

# A Predictor–Corrector–Type Technique for the Approximate Parameterization of Intersection Curves

Bert Jüttler, Pavel Chalmovianský

<sup>1</sup> Institute of Applied Geometry, Johannes Kepler University, Linz, Austria

e-mail: Bert.juetzler@jku.at,

homepage: <http://www.ag.jku.at/>

<sup>2</sup> Radon Institute of Computational and Applied Mathematics

Austrian Academy of Science

e-mail: pavel.chalmoviansky@ricam.oeaw.ac.at,

homepage: <http://www.ricam.oeaw.ac.at/>

Received: date / In revised form: May 3, 2006 / Accepted: September 19, 2006

**Abstract** We describe a method to approximate a segment of the intersection curve of two implicitly defined surfaces by a rational parametric curve. Starting from an initial solution, the method applies predictor and corrector steps in order to obtain the result.

Based on a preconditioning of the two given surfaces, the corrector step is formulated as an optimization problem, where the objective function approximates the integral of the squared Euclidean distance of the curve to the intersection curve. An SQP-type method is used to solve the optimization problem numerically. Two different predictor steps, which are based on simple extrapolation and on a differential equation, are formulated.

Error bounds are needed in order to certify the accuracy of the result. In the case of the intersection of two algebraic surfaces, we show how to bound the Hausdorff distance between the intersection curve (an algebraic space curve) and its rational approximation.

## 1 Introduction

Implicitly defined space curves, which are obtained by intersecting two implicitly defined surfaces, arise in various ways in geometric modeling. As a special case, they include algebraic space curves, which are intersections of two (or more) algebraic surfaces<sup>1</sup>. For instance, the intersection curves of

---

<sup>1</sup> In some cases, as for a space cubic, two surfaces are not sufficient to define the curve globally, due to the presence of additional components of the intersection,

two quadric surfaces have thoroughly been analyzed in the literature, see [WJG02] and the references cited therein. See also [PM02] for a description of the state of the art in surface–surface intersection in Computer–Aided Design.

In the case of algebraic space curves, several computational techniques relying on symbolic and hybrid symbolic-numeric algorithms exist. These include methods for computing the topology [AS05, GLMT05] and for generating – in the case of rational space curves – a rational parameterization. Indeed, by using a birational transformation, the problem can be reduced to that of parameterizing a planar algebraic curve [Ber97]. The latter problem has thoroughly been analyzed in the literature, including the choice of minimal field extensions and the construction of proper parameterizations [ASS05, Sch92, Sen02, SW97, SW01].

Clearly, these techniques for generating a rational parameterization are limited to the class of *rational* curves (algebraic curves of genus zero). In the *general* situation, *approximate* techniques have to be used. Various related results for planar curves exist [BX97, BR00, GL04]. Numerical methods for space curves and more general varieties have been discussed in [AC04, Har00].

The present paper presents a numerical method which generates an approximate rational parameterization of an implicitly defined space curve. Clearly, a numerical method for tracing a point on the space curve by a predictor–corrector type algorithm can easily be formulated by combining a *predictor* step (e.g., a Runge–Kutta method) with a suitable *corrector*, such as several alternating projections on the two implicitly defined surfaces which define the space curve. In a certain sense, we extend this technique to the case of a rational parameterization. Instead of tracing a just single point on the curve, we formulate a predictor–corrector type method that traces an entire rational curve segment, whose length increases and which represents approximately the algebraic curve between a fixed starting point and a variable end point.

The paper is organized as follows. The next section formulates the problem in more detail and derives a scalar field which approximates the distance function to the given space curve. Section 3 formulates the approximation algorithm and describes its components in some detail. Error bounds are derived in Section 4, and the extension to singular points is discussed in Section 5. Finally we conclude this paper.

## 2 Preliminaries

We describe the parameterization problem and discuss approximations to the distance function of a space curve.

---

cf. [Har83]. In order to obtain a global definition, an ideal of trivariate polynomials has to be used. However, in a neighborhood of a regular point of an intersection curve, two surfaces are locally sufficient.

### 2.1 The parameterization problem

For any  $C^3$  function  $f: \mathbb{R}^3 \rightarrow \mathbb{R}$ , let

$$Z(f) = \{\mathbf{p} \in \mathbb{R}^3 \mid f(\mathbf{p}) = 0\} \quad (1)$$

be its *zero set*. Points  $\mathbf{p} \in Z(f)$  with  $\nabla f(\mathbf{p}) \neq \mathbf{0}$  (resp.  $= \mathbf{0}$ ) are called regular (resp. singular). We consider the intersection curve of two zero sets of  $C^3$  functions  $f$  and  $g$ ,

$$C(f, g) = Z(f) \cap Z(g). \quad (2)$$

If both  $f$  and  $g$  are polynomials, then  $C$  is called an *algebraic curve*. A point  $\mathbf{p} \in C$  is called *regular*, if the two gradient vectors  $\nabla f(\mathbf{p})$  and  $\nabla g(\mathbf{p})$  are linearly independent (and *singular* otherwise). At a regular point  $\mathbf{p}$ , the vectors

$$\mathbf{t}_\ell(\mathbf{p}) = (-1)^{(\ell-1)} \frac{\nabla f(\mathbf{p}) \times \nabla g(\mathbf{p})}{\|\nabla f(\mathbf{p}) \times \nabla g(\mathbf{p})\|}, \quad \ell \in \{1, 2\}, \quad (3)$$

are unit tangent vectors of the intersection curve  $C$  at  $\mathbf{p}$ .

We assume that a regular point  $\mathbf{p}$  of the intersection curve and one of the associated tangent vectors  $\mathbf{t}_\ell(\mathbf{p})$ , where the flag  $\ell \in \{1, 2\}$  specifies the desired orientation, are given.

