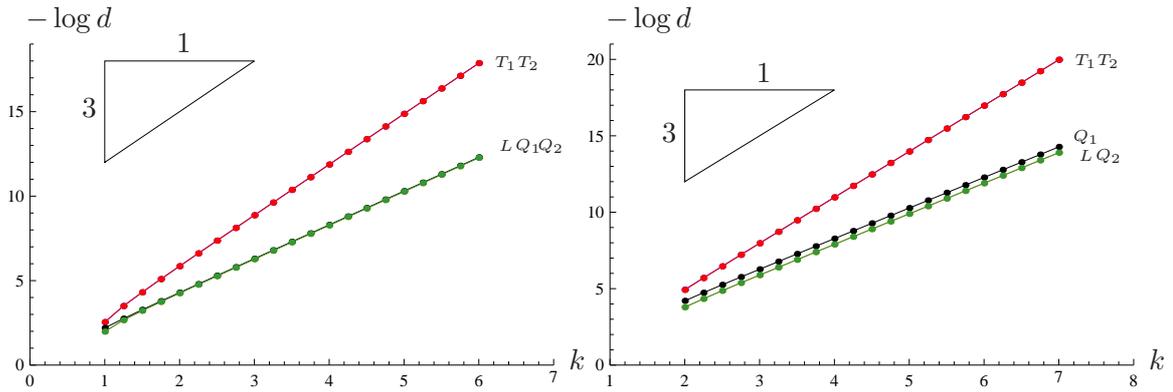


Figure 3.5: Comparison of relation between the fat arc diameter and the domain diameter for five median arc generation strategies. The negative logarithm of the associated fat arc diameter are shown in a doubly-logarithmic plot, in the left side for the polynomials f_2 and g_2 , in the right side for f_3 and g_3 . The red lines, denoted by T_1 and T_2 , show the results from the Taylor expansion modification by different parameter values. The lines denoted in the picture by Q_1 (black) and Q_2 (green) show the output of least-square approximation with quadratic normalization conditions ((3.7) and (3.8)). The result of least-square approximation with linear condition is denoted by L . The lines, denoted by T_1 and T_2 overlap each other as well as the ones denoted by L and Q_2

3.6.1 Global Algorithm

The algorithm `GenerateFatArcs` (see Algorithm 6) generate fat arcs for general algebraic space curves. It combines the fat arc generation for single curve segments with recursive subdivision.

First it analyzes the Bernstein–Bézier coefficients of the polynomials with respect to the current sub-domain. If no sign change is present for one or both of the polynomials, then the current domain does not contain any components of the algebraic curve. Otherwise the algorithm tries to apply the fat arc generation for the algebraic curve segment. If this is not successful, then the algorithm either subdivides the current domain into eight sub-domains, or returns the entire domain, if its diameter is already below the user-defined threshold ε .

Note that the algorithm may return domains which do not contain any segments of the implicitly defined curve (“false positive boxes”). However, it is guaranteed that the method returns a collection of bounding regions, which contains the implicitly defined curve.

3.6.2 Examples

Example 3.22. The implicitly defined curve is described by the equation system

$$\begin{aligned} y^2 + 2x - 1 &= 0, \\ z + x^2 - 0.4 &= 0. \end{aligned}$$

It is represented in the unit box $\Omega = [0, 1]^3$. The first figure of Fig.3.6 shows the generated median arcs, the second presents the generated fat arcs within the computational domains. The tolerance was set to $\varepsilon = 0.05$. The number of generated fat arcs is five. In the further examples we do not visualize the whole fat arcs, only the median arcs and bounding boxes,

Algorithm 6 $\text{GenerateFatArcs}(f, g, \Omega, \varepsilon)$

```

1: if Obs.3.2 true for  $f$  and  $g$  then
2:   return  $\emptyset$                                      {the sub-domain is empty}
3: end if
4: if the curve is regular in  $\Omega$  at least in one coordinate then
5:    $\mathcal{A} \leftarrow \text{FatArcLocal\_3d}(f, g, \Omega, \varepsilon)$    {single fat arc generation}
6:   if  $\mathcal{A} \neq \emptyset$  then
7:     return  $\mathcal{A}$                                        {... has been successful}
8:   end if
9: end if
10: if diameter of  $\Omega > \varepsilon$  then
11:   subdivide the domain into 8 sub-domains  $\Omega_1, \dots, \Omega_8$    {subdivision}
12:   return  $\bigcup_{i=1}^8 \text{GenerateFatArcs}(f, \Omega_i, \varepsilon)$    {recursive call}
13: end if
14: return  $\Omega$                                        {current domain is small enough}

```

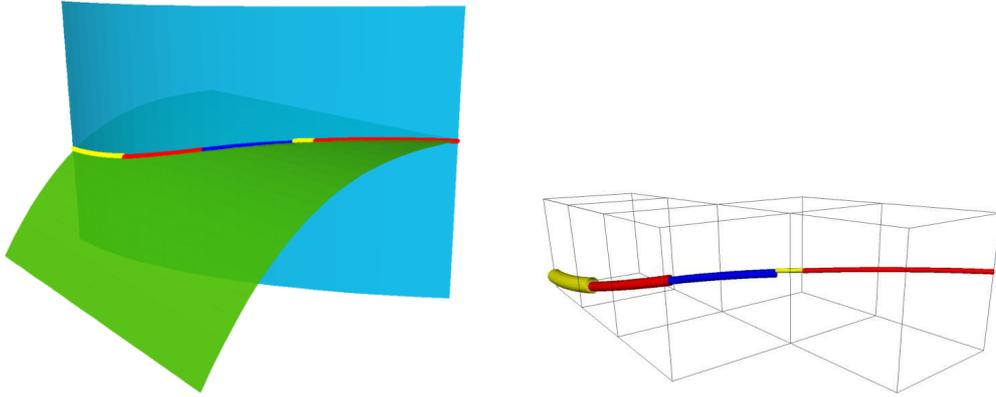


Figure 3.6: Median arcs and fat arcs for implicitly defined space curve.

since the fat arc shows only a slight difference compared to the thickened median arc by relatively small tolerance.

Example 3.23. In this example we approximate the intersection curve of quadric surfaces. We apply the algorithm GenerateArcs for three different intersections of four different pairs of quadric surfaces. The outputs are represented in Fig. 3.7. The numbers of used approximating primitives are given in Tab. 3.1 for each intersection curve. If the curve has a singular point (here in 1.(b), 2.(c), 3.(b) and 4.(c)), then the algorithm returns not only fat arcs but also sub-domains as abounding regions. All the examples are represented in the unit cube $[0, 1]^3$. The intersection curves are approximated within the tolerance $\varepsilon = 0.01$. In example 1.(b) we approximate an algebraic curve, which has a singular point (the tangent vector vanishes). This curve is the so called Viviani curve, which is defined by the intersection of a sphere and a cylinder, which is touching the sphere. Since the fat arc generation is not possible in a domain where the curve has singular point, the approximating algorithm uses not only fat arcs but also a few sub-domains to bound the space curve. In Fig.3.8 we show the result of

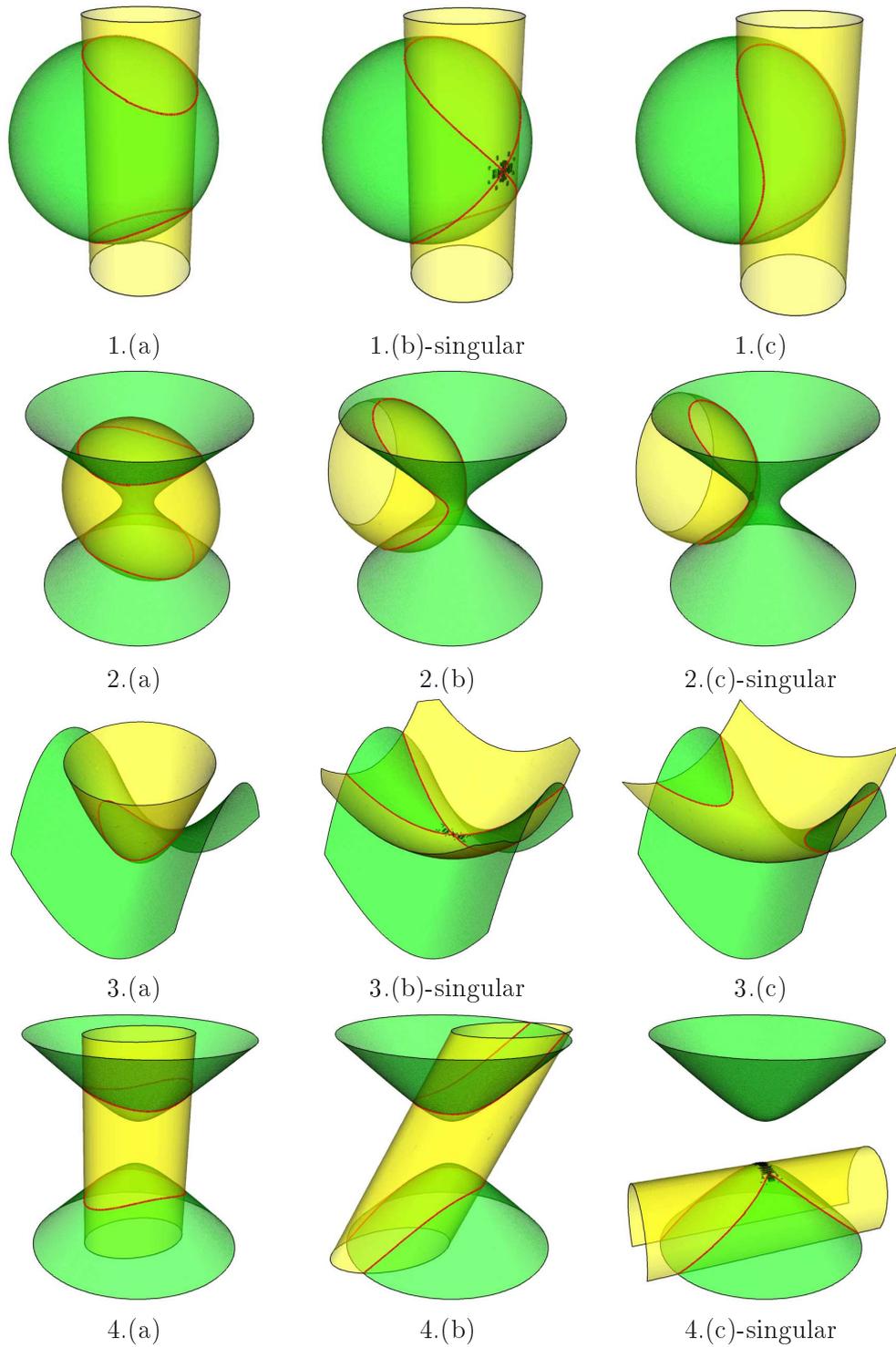


Figure 3.7: Approximation of the intersection curves of quadric surfaces.

Table 3.1: Approximating intersection curve of quadric surfaces. The number of used approximating primitives are given for the examples shown in Fig. 3.7.

Quadric Surfaces	Position (see Fig. 3.7)	Number of Arcs	Number of Boxes
1. sphere + cylinder	(a)	80	0
	(b)-singular	104	248
	(c)	52	0
2. ellipsoid + hyperboloid of one sheet	(a)	80	0
	(b)	76	0
	(c)-singular	96	76
3. rotational paraboloid + hyperbolic paraboloid	(a)	60	0
	(b)-singular	108	156
	(c)	50	0
4. hyperboloid of two sheets + elliptic cylinder	(a)	80	0
	(b)	80	0
	(c)-singular	88	612

the fat arc generation algorithm in comparison with bounding box generation for the curve by different tolerance bounds $\varepsilon = 0.1, 0.01$ and $\varepsilon = 0.001$. In the first row of the figure the output of the fat arc generation method is visualized. The median arcs of the generated fat arcs are shown in red and the boxes, which are used themselves as bounding primitives, are shown as gray cubes. In the second row the results of bounding box generation algorithm are shown (gray cubes) for the same tolerances. The numbers of used bounding primitives are shown in Table 3.2.

Example 3.24. In this example we approximate the isophotes of surfaces for different light directions. Isophotes are curves on a surface, where all points are exposed with equal light intensity from a given light source. An isophote of an implicitly defined surface $f = 0$ for a fixed direction vector \mathbf{d} and angle φ traces the point set

$$\mathcal{I}(f, \mathbf{d}, \varphi) = \{\mathbf{p} : f(\mathbf{p}) = 0, \langle \mathbf{d}, \nabla f(\mathbf{p}) \rangle = \cos(\varphi) \|\nabla f(\mathbf{p})\|\},$$

if we suppose that the direction vector is a unit vector. In order to describe an isophote for a given \mathbf{d} and φ , we used the algebraic equation system

$$\begin{aligned} f &= 0, \\ (f_x d^x + f_y d^y + f_z d^z)^2 - \cos^2 \varphi (f_x^2 + f_y^2 + f_z^2) &= 0, \end{aligned}$$

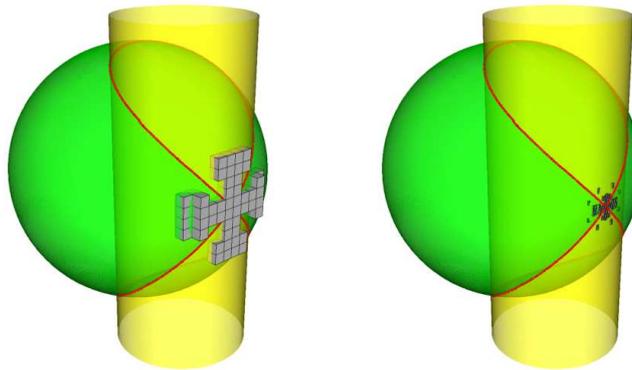
where $\mathbf{d} = (d^x, d^y, d^z)$. These two equations allocate the points of the isophotes, which belong to the direction \mathbf{d} and the angles φ and $(\pi - \varphi)$. We approximate some isophotes of three different implicitly defined surfaces

$$\begin{aligned} \mathcal{S}_1 &= \{(x, y, z) : xy - z + 0.5 = 0\}, \\ \mathcal{S}_2 &= \{(x, y, z) : x^2 + 2y^2 + 2z^2 - 1 = 0\}, \\ \mathcal{S}_3 &= \{(x, y, z) : x^3 + 0.5y^3 + z - 0.5 = 0\}, \end{aligned}$$

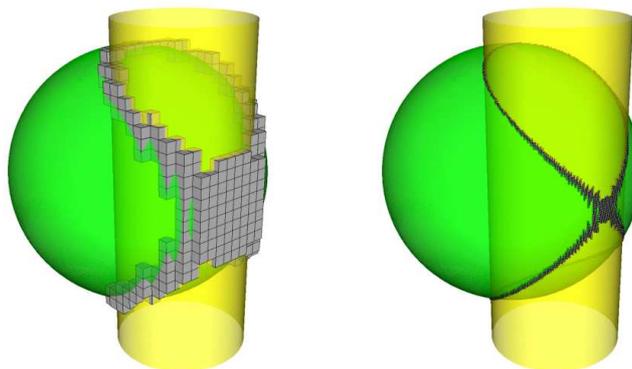
Table 3.2: Approximating intersection curve with singular point. The numbers of used approximating primitives are given for the examples shown in Fig. 3.8.

ε	Number of Primitives		Number of Boxes
	Num. of Arcs	Num. of Boxes	
0.1	68	56	284
0.01	104	248	2840
0.001	212	1592	26411

Fat arc generation



Bounding boxes



$\varepsilon = 0.1$

$\varepsilon = 0.01$

Figure 3.8: Fat arc generation for the Viviani curve. In the first row the outputs of the fat arc generation method are shown for three different tolerance bounds. In the second row the result of the bounding box generation is shown for the same tolerance bounds as in the first row.

Table 3.3: Number of used approximating primitives in the isophote approximations (see example in Fig.3.9).

\mathcal{S}_1	$(0, 0, -1)$		$(-1, 1, -4)$		$(-2, 0, -3)$	
	$\cos \varphi$	Num. of Arcs	$\cos \varphi$	Num. of Arcs	$\cos \varphi$	Num. of Arcs
	0.8	66	0.7	19	0.5	15
	0.85	44	0.8	25	0.65	18
	0.9	48	0.88	56	0.8	28
	0.95	32	0.95	54	0.9	22
	0.99	28	0.99	26	0.97	31
\mathcal{S}_2	$(0, -1, 0)$		$(-1, 0, -1)$		$(-1, -2, -1)$	
	$\cos \varphi$	Num. of Arcs	$\cos \varphi$	Num. of Arcs	$\cos \varphi$	Num. of Arcs
	0.4	112	0.2	152	0.4	152
	0.6	96	0.3	132	0.6	106
	0.8	128	0.6	104	0.7	82
	0.9	104	0.8	52	0.8	80
	0.99	80	0.9	80	0.9	80
\mathcal{S}_3	$(-1, -1, -1)$		$(-1, 1, 0)$		$(0, -1, -1)$	
	$\cos \varphi$	Num. of Arcs	$\cos \varphi$	Num. of Arcs	$\cos \varphi$	Num. of Arcs
	0.6	28	0.2	35	0.3	16
	0.7	32	0.4	52	0.4	32
	0.75	58	0.52	75	0.5	44
	0.8	107	0.7	80	0.7	70
	0.85	120	0.82	58	0.99	79

with the help of the fatarc generation algorithm. For all the surfaces we compute isophotes for three different light directions (see Fig. 3.9). In Tab. 3.3 we show the number of used approximating arcs for each isophote, along with the light directions and angles. We approximated the isophotes in the domain $[-1, 1]^3$ within the tolerance $\varepsilon = 0.05$.

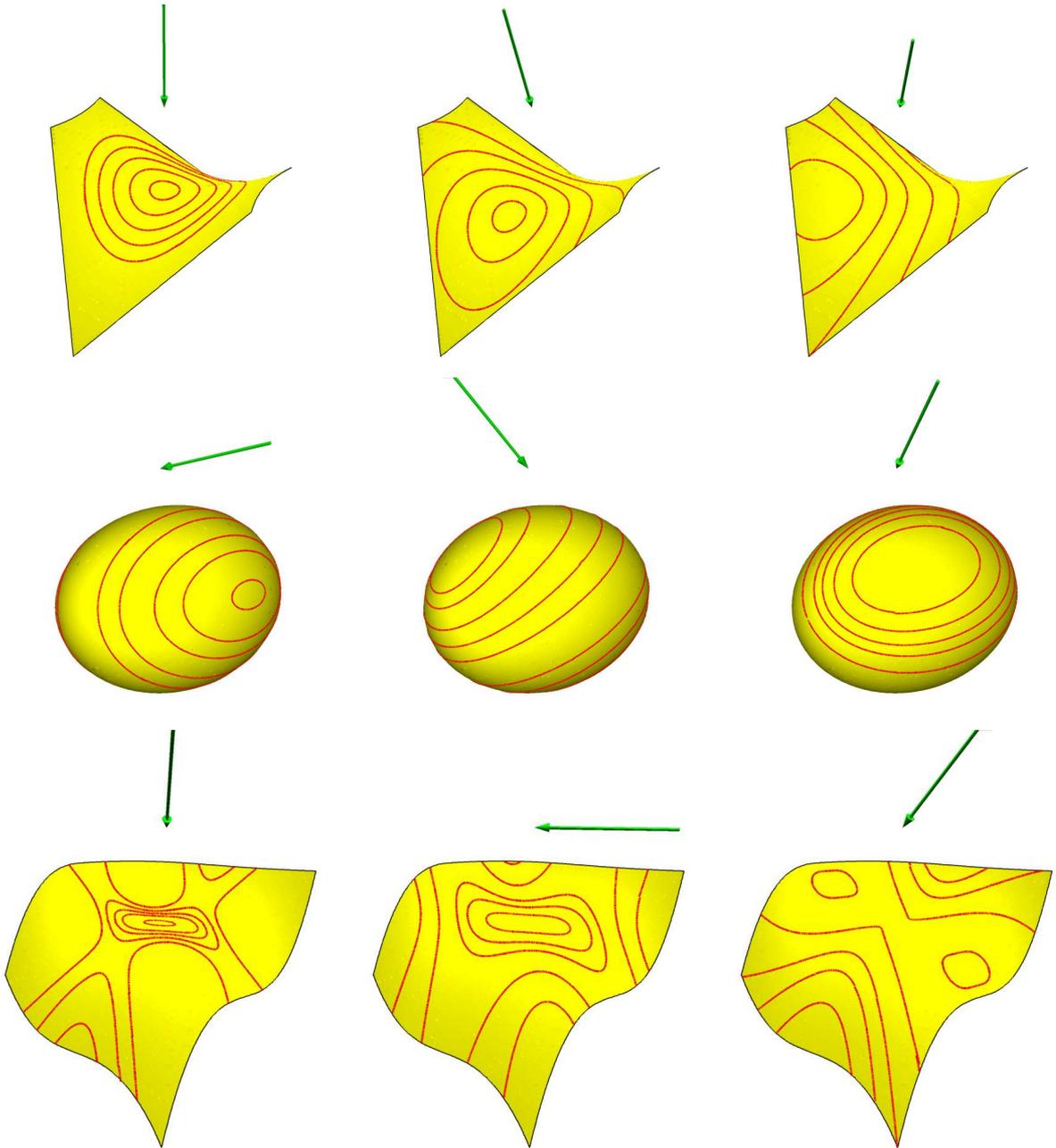


Figure 3.9: Approximation of isophotes for different light directions.

Chapter 4

Fat Arcs for Implicitly Defined Algebraic Curves

In this chapter we present an algorithm, which generalizes the fat arc generation method to bound one-dimensional algebraic sets. We consider algebraic systems consist of $n - 1$ linearly independent polynomials, which define one dimensional algebraic set (a set of curves) in \mathbb{R}^n .

4.1 Generalized Fat Arcs

We summarize in this section first the related results in approximation of one-dimensional algebraic sets. Then we introduce the definition of fat arcs in \mathbb{R}^n . In the end of the section we will state the approximation problem, such that one-dimensional algebraic sets are defined by polynomials represented in Bernstein–Bézier form.

4.1.1 Approximation of Algebraic Space Curves

Recently the interest for higher dimensional algebraic objects has been increased in research. The reason is the wide variety of applications, which appear in practice or in physics, for instance the description of physical fields with several free variables, movements or deformation of surfaces and volumes. Some of these problems are formulated with the help of multivariate polynomial systems. In particular we consider such systems, which have one dimensional set of solutions.

The first numerical approaches were formulated to approximate the solution set of univariate and bivariate polynomials. However, even in the univariate case these computations are very unstable for higher degree. In order to develop robust approximation algorithms a great leap forward was to use Bernstein–Bézier polynomials. The stability of this representation form allows to generalize the approximation algorithms for algebraic sets given in higher dimensional space. The first general numerical algorithms, which computed with polynomials given in BB-form, were developed by Sherbrooke and Patrikalakis [39]. These are subdivision methods for finding zero dimensional solution set of multivariate polynomial equations. A more sophisticated algorithm was presented by Elber and Kim in [12]. It uses multivariate Newton-Raphson method combined with subdivision, in order to reduce the number of subdivision steps during the computations. Moreover this method can be applied to under-determined systems, where the set of solutions has arbitrary many dimensions, although it

requires to compute some topological information about the solution set. The method of Elber and Kim has the additional advantage, that it can be extended to detect semi algebraic sets.

4.1.2 Definition of Generalized Fat Arcs

We would like to bound implicitly defined algebraic curves in \mathbb{R}^n with a set of regions. We generalize the fat arc construction, which we defined formerly in two- (Section 2.1.2) and three-dimensional space (Section 3.1.2). Generally a fat arc is a bounding region, which is a tubular neighborhood with a certain radius of an approximating arc.

Definition 4.1. *A fat arc is defined in \mathbb{R}^n by*

- *a segment of a circular arc (median arc) $\mathcal{S} \subset \Omega \subset \mathbb{R}^n$.*
- *and a distance $\varrho \in \mathbb{R}$.*

The fat arc is the point set

$$\mathcal{F}(\mathcal{S}, \varrho) = \{\mathbf{x} : \exists \mathbf{x}_0 \in \mathcal{S}, \|\mathbf{x} - \mathbf{x}_0\|_2 \leq \varrho\}.$$

As we saw in Section 2.1.2 in the two-dimensional case, the fat arc is a bounding region, which consists of a part of an annulus and two circular disks. In the three-dimensional case (see Section 3.1.2) it is bounded by a segment of a torus and two spherical caps. Generally we can say that in the n -dimensional space a fat arc is a thickened circular arc, which is bounded by a toroidal part and two spherical cups in the end.

The median arc can be represented as the zero set of spheres. This algebraic representation form is advantageous, since it simplifies the computation of the intersection of spheres.

4.1.3 Algebraic Space Curves

Since the visualization of higher dimensional space curves is difficult, and the number of required computational steps climbs fast with the raise of the dimension, the choice of the representation plays an important role. The most widely used representations of polynomials in geometric computing are the monomial, Lagrange, Hermite, B-Spline and Bézier forms. In order to construct fat arcs for algebraic curves, we shall use the properties of the defining polynomials. It is the most advantageous option, if the polynomials are given in the Bernstein-Bézier tensor product form. It provides the convex-hull property, the de Casteljau-algorithm, degree manipulation formulas etc. Therefore we suppose that the input polynomials are defined in the form

$$f(\mathbf{x}) = \sum_{\mathbf{k}=\mathbf{0}}^{\mathbf{l}} d_{\mathbf{k}} B_{\mathbf{k},\mathbf{l}}(\mathbf{x}), \tag{4.1}$$

with respect to an axis aligned domain

$$\Omega_0 = \times_{i=1}^n [\alpha_i, \beta_i] \subset \mathbb{R}^n.$$

The coordinates of the vector $\mathbf{l} = (l^i)_{i=1}^n$ denote the maximal degree of the basis polynomials in each variables x^i . The vector of indexes \mathbf{k} is

$$\mathbf{k} = (k^i)_{i=1}^n, \quad \text{such that } k^i \in \{0, \dots, l^i\}.$$

The coefficients are given as $d_{\mathbf{k}} \in \mathbb{R}$, and the functions are

$$B_{\mathbf{k},1}(\mathbf{x}) = \prod_{i=1}^n B_{k^i,l^i}(x^i),$$

where $B_{i,n}^j$ is defined as (2.2). For such functions $f : \mathbb{R}^n \rightarrow \mathbb{R}$, let us denote the zero level set with

$$\mathcal{Z}(f, \Omega_0) = \{\mathbf{x} : f(\mathbf{x}) = 0\} \cap \Omega_0.$$

An algebraic curve is given as the intersection of the zero sets of the polynomials $F = \{f_1, \dots, f_{n-1}\}$

$$\mathcal{C}(F, \Omega_0) = \bigcap_{i=1}^{n-1} \mathcal{Z}(f_i, \Omega_0) = \{\mathbf{x} : \forall i = 1, \dots, n-1, f_i(\mathbf{x}) = 0\} \cap \Omega_0 \quad (4.2)$$

with respect to the domain Ω_0 . Clearly, the curve may be the empty set, or it may consist of higher dimensional segments.

4.2 Approximation of Regular Curve Segments

In order to generate fat arcs for algebraic curves in \mathbb{R}^n , we present a local algorithm, which generates fat arcs only for regular segments of the curve. In this section first we describe the general definition of regular and coordinate-regular curve segments. Then we present a local algorithm to generate fat arcs, and we analyze the behavior of this algorithm. Later on we will combine this local bounding region generation with a subdivision technique.

4.2.1 Regularity Criterion

In order to bound an algebraic curve, we analyze the behavior of the curve in the computational domain. We identify first empty sub-domains of the computational domain as in Section 2.2.1 and Section 3.2.1. It is obvious, that Observation 3.2 is true in general for multivariate polynomials in BB-representation. Therefore we can apply it in general for detecting the domains without any segment of the algebraic curve.

In the two and three dimensional cases we used certain regularity criteria to find single segments of the algebraic curve. For the verification of such criteria we used the convex hull property of the Bernstein polynomials. Here we state similar conditions as in the case of two and three dimensional algebraic curves. Therefore we use the definitions:

Definition 4.2. A point $\mathbf{p} \in \Omega$ of an algebraic curve $\mathcal{C}(F, \Omega) \subset \mathbb{R}^n$ defined by the polynomial system $F = \{f_1, \dots, f_{n-1}\}$ is called regular, if the Jacobian-matrix

$$\mathbf{J}(F)(\mathbf{p}) = (\nabla f_1(\mathbf{p}), \dots, \nabla f_{n-1}(\mathbf{p}))$$

has full rank (and called singular otherwise). An algebraic curve segment is regular on $\Omega \subset \mathbb{R}^n$, if each point of the segment is regular in the domain.

Definition 4.3. Suppose that an algebraic curve is defined by the polynomial system $F = \{f_1, \dots, f_{n-1}\}$. In any point \mathbf{p} of the intersection curve $\mathcal{C}(F, \Omega) \subset \mathbb{R}^n$ we define the sub-matrices of the Jacobian $\mathbf{J}(F)(\mathbf{p})$ as

$$\mathbf{J}_i(F)(\mathbf{p}) = ((\nabla f_1(\mathbf{p})^j)_{j \neq i}, \dots, (\nabla f_{n-1}(\mathbf{p})^j)_{j \neq i}),$$

which are the $(n-1) \times (n-1)$ matrices, we get from $\mathbf{J}(F)(\mathbf{p})$ with deleting the i th row. Then a point \mathbf{p} of the algebraic curve $\mathcal{C}(F, \Omega) \subset \mathbb{R}^n$ is called i -regular for $i \in \{1, \dots, n\}$, if

$$\det(\mathbf{J}_i(F)(\mathbf{p})) \neq 0,$$

and called i -singular otherwise. An algebraic curve segment is i -regular in the domain $\Omega \subset \mathbb{R}^n$, if each point of the segment is i -regular in Ω .

Lemma 4.4. If there exists i , such that the algebraic curve segment is i -regular (regular at least in one coordinate) in the sub-domain $\Omega \subset \mathbb{R}^n$, then the curve segment is regular on Ω . Moreover it is not a loop.

Proof. If the algebraic curve is i -regular in a point \mathbf{p} , then

$$\det(\mathbf{J}_i(F)(\mathbf{p})) \neq 0.$$

Thus $\mathbf{J}_i(F)$, which is the sub-matrix of $J(F)$ has full rank: $n-1$. Therefore also the matrix $J(F)$ has at least rank $n-1$. Since $J(F) \in \mathbb{R}^{n \times (n-1)}$, it implies that $J(F)$ has full rank.

The tangent vector of the curve $\mathcal{C}(F, \Omega)$ in a point \mathbf{p} is the unit vector, which is perpendicular to the sub-space span by gradient vectors $\nabla f_i(\mathbf{p})$, $i = 1, \dots, n-1$. If the curve is i -regular on a sub-domain Ω , then the i th coordinate of the tangent vector is not zero in any point of Ω . Therefore the regularity in the i th coordinate excludes the situation, that the tangent vector returns to the same position if we trace the curve. So we cannot have loops in the domain. \square

Control of coordinate regularity. In order to identify domains with i -regular curve segments, we use the convex hull property of the defining polynomials. We give here a sufficient condition for detecting such domains. Namely it is sufficient to show, that there exists a positive lower bound for the absolute value of one of the determinants $\mathbf{J}_i(F)(\mathbf{x})$ in the domain Ω , which bounds the value of the determinant away from zero. Since we compute with BB-polynomials we can represent the coordinates of the gradient vector of the polynomial $f \in F$ in BB-tensor product form using the notation of Section 4.1.3

$$\nabla f^j(\mathbf{x}) = \sum_{\mathbf{k}}^{1-\mathbf{e}_j} (d_{\mathbf{k}} - d_{\mathbf{k}-\mathbf{e}_j}) B_{\mathbf{k}-\mathbf{e}_j, 1-\mathbf{e}_j}(\mathbf{x}),$$

where $d_{\mathbf{k}}$ are the coefficients of $f \in F$ and

$$\mathbf{e}_j = (0, \dots, \underbrace{1}_j, \dots, 0).$$

Thus the determinant of each $\mathbf{J}_i(F)$ matrix can be also given in a BB-tensor product form. If we denote by m_i the minimum and by M_i the maximum coefficient of $\det \mathbf{J}_i(F)$, then for $M_i m_i > 0$

$$|\det \mathbf{J}_i(F)| \geq \min\{|M_i|, |m_i|\} > 0.$$

If such m_i and M_i exist for an i , it implies that the curve is regular with respect to the i th coordinate in Ω .

However, one has to be careful to apply this regularity check, because it is computationally expensive. Suppose that each polynomial has the same degree bound $L = \max\{l_1, l_2, \dots, l_n\}$, then the cost of computation for one determinant is $\mathcal{O}(n^3 L^{n+1})$. In comparison, the cost of the de Casteljau algorithm for a polynomial in one coordinate direction is $\mathcal{O}(L^{n+1})$ such as a multiplication of two polynomials. To find empty domains has less computational cost. It is comparable with the size of our input polynomials, it needs $\mathcal{O}(L^n)$ steps for each polynomials, so to find an empty domain costs $\mathcal{O}(nL^n)$.

4.2.2 Local Algorithm

We present here an algorithm (Algorithm 7), which is the generalization of the local fat arc generation for two and three dimensional regular curve segments. It generates bounding regions in sub-domains, where the n -dimensional curve is regular with respect to at least one coordinate. Later on this local method will be combined with subdivision technique like the two and three-dimensional local fat arc generation methods (Algorithm 3 and 5).

In order to detect sub-domains with regular algebraic curve segments, we use the approach described in Section 4.2.1.

The fat arc generation is similar to the low dimensional cases described in Section 2.3.1 and Section 3.2.2. First we compute the median arc. Therefore we generalize the arc generation technique, which is computing polynomials with special Taylor expansions (see Section 2.3.3, Section 3.3.2). This method computes the median arc in algebraic form, as the zero set of special quadratic equations, which are simply equations of spheres.

The distance estimation method generalizes the approach from Section 4.4. It bounds the BB-distance in between each polynomial and the associated quadratic Taylor expansion. Then an upper bound is generated for the one-sided Hausdorff distance of the median arc and the algebraic space curve.

The algorithm is successful, if the median arc is found, the fat arc thickness can be computed, and it is smaller than the prescribed tolerance ε . Then the algorithm returns a fat arc, which bounds the curve segment in the appropriate sub-domain. If the local algorithm fails then it returns the empty set.

4.3 Median Arc Generation

The local algorithm generates first an approximating arc (median arc) for the implicitly defined curve. The median arc generation methods for two- and three-dimensional curves can be found in Section 2.2.3, 2.3.2, 2.3.3 and Section 3.3. In Section 2.2.3 we described a method, which generates parametric approximation of implicitly defined curves. All the other methods, we presented, generate approximating arc in implicit form. These methods can be generalized to higher dimensional curves. However, we generalize in this section only the strategy, which modifies the Taylor expansion of the polynomials.

4.3.1 Median Arc Generation Using Taylor Expansion

We suppose that the polynomial system $F = \{f_1, f_2, \dots, f_{n-1}\}$ describes a one dimensional algebraic set in \mathbb{R}^n . Let us denote the algebraic variety of the system in the sub-domain Ω

Algorithm 7 FatArcLocal_nd(F, Ω, ε)

Require: The curve is regular with respect to at least one coordinate in Ω .

```

1:  $\hat{f}_i$  modified polynomials
2:  $p_i \leftarrow T_{\mathbf{c}}^2(\hat{f}_i)$  spherical approximations
3: if  $p_i \neq 0$  then
4:    $\mathcal{P}_i \leftarrow$  zero contour of  $p_i$ 
5:    $\mathcal{S} \leftarrow \bigcap_{i=1}^{n-1} \mathcal{P}_i$  {median circle}
6:   if  $\mathcal{S} \neq \emptyset$  then
7:      $G \leftarrow \forall i \in \{1, \dots, n-1\}, 0 < G \leq \|\nabla \hat{f}_i\|$ 
8:      $K \leftarrow$  upper bound for  $|\nabla \hat{f}_i \cdot \nabla \hat{f}_j|$ 
9:     if  $0 < G$  and  $0 < G^2 - \binom{n-1}{2}K$  then
10:       $\varrho \leftarrow$  upper bound of  $\text{HD}_{\Omega}(\mathcal{S}, \mathcal{C}(\hat{f}_i, \Omega))$  {see Lemma 4.18}
11:      if  $\varrho \leq \varepsilon$  then
12:         $\mathcal{F} = \{\mathbf{x} : \exists \mathbf{y} \in \mathcal{S}, |\mathbf{x} - \mathbf{y}| \leq \varrho\} \cap \Omega$  {fat arc}
13:        return  $\mathcal{F}$  {fat arc has been found}
14:      end if
15:    end if
16:  end if
17: end if
18: return  $\emptyset$  {no fat arc has been found}
    
```

by

$$\mathcal{C}(F, \Omega) = \{\mathbf{x} : \forall f \in F, f(\mathbf{x}) = 0\} \cap \Omega.$$

We further assume that this curve segment is regular at least in one coordinate in the sub-domain Ω . In order to generalize the median arc generation method from Section 2.3.3 and Section 3.3.2, we reformulate the polynomial system F . We compute an algebraic system \hat{F} with $n-1$ new polynomials, such that for all $\hat{f} \in \hat{F}$

$$\forall \hat{f} \in \hat{F}, \quad \mathcal{C}(F, \Omega) \subset \mathcal{Z}(\hat{f}, \Omega) = \{\mathbf{x} : \hat{f}(\mathbf{x}) = 0\} \cap \Omega.$$

Moreover each new polynomial $\hat{f} \in \hat{F}$ has to satisfy the condition

$$\mathcal{H}(\hat{f})(\mathbf{c}) = \begin{pmatrix} \lambda & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda \end{pmatrix} = \lambda \mathbf{I}^{n \times n}, \quad \lambda \in \mathbb{R}, \quad (4.3)$$

where \mathbf{c} denotes the center of the sub-domain Ω . The quadratic Taylor expansion of \hat{f} about \mathbf{c}

$$p(\mathbf{x}) = T_{\mathbf{c}}^2(\hat{f}) = \hat{f}(\mathbf{c}) + \sum_{i=1}^n \frac{\partial \hat{f}(\mathbf{c})}{\partial x^i} (x^i - c^i) + \frac{\lambda}{2} \sum_{i=1}^n (x^i - c^i)^2$$

possesses a spherical zero contour $p(\mathbf{x}) = 0$.

Definition 4.5. A polynomial f is called a polynomial with special Hessian in the point \mathbf{c} , if the Hessian matrix of f in the point \mathbf{c} is equal to a matrix $\lambda \mathbf{I}^{n \times n}$, where $\lambda \in \mathbb{R}$.

A polynomial \hat{f} with special Hessian in the point \mathbf{c} satisfies the equations

$$\frac{\partial^2 \hat{f}(\mathbf{c})}{\partial x^i \partial x^i} - \frac{\partial^2 \hat{f}(\mathbf{c})}{\partial x^{i+1} \partial x^{i+1}} = 0, \quad i = 1, \dots, n-1$$

and

$$\frac{\partial^2 \hat{f}(\mathbf{c})}{\partial x^i \partial x^j} = 0, \quad 1 \leq i < j \leq n. \quad (4.4)$$

We compute the polynomial $\hat{f} \in \hat{F}$ as certain combination of the original polynomials $f_i \in F$. In order to provide sufficiently many degrees of freedom in the system (4.4), we consider polynomial multipliers in the combinations. On the other hand we restrict ourselves to linear multipliers. So each polynomial with special Hessian is computed as

$$\hat{f}(\mathbf{x}) = \sum_{j \in J} k_j(\mathbf{x}) f_j(\mathbf{x}), \quad (4.5)$$

where $J \subseteq \{1, \dots, n-1\}$. The linear multipliers are given for each $j \in J$ as

$$k_j(\mathbf{x}) = u_j + \sum_{i=1}^n k_j^i (x^i - c^i), \quad k_j^i, u_j \in \mathbb{R}, \quad (4.6)$$

where $\mathbf{c} = (c^i)_{i=1}^n$ denotes the center point of the sub-domain Ω . The coefficients of k_j can be computed by solving a linear system. In order to avoid that the system (4.4) becomes overdetermined, the number of unknowns in the multipliers has to be greater than equal as the number of equations.

In the former chapters for $n = 2, 3$ we used all polynomials f_i in the computation of the new polynomials. For $n = 2$ this was obvious, since we had only one polynomial, and we used a single linear polynomial as a multiplier, which has three coefficients. The system characterizing the special Hessian matrix has two equations. If we fix the constant term of the linear multiplier, we arrive at a non-homogeneous system, which has full rank under certain conditions (see Lemma 2.9). Thus the system has a unique solution.

In three dimensions, the system (4.4) has five equations. If we use the combination of both input polynomials and fix the constant terms of the linear multipliers, we have six free variables. So the system is under-determined. We observed, that our equation system in (3.11) has a full rank, if the condition of Lemma 3.6 for the gradients is satisfied. In order to compute the multipliers k_j uniquely, we chose from the solution space of the coefficients the shortest solution vector (3.15). This minimization problem always has a unique solution.

In the n -dimensional case a polynomial \hat{f} defined as in (4.5) can be found by solving a homogeneous system with $\frac{n(n+1)}{2} - 1$ equations. To avoid that the system (4.4) becomes overdetermined, the number of equations should not exceed the number of variables. In the n -dimensional space a linear polynomial has $n+1$ coefficients. In order to obtain non-trivial solution, we always choose the constant term u_j of the linear multipliers k_j as arbitrary but fixed parameter values for all j . Thus the equation system reduces to a non-homogeneous one, where each linear multiplier provides n free variables. By this assumption, if the trivial solution satisfies the system (4.4), it implies that a certain linear combination of the original polynomials also satisfy (4.3).

Observation 4.6. Let N_J denote the number of the elements of the index set $J \subseteq \{1, \dots, n\}$. In order to avoid that the system (4.4) is overdetermined for an arbitrary but fixed parameter vector $\mathbf{u} = (u_j)_{j \in J}$,

$$\frac{n(n+1)}{2} - 1 - n \cdot N_J \leq 0$$

should be satisfied. This implies that

$$\left\lceil \frac{n+1}{2} \right\rceil \leq N_J. \quad (4.7)$$

If a new polynomial \hat{f} is computed as

$$\hat{f}(\mathbf{x}) = \sum_{j \in J} k_j(\mathbf{x}) f_j(\mathbf{x}), \quad (4.8)$$

and \mathbf{c} is the center of the computational domain Ω , then

$$\mathcal{H}(\hat{f})(\mathbf{c}) = \sum_{j \in J} \nabla k_j(\mathbf{c}) \nabla f_j(\mathbf{c})^T + \nabla f_j(\mathbf{c}) \nabla k_j(\mathbf{c})^T + k_j(\mathbf{c}) \mathcal{H}(f_j)(\mathbf{c}).$$

Since we know that the value $k_j(\mathbf{c}) = u_j$ are arbitrary but fixed, the system (4.4) can be written as

$$\mathbf{A}_J \mathbf{k} = \mathbf{b}_J, \quad (4.9)$$

where \mathbf{k} consists of the coefficient vectors of the polynomials k_j for $j \in J$

$$\mathbf{k}^T = ((k_j^1, \dots, k_j^n)_{j \in J}^T).$$

The rows of the system are the equations derived from (4.4) by substituting (4.5) and (4.6). The vector $\mathbf{b}_J \in \mathbb{R}^{n(n+1)/2-1}$ consists of the coordinates

$$\left\{ \begin{array}{ll} -\sum_{j \in J} u_j \left(\frac{\partial^2 f_j(\mathbf{c})}{\partial x^i \partial x^i} - \frac{\partial^2 f_{j+1}(\mathbf{c})}{\partial x^{i+1} \partial x^{i+1}} \right), & \text{for all } 1 \leq i \leq n-1 \\ -\sum_{j \in J} u_j \frac{\partial^2 f_j(\mathbf{c})}{\partial x^k \partial x^l}, & \text{for all } 1 \leq k \leq n, k+1 \leq l \leq n \end{array} \right. \quad (4.10)$$

depending on the order of the equations in (4.4). All entries of the matrix \mathbf{A}_J are equal to zero or to

$$\pm \frac{\partial f_j(\mathbf{c})}{\partial x^i}, \text{ for certain } 1 \leq i \leq n-1, j \in J. \quad (4.11)$$

In the next section, in Section 4.3.2, we consider in details the structure and the solvability of such systems. In Lemma 4.10 we show, that a polynomial \hat{f} with special Hessian in the center point \mathbf{c} of a sub-domain Ω can always be computed, if the gradient vectors $\nabla f_i(\mathbf{c})$ are linearly independent and \hat{f} is the combination of all polynomials in F

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^{n-1} k_i(\mathbf{x}) f_i(\mathbf{x}), \quad (4.12)$$

Table 4.1: Construction of new polynomials with special Hessian.

dimension	num. of equ.	num. of coeff.	dim. of sol.sys.
n	$\frac{1}{2}n(n+1) - 1$	$n(n-1)$	$\frac{1}{2}(n-1)(n-2)$
2	2	2	0
3	5	6	1
4	9	12	3
5	14	20	6
6	20	30	10
100	5049	9900	4851
1000	500499	999000	498501

i.e. if $J = \{1, \dots, n-1\}$. Therefore we denote the linear system (4.9) for the index set $J = \{1, \dots, n-1\}$ by

$$\mathbf{A} := \mathbf{A}_J \quad \text{and} \quad \mathbf{b} := \mathbf{b}_J.$$

In order to find the coefficients of k_i we solve the non-homogeneous system

$$\mathbf{A}\mathbf{k} = \mathbf{b},$$

with an $(n(n-1)/2) \times (n(n-1))$ matrix which has full rank (see Lemma 4.10). We show in Table 4.1 the behavior of this linear system for different value of the dimension n . It is obvious, that the number of coefficients and the dimension of the solution system increases drastically if we increase the dimension n . However, in low dimensional cases, like $n = 2, 3$, the size of the linear system is still small.