The solution of the differential equation

$$\mathbf{c}'(s) = \mathbf{t}_\ell(\mathbf{c}(s)), \quad \text{with } \mathbf{c}(0) = \mathbf{p}, \quad (4)$$

defines the *arc-length parameterization* of the intersection curve, starting at the given point  $\mathbf{p}$ . We assume that a suitable numerical method for evaluating this arc-length parameterization is available. For instance, one may combine a Runge–Kutta–type predictor with a suitable corrector step, which projects a point back onto the intersection curve, see [HL93]. A detailed discussion of this numerical method is beyond the scope of the present paper.

In order to represent it in a form which is suitable for further applications, in particular in Computer Aided Design, we want to construct an *approximate rational parameterization of a curve segment of the intersection curve*. The segment starts at the point  $\mathbf{p}$  and has one of the two unit tangent vectors  $\mathbf{t}_\ell$  there, depending on the desired orientation. The segment end point is not known a priori; it is determined by the algorithm. In order to be able to generate a globally smooth approximation by collecting several segments, we shall enforce  $G^1$  boundary tangents, see the next section for details.

## 2.2 The approximating rational curve

The approximating curve is represented as a rational Bézier curve  $\mathbf{x}(t) = (x_1(t)/x_4(t), x_2(t)/x_4(t), x_3(t)/x_4(t))^\top$  of degree  $n$ , where

$$(x_1(t), x_2(t), x_3(t), x_4(t))^\top = \tilde{\mathbf{x}}(t) = \sum_{i=0}^n B_i^n(t) \tilde{\mathbf{p}}_i, \quad (5)$$

with the unknown control points (in homogeneous coordinates)

$$\tilde{\mathbf{p}}_i = (b_{4i}, b_{4i+1}, b_{4i+2}, b_{4i+3})^\top \in \mathbb{R}^4 \setminus \{\mathbf{0}\} \quad (6)$$

and the Bernstein polynomials  $B_i^n(t) = \binom{n}{i} (1-t)^{n-i} t^i$ . All control points can be gathered in one vector

$$\mathbf{b} = (b_0, \dots, b_{4n+3})^\top \in \mathbb{R}^{4n+4}. \quad (7)$$

The approximating curve satisfies  $G^1$  boundary conditions, i.e.,

$$\begin{aligned} \mathbf{x}(0) &= \mathbf{p}, & \mathbf{x}(1) &= \mathbf{q} = \mathbf{c}(s), \\ \mathbf{x}'(0) &= \lambda \mathbf{t}_\ell(\mathbf{p}), & \mathbf{x}'(1) &= \mu \mathbf{t}_\ell(\mathbf{q}) \end{aligned} \quad (8)$$

where  $\mathbf{q} = \mathbf{c}(s)$  with  $s > 0$  is the segment end point, and  $\lambda, \mu > 0$  is needed in order to preserve the orientation of the boundary tangents. In addition, we require that the curve is described in *standard form*,  $b_3 = b_{4n+3} = 1$ , see [HL93]. We use these conditions in order to eliminate some components of the unknown control points by introducing new variables

$$\mathbf{r} = (r_0, \dots, r_{4n-9})^\top. \quad (9)$$

The vector of control points  $\mathbf{b}$  depends linearly on these new variables. The two boundary control points  $\tilde{\mathbf{p}}_0$  and  $\tilde{\mathbf{p}}_n$  are fully determined by  $\mathbf{p}$  and  $\mathbf{q}$ , and the next two inner control points  $\tilde{\mathbf{p}}_1$  and  $\tilde{\mathbf{p}}_{n-1}$  are obtained from the boundary points, the tangents and two free parameters  $r_0, r_1$  and  $r_{4n-8}, r_{4n-9}$  for each of them. The remaining control points are completely free and their components are parameterized by  $r_2, \dots, r_{4n-7}$ . More precisely, we get

$$\mathbf{b}(\mathbf{r}) = A\mathbf{r} + \mathbf{b}_{\text{boundaries}}, \quad (10)$$

where  $A$  is the  $(4n+4) \times (4n-8)$  matrix

$$A(s) = \begin{pmatrix} A_{11} & \mathbf{0}_{8 \times 4(n-3)} & \mathbf{0}_{8 \times 2} \\ \mathbf{0}_{4(n-3) \times 2} & I_{4(n-3) \times 4(n-3)} & \mathbf{0}_{4(n-3) \times 2} \\ \mathbf{0}_{8 \times 2} & \mathbf{0}_{8 \times 4(n-3)} & A_{33} \end{pmatrix} \quad (11)$$

with the blocks

$$A_{11} = \begin{pmatrix} \mathbf{0}_4 & \mathbf{0}_4 \\ \mathbf{t}_\ell(\mathbf{p}) & \mathbf{p} \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad A_{33} = \begin{pmatrix} \mathbf{q} & -\mathbf{t}_\ell(\mathbf{q}) \\ 1 & 0 \\ \mathbf{0}_4 & \mathbf{0}_4 \end{pmatrix}, \quad (12)$$

and

$$\mathbf{b}_{\text{boundaries}} = (\mathbf{p}^\top, 1, 0, \dots, 0, \mathbf{q}^\top, 1)^\top. \quad (13)$$

In order to preserve the orientation of the boundary tangents, only vectors (9) satisfying

$$r_0 = \lambda > 0 \quad \text{and} \quad r_{4n-9} = \mu > 0 \quad (14)$$

are feasible. For later reference we note the relations between gradients and Hessians

$$\nabla G = A^\top \nabla H \quad \text{and} \quad \nabla^2 G = A^\top \nabla^2 H A, \quad (15)$$

where  $H : \mathbf{b} \mapsto H(\mathbf{b})$  is an arbitrary  $C^2$  function on  $\mathbb{R}^{4n+4}$ , and the function  $G : \mathbf{r} \mapsto G(\mathbf{r}) = H(\mathbf{b}(\mathbf{r}))$  on  $\mathbb{R}^{4n-8}$  is obtained by substituting (10).

### 2.3 Approximating the distance to the intersection curve

Following [Sam82], one may use the *Sampson distance*

$$\text{dist}_f(\mathbf{x}) = |f(\mathbf{x})| / \|\nabla f(\mathbf{x})\| \quad (16)$$

in order to approximate the distance between a point  $\mathbf{x}$  and the implicitly defined surface  $Z(f)$ . If  $\mathbf{f}_\mathbf{x} \in Z(f)$  is the closest point to  $\mathbf{x}$ , then

$$\|\mathbf{x} - \mathbf{f}_\mathbf{x}\| = \text{dist}_f(\mathbf{x}) + \mathcal{O}(\|\mathbf{x} - \mathbf{f}_\mathbf{x}\|^2). \quad (17)$$

Consequently, in the vicinity of the surface, (16) is a fairly accurate estimate of the true distance.

The distance to the intersection curve can be approximated by

$$\text{Dist}_{f,g}(\mathbf{x}) = \sqrt{\text{dist}_f(\mathbf{x})^2 + \text{dist}_g(\mathbf{x})^2}, \quad (18)$$

but this does not give a precise estimate. In order to improve it, we consider the functions

$$F(\mathbf{x}) = \bar{F}(\mathbf{x}) / \|\nabla \bar{F}(\mathbf{x})\|, \quad G(\mathbf{x}) = \bar{G}(\mathbf{x}) / \|\nabla \bar{G}(\mathbf{x})\| \quad (19)$$

where

$$\left. \begin{array}{l} \bar{F}(\mathbf{x}) \\ \bar{G}(\mathbf{x}) \end{array} \right\} = f(\mathbf{x}) \|\nabla g(\mathbf{x})\| \left\{ \begin{array}{l} + \\ - \end{array} \right\} g(\mathbf{x}) \|\nabla f(\mathbf{x})\|. \quad (20)$$

**Lemma 1** *For any regular point  $\mathbf{r}$  of the intersection curve, there exists a neighborhood  $N$  such that for all points  $\mathbf{x} \in N$*

$$\|\mathbf{x} - \mathbf{f}_\mathbf{x}\| = \underbrace{\sqrt{F(\mathbf{x})^2 + G(\mathbf{x})^2}}_{= \text{Dist}_{F,G}(\mathbf{x})} + \mathcal{O}(\|\mathbf{x} - \mathbf{f}_\mathbf{x}\|^2). \quad (21)$$

holds, where  $\mathbf{f}_\mathbf{x} \in C$  is the closest point of  $\mathbf{x}$ .

*Proof* If  $N$  is sufficiently small, then the closest points of all  $\mathbf{x} \in N$  are regular points of  $C$ . Then, the two functions  $F$  and  $G$  have unit and mutually perpendicular gradients along  $C$ . The result follows by analyzing the Taylor expansion (and its remainder term) of

$$D(d) = \sqrt{F(\mathbf{f}_\mathbf{x} + d\mathbf{v})^2 + G(\mathbf{f}_\mathbf{x} + d\mathbf{v})^2}, \quad (22)$$

where  $\mathbf{v} = (\mathbf{x} - \mathbf{f}_\mathbf{x}) / \|\mathbf{x} - \mathbf{f}_\mathbf{x}\|$  and  $d$  is the distance, at  $d = 0$ .  $\square$

### 3 Approximate parameterization of the intersection curve

This section starts with an outline of the predictor–corrector–type algorithm for approximating a regular segment of the intersection curve. Then we describe its components in more detail.

#### 3.1 Outline

We assume that the two functions  $f, g$  defining the intersection curve  $C$ , a regular point  $\mathbf{p} \in C$ , the flag  $\ell \in \{1, 2\}$  which specifies the desired orientation, the degree  $n$ , and the numerical procedure for evaluating the arc-length parameterization  $s \mapsto \mathbf{c}(s)$  are available. The approximation algorithm consists of four steps.

1. Use the *initial predictor* to generate the segment end point  $\mathbf{q}$  and the initial curve segment  $\mathbf{x}(t)$ .
2. Apply the *corrector* in order to optimize the current solution. If the accuracy of the result is insufficient, then continue with step 4.
3. Use one of the two *predictors* in order to extend the current approximating curve  $\mathbf{x}(t)$ . If an extension is possible, then continue with step 2.
4. Use the rational curve  $\mathbf{x}(t)$  which was generated prior to applying the last predictor step as the final result.

If the length of the curve  $\mathbf{x}(t)$  is too small, then one can now set  $\mathbf{p} = \mathbf{x}(1)$  and apply the algorithm again. This gives a  $G^1$  rational spline curve which approximates the intersection curve  $C$ .

*Example 1* An example is shown in Fig. 1. Two quadrics intersect in an algebraic curve of order 4, which is approximated by a rational cubic curve. Three curves obtained by the iterative algorithm are shown.