The solution of system (4.4) for three- or higher dimensional problems has an at least one dimensional solution space. However, we need only one set of coefficients, which defines the multipliers k_i . Therefore we compute the solution vector \mathbf{k} , which has the smallest l_2 norm

$$\|\mathbf{k}\|_2 \rightarrow \min \text{ subject to } \mathbf{A}\mathbf{k} = \mathbf{b}. \quad (4.13)$$

Therefore the multipliers k_i obtained by the construction are unique for each parameter vector \mathbf{u} . So we can introduce function \mathcal{G} , whose value depends on the set of polynomials $F = \{f_1, \dots, f_{n-1}\}$, a value of \mathbf{u} and the center point \mathbf{c} of a domain Ω . The associated value of the function is given as the solution of the minimization problem (4.13)

$$\mathcal{G}(F, \mathbf{u}, \mathbf{c}) = \sum_{i=1}^{n-1} k_i f_i. \quad (4.14)$$

Remark 4.7. If the right hand side of the system (4.9), vector \mathbf{b} , vanishes for a certain parameter vector \mathbf{u} , then the solution set of (4.9) is a subspace of $\mathbb{R}^{n(n-1)}$. It implies that also the trivial solution is a solution of the system. Therefore the linear combination of $f_i \in F$ fulfills the condition (4.4). According to (4.13) we always choose the solution of the system (4.9) which has the smallest length. So in this special case all k_j are constants.

The polynomial $\hat{f} = \mathcal{G}(F, \mathbf{u}, \mathbf{c})$ is a polynomial with special Hessian in the point \mathbf{c} . Thus the quadratic Taylor expansion of \hat{f} about \mathbf{c} has a spherical zero level set. We compute

$n - 1$ polynomials $\hat{F} = \{\hat{f}_1, \dots, \hat{f}_{n-1}\}$ for different parameter vectors $\mathbf{u}_i, i = 1, \dots, n - 1$. The quadratic Taylor expansion of each polynomial $\hat{f}_i \in \hat{F}$ about \mathbf{c} , denoted by

$$p_i = T_{\mathbf{c}}^2 \hat{f}_i,$$

has a spherical zero level set. If $P = \{p_1, \dots, p_{n-1}\}$, then these quadratic polynomials define the algebraic set

$$\mathcal{S}(P, \Omega) = \{\mathbf{x} : \forall p_i \in P, p_i(\mathbf{x}) = 0\} \cap \Omega.$$

If this algebraic set is one dimensional, then it forms a circular arc. Later we show in Section 4.5.2, that asymptotically this arc exists. The arc can be used as an approximating circular arc of the curve $\mathcal{C}(F, \Omega)$. We estimate the error of this approximation by bounding the distance of the algebraic sets $\mathcal{C}(\hat{F}, \Omega)$ and $\mathcal{S}(P, \Omega)$.

4.3.2 Computing Polynomials with Special Hessian

In order to compute a polynomial with special Hessian, one has to solve a linear systems (4.9) which is derived from (4.4) with substituting (4.5) and (4.6). To describe the matrix of such linear system, we introduce the following operator $\mathcal{A} : \mathbb{R}^n \rightarrow \mathbb{R}^{(n(n+1)/2-1) \times n}$, such that

$$\mathcal{A}(\mathbf{v}) = \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{pmatrix},$$

where \mathbf{A}_1 is the $(n - 1) \times n$ dimensional matrix

$$\mathbf{A}_1 = \begin{pmatrix} v^1 & -v^2 & 0 & \dots & 0 \\ 0 & v^2 & -v^3 & 0 & \dots & 0 \\ 0 & & & \ddots & & 0 \\ 0 & \dots & & 0 & v^{n-1} & -v^n \end{pmatrix}$$

and the matrix \mathbf{A}_2 is $\frac{n(n-1)}{2} \times n$ dimensional

$$\mathbf{A}_2 = \begin{pmatrix} v^2 & v^1 & 0 & \dots & 0 \\ v^3 & 0 & v^1 & 0 & \dots & 0 \\ \vdots & & & \ddots & & \\ v^n & 0 & \dots & & 0 & v^1 \\ 0 & v^3 & v^2 & 0 & \dots & 0 \\ 0 & v^4 & 0 & v^2 & \dots & 0 \\ & \vdots & & \ddots & & \\ 0 & v^n & 0 & \dots & 0 & v^2 \\ & & & \ddots & & \\ 0 & \ddots & & 0 & v^n & v^{n-1} \end{pmatrix}.$$

According to the definition of \mathcal{A} the sub-matrices of \mathbf{A}_J can be given as the concatenation of the matrices

$$\mathbf{A}_J = (\mathcal{A}(\nabla f_j(\mathbf{c})))_{j \in J}.$$

Lemma 4.8. *If $\mathbf{v} \in \mathbb{R}^n$ is not the null-vector, then*

$$\text{rank}(\mathcal{A}(\mathbf{v})) = n.$$

Proof. The vector $\mathbf{v} = (v^i)_{i=1}^n$ is not the null-vector, thus there exists a coordinate $v^j \neq 0$. According to the definition of the operator \mathcal{A} the following $n \times n$ sub-matrix can be selected from $\mathcal{A}(\mathbf{v})$ for $j < n$

$$\mathbf{S}_j = \begin{pmatrix} v^j & 0 & \dots & v^1 & & \dots & 0 \\ 0 & v^j & & v^2 & & & \\ & & \ddots & \vdots & & & \\ & & & v^j & -v^{j+1} & & \\ & & & v^{j+1} & v^j & & \\ & & & \vdots & & \ddots & 0 \\ 0 & \dots & & v^{n-1} & & 0 & v^j \end{pmatrix}.$$

If $j = n$, then

$$\mathbf{S}_n = \begin{pmatrix} v^n & 0 & \dots & v^1 \\ 0 & v^n & & v^2 \\ & & \ddots & \vdots \\ & & & v^n & v^{n-1} \\ & & & v^{n-1} & -v^n \end{pmatrix}.$$

Therefore we observe, that

$$\det(\mathbf{S}_j) = \begin{cases} (v^j)^{n-2} ((v^j)^2 + (v^{j+1})^2), & \text{if } j < n \\ (v^n)^{n-2} (-(v^{n-1})^2 - (v^n)^2), & \text{if } j = n \end{cases}$$

Since we supposed that v^j is non-zero

$$\det(\mathbf{S}_j) \neq 0.$$

Thus $\mathcal{A}(\mathbf{v})$ always has a non-singular $n \times n$ sub-matrix. Since $\mathcal{A}(\mathbf{v})$ is a matrix with n columns we arrive at

$$\text{rank}(\mathcal{A}(\mathbf{v})) = n.$$

□

Lemma 4.8 guarantees, that $\mathcal{A}(\mathbf{v})$ is a matrix with full rank if $\mathbf{v} \neq \mathbf{0}$.

Remark 4.9. Each coordinate of the vector $\mathcal{A}(\mathbf{v})\mathbf{u}$ can be given as

$$v^i u^i - v^{i+1} u^{i+1} \text{ for certain } 1 \leq i \leq n-1$$

or as

$$v^k u^l + v^l u^k \text{ for certain } 1 \leq k \leq n-1, k+1 \leq l \leq n.$$

Since these coordinates are symmetric in \mathbf{v} and \mathbf{u} it implies that

$$\mathcal{A}(\mathbf{v})\mathbf{u} = \mathcal{A}(\mathbf{u})\mathbf{v}.$$

Therefore we can establish that, if we multiply it by the vector $\mathbf{u} \in \mathbb{R}^n$

$$\mathcal{A}(\mathbf{v})\mathbf{u} = \mathbf{0} \quad \text{iff} \quad \mathbf{v} = \mathbf{0} \text{ or } \mathbf{u} = \mathbf{0}.$$

Now we define an operator \mathcal{B} , which combines the image matrices of the operator \mathcal{A} for a certain set of vectors. If $\mathbf{v}_i \in \mathbb{R}^n$ for $i = 1, \dots, k$, then

$$\mathcal{B} : \mathbb{R}^{n \times k} \rightarrow \mathbb{R}^{(n(n+1)/2-1) \times nk}, \quad \mathcal{B}(\mathbf{v}_1, \dots, \mathbf{v}_k) = (\mathcal{A}(\mathbf{v}_1) | \mathcal{A}(\mathbf{v}_2) | \dots | \mathcal{A}(\mathbf{v}_k)).$$

The definition of \mathcal{A} and \mathcal{B} implies, that the matrix of the system (4.9) can be given as

$$\mathbf{A}_J = (\mathcal{A}(\nabla f_j(\mathbf{c})))_{j \in J} = \mathcal{B}((\nabla f_j(\mathbf{c}))_{j \in J}).$$

Theorem 4.10. *Suppose that the vectors $\mathbf{v}_i \in \mathbb{R}^n$ for $i = 1, \dots, k$ are linearly independent. Then*

$$\dim(\text{Ker}(\mathcal{B}(\mathbf{v}_1, \dots, \mathbf{v}_k))) \geq \binom{k}{2}.$$

Proof. A vector $\mathbf{u} \in \mathbb{R}^{nk}$ is the element of the kernel of the matrix $\mathcal{B}(\mathbf{v}_1, \dots, \mathbf{v}_k)$ if

$$\mathcal{B}(\mathbf{v}_1, \dots, \mathbf{v}_k)\mathbf{u} = \mathbf{0}.$$

We consider the set of vectors N

$$N = \left\{ \underbrace{(0, \dots, 0)}_{n(j-1)}, \mathbf{v}_i^T, \underbrace{(0, \dots, 0)}_{n(i-j-1)}, -\mathbf{v}_j^T, \underbrace{(0, \dots, 0)}_{n(j-i)} \right\}^T : 1 \leq j < i \leq k.$$

These vectors are linearly independent, since any linear combinations of them forms a vector, which is a certain linear combination of the vectors \mathbf{v}_i in between the $(j-1)n+1$ -th and jn -th coordinates. Moreover according to Remark 4.9 for any $\mathbf{u} \in N$

$$\mathcal{B}(\mathbf{v}_1, \dots, \mathbf{v}_k)\mathbf{u} = \mathcal{A}(\mathbf{v}_i)(\mathbf{v}_j) + \mathcal{A}(\mathbf{v}_j)(-\mathbf{v}_i) = \mathcal{A}(\mathbf{v}_i)(\mathbf{v}_j) - \mathcal{A}(\mathbf{v}_j)(\mathbf{v}_i) = \mathbf{0}.$$

Since the set of vectors N consists of $\binom{k}{2}$ elements, it implies that

$$\dim(\text{Ker}(\mathcal{B}(\mathbf{v}_1, \dots, \mathbf{v}_k))) \geq \binom{k}{2}.$$

□

Corollary 4.11. *The matrix of the system (4.9) for an index set J is as*

$$\mathbf{A}_J = \mathcal{B}((\nabla f_j(\mathbf{c}))_{j \in J}).$$

If $\nabla f_j(\mathbf{c})$ are linearly independent for $j \in J$, then the rank of \mathbf{A}_J can be bounded by

$$\text{rank}(\mathbf{A}_J) \leq n \cdot N_J - \binom{N_J}{2},$$

where N_J the size of the index set J .

Observation 4.12. Given a set of $(n - 1)$ polynomials $F = \{f_i\}_{i=1}^{n-1}$ in a certain domain Ω . We choose an index set $J \subseteq \{1, \dots, n - 1\}$ which specifies the set of polynomials $\{f_j\}_{j \in J}$. Suppose that the gradient vectors $\nabla f_j(\mathbf{c})$ are linearly independent for $j \in J$ in the center point \mathbf{c} of Ω . In order to compute a polynomial with special Hessian in the point \mathbf{c} , we need to solve the linear system (4.9). If $J \subset \{1, \dots, n - 1\}$, then the rank of the system matrix is smaller than the number of rows (the matrix does not have full rank).

We compute a new polynomial with special Hessian by solving the linear system (4.9). The right-hand side of the system \mathbf{b}_J is computed as the combination of the parameter values $\mathbf{u} = (u_j)_{j \in J}$ and the second derivatives of the polynomials evaluated in the center of the computational domain (see (4.10)). If the system matrix \mathbf{A}_J has full rank and there are more variables than equations, it guarantees that for any choice of $\mathbf{u} = (u_j)_{j \in J}$ the linear system has a solution. According to Observation 4.12 \mathbf{A}_J has full rank if the new polynomial \hat{f} is computed for the index set $J = \{1, \dots, n - 1\}$, i.e. as the combination of all polynomials in F .

Note that Corollary 4.11 only implies for the index set $J = \{1, \dots, n - 1\}$, that

$$\text{rank}(\mathbf{A}_J) \leq \frac{n(n+1)}{2} - 1.$$

Therefore we still have to prove, that for $J = \{1, \dots, n - 1\}$

$$\text{rank}(\mathbf{A}_J) \geq \frac{n(n+1)}{2} - 1.$$

Lemma 4.13. Given a set of linearly independent vectors $\{\mathbf{v}_i\}_{i=1}^{n-1} \in \mathbb{R}^n$, then

$$\text{rank}(\mathcal{B}(\mathbf{v}_1, \dots, \mathbf{v}_{n-1})) = \frac{n(n+1)}{2} - 1.$$

Proof. We describe here how to prove that the rows of the matrix $\mathcal{B}(\mathbf{v}_1, \dots, \mathbf{v}_{n-1})$ are linearly independent for $n = 3$. Analogously the same can be proven for higher dimensional cases.

If $n = 3$ the matrix $\mathcal{B}(\mathbf{v}_1, \mathbf{v}_2)$ is defined by the vectors $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^3$, which are both not the zero vectors. The matrix is

$$\mathcal{B}(\mathbf{v}_1, \mathbf{v}_2) = (\mathcal{A}(\mathbf{v}_1) | \mathcal{A}(\mathbf{v}_2)) = \begin{pmatrix} v_1^1 & -v_1^2 & 0 & v_2^1 & -v_2^2 & 0 \\ 0 & v_1^2 & -v_1^3 & 0 & v_2^2 & -v_2^3 \\ v_1^2 & v_1^1 & 0 & v_2^2 & v_2^1 & 0 \\ v_1^3 & 0 & v_1^1 & v_2^3 & 0 & v_2^1 \\ 0 & v_1^3 & v_1^2 & 0 & 0 & v_2^2 \end{pmatrix},$$

where $\mathbf{v}_i^T = (v_i^1, v_i^2, v_i^3)$. Since $\mathbf{v}_1, \mathbf{v}_2 \in \mathbb{R}^3$ are linearly independent there exists only one unit vector $\mathbf{m} \in \mathbb{R}^3$, which is perpendicular to both vectors \mathbf{v}_1 and \mathbf{v}_2 . We assume, that there is a vector $\mathbf{u}^T = (u^1, u^2, u^3, u^4, u^5) \neq \mathbf{0}$, which satisfies

$$\mathbf{u}^T \mathcal{B}(\mathbf{v}_1, \mathbf{v}_2) = \mathbf{0},$$

i.e. the rows of the matrix are linearly dependent. Let us denote by \mathbf{B} the matrix, which we derive from $\mathcal{B}(\mathbf{v}_1, \mathbf{v}_2)$ by changing the order of the columns

$$\mathbf{B} = \begin{pmatrix} v_1^1 & v_2^1 & -v_1^2 & -v_2^2 & 0 & 0 \\ 0 & 0 & v_1^2 & v_2^2 & -v_1^3 & -v_2^3 \\ v_1^2 & v_2^2 & v_1^1 & v_2^1 & 0 & 0 \\ v_1^3 & v_2^3 & 0 & 0 & v_1^1 & v_2^1 \\ 0 & 0 & v_1^3 & v_2^3 & v_1^2 & v_2^2 \end{pmatrix}.$$

If the first two coordinates of $\mathbf{u}^T \mathbf{B}$ are zero, then the equations

$$\begin{aligned} u^1 &= c_1 m^1, \\ u^3 &= c_1 m^2, \\ u^4 &= c_1 m^3, \end{aligned} \tag{4.15}$$

have to be satisfied for a certain $c_1 \in \mathbb{R}$, where $\mathbf{m} = (m^1, m^2, m^3)$. If the third and the fourth coordinates of $\mathbf{u}^T \mathbf{B}$ are zero then

$$\begin{aligned} u^2 - u^1 &= c_2 m^2, \\ u^3 &= c_2 m^1, \\ u^5 &= c_2 m^3, \end{aligned} \tag{4.16}$$

and if the last two coordinates are zero, then

$$\begin{aligned} -u^2 &= c_3 m^3, \\ u^4 &= c_3 m^1, \\ u^5 &= c_3 m^2, \end{aligned} \tag{4.17}$$

have to be satisfied for certain $c_2, c_3 \in \mathbb{R}$. The second and the third equation of each system (4.15), (4.16) and (4.17) imply for all $i, j \in \{1, 2, 3\}$ that

$$c_j m^i = c_i m^j.$$

If we add the first three equations from (4.15), (4.16) and (4.17), then we obtain that

$$c_1 m^1 + c_2 m^2 + c_3 m^3 = 0. \tag{4.18}$$

Suppose that $c_i \neq 0$, then we substitute each m^j for $(c_j/c_i)m^i$ where $i \neq j$ in (4.18), which results that

$$m_i(c_1^2 + c_2^2 + c_3^2) = 0.$$

Therefore m_i has to be zero. This implies, that for all $i = 1, 2, 3$ either c_i is zero or m_i is zero. Therefore we can derive from (4.15), (4.16) and (4.17), that all coordinates of the vector \mathbf{u} are zero. It is in contradiction with our assumption, that $\mathbf{u} \neq \mathbf{0}$. Thus the rows of \mathbf{B} and also the rows of $\mathcal{B}(\mathbf{v}_1, \mathbf{v}_2)$ are linearly independent. Since the number of rows in $\mathcal{B}(\mathbf{v}_1, \mathbf{v}_2)$ is less than the number of columns, the rank of the matrix is equal to the number of rows

$$\text{rank}(\mathcal{B}(\mathbf{v}_1, \mathbf{v}_2)) = \frac{3(3+1)}{2} - 1 = 5.$$

□

This lemma implies, that the matrix of the linear system (4.9)

$$\mathbf{A}_J = \mathcal{B}((\nabla f_j(\mathbf{c}))_{j \in J})$$

for $J = \{1, \dots, n-1\}$ has the rank

$$\text{rank}(\mathbf{A}_J) = \frac{n(n+1)}{2} - 1,$$

if we suppose that the gradient vectors of the original polynomial system F are linearly independent in the center of the computational domain.

4.3.3 Orthogonalization of the Polynomial System

In order to get efficient error bound for the algebraic curves $\mathcal{C}(\hat{F}, \Omega)$ and $\mathcal{S}(P, \Omega)$, we re-define the set of polynomials \hat{F} . We use the linear combinations of them to keep the special form of each polynomials.

Remark 4.14. If each $\hat{f}_i \in \hat{F}$ fulfills the property of functions with special Hessians (4.4), then any linear combination of them

$$\hat{h} = \sum_{\hat{f}_i \in \hat{F}} c_i \hat{f}_i, \quad c_i \in \mathbb{R},$$

also fulfills the condition of special Hessian (see (4.4)). Thus $T_{\mathbf{c}}^2(\hat{h})(\mathbf{x}) = 0$ defines a sphere in \mathbb{R}^n .

Therefore a new set of polynomials can be generated F^* , such that

$$\forall i, j = 1, \dots, n-1, i \neq j, \quad \nabla f_i^*(\mathbf{c}) \perp \nabla f_j^*(\mathbf{c}). \quad (4.19)$$

in the center of the domain. This new system can be deduced with the help of the Gram-Schmidt orthogonalization of the vectors $\nabla \hat{f}_i(\mathbf{c})$. If the vectors $\nabla \hat{f}_i(\mathbf{c})$ are linearly independent, then we can compute $n-1$ vectors \mathbf{v}_i , which are pairwise orthogonal, and each new vector is the linear combination of $\nabla \hat{f}_i(\mathbf{c})$

$$\mathbf{v}_i = \sum_{i=1}^{n-1} \mu_i \nabla \hat{f}_i(\mathbf{c}), \quad \mu_i \in \mathbb{R}.$$

If we compute the linear combination of the polynomials \hat{f}_i with the same coefficients

$$f_i^* = \sum_{i=1}^{n-1} \mu_i \hat{f}_i,$$

then they also fulfill the condition of special Hessians according to Remark 4.14. Moreover (4.19) is satisfied for each pair of polynomials. Thus we introduce the function \mathcal{O} , which assigns to the polynomials \hat{f}_i and the center point \mathbf{c} of a domain Ω . It generates a set of polynomials

$$F^* = \mathcal{O}(\hat{F}, \mathbf{c}) = \{f_i^* : i = 1, \dots, n-1\} \quad (4.20)$$

as it is described above.

4.3.4 Connection with the Osculating Circle

Now we consider the case, where the center of the computational domain Ω is a point of the algebraic curve \mathcal{C} defined by the polynomials f_i , $i = 1, \dots, n-1$. If the center point is denoted by \mathbf{c} , then

$$\forall i \in \{1, \dots, n-1\}, \quad f_i(\mathbf{c}) = 0. \quad (4.21)$$

As we saw it in the three dimensional case, this special case plays an important role during the computations.

A new set of polynomials \hat{F} is computed as the combination of f_i using a the strategy from Section 4.3.1. The quadratic approximation of each polynomial is

$$s_i = T_{\mathbf{c}}^2(\hat{f}_i).$$

If (4.21) is satisfied, then the quadratic approximating polynomial s_i has the following form

$$s_i(\mathbf{x}) = \nabla \hat{f}_i(\mathbf{c})^T (\mathbf{x} - \mathbf{c}) + \lambda_i (\mathbf{x} - \mathbf{c})^T (\mathbf{x} - \mathbf{c}), \quad (4.22)$$

where

$$\mathcal{H}(\hat{f}_i)(\mathbf{c}) = \lambda_i \mathbf{I}^n,$$

like in (4.3). So we can represent the zero set of each s_i in the form

$$\left\langle \mathbf{x} - \left(\mathbf{c} + \frac{1}{\lambda} \nabla \hat{f}_i(\mathbf{c}) \right), \mathbf{x} - \left(\mathbf{c} + \frac{1}{\lambda} \nabla \hat{f}_i(\mathbf{c}) \right) \right\rangle = \frac{\|\nabla \hat{f}_i(\mathbf{c})\|^2}{\lambda_i^2},$$

as we already observed in the three-dimensional case. The radius of the sphere $s_i = 0$ has the length

$$r = \frac{\|\nabla h(\mathbf{c})\|}{\lambda}.$$

Lemma 4.15. *Suppose we have a system of polynomials $F = \{f_i, i = 1, \dots, n - 1\}$ and $J = \{1, \dots, n - 1\}$. The polynomial system defines an algebraic curve in the domain Ω*

$$\mathcal{C}(F, \Omega) = \{\mathbf{x}, \forall i f_i(\mathbf{x}) = 0\} \cap \Omega.$$

For any polynomial \hat{f} is computed as

$$\hat{f} = \mathcal{G}(\mathcal{F}_J, \mathbf{u}, \mathbf{c}).$$

The sphere defined as the zero set of the polynomial

$$s = T_{\mathbf{c}}^2(\hat{f})$$

has at least second order contact with the algebraic curve \mathcal{C} . Moreover the intersection curve of the affine subspace defined by the tangent and the normal direction of the curve (the osculating plane) in the point \mathbf{c} and the sphere is the osculating circle of the curve in the point \mathbf{c} .