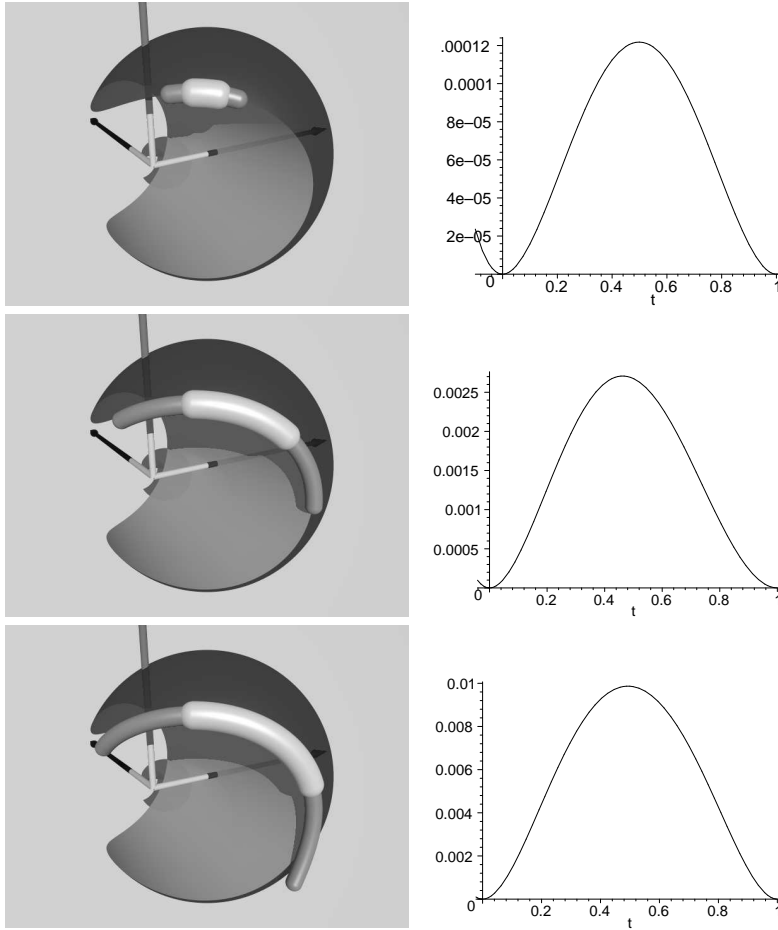
#### 3.2 Initial predictor

Using the numerical procedure for evaluating the arc length parameterization, we generate the segment end point  $\mathbf{q} = \mathbf{c}(h)$  and the unit tangent vectors at  $\mathbf{p} = \mathbf{t}_\ell(\mathbf{p})$  and  $\mathbf{q} = \mathbf{t}_\ell(\mathbf{q})$ . The step size  $h$  should be chosen proportional to the radius of curvature of  $C$  at  $\mathbf{p}$ . The initial solution is the cubic polynomial interpolant  $\mathbf{x}(t)$  (see (5), but with degree  $n = 3$  and weights  $b_3 = b_7 = b_{11} = b_{15} = 1$ ) of the  $C^1$  Hermite boundary data

$$\mathbf{x}(0) = \mathbf{p}, \quad \mathbf{x}'(0) = L \mathbf{t}_\ell(\mathbf{p}), \quad \mathbf{x}(1) = \mathbf{q}, \quad \mathbf{x}'(1) = L \mathbf{t}_\ell(\mathbf{q}). \quad (23)$$

where  $L = \|\mathbf{p} - \mathbf{q}\|$ . If  $n > 3$ , then we use degree elevation (see [HL93]) in order to generate the control points of the initial solution.

*Example 2* Fig. 2 (top row) shows a the initial solution obtained for the intersection curve of a quadric and a cubic surface.

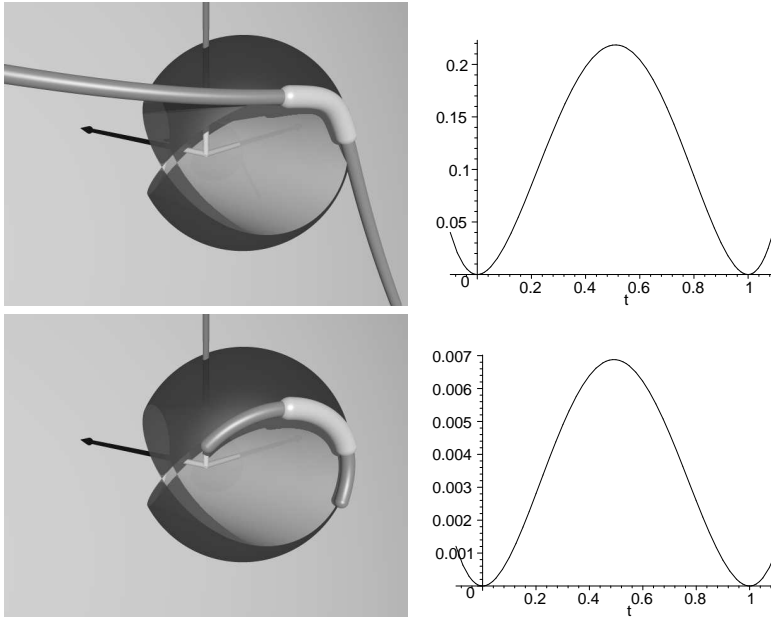


**Fig. 1** Left column: Rational cubic curves approximating the intersection of a sphere (radius 2) and a cylinder. The thick parts represent the the curve obtained for  $t \in [0, 1]$ . From top to bottom: While the left end point remains fixed, the right end point moves along the intersection curve. Right column: Orthogonalized distance functions  $\text{Dist}_{f,g}(\mathbf{x}(t))$ . The error increases with the length of the curve segment.

### 3.3 Corrector step

We generate the optimized rational curve by minimizing a suitable objective function (see below) among all rational curves  $\mathbf{x}(t)$  in standard form, which are subject to the  $G^1$  boundary conditions (8) for fixed boundary points  $\mathbf{p}$ , and  $\mathbf{q}$ . The objective function  $H = H(\mathbf{b})$  is chosen as

$$H = H_0 + w_1 H_1 + w_2 H_2 \quad (24)$$



**Fig. 2** Approximate parameterization of the intersection of a sphere with a cubic surface. Initial (top) and optimized (bottom) solutions, and estimated distance  $\text{Dist}_{F,G}$  (right). The sphere has radius 2. The thicker (resp. thinner) part corresponds to the curve segment  $t \in [0, 1]$  (resp.  $t \in \mathbb{R} \setminus [0, 1]$ ).

with

$$H_0 = \int_0^1 F(\mathbf{x}(t))^2 + G(\mathbf{x}(t))^2 dt, \quad (25)$$

and

$$H_1 = \int_0^1 h(x_0(t)) dt, \text{ where } h(\xi) = (\xi - 1)^8, \quad (26)$$

and the regularization terms

$$H_2 = \sum_{i=0}^{n-1} \|\tilde{\mathbf{p}}_{i+1} - \tilde{\mathbf{p}}_i\|^2, \text{ or } H_2' = \sum_{i=0}^{n-2} \|\tilde{\mathbf{p}}_{i+2} - 2\tilde{\mathbf{p}}_{i+1} + \tilde{\mathbf{p}}_i\|^2. \quad (27)$$

The first term  $H_0$  approximates the integral of the squared Euclidean distance between the intersection curve and its rational approximation, cf. Lemma 1. It will keep the approximating rational curve close to the intersection curve.