Proof. In this proof we use similar reasoning as in the three-dimensional case in Lemma 3.12. Suppose that in a certain neighborhood of the point \mathbf{c} the algebraic curve can be parametrized with arc length parametrization. It is not a restriction, since we are computing only with regular segment of the algebraic curve. The parametrization is denoted by

$$\mathbf{p}(t), \quad \text{where} \quad \mathbf{p}(t_0) = \mathbf{c}.$$

This curve is a curve on the surface $\hat{f} = 0$ according to the definition, therefore it satisfies

$$\frac{d^i \hat{f}(\mathbf{p}(t))}{dt^i} = 0,$$

for any i . Since s is the quadratic Taylor expansion of \hat{f} about \mathbf{c} , therefore

$$\langle \nabla s(\mathbf{c}), \mathbf{p}'(t_0) \rangle = 0$$

and

$$\frac{ds(\mathbf{p}(t))}{dt} = \frac{d^2s(\mathbf{p}(t))}{dt^2} = 0. \quad (4.23)$$

Moreover if the second derivative vanishes, then

$$\langle \nabla s(\mathbf{c}), \mathbf{p}''(t_0) \rangle = \lambda.$$

Since we parameterized the curve $\mathbf{p}(t)$ with arc length parametrization, therefore

$$\langle \nabla s(\mathbf{c}), \kappa \mathbf{n}(t_0) \rangle = \lambda,$$

where $\mathbf{n}(t_0)$ denotes the unit normal vector of the curve in the point \mathbf{c} . Thus the orthogonal projection of the radius of the sphere starting from the curve point \mathbf{c} to the normal of the curve has a fixed length

$$\left\langle \frac{\nabla s(\mathbf{c})}{\lambda}, \mathbf{n}(t_0) \right\rangle = \frac{1}{\kappa}.$$

It is exactly the radius of the osculating circle of the curve at the point \mathbf{c} . We know, that the intersection of a sphere and a two dimensional affine sub-space is a circle. The radius of such a circle has the same length as the orthogonal projection of the radius of the sphere starting from a point of the circle into the affine sub-space. The tangent and normal directions of the intersection circle of the sphere with the osculating plane are the same as the osculating circle of the curve, therefore the second statement of the lemma is confirmed. \square

4.4 Distance Estimate

In this section we describe a method to estimate the distance of two algebraic space curves. Since the curves are defined as the intersection of algebraic surfaces, the method is based on the distance estimation of the implicitly defined surfaces. First we generalize the distance estimation technique from Section 3.4.2 for implicitly defined and parametric curves. In order to get a sufficient distance estimation for algebraic space curves, we use pairwise distance estimation between the defining algebraic surfaces.

4.4.1 Distance of Algebraic and Parametric Space Curves

In order to bound the distance of algebraic space curves, we generalize the result from [20], which we used in the three dimensional case to bound the distance of parametric and implicitly defined space curves.

We assume that the a curve segment $\mathbf{r}(t)$ is defined with the parameter domain $t \in [0, 1]$ in $\Omega \subset \mathbb{R}^n$. The curve traces the point set

$$\mathcal{R} = \{\mathbf{r}(t) : t \in [0, 1]\}.$$

The algebraic curve $\mathcal{C}(F, \Omega)$ is defined by the zero set of $F = \{f_i, i = 1, \dots, n-1\}$ in the sub-domain Ω . In order to avoid certain technical difficulties, we bound the distance between the point set \mathcal{R} and

$$\mathcal{C}^* = \mathcal{C}(F, \Omega) \cup \partial\Omega,$$

where $\partial\Omega$ denotes the boundary of the domain. The one-sided Hausdorff-distance is defined as

$$\text{HD}_\Omega(\mathcal{R}, \mathcal{C}^*) = \sup_{t \in [0,1]} \inf_{\mathbf{x} \in \mathcal{C}^*} \|\mathbf{x} - \mathbf{r}(t)\|. \quad (4.24)$$

Lemma 4.16. *Consider the function $h = \sqrt{\sum_{i=1}^{n-1} f_i^2}$ defined by the polynomials $f_i \in F$. We assume that positive constants G and K exist, such that*

$$G \leq \|\nabla f_i\| \quad i = 1, \dots, n-1$$

and

$$|\nabla f_i \cdot \nabla f_j| \leq K \quad i \neq j, \quad i, j = 1, \dots, n-1,$$

then the length of the gradient is bounded by

$$\|\nabla h\|^2 \geq G^2 - \binom{n-1}{2} K \quad \forall \mathbf{x} \in \Omega.$$

Proof. Since

$$\nabla h = \frac{\sum_{i=1}^{n-1} f_i \nabla f_i}{\sqrt{\sum_{i=1}^{n-1} f_i^2}},$$

we obtain

$$\begin{aligned} \|\nabla h\|^2 &= \frac{1}{\sum_{i=1}^{n-1} f_i^2} \left(\sum_{i=1}^{n-1} f_i^2 \|\nabla f_i\|^2 \right) + \frac{1}{\sum_{i=1}^{n-1} f_i^2} \left(\sum_{1 \leq i < j \leq n-1} 2f_i f_j \nabla f_i \cdot \nabla f_j \right) \geq \\ &\geq \left| \frac{1}{\sum_{i=1}^{n-1} f_i^2} \left(\sum_{i=1}^{n-1} f_i^2 \|\nabla f_i\|^2 \right) \right| - \sum_{1 \leq i < j \leq n-1} \underbrace{\left| \frac{2f_i f_j}{\sum_{i=1}^{n-1} f_i^2} \right|}_{\leq 1} \underbrace{|\nabla f_i \cdot \nabla f_j|}_{\leq K} \geq \\ &\geq G^2 - \binom{n-1}{2} K. \end{aligned}$$

□

Theorem 4.17. *Consider a curve segment $\mathbf{r}(t) : t \rightarrow \Omega$, which traces the point set \mathcal{R} . The polynomials $f_i \in F$ define the algebraic curve $\mathcal{C}(F, \Omega)$. We assume that positive constants G and K exist, such that*

$$G \leq \|\nabla f_i\| \quad i = 1, \dots, n-1$$

and

$$|\nabla f_i \cdot \nabla f_j| \leq K \quad i \neq j, \quad i, j = 1, \dots, n-1.$$

If $G^2 - \binom{n-1}{2} K > 0$ and $h = \sqrt{\sum_{i=1}^{n-1} f_i^2}$, then

$$\forall \mathbf{x} \in \Omega \quad \|\nabla h\| \geq \sqrt{G^2 - \binom{n-1}{2} K}.$$

Moreover if there exists a positive constant M , that $\sum_{i=1}^{n-1} f_i(\mathbf{r}(t))^2 \leq M^2$, then the one-sided Hausdorff-distance is bounded by

$$HD_{\Omega}(\mathcal{R}, \mathcal{C}^*) \leq \frac{M}{\sqrt{G^2 - \binom{n-1}{2}K}}. \quad (4.25)$$

Proof. We compute a modified polynomial $\hat{f}_i = \mathcal{G}(F_i, \mathbf{u}_i, \mathbf{c})$ with respect to the domain Ω as the combination of polynomials F_i . Suppose that \mathbf{p} is a point from the parametric curve $\mathbf{r}(t)$. Let us define the function $h = \sqrt{\sum_{i=1}^n \hat{f}_i^2}$. We consider the integral curves defined in Ω by the vector field $-h/\|\nabla h\|$. From Lemma 4.16 we know, that the integral curves are regular in all inner points of Ω . We assume, that the integral curves $\mathbf{u}(s)$ are parametrized by arc length. According to the mean value theorem there exists s_0

$$\begin{aligned} h(\mathbf{u}(s)) &= h(\mathbf{u}(0)) + s \nabla h(\mathbf{u}(s_0)) \cdot \dot{\mathbf{u}}(s_0) = \\ &= h(\mathbf{u}(0)) - s \|\nabla h(\mathbf{u}(s_0))\| \leq M - s \sqrt{G^2 - \binom{n-1}{2}K}. \end{aligned}$$

Since $h(\mathbf{x}) \geq 0$, then $s \in [0, s^*]$, where $s^* = \frac{M}{\sqrt{G^2 - \binom{n-1}{2}K}}$. For a point \mathbf{y} the function $h(\mathbf{y}) = 0$ if $\mathbf{y} \in \mathcal{C}$. Since the integral curves are regular there exists a limit

$$\lim_{s \rightarrow s^*} \mathbf{u}(s) = \mathbf{y},$$

such that $\mathbf{y} \in \mathcal{C}$. Since we supposed that $\mathbf{u}(s)$ is arc length parametrized for all \mathbf{p} from the parametric curve $\mathbf{r}(t)$, there exists $\mathbf{y} \in \mathcal{C}$ such that

$$\|\mathbf{u}(0) - \mathbf{u}(s^*)\| = \|\mathbf{p} - \mathbf{y}\| \leq s^* = \frac{M}{\sqrt{G^2 - \binom{n-1}{2}K}}.$$

The same reasoning is applied in [20] to bound the distance of planar curves. \square

4.4.2 Distance of Algebraic Space Curves

If we would like to estimate the distance of algebraic space curve, we can measure first the distance of the defining algebraic surfaces. Suppose that an algebraic curve \mathcal{C} is defined by the polynomials $f_i \in F$ in the domain Ω

$$\mathcal{C}(F, \Omega) = \{\mathbf{x} : \forall i = 1, \dots, n-1, f_i(\mathbf{x}) = 0\} \cap \Omega.$$

An approximating space curve \mathcal{S} is given by the zero set of approximating algebraic surfaces $p_i \in P$

$$\mathcal{S}(P, \Omega) = \{\mathbf{x} : \forall i = 1, \dots, n-1, p_i(\mathbf{x}) = 0\} \cap \Omega.$$

The polynomial p_i approximates f_i .

We estimate the distance between the algebraic surfaces and the approximating surfaces pairwise. We consider the BB-norm, which is the maximum absolute value of the coefficients in the BB-representation. With the help of the norm, a distance bound can be defined

between an arbitrary polynomial f_i and an approximating polynomial p_i in the domain Ω as we observed in Section 3.4.1

$$\varepsilon_i = \|f_i - p_i\|_{\text{BB}}^{\Omega}. \quad (4.26)$$

Due to the convex hull property

$$|f_i(\mathbf{x}) - p_i(\mathbf{x})| \leq \varepsilon_i, \quad \forall \mathbf{x} \in \Omega.$$

Lemma 4.18. *Consider two algebraic curves $\mathcal{C}(F, \Omega)$ and $\mathcal{S}(P, \Omega)$, defined by the polynomials $f_i \in F$ and $p_i \in P$ in the domain $\Omega \subset \mathbb{R}^n$. We denote by ε_i the norm*

$$\varepsilon_i = \|f_i - p_i\|_{\text{BB}}^{\Omega} \quad i = 1, \dots, n-1.$$

Assume, that the constants G and K exist such that

$$0 < G \leq \|\nabla f_i\| \quad i = 1, \dots, n-1$$

and

$$|\nabla f_i \cdot \nabla f_j| \leq K \quad i \neq j, i, j = 1, \dots, n-1.$$

If $G^2 - \binom{n-1}{2}K > 0$, then for all points $\mathbf{x} \in \mathcal{S}$ exists a point $\mathbf{y} \in \mathcal{C}$ such that

$$\|\mathbf{x} - \mathbf{y}\| \leq \sqrt{\frac{\sum_{i=1}^{n-1} \varepsilon_i^2}{G^2 - \binom{n-1}{2}K}} = \varrho. \quad (4.27)$$

Proof. It is the consequence of Theorem 4.17 and (4.26). □

Lemma 4.18 gives us an upper bound of the distance between two algebraic space curves. So the bounding fat region can be defined as the point set

$$\mathcal{F}(P, \varrho, \Omega) = \{\mathbf{x} : \exists \mathbf{x}_0 \forall i = 1, \dots, n-1, p_i(\mathbf{x}_0) = 0, |\mathbf{x} - \mathbf{x}_0| \leq \varrho\} \cap \Omega.$$

The defining polynomials of the algebraic curves f_i and p_i are given in the BB-tensor product form. In order to find the constants in Lemma 4.18, we use the convex hull property of these polynomials.

4.5 Convergence and Global Algorithm

Since we generate quadratic approximating curves, we expect that the fat arc generation algorithm has cubic convergence rate. We analyze in this section the convergence rate of the method, and certify the third order convergence of the fat arcs in Lemma 4.24. Then we combine the local fat arc generation with subdivision.

4.5.1 Continuity of Taylor Expansion Modification

The local fat arc generation technique approximates the intersection curve of algebraic surfaces in the domain $\Omega \subseteq \Omega_0 \subset \mathbb{R}^n$ by a circular arc. This arc is defined as the intersection of spheres, which are given as the zero set of the quadratic Taylor expansion of polynomials with special Hessian. In order to prove that these arcs converge to a limit position in the sub-domains, we have to show, that the computed spheres depend continuously on the points of Ω_0 for a fixed choice of the parameter vector \mathbf{u} . This means, that the polynomial $\hat{f} = \mathcal{G}(F, \mathbf{u}, \mathbf{c})$ depends continuously on the choice of the point \mathbf{c} .

Lemma 4.19. *Given the set of polynomials $F = \{f_i : i = 1, \dots, n-1\}$ over the domain $\Omega \subseteq \Omega_0$. We suppose that for any point $\mathbf{c} \in \Omega_0$ the vector set $\{\nabla f_i(\mathbf{c}) : f_i \in F\}$ is linearly independent. For an arbitrary but fixed vector of parameters \mathbf{u} , where $u^i \in \mathbb{R} \setminus \{0\}$, we compute the polynomial*

$$\hat{f} = \mathcal{G}(F, \mathbf{u}, \mathbf{c})$$

with a special Hessian under the condition (4.13). Then \hat{f} depends continuously on the points of the domain Ω_0 .

Proof. We have to show that the computed linear factors l_i depend continuously on the point \mathbf{c} . We computed the coefficient vector $\mathbf{k} = (k_1^1, k_2^2, \dots, k_1^n, \dots, k_{n-1}^n)$, such that it satisfies the linear system $\mathbf{A}\mathbf{k} = \mathbf{b}$ in (4.9) and minimizes the l_2 -norm of the vector \mathbf{k} . If the vector set $\{\nabla f_i(\mathbf{c}) : f_i \in F\}$ is linearly independent for any \mathbf{c} , then \mathbf{A} has full rank (see Corollary 4.11 and Theorem 4.13). In this situation the vector, which satisfies (4.9) and (4.13), can be computed as

$$\mathbf{k} = \underbrace{\mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1}}_{\mathbf{A}^\dagger} \mathbf{b}.$$

The matrix \mathbf{A}^\dagger is the so called Moore-Penrose generalized inverse of \mathbf{A} (see [9]). Since f_i is a polynomial the entries of the matrix \mathbf{A} and the vector \mathbf{b} depend continuously on the point \mathbf{c} . Therefore the vector \mathbf{k} also depends continuously on the point \mathbf{c} . The values of $u^i \neq 0$ are fixed real numbers. So all coefficients u^i, k_i^j $i = 1 \dots n-1$ and $j = 1, \dots, n$ depend continuously on \mathbf{c} . Therefore also \hat{f} depends continuously on the point \mathbf{c} . \square

If we modify the Taylor expansion, then we can establish the following result considering the behavior of a sequence of the generated median circles.

Corollary 4.20. *Suppose we have a nested sequence of sub-domains $(\Omega_i)_{i=1,2,3,\dots} \subset \Omega_0 \subset \mathbb{R}^n$*

$$\Omega_{i+1} \subset \Omega_i,$$

which have decreasing diameters δ_i , such that

$$\lim_{i \rightarrow \infty} \delta_i = 0,$$

and \mathbf{c}_i denotes the center point of Ω_i . Consider a set of $n-1$ polynomials F , which defines an algebraic curve in \mathbb{R}^n

$$\mathcal{C}(F, \Omega_0) = \{\mathbf{x} : \forall f \in F, f(\mathbf{x}) = 0\} \cap \Omega_0.$$

Suppose that there exists a point $\mathbf{p} \in \Omega_i$ for all i , which satisfies

$$f_j(\mathbf{p}) = 0, \quad \forall j = 1, \dots, n-1,$$

and it is not an inflection point of the curve $\mathcal{C}(F, \Omega_0)$. We compute the set of $n-1$ new polynomials \hat{F}_i for each \mathbf{c}_i , such that each $\hat{f}_j^i \in \hat{F}_i$ is computed as $\hat{f}_j^i = \mathcal{G}(F, \mathbf{u}_j, \mathbf{c}_i)$ for different but fixed vectors of parameters \mathbf{u}_j , where $u_j^k \neq 0$. We consider the circle defined by the set of polynomials S_i , where each $s_j^i \in S_i$ defined as the quadratic Taylor expansion of a $\hat{f}_j^i \in \hat{F}_i$. Then the sequence of these circles converges to a limit circle, which is the osculating circle of the curve $\mathcal{C}(F, \Omega_0)$ in the point \mathbf{p} .

The next corollary follows from the fact that in the limit position, the zero sets of all quadratic polynomials intersect the osculating plane of the curve at the point \mathbf{p} in the osculating circle of the curve.

Corollary 4.21. *Given a set of polynomials $F = \{f_i : i = 1, \dots, n-1\}$ in the domain Ω_0 . Suppose we construct the polynomial \hat{f} , such that*

$$\hat{f} = \mathcal{G}(F, \mathbf{u}, \mathbf{c}) = \sum_{i=1}^{n-1} k_i f_i$$

for an arbitrary but fixed parameter vector \mathbf{u} , where $u^i \neq 0$. For all $\mathbf{c} \in \Omega_0$ the norm of the common coefficient vector of k_i , $\mathbf{k} = (k_1^1, k_1^2, \dots, k_{n-1}^n)$ can be bounded by a constant

$$\|\mathbf{k}\|_2 < L,$$

which depends only on F, Ω_0 and the choice of \mathbf{u} .

4.5.2 General Lower Bound for the Gradient Length

The following lemma (Lemma 3.19) ensures, that $\mathcal{G}(F, \mathbf{u}, \mathbf{c})$ has also a non-vanishing gradient if we compute fat arcs in sufficiently small sub-domains, which enclose the algebraic curve.

Lemma 4.22. *Suppose that there exists G in Ω_0 for the polynomials $f_i \in F$ such that for all $i = 1, \dots, n-1$*

$$\forall \mathbf{x} \in \Omega_0, \quad \|\nabla f_i(\mathbf{x})\| \geq G > 0. \quad (4.28)$$

Consider a domain $\Omega \subset \Omega_0$, which has a diameter $\delta_\Omega < \varepsilon$. Suppose that there is a point $\mathbf{p} \in \Omega$ such that for all $i = 1, \dots, n-1$, $f_i(\mathbf{p}) = 0$. The vector of parameters \mathbf{u} is arbitrary but fixed, such that $u^i \neq 0$. We compute $\hat{f} = \mathcal{G}(F, \mathbf{u}, \mathbf{c})$. If ε is sufficiently small, then there exists $\hat{G} > 0$ constant, such that for any $\mathbf{x} \in \Omega$

$$\|\nabla \hat{f}(\mathbf{x})\| \geq \hat{G} > 0.$$

Proof. If $\mathbf{x} \in \Omega \subseteq \Omega_0$ then

$$\nabla \hat{f}(\mathbf{x}) = \sum_{i=1}^{n-1} k_i(x) \nabla f_i(\mathbf{x}),$$

k_i are computed as described in Section 4.3.1. According to the triangle inequality

$$\|\nabla \hat{f}(\mathbf{x})\| \geq \left\| \sum_{i=1}^{n-1} k_i(x) \nabla f_i(\mathbf{x}) \right\| - \left\| \sum_{i=1}^{n-1} \nabla k_i(x) f_i(\mathbf{x}) \right\| \geq \left\| \sum_{i=1}^{n-1} k_i(x) \nabla f_i(\mathbf{x}) \right\| - \sum_{i=1}^{n-1} |f_i(\mathbf{x})| \|\nabla k_i(x)\|. \quad (4.29)$$

Since we know that there exists a point $\mathbf{p} \in \Omega$ such that $f_i(\mathbf{p}) = 0$, then

$$\forall i = 1, \dots, n-1, \quad |f_i(\mathbf{x})| \leq \frac{\varepsilon}{G}, \quad (4.30)$$

where ε is an upper bound of the diameter of Ω . In Corollary 4.5.2 we also observed, that there exists $L > 0$

$$|\mathbf{k}| \leq L,$$

which only depends on F, Ω_0 and the choice of \mathbf{u} . Therefore also

$$\forall i = 1, \dots, n-1, \quad \|\nabla k_i(\mathbf{x})\| \leq L.$$

We can bound the value of the linear polynomials l_i on a sufficiently small sub-domain Ω . Suppose that the diameter of Ω is smaller than ε . If $\mathbf{x} \in \Omega$, then for all $i = 1, \dots, n-1$

$$|k_i(\mathbf{x})| = \left| u^i + \sum_{j=1}^n k_i^j (x^j - c^j) \right| > |u^i| - \frac{\varepsilon}{2} \sqrt{n}L$$

where $\mathbf{c} = (c^j)_{j=1}^n$ denotes the center of Ω . Since u^i are non-zero, if

$$\varepsilon < \frac{\min_{i=1, \dots, n-1} \{u^i\}}{\sqrt{n}L}, \quad (4.31)$$

then $|k_i(x)| \geq u^i/2$.

We supposed that $\nabla f_i(\mathbf{x})$ are linearly independent in any point of Ω_0 . If (4.31) is satisfied for an $\Omega \subseteq \Omega_0$, then there exists a general bound \tilde{G} depending on \mathbf{u} and G , such that

$$\left\| \sum_{i=1}^{n-1} k_i(\mathbf{x}) \nabla f_i(\mathbf{x}) \right\| \geq \tilde{G} > 0, \quad \forall \mathbf{x} \in \Omega.$$

Therefore for all $\mathbf{x} \in \Omega$

$$\|\nabla \hat{f}(\mathbf{x})\| \geq \tilde{G} - \sum_{i=1}^{n-1} \|f_i(\mathbf{x}) \nabla k_i(\mathbf{x})\| \stackrel{(4.13)}{\geq} \tilde{G} - L \sum_{i=1}^{n-1} |f_i(\mathbf{x})|.$$

Since we know that there exists a point $\mathbf{p} \in \Omega$ such that $f(\mathbf{p}) = g(\mathbf{p}) = 0$, then because of (4.30)

$$\|\nabla \hat{f}(\mathbf{x})\| \geq \tilde{G} - \frac{(n-1)\varepsilon L}{G}.$$

Suppose that

$$\varepsilon = \min \left\{ \frac{\tilde{G}G}{(n-1)L}, \frac{u^1}{\sqrt{n}L}, \dots, \frac{u^{n-1}}{\sqrt{n}L} \right\}. \quad (4.32)$$

If the diameter of Ω denoted by δ_Ω satisfies

$$\delta_\Omega < \frac{\varepsilon}{2},$$

then

$$\|\nabla \hat{f}(\mathbf{x})\| \geq \frac{\tilde{G}}{2} = \hat{G} > 0.$$

□

Corollary 4.23. *Suppose that the conditions of Lemma 4.22 are satisfied for a set of polynomials F in the domain Ω_0 . If $\hat{f} = \mathcal{G}(F, \mathbf{u}, \mathbf{c})$ is computed in a sufficiently small sub-domain Ω of Ω_0 , then $T_{\mathbf{c}}^2(\hat{f}) \neq 0$.*

4.5.3 Convergence of Taylor Expansion Modification

Now we have to show that the fat arc thickness is sufficiently small compared with the diameter of the computational domain. The following lemma shows, how the computed fat arc thickness behaves as the size of the domain tends to zero.