The second term  $H_1$  measures the deviation of the rational Bézier curve  $\mathbf{x}(t)$  from a polynomial one. It is introduced in order to avoid potential problems with a vanishing denominator  $x_4(t)$ , cf. (5). At the same time, it allows the denominator to deviate significantly from 1. Hence, the method uses the greater flexibility of rational curves, as compared to polynomial



ones. The function  $h$  can be replaced with other suitable  $C^2$  convex function with minimum at 1 in order to get similar results.

The regularization term  $H_2$  represents the length of the homogeneous control polygon. The alternative regularization term  $H'_2$  forces the distribution of the control points on the control polygon to be as uniform as possible. By using a regularization one may avoid potential problems with degeneracies of the linear systems which have to be solved in the corrector and predictor steps of the algorithm. For instance, such degeneracies could be caused by using a too high polynomial degree, such as, e.g., trying to generate a cubic parameterization of a straight line. The terms  $H_2$  and  $H'_2$  can also be used simultaneously.

We assume that the weights  $w_1$  and  $w_2$  are specified by the user. They should satisfy

$$0 < w_1 \ll w_2 \ll 1 \quad (28)$$

in order to match the importance of the objectives represented by the corresponding terms. They can semi-automatically be adjusted during the optimization, such that both  $w_1$  and  $w_2$  tend to zero.

The objective function depends on the parameters  $\mathbf{r} = (r_0, \dots, r_{4n-9})$ , see (9), which are subject to the constraints  $r_0 > 0, r_{4n-9} > 0$ , see (14). In order to obtain a closed convex domain, we replace the constraints by

$$r_0 \geq \varepsilon, \quad r_{4n-9} \geq \varepsilon, \quad (29)$$

where  $0 < \varepsilon \ll 1$  is a suitable constant.

The optimized rational approximation of the intersection curve is found by solving the non-linear optimization problem

$$\mathbf{r}^* = \arg \min_{\{\mathbf{r} \mid r_0 \geq \varepsilon, r_{4n-9} \geq \varepsilon\}} G(\mathbf{r}), \quad (30)$$

where  $G(\mathbf{r}) = H(\mathbf{b}(\mathbf{r}))$ , via an SQP<sup>2</sup>-type method. More precisely, we generate a sequence of parameters  $\{\mathbf{r}^i\}_{i=0}^\infty$ , where  $\mathbf{r}^0$  is the previously generated initial solution, and

$$\mathbf{r}^{i+1} = \mathbf{r}^i + \gamma_i (\mathbf{r}^i - \text{sgn}(\nabla G(\mathbf{r}^i) \cdot \mathbf{q}^i) \mathbf{q}^i), \quad (31)$$

where

$$\mathbf{q}^i = [\nabla^2 G(\mathbf{r}^i)]^{-1} \nabla G(\mathbf{r}^i). \quad (32)$$

The derivatives in  $\nabla G(\mathbf{r}^i)$ ,  $\nabla^2 G(\mathbf{r}^i)$  can be evaluated by applying the differentiation to the integrands in  $H_0$  and  $H_1$ , and using numerical integration. In addition, the equations (15) may be useful.

Using bisection starting with 1, the step sizes  $\gamma_i \in (0, 1]$  are chosen such that the values  $H(\mathbf{r}^i)$  decrease and such that the new iterate respects the constraints (29). If the decrease of  $H(\mathbf{r}^i)$  falls below a user-defined threshold, then the iteration (31) terminates and the current vector  $\mathbf{r}^i$  serves as approximation to  $\mathbf{r}^*$ .

---

<sup>2</sup> sequential quadratic programming

*Example 3* This corrector step has been applied to the initial solution described in the previous example, see Fig. 2 (bottom row). The maximum distance error is reduced to approx. 3% of its initial value. The rational approximation has degree  $n = 3$ .

### 3.4 Predictor steps

Two different predictor steps, which generate the homogeneous control points  $(\tilde{\mathbf{p}}_i^+)_{i=0,\dots,n}$  describing the initial guess for the next corrector step, may be used.

The first predictor (by extrapolation) is simpler and uses a more heuristic approach. The second predictor (by evolution) tries to trace the optimal solution, which is slightly more complicated, but automatically guarantees the preservation of the boundary conditions.

*Extrapolation predictor.* We choose an extrapolation parameter  $E \in [0, 1]$  and extrapolate the current rational curve segment to the parameter domain  $[0, 1 + E]$ . The extrapolated segment has the homogeneous control points

$$\tilde{\mathbf{p}}_k^* = \sum_{i=0}^k B_i^k(1 + E) \tilde{\mathbf{p}}_i, \quad k = 0, \dots, n. \quad (33)$$

They can be computed using the algorithm of de Casteljau, see [HL93]. By transforming this extrapolated curve into standard form, we obtain

$$\tilde{\mathbf{p}}_k^\circ = \frac{1}{(\sqrt[n]{\tilde{p}_{i,4}^*})^i} \tilde{\mathbf{p}}_k^*, \quad (34)$$

where  $\tilde{p}_{i,4}^*$  is the weight of the last control point.

Finally, we modify the last two control points in order to satisfy the  $G^1$  boundary conditions at  $\mathbf{x}(1)$ . We need to consider the Cartesian coordinates  $\mathbf{p}_i = (p_{i,1}, p_{i,2}, p_{i,3})^\top$  of these control points, which satisfy  $p_{i,j} = \tilde{p}_{i,j}/\tilde{p}_{i,4}$ . First, we project the last control point into a point  $\mathbf{q} = \mathbf{c}(s^*)$  and replace the last homogeneous control point by  $\tilde{\mathbf{p}}_n^+ = (q_1, q_2, q_3, 1)$ . Second, we project the last but one control point onto the tangent at  $\mathbf{q}$ ,

$$\mathbf{p}_{n-1}^+ = \mathbf{q} + [(\mathbf{p}_{n-1}^\circ - \mathbf{q}) \cdot \mathbf{t}_\ell(\mathbf{q})] \mathbf{t}_\ell(\mathbf{q}) \quad (35)$$

and replace the corresponding homogeneous control point by

$$\tilde{\mathbf{p}}_{n-1}^+ = \tilde{\mathbf{p}}_{n-1,4}^\circ (p_{n-1,1}^+, p_{n-1,2}^+, p_{n-1,3}^+, 1)^\top. \quad (36)$$

All other control points remain unchanged,  $\tilde{\mathbf{p}}_i^+ = \tilde{\mathbf{p}}_i^\circ$  for  $i = 0, \dots, n - 2$ .

If the extrapolation parameter  $E$  is too large, then the homogeneous control points  $\tilde{\mathbf{p}}_i^+$ , which are generated by the extrapolation predictor, may describe an infeasible curve, or the error (measured by (25)) may be too large. We then use bisection towards 0 in order to decrease  $E$ . As an heuristic initial estimate we choose  $E = \min\{E_{\max}, \tan \alpha / \kappa_1\}$  where  $E_{\max}$  is a user-defined maximum value and  $\kappa_1$  is curvature of the current curve  $\mathbf{x}(1) = \mathbf{q}$ . The angle  $\alpha$  is defined by user. In our experiments, we chose  $\alpha = \frac{\pi}{6}$ .

*Evolution predictor.* In order to derive the second predictor, we consider the solution of the optimization problem  $G(\mathbf{r}) \rightarrow \min$ , see (30), as a function of the arc length  $s$  of the curve segment. For any fixed value of  $s$ , i.e., for any fixed segment end point  $\mathbf{q} = \mathbf{c}(s)$ , the solution satisfies  $\nabla G = \mathbf{0}$ . If  $s$  varies, then we obtain using (10) and (15) the ordinary differential equation

$$\mathbf{0} = \frac{d}{ds} \nabla G = \frac{d}{ds} (A(s)^\top) \nabla H(A(s) \mathbf{r}(s) + \mathbf{b}_{\text{boundaries}}(s)) \quad (37)$$

or, equivalently,

$$\mathbf{0} = C(\mathbf{r}, s) \mathbf{r}'(s) + D(\mathbf{r}, s) \mathbf{r}(s) + E(\mathbf{r}, s) \quad (38)$$

with the matrices

$$C = A^\top \nabla^2 H A, \quad D = A^\top \nabla^2 H A' \quad \text{and} \quad (39)$$

$$E = A'^\top \nabla H + A^\top \nabla^2 H \mathbf{b}'_{\text{boundaries}}. \quad (40)$$

where  $' = d/ds$ . In order to extend the current solution, where  $s = s_0$ , to some bigger curve segment  $s_0(1 + E)$  (where  $E$  is again the extrapolation parameter), we trace the solutions of the differential equation (38), which is equivalent to

$$\mathbf{r}' = \phi(\mathbf{r}, s) = -C(\mathbf{r}, s)^{-1} D(\mathbf{r}, s) \mathbf{r} - C(\mathbf{r}, s)^{-1} E(\mathbf{r}, s) \quad (41)$$

using the standard Runge–Kutta method, see e.g. [SB05]. Clearly, this works only if  $C$  is regular. This can be guaranteed by using the following result.

**Lemma 2** *If  $w_2$  is sufficiently large, then the matrix  $C(\mathbf{r}, s)$  is regular.*

*Proof* Eq. (24) gives

$$C = A^\top \nabla^2 H_0 A + w_1 A^\top \nabla^2 H_1 A + w_2 A^\top \nabla^2 H_2 A \quad (42)$$

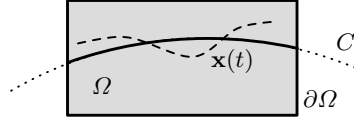
Since  $H_2$  is a semidefinite quadratic form of the control point vector  $\mathbf{b}$  (see (6), (7) and (27)), its Hessian is also semidefinite. The null space of the Hessian consists of all vectors of the form

$$\mathbf{b} = (\xi_0, \xi_1, \xi_2, \xi_3, \xi_0, \xi_1, \xi_2, \xi_3, \dots, \xi_0, \xi_1, \xi_2, \xi_3)^\top \quad (43)$$

which correspond to curves where all homogeneous control points (6) coincide. On the other hand, the space spanned by the columns of  $A$  (see (11)) does not contain any of these vectors, except for the null vector. Consequently, the matrix  $A^\top \nabla^2 H_2 A$  is positive definite and therefore regular. Hence, if  $w_2$  is sufficiently large, the matrix  $C$  is regular, too. More precisely,  $\lim_{w_2 \rightarrow \infty} \det C = \infty$ .  $\square$

#### 4 Error bound

We discuss how to obtain an error bound for the rational approximation of the intersection curve.



**Fig. 3** Eq. (44) defines the one-sided Hausdorff distance between the dashed curve and the solid curves.

#### 4.1 Hausdorff distance bound

We assume that the approximating curve segment  $\mathbf{x}(t)$  is contained in some closed domain  $\Omega \subset \mathbb{R}^3$ , and let  $C^* = C \cup \partial\Omega$ , where  $\partial\Omega$  is the boundary of  $\Omega$ . The one-sided Hausdorff distance of the curve segment and  $C^*$  is defined as

$$\text{HD}_\Omega(\mathbf{x}, C) = \sup_{t \in [0,1]} \inf_{\mathbf{y} \in C^* \cap \Omega} \|\mathbf{x}(t) - \mathbf{y}\|, \quad (44)$$

see Fig. 3 for a schematic illustration.

*Remark 1* The usual Hausdorff distance is the symmetrized version of (44). The boundary of  $\Omega$  had to be added to  $C$ , in order to avoid certain technical problems.