Lemma 4.24. *Given a set of polynomials F defined over the domain $\Omega_0 \subset \mathbb{R}^n$. We suppose that the conditions of Lemma 4.22 are satisfied. We compute a set of polynomials \hat{F} with special Hessian for arbitrary but fixed vectors of parameters \mathbf{u}_i and apply the orthogonalization function*

$$\hat{F} = \mathcal{O}(\{\mathcal{G}(F, \mathbf{u}_i, \mathbf{c}) : i = 1 \dots n-1\}, \mathbf{c}),$$

in the center point \mathbf{c} of the sub-domain $\Omega \subset \Omega_0$. If the sub-domain Ω has a sufficiently small diameter δ_Ω , then there exists a constant $C \in \mathbb{R}$, which does not depend on the choice of Ω and satisfies

$$\varrho \leq C\delta_\Omega^3, \quad (4.33)$$

where ϱ is the corresponding fat arc thickness computed as in (4.27).

Proof. Since the conditions of Lemma 4.22 are satisfied, we know that for all $\hat{f}_i \in \hat{F}$ there exists \hat{G} such that

$$\|\nabla \hat{f}_i\| \geq \hat{G},$$

for any sufficiently small sub-domain Ω , which encloses the curve. We denote by p_i the quadratic Taylor expansion of $\hat{f}_i \in \hat{F}$ about the center \mathbf{c} of the domain Ω . Then

$$\|\hat{f}_i - p_i\|_\infty = \|\hat{f}_i - T_{\mathbf{c}}^2(\hat{f}_i)\|_\infty < \underbrace{\frac{1}{6} \max_{\mathbf{v} \in S^1, \mathbf{x} \in \Omega} \left| \frac{d^3 \hat{f}_i}{d\mathbf{v}^3}(\mathbf{x}) \right|}_{*} \delta_\Omega^3.$$

Recall from Lemma 4.19 that \mathcal{G} depends continuously on the points of the computational domain Ω_0 for each \mathbf{u}_i , where $\mathbf{u}_i \neq \mathbf{0}$. Therefore also \hat{f}_i depend continuously on the points of the computational domain Ω_0 . Thus for all \hat{f}_i a general upper bound C_i can be given for (*).

The fat arc thickness is defined by

$$\varrho_\Omega = \sqrt{\frac{\sum_{i=1}^{n-1} \varepsilon_i^2}{G_\Omega^2 - \binom{n-1}{2} K_\Omega}},$$

where

$$\varepsilon_i = \|\hat{f}_i - p_i\|_{\text{BB}}.$$

Because of the norm equivalences there exist D_i , such that

$$\varepsilon_i \leq D_i \|\hat{f}_i - p_i\|_\infty.$$

So we observe, that

$$\sqrt{\sum_{i=1}^{n-1} \varepsilon_i^2} \leq \underbrace{\frac{1}{6} \sqrt{\sum_{i=1}^{n-1} (C_i D_i)^2}}_M \delta_\Omega^3.$$

We assumed that $\hat{G} < G_\Omega$ is a general lower bound for each $\|\nabla \hat{f}_i\|$ independent of the choice of the sub-domain Ω . Since we also applied the orthogonalization step to the polynomials \hat{f}_i for any $i \neq j$

$$\left| \nabla \hat{f}_i(\mathbf{c}) \cdot \nabla \hat{f}_j(\mathbf{c}) \right| = 0$$

in the center point \mathbf{c} of a domain Ω . If the diameter of a sub-domain Ω is sufficiently small, then there exists $K > 0$, which does not depend on Ω and each $\mathbf{x} \in \Omega$ satisfies

$$\left| \nabla \hat{f}_i(\mathbf{x}) \cdot \nabla \hat{f}_j(\mathbf{x}) \right| \leq K$$

for any $i \neq j$. If the diameter of the sub-domains is sufficiently small, then the general bound K satisfies $\binom{n-1}{2}K < \hat{G}^2$. Thus this implies that

$$\varrho_\Omega \leq \frac{M \delta_\Omega^3}{\sqrt{\hat{G}_\Omega^2 - \binom{n-1}{2}K_\Omega}} \leq \frac{M \delta_\Omega^3}{\sqrt{\hat{G}^2 - \binom{n-1}{2}K}} \leq C \delta_\Omega^3.$$

□

4.5.4 Global Algorithm

Subdivision is a frequently used technique, and it is often combined with local approximation methods. Such hybrid algorithms subdivide the computational domain in order to separate regions where the local curve approximation techniques can be applied. The regions with unknown curve behavior can be made smaller and smaller with subdivision.

The local algorithm `FatArcLocal_nd` (see Algorithm 7) generates fat arcs for regular algebraic space curves. As we saw in the two and three-dimensional case, this local method can be combined with recursive subdivision.

First the Bernstein–Bézier coefficients of the polynomials are analyzed with respect to the computational domain. If no sign changes are present for one or both of the polynomials, then the current domain does not contain any components of the algebraic curve. Otherwise the fat arc generation technique can be applied. If it is not successful, then the algorithm either subdivides the current domain into sub-domains, or returns the entire domain, if its diameter is already below the user-defined threshold ε . It is guaranteed that during the process no region will be eliminated, which contains the implicitly defined curve. However, it may happen, that the output contains domains without any segments of the implicitly defined curve (“false positive boxes”).

Chapter 5

Fat Spheres for Solving Multivariate Polynomial Systems

Solving multivariate polynomial systems has several applications in algebra and geometry. Therefore various methods exist to find or to isolate the roots of polynomial systems. They are using symbolic, numeric or combined techniques in order to find solutions. In this chapter first we give a brief summary on the topic. We discuss the main classes of solvers and the existing results. Then we introduce fat spheres, which are multidimensional bounding regions for implicitly defined algebraic objects. With the help of fat spheres we describe a local domain reduction strategy, which bounds the intersection of algebraic objects. We combine this local algorithm with iterative subdivision. This hybrid algorithm can be applied for approximating the solution of multivariate polynomial systems.

5.1 Fat Arcs and Fat Spheres

In this section we summarize the related work in solving multivariate polynomial systems. First we describe the different families of solvers. Then we define the fat spheres. Finally we formulate the root finding problem with polynomials represented in Bernstein-Bézier form.

5.1.1 Real Root Finding Algorithms

Real root finding is considered as a difficult task. A general overview about the multivariate root finding algorithms is given in [13, 38]. The solvers, described in the literature, are using either algebraic or geometric tools.

Algebraic approaches, such as the Gröbner-basis technique [5], resultant based methods or continuous fractions methods assure exact and efficient solution algorithms. These algorithms frequently provide more information about the solutions than we need. It is often unnecessary to compute all solutions. For instance CAD-systems usually require information only about real solutions, which lie in a certain domain. Moreover these symbolic methods are not really suitable for numerical computations.

An algebraic solver, which is using the Gröbner-basis technique, was developed for instance by Rouillier [33] for bi-variate polynomial systems. Busé et al. considered resultant

based methods in [6, 7]. In [14] an algebraic method is described, which is using Sturm-Habicht sequences.

Homotopy solvers compute a family of root-finding problems. The method transforms a simple problem to the original one in several steps, and compute the roots of each intermediate problem. The computed sequence of roots converges to the solutions of the original root-finding problem. However, such computations are usually require inefficient memory and time.

Polynomial solvers based on homotopy methods can be found in [24, 28].

Subdivision algorithms are based on the “divide and conquer” paradigm. They compute in a certain domain (usually in an axis-aligned box) and provide information only about real roots. If we are interested in certain properties of a root, like multiplicity, then further computations are necessary. These algorithms decompose the problem into several sub-problems. The decomposition terminates if suitable approximating primitives can be generated in each sub-problem [29]. In order to construct these approximating primitives, several local domain reduction strategies can be applied. These reduction methods are usually based on interpolation, bounding region generation or least-squares approximation.

The first subdivision solvers were developed by Sederberg et al. for bivariate polynomials represented in Bernstein-Bézier tensor product form. They are using clipping and subdivision techniques [35, 36]. Later on a family of algorithms were invented, which is using projection techniques [39]. The most recently developed solvers are published by Mourrain et al. [13] and Elber et al. [12].

5.1.2 Definition of Fat Arcs and Fat Spheres

We present in this chapter an algorithm, which combines iterative domain reduction with a subdivision technique to solve multivariate polynomial systems. The domain reduction strategy is based on bounding region generation. In Chapter 4 we generated fat arcs as bounding regions for n -dimensional algebraic curves. These bounding regions consist of a one-dimensional approximating primitive (a circular arc) and a certain neighborhood of this curve. In this chapter we consider bounding regions, which are generated as a thickened neighborhood of a multi-dimensional object. The multi-dimensional object approximates a part of an algebraic surface, and it is defined as a segment of a sphere. The thickened neighborhood of the sphere-segment contains each point of the algebraic surface. Therefore it is a bounding region for the patch of the algebraic surface. So we extend the definition of fat arcs (see in Definition 4.1) to the concept of fat spheres.

Definition 5.1. *A fat sphere is defined in \mathbb{R}^n by*

- *a patch of an arbitrary dimensional sphere (median sphere) $\mathcal{S} \subset \Omega \subset \mathbb{R}^n$,*
- *and a distance $\varrho \in \mathbb{R}$.*

Then the fat region is the point set

$$\mathcal{F}(\mathcal{S}, \varrho) = \{\mathbf{x} : \exists \mathbf{x}_0 \in \mathcal{S}, \|\mathbf{x} - \mathbf{x}_0\|_2 \leq \varrho\}.$$

The fat spheres with one-dimensional median sphere are the standard fat arc in \mathbb{R}^n (see in Definition 4.1). In this case the median sphere is a circular arc.

A multi-dimensional median sphere \mathcal{S} can always be defined as an algebraic set. It is the intersection of implicitly defined spheres, each possessing the form

$$p_i = a_i \langle \mathbf{x}, \mathbf{x} \rangle + \langle \mathbf{b}_i, \mathbf{x} \rangle + c_i = 0, \quad a_i, c_i \in \mathbb{R}, \mathbf{b}_i \in \mathbb{R}^n.$$

The median spherical patch is defined algebraically as a whole sphere restricted to an axis-aligned domain

$$\mathcal{S} = \{\mathbf{x} : \forall i, p_i(\mathbf{x}) = 0\} \cap \Omega.$$

Median spheres can also be represented in parametric form with the help of rational functions. It is an advantageous property of arcs and spheres, that they possess exact parametric and implicit representation form. The implicit representation provide us a simple way to represent the offset of the spheres and to compute the intersection of them, while the parametric form simplifies the visualization.

5.1.3 Multivariate Polynomial Systems

In order to compute efficiently the real roots of a polynomial system with subdivision technique, we assume that the polynomials are given in tensor-product Bernstein-Bézier(BB) form (see (4.1)), with respect to the axis-aligned domain

$$\Omega_0 = \times_{i=1}^n [\alpha_i, \beta_i] \subset \mathbb{R}^n.$$

For such a polynomial $f : \mathbb{R}^n \rightarrow \mathbb{R}$, let us denote the zero level set with respect to the domain Ω_0 as

$$\mathcal{Z}(f, \Omega_0) = \{\mathbf{x} : f(\mathbf{x}) = 0\} \cap \Omega_0.$$

The solution set of a polynomial system $F = \{f_1, \dots, f_n\}$ is the intersection of the zero set of the polynomials

$$\mathcal{R}(F, \Omega_0) = \bigcap_{i=1}^n \mathcal{Z}(f_i, \Omega_0) \tag{5.1}$$

with respect to the domain Ω_0 . If the system of polynomials is zero-dimensional, then this algebraic set consists of distinct points or it is the empty set.

5.2 Bounding Region Generation

In order to generate fat spheres for solving polynomial systems, we present first a local domain reduction strategy. This local algorithm is applied in the sub-domains of the computational domain Ω_0 . In each sub-domain the zero set of the polynomial system is bounded by intersecting fat spheres. Later on we will combine this local domain reduction strategy with a subdivision technique.

Algorithm 8 DomainReduction(F, Ω)

Require: Each polynomial has sign change in its BB-coefficients in Ω .

- 1: \hat{f}_i modified polynomials with spherical quadratic Taylor expansion p_i
- 2: $\mathcal{S}_i = \{\mathbf{x} : p_i(\mathbf{x}) = 0\} \cap \Omega$ {median spheres}
- 3: $\varepsilon_i = \left\| \hat{f}_i - p_i \right\|_{\text{BB}}$
- 4: $\mathcal{P}_i^\pm = \{\mathbf{x} : \forall i, p_i(\mathbf{x}) = \pm \varepsilon_i\} \cap \Omega \leftarrow$ boundaries of fat spheres
- 5: $\mathcal{C} \leftarrow$ extremal points of fat sphere intersection
- 6: **if** $\mathcal{C} \neq \emptyset$ **then**
- 7: $\mathcal{M} \leftarrow$ min-max box around the points \mathcal{C} {new bounding domain}
- 8: **return** \mathcal{M}
- 9: **end if**
- 10: **return** \emptyset {no bounding domain has been found}

5.2.1 Local Algorithm

In order to bound the zero set of a polynomial system, first we detect the empty sub-domains in the computational domain and eliminate them. Therefore we analyze the sign changes of the BB-coefficients in the representation of the polynomials. If one of the polynomials has only negative or only positive BB-coefficients over the sub-domain, then no point of the sub-domain belongs to the solution set of the polynomial system (see Observation 3.2). Such sub-domains can be neglected during further computations.

In order to bound the zero set of the polynomials $F = \{f_1, \dots, f_n\}$, we generate fat spheres as bounding regions. First we compute a new system of polynomials with modified Taylor expansion. The technique, we described in Section 4.3.1, provide us a method to compute polynomials \hat{f}_i , which has special Hessian matrix in the center point of the sub-domain. The set of modified polynomials $\hat{F} = \{\hat{f}_1, \dots, \hat{f}_n\}$ has a zero set, which contains the solution set of the polynomials F

$$\mathcal{Z}(F, \Omega) \subseteq \mathcal{Z}(\hat{F}, \Omega)$$

in the sub-domain Ω . The quadratic Taylor expansion of the modified polynomials about the center point \mathbf{c} of Ω

$$p_i = T_{\mathbf{c}}^2(f_i)$$

has a zero level set, which is a part of a sphere. Each sphere is used as a median sphere to generate a fat sphere \mathcal{F}_i . Such a fat sphere is the thickened neighborhood of the median sphere $p_i = 0$, and it contains the zero set of \hat{f}_i in the sub-domain Ω . If all the fat spheres intersect in Ω , then a min-max box is constructed around this intersection (see details Section 5.2.3). The local algorithm returns this min-max box as a bounding region of the zero set of the polynomials $f_i \in F$.

If the fat spheres have no intersection, then the sub-domain Ω does not contain any point of $\mathcal{Z}(\hat{F}, \Omega)$, so as no point of $\mathcal{Z}(F, \Omega)$. This implies, that no solution of the polynomial system lies in the sub-domain Ω . Thus such a sub-domain with non-intersecting fat spheres can be neglected in the further computations.

The two-dimensional real root finding algorithm approximates the solution of two bivariate polynomials. In this low dimensional case the definition of fat spheres coincides with the concept of fat arcs. The median sphere is always a circular arc. In each sub-domain, which is

not detected as a region without any root inside, the local algorithm generates two fat arcs. These are the bounding regions of the two different algebraic curves. Fig.5.1 presents some examples of these fat arcs and the bounding box around their intersection. Each figure has been generated with the help of Algorithm 8. In the second figure one can see, that the fat arcs intersect each other, however the polynomials have no solution point in the sub-domain. Such “false positive regions” can be eliminated if we apply the domain reduction iteratively.

In the three-dimensional space the domain reduction algorithm bounds the intersection of three algebraic surfaces. The fat spheres are generated as thickened three dimensional spheres. Fig.5.2 presents some examples generated with the help of Algorithm 8.

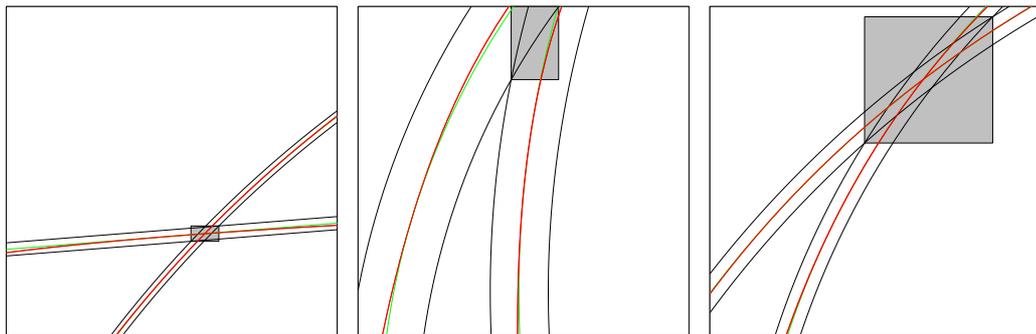


Figure 5.1: Examples for fat arc intersection with the help of algorithm `DomainReduction`. The red curves are the implicitly defined curves. The median circles are shown in green. The gray regions represent the generated bounding regions: the min-max boxes around the intersections of fat arcs.

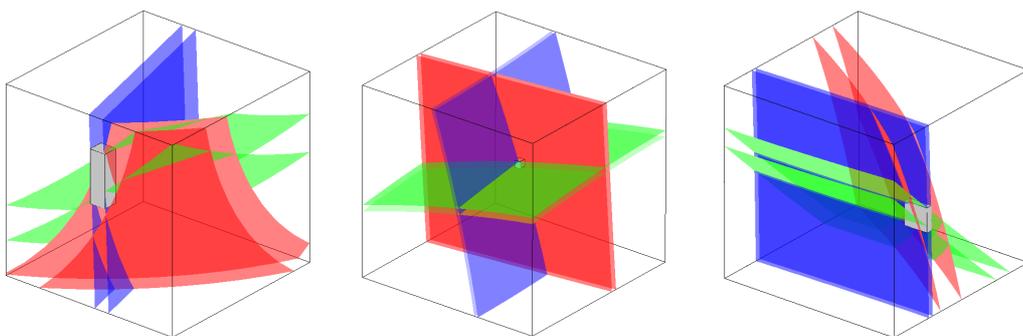


Figure 5.2: Examples for fat sphere intersection generated by the algorithm `DomainReduction` in three-dimensional space. The boundary patches of the three fat spheres are represented in red, green and blue. The gray regions represent the generated bounding regions: the min-max boxes around the intersections of fat spheres.

In the next sections we consider the most important steps of the local algorithm. We will first describe the fat sphere generation technique. Then we will show how to generate min-max box around the intersection of fat spheres.

5.2.2 Fat Sphere Generation

We approximate the zero set of the polynomials $F = \{f_1, \dots, f_n\}$ in the sub-domain $\Omega \subset \mathbb{R}^n$. The geometric interpretation of this problem is to find the intersection points of algebraic hyper-surfaces in the sub-domain Ω . According to this approach we generate fat spheres as bounding regions for each algebraic surface. Due to the definition of fat spheres first we compute an approximating sphere segment, the median sphere, for each algebraic surface. Then we bound the distance of the approximating sphere and the algebraic surface.

With combining the polynomials $f \in F$ we compute a polynomial \hat{f} , which has a special Hessian matrix in the center point \mathbf{c} of the sub-domain Ω ,

$$\mathcal{H}(\hat{f})(\mathbf{c}) = \begin{pmatrix} \lambda & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda \end{pmatrix} = \lambda \mathbf{I}^{n \times n}, \quad \lambda \in \mathbb{R}. \quad (5.2)$$

We apply the same technique as in the fat arc generation method (see Section 4.3.1). We compute a polynomial \hat{f} as the combination of the polynomials $f_i \in F$ with respect to the index set $i \in J \subseteq \{1, \dots, n\}$ as

$$\hat{f} = \sum_{i \in J} k_i f_i \quad (5.3)$$

multiplying with the linear polynomials k_i defined as in (4.6). The new polynomial has to satisfy (5.2). The coefficients of k_i can be computed by solving a linear system. In order to avoid to have only the trivial solution for the coefficient of the multipliers k_i , we choose the constant term of the multipliers arbitrary but fixed, non-zero parameter values. Then the number of the free coefficients in the multipliers has to be more than the number of equations in the linear system (see (4.9)). So we avoid to have an overdetermined system.

According to the observations on the solvability of this system in Section 4.3.1 we consider only the cases, when the combination (5.3) involves $n-1$ or n polynomials. In these cases the polynomials k_i exist, and they are non-zero linear polynomials. We describe the behavior of the linear system in Table 5.1 similarly to the case of fat arcs in Table 4.1. If we combine all polynomials (not only $n-1$ ones), then the solution space of k_i has an even higher number of dimension. However, according to our experiences, using all polynomials gives better approximations and speeds up the shrinking of the bounding regions.

The solution space of the coefficients of k_i is at least one-dimensional for the combination of $n-1$ and n polynomials too. However, we need only one collection of coefficients, which defines the multipliers k_i . Therefore we compute the solution vector of coefficients, which has the minimal l_2 -norm (as for fat arcs (4.13)).

A modified polynomial with special Hessian matrix has special quadratic Taylor expansion $p = T_{\mathbf{c}}^2 \hat{f}$. This quadratic polynomial defines the algebraic set

$$\mathcal{S} = \{\mathbf{x} : p(\mathbf{x}) = 0\},$$

which can be used as median sphere. We estimate the error of the approximation by bounding the difference of the polynomials \hat{f} and p . With the help of the BB-norm the distance bound can be given as

$$\varepsilon = \left\| \hat{f} - p \right\|_{\text{BB}}^{\Omega}. \quad (5.4)$$

Table 5.1: Comparison of strategies to construct polynomials with special Hessian for different number of variables. The table shows the number of coefficients and the dimension of their solution space in the construction of a new function \hat{f} . For each number of dimension n , the first row shows the results if we combine $n - 1$ polynomials, the second one if we combine n polynomials.

dimension	num. of equ.	num. of multipliers	num. of coeff.	dim. of sol. sys.
3	5	2	6	1
		3	9	4
4	9	3	12	3
		4	16	7
5	14	4	20	10
		5	25	11
6	20	5	30	10
		6	36	16
100	5049	99	9900	4851
		100	10000	4951

Due to the convex hull property

$$\left| \hat{f}(\mathbf{x}) - p(\mathbf{x}) \right| \leq \varepsilon, \quad \forall \mathbf{x} \in \Omega,$$

which implies that

$$p(\mathbf{x}) - \varepsilon \leq \hat{f}(\mathbf{x}) \leq p(\mathbf{x}) + \varepsilon, \quad \forall \mathbf{x} \in \Omega. \quad (5.5)$$

A fat sphere as bounding region can be defined in Ω for $\hat{f} = 0$ as

$$\mathcal{F}(p, \varepsilon, \Omega) = \{\mathbf{x} : |p(\mathbf{x})| \leq \varepsilon\} \cap \Omega.$$

The boundaries of this region are the offsets of the median sphere $p = 0$. This fat sphere is bounding the zero level set of f .

In the two-dimensional case the fat sphere generation is the same as the fat arc generation. The zero level set of polynomials and their approximations are given as implicitly defined curves in \mathbb{R}^2 . In the three-dimensional space we have two different strategies to generate modified polynomials. We can use either two or all three polynomials from F to generate a new polynomial \hat{f} . Then a fat sphere is defined as a thickened region of a three-dimensional spherical patch.

5.2.3 Min-max Box of the Intersection of Fat Spheres

We compute a set of polynomials with modified Taylor expansion $\hat{F} = \{\hat{f}_1, \dots, \hat{f}_n\}$. The set of modified polynomials has the same or a larger solution set as F

$$\mathcal{Z}(F, \Omega) \subseteq \mathcal{Z}(\hat{F}, \Omega).$$

Each polynomial has a special quadratic Taylor expansion $p_i = T_{\mathbf{c}}^2 \hat{f}_i$ about the center of the sub-domain Ω . These quadratic polynomials define the algebraic sets

$$\mathcal{S}_i = \{\mathbf{x} : p_i(\mathbf{x}) = 0\},$$

which can be used as median spheres. We bound the distance of the polynomials pairwise

$$\varepsilon_i = \left\| \hat{f}_i - p_i \right\|_{\text{BB}}^{\Omega}. \quad (5.6)$$

In order to bound the zero set of the polynomials \hat{F} , we consider the intersection of the generated fat spheres. Each fat sphere

$$\mathcal{F}_i(p_i, \varepsilon_i, \Omega) = \{\mathbf{x} : |p_i(\mathbf{x})| \leq \varepsilon_i\} \cap \Omega$$

bounds the zero level set of the polynomial \hat{f}_i in the sub-domain Ω . If the intersection of fat spheres is not empty,

$$\mathcal{I} = \bigcap_{i=1}^n \mathcal{F}_i(p_i, \varepsilon_i, \Omega) \neq \emptyset,$$

then it contains the zero set of the polynomials \hat{f}_i in the sub-domain Ω .

Observation 5.2. If the intersection is empty

$$\mathcal{I} = \bigcap_{i=1}^n \mathcal{F}_i(p_i, \varepsilon_i, \Omega) = \emptyset,$$

then it implies, that the intersection of the zero sets $\mathcal{Z}_i = \{\mathbf{x} : \hat{f}_i(\mathbf{x}) = 0\}$ is also empty

$$\bigcap_{i=1}^n \mathcal{Z}_i \cap \Omega = \emptyset.$$

Thus the domain Ω has no common point with the zero set of the polynomial system \hat{F} and also with the zero set of F . These sub-domains can be neglected during the further computations.

Now we consider the case

$$\mathcal{I} = \bigcap_{i=1}^n \mathcal{F}_i(p_i, \varepsilon_i, \Omega) \neq \emptyset.$$

The region \mathcal{I} is a „curved polytope“, which is bounded by spherical patches and linear subspaces. The spherical patches are a part of the boundary surfaces of the fat spheres. The pair of bounding spheres of the fat sphere $\mathcal{F}_i(p_i, \varepsilon, \Omega)$ can be described as the point set

$$\mathcal{P}_i = \{\mathbf{x} : p_i(\mathbf{x}) = \pm \varepsilon_i\} \cap \Omega.$$

The segments of linear subspaces, which bound the fat sphere intersection, are a part of the boundaries of the sub-domain Ω .

An example for two-dimensional fat arc intersection is shown in Fig.5.3. Each fat arc is the intersection of the computational domain and an annulus. The intersection of two fat arcs is bounded by a curved polygon. The boundaries of the polygon are circular arcs and line segments.

In general the intersection of fat spheres is a curved polytope. It is not practical to use it as computational domain in further domain reductions. In order to reduce iteratively the

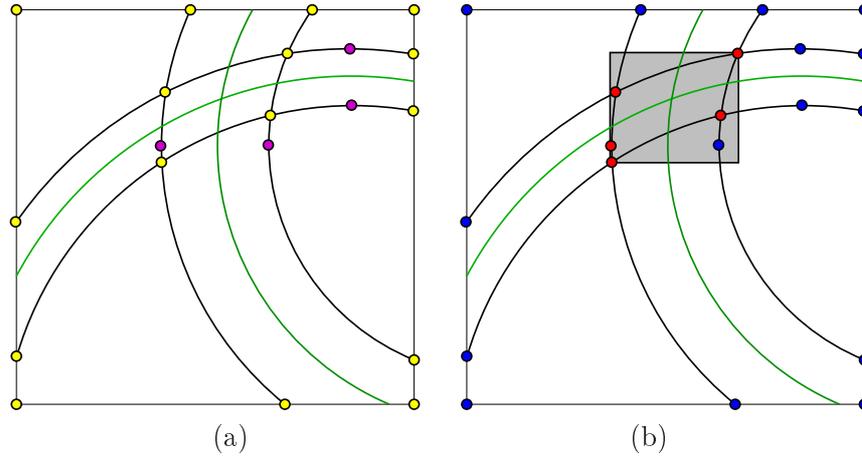


Figure 5.3: Extremal points of fat arc intersection. The fat arcs are represented by their bounding arcs (black) and the median arcs (green). In figure (a) the fat sphere corner points are marked by yellow dots and the fat sphere extreme points (fat sphere 1-extrema) by pink ones. The important fat sphere extrema from the fat sphere extrema are marked by red dots in figure (b). The bounding region of the fat arc intersection is the min-max box generated around the important fat sphere extrema (gray rectangle).

bounding regions, the output of the domain region has to be an axis-aligned box. Therefore we compute the min-max box, which bounds the fat sphere intersection \mathcal{I} . This box can be computed exactly, by finding the extremal points of the fat sphere intersection. For instance in Fig.5.3 (b) the extremal points of the fat arc intersection are marked by red dots. Four of these extrema are the intersection points of the fat arc boundaries, while another one is an extremal point of a boundary arc. In order to find the extremal points of the fat sphere intersection in general, we use the following definitions.

Definition 5.3. Given a system of polynomials \hat{F} in the sub-domain $\Omega = \times_{i=1}^n [\alpha_i, \beta_i]$. For each polynomial $\hat{f}_i \in \hat{F}$ we can compute the fat sphere

$$\mathcal{F}_i(p_i, \varepsilon_i, \Omega),$$

where p_i is the quadratic Taylor expansion of \hat{f}_i about the center point of the sub-domain Ω . The fat sphere $\mathcal{F}_i(p_i, \varepsilon_i, \Omega)$ bounds $\hat{f}_i = 0$ in Ω . The i th boundary pair of the domain Ω is defined as

$$\partial\Omega_i = \{\mathbf{x} : x^i = \alpha_i \vee x^i = \beta_i\}.$$

The boundary points of the fat sphere \mathcal{F}_i are contained in the set

$$\mathcal{P}_i = \{\mathbf{x} : p_i(\mathbf{x}) = \varepsilon_i \vee p_i(\mathbf{x}) = -\varepsilon_i\}.$$

Let N_S denote the number of the elements of an index set $S \subseteq \{1, \dots, n\}$. A point $\mathbf{x} \in \Omega$ is called

(i) **fat sphere corner point** if $I, J \subset \{1, \dots, n\}$, $N_I = k$, $N_J = n - k$:

$$\mathbf{x} \in \mathcal{X} = \bigcap_{i \in I} \partial\Omega_i \bigcap_{j \in J} \mathcal{P}_j$$

(ii) **fat sphere m -extreme point** if $I, J \subset \{1, \dots, n\}$, $N_I = k$, $N_J < n - k$

$$\mathbf{x} \in \mathcal{Y} = \bigcap_{i \in I} \partial\Omega_i \bigcap_{j \in J} \mathcal{P}_j,$$

where \mathcal{Y} is an m -dimensional algebraic object, and there exists $n - k - m$ different indexes $l \in \{1, \dots, n\} \setminus I$, such that

$$\sum_{j \in J} \left(\frac{\partial p_j}{\partial x_l} \right)^2 = 0.$$

All corner points \mathbf{x} of the sub-domain Ω are fat sphere corner points for $k = n$

$$\mathbf{x} \in \bigcap_{i=1}^n \partial\Omega_i \subset \mathcal{X}.$$

All intersection points \mathbf{x} of the fat sphere boundaries, which lie in the interior of the domain Ω are in the point set

$$\mathbf{x} \in \bigcap_{j=1}^n \mathcal{P}_j \subset \mathcal{X},$$

These points are fat sphere corner points with $k = 0$.

Observation 5.4. A fat sphere m -extreme point is always an extreme point of an m -dimensional object defined by the intersection of spherical patches and linear subspaces. A fat sphere m -extreme point is a point on the object, where the tangent space (the linear subspace spanned by the gradient vectors of the intersecting algebraic surfaces) in the point is perpendicular to $n - k - m$ coordinate directions. For instance the fat sphere $(n - 1)$ -extreme points are the extremal points of the fat sphere boundary patches defined by the equations $p_i = \pm \varepsilon_i$.

Observation 5.5. All fat sphere corner points and fat sphere extreme points are defined by an equation system with n equation in n variables, where all equations are linear or quadratic ones. The quadratic equations are the equations of spheres. Therefore each fat sphere corner point and fat sphere extreme point can be computed as the solution of an equation system consists of $n - 1$ linear equations and a single quadratic equation.

Definition 5.6. We call a fat sphere corner point or a fat sphere extreme point \mathbf{x} an important fat sphere extrema, if it satisfies for all $i \in \{1, \dots, n\}$

$$-\varepsilon_i \leq p_i(\mathbf{x}) \leq \varepsilon_i,$$

thus the point $\mathbf{x} \in \mathcal{I}$ belongs to the intersection of the fat spheres.

Observation 5.7. Definition 5.3 and Observation 5.5 imply, that all important fat sphere extrema can be computed by solving a finite number of algebraic systems, where each system consists $n - 1$ linear and one linear or quadratic equations, and at most $2n$ inequality tests.

Lemma 5.8. The min-max box around the region $\mathcal{I} = \bigcap_{i=1}^n \mathcal{F}_i(p_i, \varepsilon_i, \Omega) \neq \emptyset$, which is the fat sphere intersection in the sub-domain Ω , is the min-max box around the important fat sphere extrema.

Proof. The min-max box around \mathcal{I} is the min-max box around the extremal points of the region. Since the fat sphere intersection is bounded by spherical patches and segments of linear sub-spaces, any extremal point is either the corner point of the region, or the local minimum/maximum point of the spherical patch or its boundaries. All corner points of the intersection \mathcal{I} are contained in the point set of fat sphere corner points. All local extremal points of \mathcal{I} are fat sphere m -extreme points. Moreover only the points of \mathcal{I} fulfill the condition for the important fat sphere extrema. \square

Fig.5.3 (b) shows a two-dimensional fat arc intersection, where the fat sphere corner points and the fat sphere extreme points are marked by red and blue dots. The red ones denote the important fat sphere extrema.

The min-max box of the fat sphere intersection is an axis-aligned box. It contains all points of the sub-domain Ω , which can lie in the zero set of F . Therefore it can be used as a reduced bounding region of the zero set of F in the sub-domain Ω .

5.3 Convergence Rate for Single Roots

We bound the zero sets of polynomials with the help of quadratic polynomial equations. Therefore we expect that the rate of convergence of the sequence of bounding regions is equal to three. These expectation is confirmed in Theorem 5.14 in the end of this section. If we assume, that the polynomials F possess a single root \mathbf{q} in a domain, then the gradient vectors of the polynomials are linearly independent in the point \mathbf{q} . Thus the implicitly defined hyper-surfaces, defined by the zero set of the polynomials, intersect each other transversely at the root. Moreover there exists a domain Ω_0 around the root \mathbf{q} , such that for any point $\mathbf{x} \in \Omega_0$ holds

$$\det(J(F)(\mathbf{x})) \neq 0. \quad (5.7)$$

Namely the gradient vectors $\nabla f_1(\mathbf{x}), \nabla f_2(\mathbf{x}), \dots, \nabla f_n(\mathbf{x})$ are linearly independent for all $\mathbf{x} \in \Omega_0$. Therefore we suppose that any point of the initial domain Ω_0 fulfills (5.7).

The fat sphere generation algorithm computes first a set of modified polynomials \hat{F} . Each point of Ω_0 fulfills (5.7), so the gradient vectors $\nabla f_i(\mathbf{x})$ do not vanish. In Section 4.5.2 we have shown, that each modified polynomial has a positive lower bound on the gradient length, if we compute in a sufficiently small sub-domain of Ω_0 . Therefore also the quadratic Taylor expansions of the modified polynomials are non-zero polynomials. The following lemma shows, that the gradient vectors of the modified polynomials are linearly independent in a sufficiently small sub-domain of Ω_0 .

Lemma 5.9. *Suppose that the gradient vectors $\nabla f_1(\mathbf{x}), \nabla f_2(\mathbf{x}), \dots, \nabla f_n(\mathbf{x})$ of the polynomials $f_i \in F$ are linearly independent for all $\mathbf{x} \in \Omega_0$. Consider a sub-domain $\Omega \subseteq \Omega_0$, which has a diameter $\delta_\Omega < \varepsilon$. We compute the set of modified polynomials \hat{F} in the sub-domain Ω for the arbitrary but fixed vectors of constants \mathbf{u}_i , which are linearly independent. If ε is sufficiently small, then for all $\mathbf{x} \in \Omega$*

$$\det(J(\hat{F})(\mathbf{x})) \neq 0.$$

Proof. The gradient vectors of f_i are linearly independent in any point of Ω_0 , therefore there exists a constant $K > 0$, such that all $\mathbf{x} \in \Omega_0$ satisfy

$$|\det(J(F)(\mathbf{x}))| \geq K > 0.$$

We compute the set of polynomials $\hat{F}_{\mathbf{c}}$ with special Hessian in a certain point $\mathbf{c} \in \Omega_0$ for the fixed vectors of constants \mathbf{u}_i . Then the gradient vectors of $\hat{f}_i \in \hat{F}_{\mathbf{c}}$ in the point \mathbf{c} can be expressed as

$$\nabla \hat{f}_i(\mathbf{c}) = \sum_{j=1}^n u_i^j \nabla f_i(\mathbf{c}).$$

The vectors of constants \mathbf{u}_i define the matrix $\mathbf{U} = (\mathbf{u}_1, \dots, \mathbf{u}_n)$. Since the vectors \mathbf{u}_i are linearly independent, the determinant of \mathbf{U} is a positive constant U

$$|\det(\mathbf{U})| = U > 0.$$

Therefore the determinant of the Jacobian of $\hat{F}_{\mathbf{c}}$ in the point \mathbf{c} satisfies

$$\left| \det(J(\hat{F}_{\mathbf{c}})(\mathbf{c})) \right| = |\det(\mathbf{U}^T \cdot J(F)(\mathbf{c}))| = |\det(\mathbf{U})| \cdot |\det(J(F)(\mathbf{c}))| \geq UK > 0.$$

Suppose that $\Omega \subseteq \Omega_0$ is a sub-domain with the center point \mathbf{c} . The set of new polynomials computed in a point \mathbf{c} is $\hat{F}_{\mathbf{c}}$. Then there exists $\varepsilon_{\mathbf{c}} > 0$, such that if the diameter δ_{Ω} of the sub-domain Ω is smaller than $\varepsilon_{\mathbf{c}}$, for all $\mathbf{x} \in \Omega$

$$\left| \det(J(\hat{F}_{\mathbf{c}})(\mathbf{x})) \right| > 0.$$

In Lemma 4.19 we have shown, that for fixed vectors of constants \mathbf{u}_i the system of polynomials $\hat{F}_{\mathbf{c}}$ depends continuously on the point \mathbf{c} . Thus there exists a general bound $\varepsilon > 0$, such that for any sub-domain $\Omega \subseteq \Omega_0$, which has the diameter $\delta_{\Omega} < \varepsilon$, any $\mathbf{x} \in \Omega$ satisfies

$$\left| \det(J(\hat{F})(\mathbf{x})) \right| > 0,$$

where \hat{F} is the set of polynomials with special Hessian in the center of the sub-domain Ω . \square

Corollary 5.10. *The median spheres are the zero set of the quadratic Taylor expansions of \hat{f}_i about the center of the sub-domain Ω*

$$p_i = T_{\mathbf{c}}^2(\hat{f}_i)(\mathbf{x}).$$

If the diameter of Ω is sufficiently small, then for all $\mathbf{x} \in \Omega$

$$\det(J(p_1, \dots, p_n)(\mathbf{x})) \neq 0.$$

Proof. The construction of p_i implies that

$$\left| \det(J(\hat{f}_1, \dots, \hat{f}_n)(\mathbf{c})) \right| = \left| \det(J(p_1, \dots, p_n)(\mathbf{c})) \right|.$$

The polynomials \hat{f}_i depend continuously on the point \mathbf{c} , so as their quadratic Taylor expansions p_i . According to the proof of Lemma 5.9 there exists a general bound $\varepsilon > 0$, such that if the diameter δ_{Ω} of the sub-domain Ω is smaller than ε , then any $\mathbf{x} \in \Omega$ satisfies

$$\left| \det(J(p_1, \dots, p_n)(\mathbf{x})) \right| > 0.$$

\square

We computed the fat sphere boundaries as concentric spheres to the median sphere $p_i = 0$. These spheres are defined by the equations

$$p_i = \pm \varepsilon_i,$$

where ε_i is computed as

$$\varepsilon_i = \left\| \hat{f}_i - p_i \right\|_{\text{BB}}^{\Omega}.$$

Lemma 5.11. *We compute a polynomial \hat{f}_i with special Hessian in the center point of the sub-domain Ω . Let ε_i denote the bound*

$$\varepsilon_i = \left\| \hat{f}_i - T_{\mathbf{c}}^2(\hat{f}_i) \right\|_{\text{BB}}^{\Omega}.$$

Then it satisfies

$$\varepsilon_i \leq C \text{diam}(\Omega)^3.$$

Proof. The sub-domain Ω is an axis-aligned box. Since all norms are equivalent on finite dimensional vector spaces, there exists a constant C_1 , such that

$$\varepsilon_i = \left\| \hat{f}_i - p_i \right\|_{\text{BB}}^{\Omega} \leq C_1 \left\| \hat{f}_i - p_i \right\|_{\infty}^{\Omega},$$

and C_1 does not depend on Ω . If the center point of Ω is denoted by \mathbf{c} , then

$$\left\| \hat{f}_i - p_i \right\|_{\infty}^{\Omega} = \left\| \hat{f}_i - T_{\mathbf{c}}^2(\hat{f}_i) \right\|_{\infty}^{\Omega} < \frac{1}{6} \underbrace{\max_{\mathbf{v} \in S^1, \mathbf{x} \in \Omega} \left| \frac{d^3 \hat{f}_i(\mathbf{x})}{d\mathbf{v}^3} \right|}_{*} \text{diam}(\Omega)^3.$$

Recall from Lemma 4.19 that \hat{f}_i depends continuously on the points of the computational domain Ω_0 for each parameter vector of \mathbf{u} , where $u^j \neq 0$. Thus for all Ω a global upper bound C_2 can be given for (*). Therefore we observe, that

$$\varepsilon_i \leq \frac{1}{6} C_1 C_2 \text{diam}(\Omega)^3 \leq C \text{diam}(\Omega)^3.$$

□

In order to measure the longest diameter of the intersection of fat spheres we give a general lower bound on the gradient of a certain function. This result is similar to the one in Lemma 4.16.

Lemma 5.12. *Consider the function and $h = \sqrt{\sum_{i=1}^n q_i^2}$ defined by the polynomials $q_i \in Q$. We assume that the Jacobian matrix is not singular in any $\mathbf{x} \in \Omega$*

$$\det(J(Q)(\mathbf{x})) \neq 0.$$

For all $\mathbf{x} \in \Omega$, which do not satisfy $q_i(\mathbf{x}) = 0$ for $i = 1, \dots, n$, there exists a positive constant L_{Ω} such that

$$\left\| \nabla h(\mathbf{x}) \right\|^2 \geq L_{\Omega} > 0.$$

Proof. Since

$$\nabla h(\mathbf{x}) = \frac{\sum_{i=1}^n q_i \nabla q_i}{\sqrt{\sum_{i=1}^n q_i^2}},$$

we obtain

$$\begin{aligned} \|\nabla h(\mathbf{x})\|^2 &= \left\langle \frac{\sum_{i=1}^n q_i \nabla q_i}{\sqrt{\sum_{i=1}^n q_i^2}}, \frac{\sum_{i=1}^n q_i \nabla q_i}{\sqrt{\sum_{i=1}^n q_i^2}} \right\rangle = \\ &= \frac{\mathbf{q}(\mathbf{x})}{\|\mathbf{q}(\mathbf{x})\|}^T J(Q)(\mathbf{x}) J(Q)(\mathbf{x})^T \frac{\mathbf{q}(\mathbf{x})}{\|\mathbf{q}(\mathbf{x})\|} \geq \min_{\|\mathbf{v}\|=1} \mathbf{v}^T \text{Gram}(\nabla q_1(\mathbf{x}), \dots, \nabla q_n(\mathbf{x})) \mathbf{v}, \end{aligned}$$

where $\mathbf{q}(\mathbf{x})^T = (q_1(\mathbf{x}), \dots, q_n(\mathbf{x}))$. We assumed, that $J(Q)(\mathbf{x})$ is not singular, therefore $\text{Gram}(\nabla q_1(\mathbf{x}), \dots, \nabla q_n(\mathbf{x}))$ is also non-singular. Moreover it is symmetric. Thus for all $\mathbf{x} \in \Omega$

$$\|\nabla h(\mathbf{x})\|^2 \geq \lambda(\mathbf{x}) > 0,$$

where $\lambda(\mathbf{x})$ is the minimal eigenvalue of the Gram matrix. Since the Gram matrix is not singular, and it depends continuously on the points of Ω , there exists a positive lower bound L_Ω depends on Ω , such that

$$\lambda(\mathbf{x}) \geq L_\Omega > 0.$$

□

Lemma 5.13. *Consider a domain Ω_0 . In each point $\mathbf{c} \in \Omega_0$ is given a set of polynomials $Q_{\mathbf{c}}$. Each polynomial $p_i \in Q_{\mathbf{c}}$ depends continuously on the point \mathbf{c} . We assume that for all \mathbf{c} there exists a sub-domain $\Omega_{\mathbf{c}} \subseteq \Omega_0$, where \mathbf{c} is the center point of the sub-domain and the Jacobian matrix of the polynomial system $Q_{\mathbf{c}}$ is not singular in any $\mathbf{x} \in \Omega_{\mathbf{c}}$. Consider the function*

$$h_{\mathbf{c}} = \sqrt{\sum_{i=1}^n q_i^2}$$

defined by the polynomials $q_i \in Q_{\mathbf{c}}$. For all \mathbf{x} from the sub-domain $\Omega_{\mathbf{c}}$, which do not satisfy $q_i(\mathbf{x}) = 0$ for $i = 1, \dots, n$, there exists a general positive constant L such that

$$\|\nabla h_{\mathbf{c}}(\mathbf{x})\|^2 \geq L > 0.$$

Proof. Each polynomial $q_i \in Q_{\mathbf{c}}$ depends continuously on the choice of the point \mathbf{c} . According to Lemma 5.12 there exists a lower bound of $\|\nabla h_{\mathbf{c}}(\mathbf{x})\|^2$ for all $\mathbf{x} \in \Omega_{\mathbf{c}}$, which bounds the minimal eigenvalue of the Gram matrix of $q_i \in Q_{\mathbf{c}}$. Therefore for all $\Omega_{\mathbf{c}}$, where $\det(J(Q_{\mathbf{c}})(\mathbf{x})) \neq 0$, there exists a general positive lower bound L , such that any $\mathbf{x} \in \Omega_{\mathbf{c}}$ satisfies

$$\|\nabla h(\mathbf{x})\|^2 \geq L > 0,$$

if \mathbf{x} does not satisfy $q_i(\mathbf{x}) = 0$ for all $q_i \in Q_{\mathbf{c}}$. □

Theorem 5.14. *Suppose that the gradient vectors of the polynomials $f_i \in F$ are linearly independent for all points $\mathbf{x} \in \Omega_0$. Consider a sub-domain $\Omega \subseteq \Omega_0$, which is sufficiently small and contains a single root \mathbf{q} of the polynomials f_i . We compute the set of polynomials \hat{F} with special Hessian in the center point of the domain Ω for the arbitrary but fixed vectors of*

constants \mathbf{u}_i , which are linearly independent. If we apply the domain shrinking step of the fat sphere generation algorithm on the sub-domain Ω , then there exists a constant C , such that the generated bounding region Ω^* satisfies

$$\text{diam}(\Omega^*) \leq C \text{diam}(\Omega)^3.$$

Proof. Suppose that Ω is a sub-domain of Ω_0 , which contains a single root \mathbf{q} . We compute the set of polynomials \hat{F} with special Hessian in the center point \mathbf{c} of Ω . The median spheres are defined as the zero set of the quadratic Taylor expansion of the polynomials $\hat{f}_i \in \hat{F}$ about the point \mathbf{c}

$$p_i(\mathbf{x}) = T_{\mathbf{c}}^2(\hat{f}_i)(\mathbf{x}).$$

We denote with ε_i the distance bound of \hat{f}_i and p_i computed in the BB-norm in Ω . A fat sphere is defined by the point set

$$\mathcal{F}_i(p_i, \varepsilon_i, \Omega) = \{\mathbf{x} : |p_i(\mathbf{x})| \leq \varepsilon_i\} \cap \Omega.$$

Let us denote the fat sphere intersection as $\mathcal{I} = \bigcap_{i=1}^n \mathcal{F}_i$. Each fat sphere bounds the hypersurface $\hat{f}_i = 0$, thus the single root $\mathbf{q} \in \Omega$ is contained in the fat sphere intersection

$$\mathbf{q} \in \mathcal{I} \cap \Omega.$$

We define the function

$$h(\mathbf{x}) = \sqrt{\sum_{i=1}^n q_i^2},$$

where $q_i(\mathbf{x}) = p_i(\mathbf{x}) - p_i(\mathbf{q})$. We consider the integral curves defined by the vector field $-\nabla h / \|\nabla h\|$ in Ω . If Ω has a sufficiently small diameter, according to Corollary 5.10 all $\mathbf{x} \in \Omega$ satisfy

$$\det(J(p_1, \dots, p_n)(\mathbf{x})) \neq 0.$$

Since $\nabla p_i(\mathbf{x}) = \nabla q_i(\mathbf{x})$, for all $\mathbf{x} \in \Omega$

$$\det(J(q_1, \dots, q_n)(\mathbf{x})) \neq 0. \quad (5.8)$$

Together with Lemma 5.12 this implies that the integral curves are regular in the inner points of $\Omega \setminus \{\mathbf{q}\}$.