**Lemma 3** *We consider the two functions  $f, g$  defining the curve  $C$ , and  $h = \sqrt{f^2 + g^2}$ . We assume that positive constants  $c, k$  exist, such that the inequalities  $\|\nabla f\| \geq c$ ,  $\|\nabla g\| \geq c$  and  $|\nabla f \cdot \nabla g| \leq k$  are satisfied for  $\mathbf{x} \in \Omega$ , and  $c^2 > k$ . Then  $\|\nabla h\| \geq \sqrt{c^2 - k}$  holds for  $\mathbf{x} \in \Omega \setminus C$ .*

*Proof* Clearly,  $\nabla h = (f\nabla f + g\nabla g)/\sqrt{f^2 + g^2}$ . Hence,

$$\begin{aligned} \|\nabla h\|^2 &= \frac{1}{f^2 + g^2} (f^2 \|\nabla f\|^2 + g^2 \|\nabla g\|^2) + \frac{1}{f^2 + g^2} (2fg \nabla f \cdot \nabla g) \\ &\geq \underbrace{\left| \frac{1}{f^2 + g^2} (f^2 \|\nabla f\|^2 + g^2 \|\nabla g\|^2) \right|}_{\geq c^2} - \underbrace{\left| \frac{2fg}{f^2 + g^2} \right|}_{\leq 1} \underbrace{|\nabla f \cdot \nabla g|}_{\leq k} \end{aligned}$$

and the result follows.  $\square$

**Theorem 1** *Consider a curve segment  $\mathbf{x} : I \rightarrow \Omega$ . Under the assumptions of Lemma 3, if  $f(\mathbf{x}(t))^2 + g(\mathbf{x}(t))^2 \leq M^2$  for  $t \in I$ , then the one-sided Hausdorff distance is bounded by*

$$\text{HD}_\Omega(\mathbf{x}, C) \leq \frac{M}{\sqrt{c^2 - k}}. \quad (45)$$

*Proof* We consider the integral curves of the vector field  $-\frac{\nabla h}{\|\nabla h\|}$  where  $h$  is defined as in the Lemma. These curves are regular at all inner points

of  $\Omega \setminus C$ . For any  $t \in I$ , consider an integral curve  $\mathbf{p}(s)$  with  $\mathbf{p}(0) = \mathbf{x}(t)$ . Applying the Mean Value Theorem to  $h(\mathbf{p}(s))$  and using Lemma 3, we get

$$\begin{aligned} h(\mathbf{p}(s)) &= h(\mathbf{p}(0)) + s \nabla h(\mathbf{p}(s_0)) \cdot \dot{\mathbf{p}}(s_0) \\ &= h(\mathbf{p}(0)) - s \|\nabla h(\mathbf{p}(s_0))\| \leq M - s \sqrt{c^2 - k}. \end{aligned}$$

Since  $h \geq 0$ , the integral curve  $\mathbf{p}(s)$  is not defined for  $s > \frac{M}{\sqrt{c^2 - k}}$ . Consequently, there exists an  $s^* \in [0, \frac{M}{\sqrt{c^2 - k}}]$ , such that

$$\lim_{\substack{s \rightarrow s^* \\ s < s^*}} \mathbf{p}(s) = \mathbf{y}, \quad \text{where } \mathbf{y} \in C^*. \quad (46)$$

As  $\mathbf{p}(s)$  is parameterized by arc length,

$$\|\mathbf{x}(t) - \mathbf{y}\| = \|\mathbf{p}(0) - \mathbf{y}\| \leq s^* \leq \frac{M}{\sqrt{c^2 - k}}. \quad \square$$

*Remark 2* In the presence of almost singular points of the gradient fields ( $c \approx 0$ ) or tangential intersections ( $c^2 \approx k$ ), the theorem gives either no bounds (if  $c^2 < k$ ) or only very poor bounds.

#### 4.2 Bounding function values and gradient norms

In the case of polynomials  $f$  and  $g$ , one may obtain the bounds  $M, c, k$  by exploiting the convex-hull property of Bernstein–Bézier representations (see [HL93]).

*Function values.* The expression  $f(\mathbf{x}(t))^2 + g(\mathbf{x}(t))^2$  is a rational function of  $t$ . It can therefore be represented in Bernstein–Bézier (BB) form as

$$f(\mathbf{x}(t))^2 + g(\mathbf{x}(t))^2 = \left( \sum_{i=0}^N B_i^N(t) a_i \right) / \left( \sum_{i=0}^N B_i^N(t) w_i \right), \quad t \in (0, 1), \quad (47)$$

where  $a_i$  and  $w_i$  are the BB coefficients of numerator and denominator, and  $N$  is the degree. If all  $w_i$  are positive (and this is very likely the case, since the numerator is a certain power of the positive polynomial  $x_4(t)$ ), the upper bound  $M$  can be chosen as

$$M = \sqrt{\max_i |a_i| / \min_i w_i}. \quad (48)$$

If some of the weights are negative, then one may generate the BB representations with respect to subintervals via subdivision. Since the numerator is a positive polynomial, this is guaranteed to give positive weights, provided that the length of the subintervals is sufficiently small.

*Length of the gradients.* We choose a tetrahedron  $\Delta$  which contains the domain  $\Omega$ . Recall that any trivariate polynomial with values in  $\mathbb{R}^d$  (for any  $d \in \mathbb{Z}_+$ ) has a Bernstein–Bézier representation with respect to this simplex, see [HL93]. In the case of the gradient of  $f$  (and similarly for  $g$ ), we get a BB representation of the form

$$\nabla f(\mathbf{y}) = \sum_{|\mathbf{i}|=n_f-1} B_{\mathbf{i}}^{n_f-1}(\mathbf{y}) \mathbf{m}_{\mathbf{i}} \quad (49)$$

with certain coefficients  $\mathbf{m}_{\mathbf{i}} \in \mathbb{R}^3$ . Here,  $n_f$  is the degree of  $f$ , and the functions  $B_{\mathbf{i}}^{n_f-1}(\mathbf{y})$  are the trivariate Bernstein polynomials with respect to  $\Delta$ . We use multi-indices  $\mathbf{i} = (i_1, i_2, i_3, i_4) \in \mathbb{Z}_+^4$  and  $|\mathbf{i}| = i_1 + i_2 + i_3 + i_4$ . If the convex hull of the coefficients  $\{\mathbf{m}_{\mathbf{i}} \mid |\mathbf{i}| = n_f - 1\}$  does not contain the origin, then the distance between the convex hull<sup>3</sup> and the origin is a lower bound for  $\|\nabla f\|$ . In the case that this assumption is violated, then one may split  $\Delta$  into sub-tetrahedra and analyze the corresponding BB representations.