Suppose that \mathbf{x} is an arbitrary point of the fat sphere intersection \mathcal{I} computed in a sufficiently small domain Ω . Such a point $\mathbf{x} \in \mathcal{I} \cap \Omega$ fulfills for all $i = 1, \dots, n$

$$|p_i(\mathbf{x})| \leq \varepsilon_i.$$

We consider the integral curve $\mathbf{u}(s)$ with the starting point $\mathbf{u}(0) = \mathbf{x} \in \mathcal{I}$, which is regular on $\Omega \setminus \{\mathbf{q}\}$. We assume, that the curve is parametrized by arc length. Such an the integral curve has a unique limit, if the computational domain is sufficiently small. Since $h(\mathbf{x}) \geq 0$ and the tangent vectors of the curve $\mathbf{u}(s)$ always point to the direction of steepest decent on h , there exists a parameter value s^* such that for $s < s^*$

$$\lim_{s \rightarrow s^*} \mathbf{u}(s) = \mathbf{q}.$$

According to the mean value theorem there exists $\xi \in (0, s^*)$ such that

$$\frac{h(\mathbf{u}(s^*)) - h(\mathbf{u}(0))}{s^*} = \nabla h(\mathbf{u}(\xi)) \cdot \dot{\mathbf{u}}(\xi) = - \|\nabla h(\mathbf{u}(\xi))\|.$$

Since $h(\mathbf{u}(s^*)) = 0$

$$s^* = \frac{h(\mathbf{u}(0))}{\|\nabla h(\mathbf{u}(\xi))\|} = \frac{h(\mathbf{x})}{\|\nabla h(\mathbf{u}(\xi))\|} \leq \sqrt{\frac{2 \sum_{i=1}^n \varepsilon_i^2}{L_\Omega}}.$$

We supposed that $\mathbf{u}(s)$ is arc length parametrized, therefore $\mathbf{x} \in \mathcal{I}$ satisfies

$$\|\mathbf{x} - \mathbf{q}\| = \|\mathbf{u}(0) - \mathbf{u}(s^*)\| \leq \sqrt{\frac{2 \sum_{i=1}^n \varepsilon_i^2}{L_\Omega}}.$$

Thus any point of \mathcal{I} is closer to \mathbf{q} than $\sqrt{\frac{2 \sum_{i=1}^n \varepsilon_i^2}{L_\Omega}}$. So the min-max box $\Omega^* \subset \Omega$, which contains \mathcal{I} , has a diameter

$$\text{diam}(\Omega^*) \leq 2\sqrt{\frac{2n \sum_{i=1}^n \varepsilon_i^2}{L_\Omega}}.$$

In Lemma 4.19 we have shown, that the system of polynomials \hat{F} depends continuously on the choice of the domain Ω . Therefore also each p_i and q_i depend continuously on the choice of Ω . The lower bound L_Ω of $\|\nabla h(\mathbf{x})\|^2$ bounds the minimal eigenvalue of the Gram matrix of q_i . According to Lemma 5.12 there exists a general positive lower bound L , such that any $\mathbf{x} \in \Omega$ satisfies

$$\|\nabla h(\mathbf{x})\|^2 \geq L > 0.$$

We have also shown in Lemma 5.11 that there exists a constant D , which does not depend on the choice of Ω , such that

$$\varepsilon_i \leq D \text{diam}(\Omega)^3.$$

Therefore the diameter of the min-max box Ω^* satisfies

$$\text{diam}(\Omega^*) \leq 2\sqrt{\frac{2n \sum_{i=1}^n \varepsilon_i^2}{L}} \leq \frac{2\sqrt{2}Dn}{\sqrt{L}} \text{diam}(\Omega)^3 = C \text{diam}(\Omega)^3,$$

where C does not depend on the choice of Ω . □

5.4 Iterative Domain Reduction Algorithm

In this section we present a subdivision algorithm combined with the local domain reduction strategy Algorithm 8. It is an iterative domain reduction, which reduces the bounding regions either by subdivision or by bounding fat sphere intersection.

Algorithm 9 $\text{GenerateBoundingBoxes}(F, \Omega, \varepsilon)$

```

1:  $\mathcal{A} \leftarrow \text{DomainReduction}(F, \Omega)$  {domain reduction}
2: if  $2 \cdot \text{diam}(\mathcal{A}) \leq \text{diam}(\Omega)$  then
3:   if  $\text{diam}(\mathcal{A}) > \varepsilon$  then
4:      $\text{GenerateBoundingBoxes}(F, \mathcal{A}, \varepsilon)$  {recursive call}
5:   else
6:      $\mathcal{B} = \mathcal{B} \cup \mathcal{A}$ 
7:   end if
8: else
9:   if diameter of  $\Omega > \varepsilon$  then
10:    subdivide the domain  $\Omega$  to  $\Omega_i$  {subdivision}
11:     $\text{GenerateBoundingBoxes}(F, \Omega_i, \varepsilon)$  {recursive call}
12:   else
13:      $\mathcal{B} = \mathcal{B} \cup \Omega$ 
14:   end if
15: end if
16: return  $\mathcal{B}$ 

```

5.4.1 Algorithm

The global root approximation algorithm (Algorithm 9) is an iterative domain reduction, which bounds the roots of a multivariate polynomial system F within a prescribed tolerance bound ε . The algorithm computes a set of axis-aligned boxes with the help of hierarchical subdivision and fat sphere intersection. Each root of the system is approximated via a nested sequence of domains, which have decreasing diameters. The algorithm reduces the domains, until each list of nested domains has an element with sufficiently small diameter. Then the algorithm returns the last element of the lists.

Each domain is analyzed, until it is detected as an empty region or it has a sufficiently small diameter. We detect empty domains via the convex hull property (see in Observation 3.2). A sub-domain is also empty, if the local algorithm generates fat spheres, which do not intersect. Then the algorithm does not analyze these domains any further. Nevertheless, it can happen that a domain without root is computed with small diameter, but it is not detected as an empty region. Thus the output can also contain empty sub-domains.

It is also important to separate the real roots of polynomials to different bounding domains. In some cases we can certify whether a domain in the output contains only one single root, although this is not always possible. If two real roots have smaller distance than the tolerance ε , they may have common bounding region in the output of the algorithm. Therefore clearly the number of bounding regions in the output is not necessarily equal to the number of real roots of the polynomial system.

Choice of Polynomial Combinations and Constants. As we described already in Section 5.2.2 we compute polynomials with special Hessian as the combination of $n - 1$ or n different polynomials from the original set of polynomials F . If we only combine $n - 1$ polynomials from F , then we can choose the set of polynomials in the construction of each new polynomial \hat{f}_i differently. However, we approximate the zero set of all polynomials in F ,

so we have to use all polynomials at least once in the computation of \hat{f}_i . Otherwise we only approximate the solution set of certain subset of F . This problem does not appear if we use all the polynomials in F to compute \hat{f}_i . According to our experiments, this strategy reduces the size of the bounding domains faster, although we have to handle larger linear systems to find polynomials with special Hessian.

In order to compute each new polynomial \hat{f}_i , we have to choose an arbitrary but fixed vector of constants \mathbf{u}_i . These vectors of constants are chosen a priori and they are kept fixed during each subdivision and domain reduction step. We have seen in Lemma 5.9, that the choice of the vectors \mathbf{u}_i is important. These vectors have to be linearly independent in order to provide the third order convergence of the bounding regions for single roots.

5.4.2 Examples

We present here several examples, which show the behavior of the root-finding algorithm `GenerateBoundingBoxes` for polynomial systems in two or three variables.

Intersection Points of Implicitly Defined Planar Curves

Example 5.15. First we present a two-dimensional example to show the behavior of the root-finding algorithm. The two implicitly defined curves are defined by the polynomials with bi-degree (9,8) and (6,9). They are represented in the unit box. The intersection points of the curves are approximated within the tolerance $\varepsilon = 10^{-4}$. The curves have five intersection points in the domain. After three subdivision steps all roots are separated into different sub-domains. Then four or five domain reduction steps are made in order to achieve the prescribed accuracy around each intersection point. The output is represented in Fig.5.4. In the left one can see the domains, generated during the domain reduction steps (either with subdivision or with the help of fat arc intersection). They are shown in different shades of gray. In the right the center point of each bounding domain from the output is marked as a red dot.

Example 5.16. This example appears in the paper of Elber et al. [17]. They present a strategy to approximate the intersection points of implicitly defined curves. Their algorithm purge away empty domains and identify domains with single solution more efficiently than the subdivision method. We compare here the fat arc generation with the simple subdivision via this example.

The two bi-cubic curves are the reflection of each other along the $x = y$ line (see Fig.5.5). They intersect each other along the reflection line in five different points and also in two other points in the domain. We represented the curves in the unit square $[0, 1]^2$, and approximate the roots using different tolerances. In Table 5.2 we compare the total number of bounding domains in the output. The fat arc generation method returns for small tolerance a number of bounding domains, which is equal to the number of the intersection points, while the subdivision method returns a large number of bounding boxes. The algorithm, which generates fat arcs, eliminates efficiently the empty sub-domains. Moreover it speeds up the convergence and uses less subdivision steps. In Fig.5.5 we show the output of the fat arc generation and the subdivision algorithms. The intersection points of the curves are marked by black crosses, while the generated bounding domains are represented by their center points marked by red dots. In the first of row of the figure we represent the outputs of the fat arc generation, while in the second row the outputs of simple subdivision method are shown.

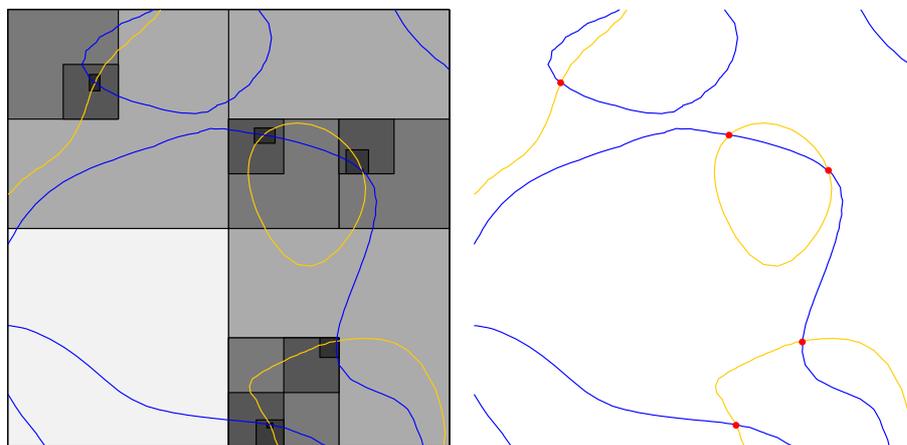


Figure 5.4: Approximation of the intersection points of implicitly defined curves given by the zero level set of polynomials with bi-degree $(9,8)$ and $(6,9)$. In the left: domains generated during the domain reduction steps, in the right: the center points of the bounding domains are marked as red dots.

Table 5.2: Approximating intersection of implicitly defined curves. The number of used bounding regions for the seven intersection points of the curves in Fig. 5.5.

Algorithm	$\varepsilon = 0.1$	$\varepsilon = 0.01$	$\varepsilon = 0.001$	$\varepsilon = 0.0001$
Fat Arcs	15	14	7	7
Subdivision	22	40	68	71

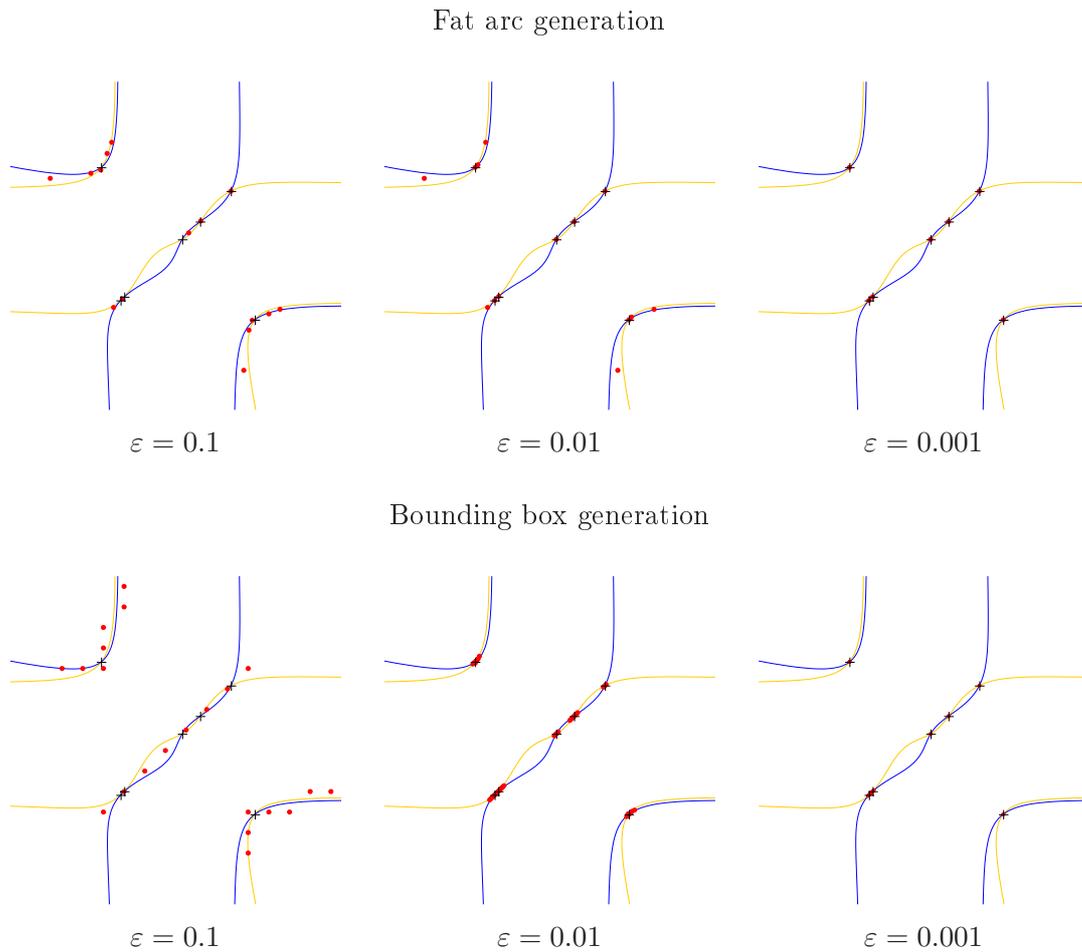


Figure 5.5: Comparison of root approximation, computed with fat arc generation and subdivision. In the first row we present the outputs of the fat arc generation, while in the second row the outputs of simple subdivision method is shown for different tolerances. The intersection points of the curves are marked by black crosses, while the generated bounding domains are represented by their center points marked by red dots.

Example 5.17. Example 5.16 indicates that the fat arc generation algorithm separates the different roots well. Therefore we present an example, where the roots of the polynomials are changing from two single roots to one double root with the translation of one of the curves. The curves are represented by the zero set of

$$f(x, y) = -(0.95 + 10^{-k}) + 0.2x + 0.4y + x^2 + y^2,$$

$$g(x, y) = -0.48 + 0.2x + 0.1y + xy.$$

We set the tolerance to $\varepsilon = 10^{-8}$ and compute the approximation in the unit box for the value of $k = 2, 5$ and $k = 10$. The distance of the exact roots (denoted by δ) is given in the first row of Table 5.3 for each value of k . On the top of the table we show the results obtained by fat arc generation, while in the bottom the outputs of simple subdivision method are shown. In each column the diameters of the bounding domains are given, which were generated step by step during the approximation methods. The bounding regions are reduced until their diameter is smaller than the tolerance or at most up to eight steps. In the last column we show the reduction of the bounding regions for one double root. Finally we present a figure, where the bounding domains are shown in the case of $k = 2, 5$ and for the double root. The shrinking regions are represented in different shades of gray (see Fig.5.6).

Intersection Points of Implicitly Defined Algebraic Surfaces

Example 5.18. This example corresponds to the two-dimensional one in Example 5.16. It compares simple subdivision method with the fat sphere intersection in a three-dimensional root-finding problem. The problem is given by the equation system

$$\begin{aligned} 0.4(x^2 + y^2 + z^2) - 0.88(x + y + z) - 4xyz + 1.452 &= 0, \\ 104(x^3 + y^3 + z^3) - 141(x^2 + y^2 + z^2) + 61.875(x + y + z) - 27.978125 &= 0, \\ x^2 + y^2 + z^2 + 0.4(x + y + z) - 1.58 &= 0, \end{aligned}$$

with respect to the unit cube. The system has six different roots in the computational domain. These roots are situated pairwise relatively close to each other. If we approximate such roots with simple subdivision, usually the root separation process is slow, and it uses high number of bounding domains in the output. According to our experiments in Example 5.16, we expect that the fat sphere generation method uses less subdivision steps and few bounding regions in the output. In Table 5.4 we compare the total number of computed bounding domains. The fat sphere generation method returns for small tolerance a number of bounding domains, which is equal to the number of the roots. Moreover it uses less subdivision steps (see in columns #1). While the subdivision method returns a large number of bounding boxes. In Fig.5.7 we show the output of the fat sphere generation and the subdivision algorithm. The generated bounding domains are represented by their center point marked by red dots. In the first row we present the outputs of the fat sphere generation, while in the second row the outputs of simple subdivision method are shown.

Example 5.19. We can approximate the ordinary singular points of an implicitly defined surface with the help of the fat sphere generation. In this example we present two different

Table 5.3: Approximating intersection points of implicitly defined curves, which are translated in three steps ($k = 2, 5, 10$) from two single roots to one double root. We represent here the diameters of bounding boxes in each step of the bounding region generation. In the cases of two single roots we marked the level of domain reduction, where the algorithms separate the roots. The distance of the two roots is given in the first row of the table (δ).

	$k = 2$ ($\delta = 1.41 \cdot 10^{-1}$)	$k = 5$ ($\delta = 4.47 \cdot 10^{-3}$)	$k = 10$ ($\delta = 1.41 \cdot 10^{-5}$)	Double root
Fat arc generation	root separation		0.707	0.707
	0.707	0.707	$9.65 \cdot 10^{-2}$	0.164
	0.128	0.151	$1.55 \cdot 10^{-2}$	$2.35 \cdot 10^{-2}$
	$1.85 \cdot 10^{-3}$	$3.00 \cdot 10^{-3}$	$4.57 \cdot 10^{-3}$	$1.28 \cdot 10^{-3}$
	$5.49 \cdot 10^{-9}$	$2.35 \cdot 10^{-8}$		$1.62 \cdot 10^{-5}$
	root separation			$1.97 \cdot 10^{-5}$
Subdivision	root separation		0.707	0.707
	0.707	0.707	0.353	0.353
	0.353	0.353	0.176	0.176
	0.176	0.176	$8.88 \cdot 10^{-2}$	$8.88 \cdot 10^{-2}$
	$8.88 \cdot 10^{-2}$	$8.88 \cdot 10^{-2}$	$4.41 \cdot 10^{-2}$	$4.41 \cdot 10^{-2}$
	$4.41 \cdot 10^{-2}$	$4.41 \cdot 10^{-2}$		$2.20 \cdot 10^{-2}$
root separation			$2.20 \cdot 10^{-2}$	$1.10 \cdot 10^{-2}$
		$2.20 \cdot 10^{-2}$	$2.20 \cdot 10^{-2}$	$1.10 \cdot 10^{-2}$
		$1.10 \cdot 10^{-2}$	$1.10 \cdot 10^{-2}$	$5.52 \cdot 10^{-3}$
			$5.52 \cdot 10^{-3}$	$5.52 \cdot 10^{-3}$

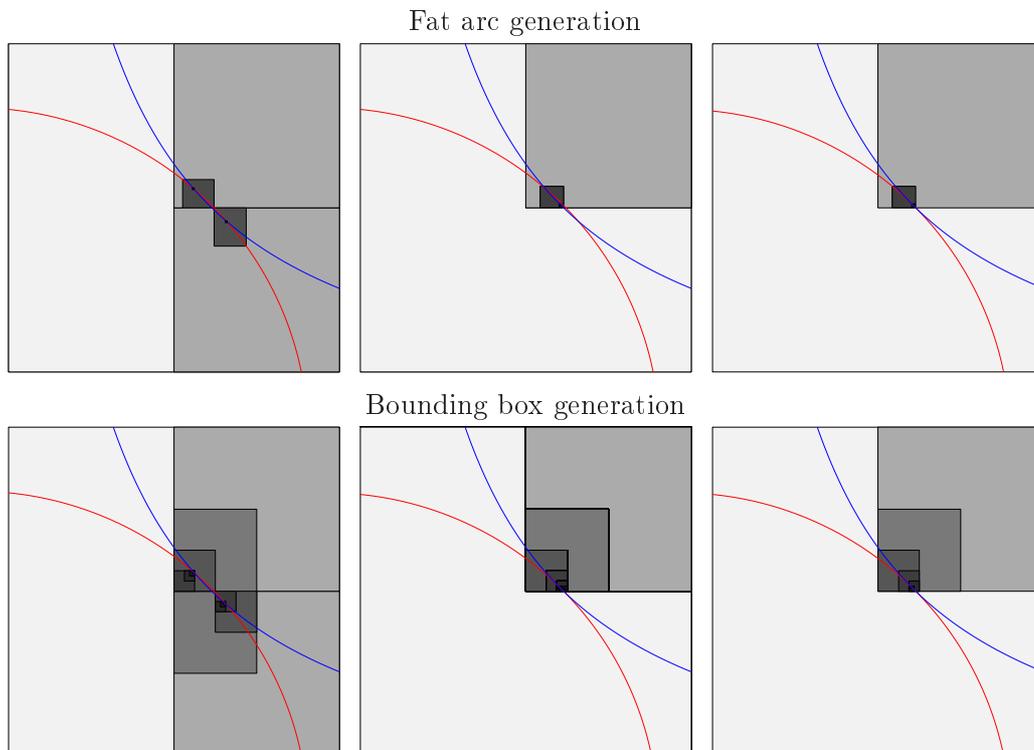


Figure 5.6: Reduction of bounding boxes in the case of $k = 2, 5$ and for the double root. In the first row we used the fat arc generation method, while in the second row simple subdivision.

Table 5.4: Approximating intersection of implicitly defined surfaces. We present the number of used bounding regions and the number of domain reduction steps (denoted by $\#l$). This number shows the maximal depth of the domain reduction or subdivision tree, which is traversed by the algorithm during the root approximation.

Algorithm	$\varepsilon = 0.1$	$\#l$	$\varepsilon = 0.01$	$\#l$	$\varepsilon = 0.001$	$\#l$
Fat Spheres	42	2	6	5	6	5
Subdivision	78	5	78	8	66	11

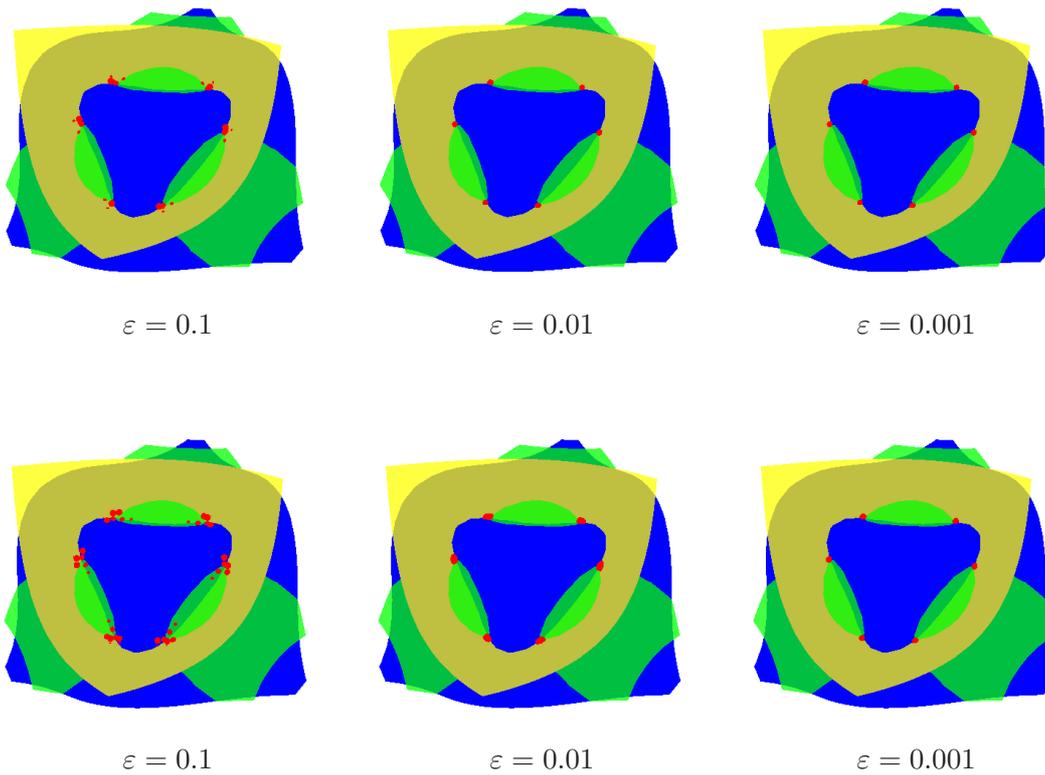


Figure 5.7: Comparison of approximate roots computed with fat sphere generation and subdivision. In the first row we represent the outputs of the fat sphere generation, while in the second row the outputs of simple subdivision method are shown for different tolerances.

algebraic surfaces given by an implicit equation $f(x, y, z) = 0$ with ordinary singularities. These singularities can be found by computing the zero set of the partial derivatives

$$f_x = 0, \quad f_y = 0, \quad f_z = 0.$$

A singular point of the surface also satisfy the equation of the surface.

In Fig.5.8 the dots mark the approximate solution points of the system of partial derivatives. The red ones are the solutions, which lie close to the implicitly defined surfaces $f = 0$. The first surface in the figure is called Cayley-cubic. It has four ordinary singularities, which are computed in the unit cube as the solution of the system

$$\begin{aligned} -250xz + 175x + 125.5z - 87.85 &= 0, \\ 250yz - 75y - 124.95z + 37.485 &= 0, \\ -125x^2 + 125y^2 + 125.5x - 124.95y + 50z - 25.275495 &= 0. \end{aligned}$$

The second surface is the Ding-dong surface, which has one ordinary singularity. It is computed in the unit cube as the solution of the system

$$\begin{aligned} 18x - 9.06 &= 0, \\ 18y - 8.994 &= 0, \\ 81z^2 - 100.08z + 29.9136 &= 0. \end{aligned}$$

The tolerance during the computations was set to 0.01 in both examples.

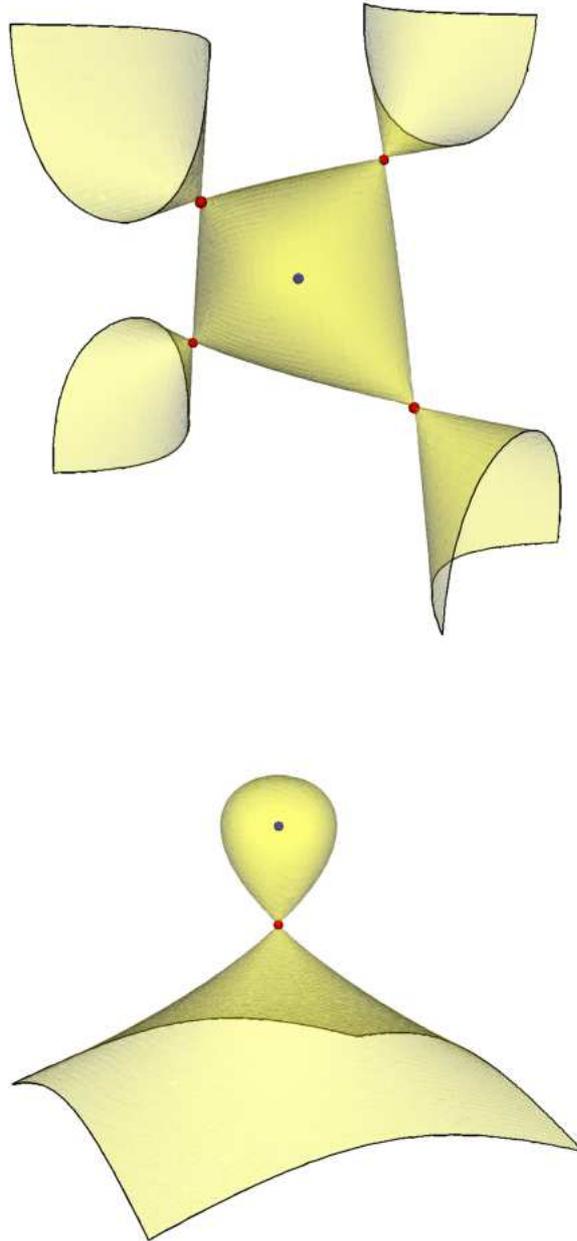


Figure 5.8: Ordinary singularities on implicitly defined surfaces.

Chapter 6

Conclusion

We presented a new family of algorithms to approximate implicitly defined algebraic curves and real roots of polynomial systems. These methods are based on the geometrical properties of polynomial systems.

In order to generate local bounding regions, we presented fat arcs, which are the tubular neighborhood of circular arcs. First we presented several techniques to generate these bounding regions for planar curves. One of these techniques computes polynomials with modified Taylor expansion. This method has several advantageous properties, therefore we generalized it to approximate algebraic curves embedded into the three- and n -dimensional space. The fat arcs, generated by this technique, have a close connection to the osculating circle of the algebraic curve. The cubic convergence order is confirmed for these bounding regions in the thesis. The local fat arc generation combined with iterative subdivision leads to a hybrid algorithm, which generates bounding regions to implicitly defined algebraic curves. We presented several examples and applications of the algorithm to approximate implicitly defined algebraic curves in two- and three-dimensional space.

Based on the definition of fat arcs we introduced fat spheres as bounding regions for algebraic objects. These regions can also be generated using polynomials with modified Taylor expansion. Intersecting these bounding regions leads us to a local domain reduction strategy, which bounds the intersection points of algebraic surfaces. We combined this strategy with iterative subdivision in order to approximate real roots of multivariate polynomial systems. This hybrid algorithm generates sequences of bounding regions, which converge with order three to the single roots of a multivariate polynomial system.

The structure of these algorithms carries two main messages. First of all, that analyzing geometric properties of algebraic objects leads to stable techniques on real algebraic set approximation. This stability is certified by the Bernstein-Bézier polynomials. In addition fat arc and sphere computations are advantageous. Although they require extra computational time compared with other bounding primitives, the generated bounding regions converge faster. Computing with quadratic bounding regions provides faster termination of the algorithm and reduces the depth of the subdivision tree.

Curriculum Vitae

Personal Information:

Name: Szilvia Béla
Date and Place of Birth: 24. September 1983., Tata, Hungary
Office: JKU, Doctoral Program “ Computational Mathematics” ,
Altenberger Str.69, A-4040 Linz, Austria,
Hochschulfondsgebäude, room 233
E-mail address: szilvia.bela@dk-compmath.jku.at
Phone: +43 (0) 732-2468-7175
Fax: +43 (0) 732-2468-7179

Education:

2002 High School Diploma, Veszprém, Hungary
2002-2007 Studies in Mathematics at Budapest University of Technology and Economics (BUTE), Budapest
2007 - Ph.D. studies in Mathematics at the Department of Applied Geometry, Johannes Kepler Universität (JKU), Linz

Career history:

2005-2007 Junior research assistant at the Department of Geometry, Institute of Mathematics, BUTE
2007-2008 Research assistant in project SFB013 “ Numerical and Symbolic Methods for Implicitly Defined Spline Surfaces”, RICAM, Linz, Austria
2008- PhD Student in the Doctoral Program “Computational Mathematics”, Johannes Kepler Universität, Linz
2009. Febr.-March guest researcher in Projet GALAAD, INRIA Sophia Antipolis, France
2010 April-May guest researcher at the Department of Mathematics, DTU, Denmark

Language knowledge:

English State Accredited Language Exam (B2), Hungary, 2002.
German Intensive Summer Course, Innsbruck, Austria, August 2006.
State Accredited Language Exam (B2), Hungary, 2007.

Publications:

M. Szilvasi-Nagy, Sz. Béla, Gy. Mátyási: About the geometry of milling paths, *Annales Mathematicae et Informaticae*, 35 (2008) 135-146.

Sz. Béla and B. Jüttler, Fat arcs for implicitly defined curves, *MMCS proceedings*, LNCS 5862, 26-40, Springer 2010, see also DK Report 2009-08.

Sz. Béla and B. Jüttler, Approximating Algebraic Space Curves by Circular Arcs, In M.-L. Mazure et al., editors, *Curves and Surfaces – Avignon 2010*, LNCS Springer, submitted, see also DK Report 2010-12.

Conferences, talks:

Participant at Spring School on Geometry and Visualization, Berlin, Germany, April 2006.

Participant at Annual DAAD project workshop, Belgrade, Germany, September 2006, 2007.

About the geometry of milling paths. 7th International Conference on Applied Informatics, Eger, Hungary, January 2007.

About the geometry of milling paths. Conference on Geometry: Theory and Applications, Vorau, Austria, June 2007.

About the geometry of milling paths. 2. Internationale Symposium "Geometrisches Modellieren, Visualisieren und Bildverarbeitung", Stuttgart, Germany, June 2007.

Approximating implicitly defined curves by fat arcs. FSP Workshop, Strobl, Austria, March 2008.

Approximating implicitly defined curves by fat arcs. MMCS'08, Tønsberg, Norway, June 2008.

Approximating implicitly defined curves by fat arcs. SNSC'08, Hagenberg, Austria, July 2008.

Approximating implicitly defined curves by fat arcs. DK Kick-Off Meeting, Strobl, Austria, October 2008.

Fat arcs for implicitly defined curves. Castro Urdiales, Spain, November 2008.

Computing with Fat Arcs. DK-Statusseminar, Pichl, Austria, June 2009.

Participant at DK-Statusseminar, Strobl, Austria, October 2009.

Fat Arcs for Algebraic Space Curves. MMCS'10, Avignon, France, June 2010.

Computing with Fat Arcs and Fat Spheres. Strobl, Austria, October 2010.

Fat arcs and fat spheres for approximating algebraic curves and for solving polynomial systems. DK Seminar, Linz, Austria, January 20, 2011.

March 2011, Linz, Austria

Bibliography

- [1] M. Aigner, I. Szilagyi, J. Schicho, and B. Jüttler. Implicitization and distance bounds. In B. Mourrain, M. Elkadi, and R. Piene, editors, *Algebraic Geometry and Geometric Modeling*, pages 71–86. Springer, 2006.
- [2] Juan Gerardo Alcazar and J. Rafael Sendra. Computation of the topology of real algebraic space curves. *Journal of Symbolic Computation*, 39(6):719 – 744, 2005.
- [3] C. L. Bajaj, C. M. Hoffmann, J. E. Hopcroft, and R. E. Lynch. Tracing surface intersections. *Comput. Aided Geom. Des.*, 5(4):285–307, 1988.
- [4] D. H. Ballard. Strip trees: a hierarchical representation for curves. *Commun. ACM*, 24(5):310–321, 1981.
- [5] B. Buchberger. Gröbner-bases: An algorithmic method in polynomial ideal theory. In N.K. Bose, editor, *Multidimensional Systems Theory - Progress, Directions and Open Problems in Multidimensional Systems*, pages 184–232. Reidel Publishing Company, The Netherlands, 1985.
- [6] L. Busé, M. Elkadi, and B. Mourrain. Generalized resultants for unirational algebraic varieties. *Journal of Symbolic Computation*, 59:515–526, 2000.
- [7] L. Busé, M. Elkadi, and B. Mourrain. Residual resultant of complete intersection. *Journal of Pure and Applied Algebra*, 164:35–57, 2001.
- [8] S. Chau, M. Oberneder, A. Galligo, and B. Jüttler. Intersecting biquadratic surface patches. In R. Piene and B. Jüttler, editors, *Geometric Modeling and Algebraic Geometry*, pages 161–180. Springer, 2008.
- [9] R. E. Cline and R. J. Plemmons. l2-solutions to undetermined linear systems. *SIAM Review*, 18(1):pp. 92–106, 1976.
- [10] O. Ruatta D.N. Daouda, B. Mourrain. On the computation of the topology of a non-reduced implicit space curve. In *Proceedings of the twenty-first international symposium on symbolic and algebraic computation*, pages 47–54, 2008.
- [11] Laurent Dupont, Daniel Lazard, Sylvain Lazard, and Sylvain Petitjean. Near-optimal parameterization of the intersection of quadrics. In *SCG '03: Proceedings of the nineteenth annual symposium on Computational geometry*, pages 246–255, New York, 2003. ACM.

- [12] G. Elber and M. Kim. Geometric constraint solver using multivariate rational spline functions. In *SMA '01: Proceedings of the sixth ACM symposium on Solid modeling and applications*, pages 1–10, New York, 2001.
- [13] M. Elkadi and B. Mourrain. Symbolic-numeric tools for solving polynomial equations and applications. In I.Z. Emiris and A. Dickenstein, editors, *Algorithms and Computation in Mathematics*, volume 14, pages 125–168. Springer-Verlag, 2005.
- [14] I.Z. Emiris and E.P. Tsigaridas. Real solving of bivariate polynomial systems. In E.W. Mayr V.G. Ganzha and E.V. Vorozhtsov, editors, *Computer Algebra in Scientific Computing, LNCS*, pages 150 –161. Springer-Verlag, 2005.
- [15] Ron Goldman. Curvature formulas for implicit curves and surfaces. *Computer Aided Geometric Design*, 22:632–658, 2005.
- [16] Laureano Gonzalez-Vega and Ioana Necula. Efficient topology determination of implicitly defined algebraic plane curves. *Computer Aided Geometric Design*, 19(9):719–743, 2002.
- [17] Iddo Hanniel and Gershon Elber. Subdivision termination criteria in subdivision multivariate solvers using dual hyperplanes representations. *Comput. Aided Des.*, 39:369–378, 2007.
- [18] M. Held and J. Eibl. Biarc approximation of polygons within asymmetric tolerance bands. *Computer-Aided Design*, 37(4):357–371, 2004.
- [19] J. Hoschek and D. Lasser. *Fundamentals of Computer Aided Geometric Design*. AK Peters, Wellesley, Mass., 1993.
- [20] B. Jüttler and P. Chalmovianský. A predictor–corrector technique for the approximate parameterization of intersection curves. *Appl. Algebra Eng. Comm. Comp.*, 18:151–168, 2007.
- [21] E. Kreyszig. *Differential Geometry*. Dover, 1991.
- [22] S. Krishnan and D. Manocha. An efficient surface intersection algorithm based on lower-dimensional formulation. *ACM Trans. Graph.*, 16(1):74–106, 1997.
- [23] Kunwoo Lee. *Principles of CAD/CAM/CAE Systems*. Addison-Wesely, 1999.
- [24] T.Y. Li. Numerical solution of multivariate polynomial systems by homotopy continuation methods. *Acta Numerica*, 6:399–436, 1997.
- [25] C. Liang, B. Mourrain, and J.-P. Pavone. *Subdivision Methods for the Topology of 2d and 3d Implicit Curves*. Springer, 2008.
- [26] Qui Lin and J. G. Rokne. Approximation by fat arcs and fat biarcs. *Computer-Aided Design*, 34(13):969–979, 2002.
- [27] K. Marciniak and B. Putz. Approximation of spirals by piecewise curves of fewest circular arc segments. *Computer-Aided Design*, 16(2):87–90, 1984.

-
- [28] Alexander Morgan and Andrew Sommese. A homotopy for solving general polynomial systems that respects m-homogeneous structures. *Applied Mathematics and Computation*, 24:101–113, 1987.
- [29] B. Mourrain and J.-P. Pavone. Subdivision methods for solving polynomial equations. *Journal of Symbolic Computation*, 44(3):292–306, 2009.
- [30] N. M. Patrikalakis and T. Maekawa. *Shape interrogation for computer aided design and manufacturing*. Springer, 2002.
- [31] M. J. Pratt and A. D. Geisow. Surface/surface intersection problem. In J.A. Gregory, editor, *The Mathematics of Surfaces II*, pages 117–142. Clarendon Press, Oxford, 1986.
- [32] H. Qiu, K. Cheng, and Y. Li. Optimal circular arc interpolation for NC tool path generation in curve contour manufacturing. *Computer-Aided Design*, 29(11):751–760, 1997.
- [33] F. Rouillier. Solving zero-dimensional systems through the rational univariate representation. *Appl. Algebra Engrg. Comm. Comput.*, 9(5):433–461, 1999.
- [34] M. A. Sabin. The use of circular arcs for the interpolation of curves through empirical data points. Technical Report VTO/MS/164, British Aircraft Corporation, 1976.
- [35] T. W. Sederberg and T. Nishita. Curve intersection using Bézier clipping. *Computer-Aided Design*, 22(9):538–549, 1990.
- [36] T. W. Sederberg and S. R. Parry. Comparison of three curve intersection algorithms. *Computer-Aided Design*, 18(1):58–63, 1986.
- [37] T. W. Sederberg, S. C. White, and A. K. Zundel. Fat arcs: a bounding region with cubic convergence. *Comput. Aided Geom. Des.*, 6(3):205–218, 1989.
- [38] E. C. Sherbrooke. *Computation of the Solutions of Nonlinear Polynomial Systems*. PhD thesis, Massachusetts Institute of Technology, 1993.
- [39] E.C. Sherbrooke and N.M. Patrikalakis. Computation of the solutions of nonlinear polynomial systems. *Computer Aided Geometric Design*, 10(5):379–405, 1993.
- [40] C.H. Tu, W. Wang, B. Mourrain, and J.Y Wang. Using signature sequences to classify intersection curves of two quadrics. *Computer Aided Geometric Design*, 26:317–335, 2009.
- [41] Z. Šír, R. Feichtinger, and B. Jüttler. Approximating curves and their offsets using biarcs and Pythagorean hodograph quintics. *Comp.-Aided Design*, 38:608–618, 2006.
- [42] D. S. Walton and D. J. Meek. Approximating quadratic NURBS curves by arc splines. *Computer-Aided Design*, 25(6):371–376, 1993.
- [43] D. S. Walton and D. J. Meek. An arc spline approximation to a clothoid. *Journal of Computational and Applied Mathematics*, 170(1):59–77, 2004.

- [44] D. S. Walton and D. J. Meek. Spiral arc spline approximation to a planar spiral. *Journal of Computational and Applied Mathematics*, 170(1):59–77, 2004.
- [45] W. Wang, R. Goldman, and C.H. Tu. Enhancing Levin’s method for computing quadric surface intersections. *Computer Aided Geometric Design*, 20:401–422, 2003.
- [46] W. Wang, B. Joe, and R. Goldman. Computing quadric surface intersections based on an analysis of plane cubic curves. *Graphical Models*, 64:335–367, 2002.
- [47] G. Xu, C. L. Bajaj, and W. Xue. Regular algebraic curve segments (i) – definitions and characteristics. *Comput. Aided Geom. Des.*, 17(6):485–501, 2000.
- [48] X. Yang. Efficient circular arc interpolation based on active tolerance control. *Computer-Aided Design*, 34(13):1037–1046, 2002.
- [49] J. H. Yong, S. M. Hu, and J. G. Sun. Bisection algorithms for approximating quadratic bézier curves by G^1 arc splines. *Computer-Aided Design*, 32(4):253–260, 2000.