*Inner product of the gradients.* This upper bound can be derived by analyzing the BB representation of the scalar-valued trivariate polynomial  $\nabla f \cdot \nabla g$  of degree  $n_f + n_g - 2$  with respect to the tetrahedron. The upper bound  $k$  can be chosen as the maximum absolute value of its BB coefficients.

*Remark 3* In the case of planar algebraic curves, a similar technique has been used in [ASJS05].

#### 4.3 Example

We applied this technique to the three examples in Figure 1, where the intersection curve  $C$  is defined by the two polynomials  $f = x^2 + y^2 - 1.44$  and  $g = (x - 1)^2 + y^2 + z^2 - 4$ . In order to obtain tight bounds, we split the curves into 1, 2 and 6 subsegments, respectively, and the table reports the values for the segment with the largest error. We chose the domain  $\Omega$  to be the convex hull of the Bézier control points of the approximating rational curve. This convex hull, which was a simplex, has also been used to generate the various Bernstein–Bézier representations.

One may compare these bounds with the graphs of the orthogonalized distance functions in Fig. 1. These functions have the same order of magnitude as the bounds, indicating that the bounds are fairly accurate.

---

<sup>3</sup> In order to avoid the convex hull computation, one may instead consider the BB representation of the scalar-valued polynomial  $\|\nabla f(\mathbf{y})\|^2$  to obtain a lower bound on it, which can then be used to infer a lower bound on  $\|\nabla f(\mathbf{y})\|$ .

	M	c	k	$\text{HD}_\Omega \leq$
top	$1.6 \cdot 10^{-4}$	1.05	$3.4 \cdot 10^{-4}$	$1.5 \cdot 10^{-4}$
center	$2.6 \cdot 10^{-3}$	0.99	$4.1 \cdot 10^{-1}$	$3.5 \cdot 10^{-3}$
bottom	$9.4 \cdot 10^{-3}$	0.73	$1.8 \cdot 10^{-4}$	$1.3 \cdot 10^{-2}$

**Table 1** Error bounds for the approximating curves in Figure 1.

## 5 Dealing with singular points

The method can be extended to curve segments starting at singular points (which are characterized by  $\nabla f \times \nabla g = \mathbf{0}$ ) of the intersection curve. For the sake of simplicity, we restrict ourselves to singular points, where the tangents of all real branches passing through this point are mutually different. Also, we do not address the question of how to compute singular points and these tangent directions. In the case of polynomials, these problems lead to (systems of) polynomial equations.

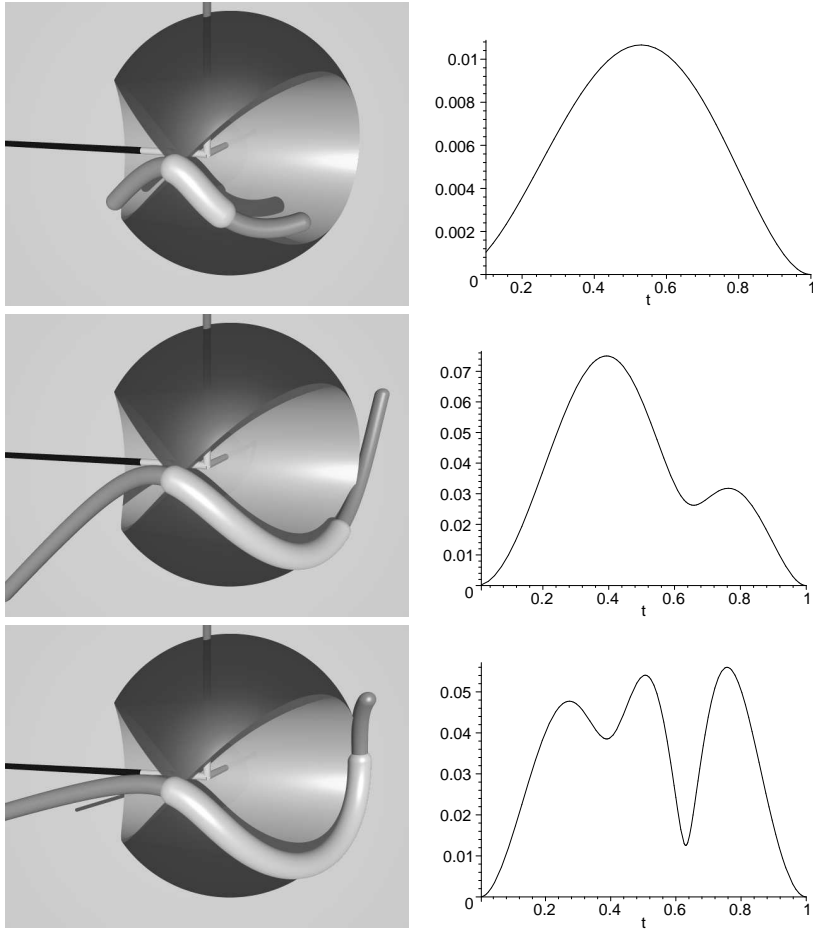
In order to apply the parameterization technique to the case, where  $\mathbf{p}$  is a singular point, it has to be modified as follows.

1. The tangent vector  $\mathbf{t}_\ell(p)$  has to be chosen as the tangent of the chosen branch of the curve starting from  $\mathbf{p}$ ; it cannot be computed from (3).
2. The corrector step of the numerical method used to solve (4) may experience difficulties for small values of  $s$ . If this is the case, then one should use a better predictor at the singular point ( $s = 0$ ). For instance, one may use a higher-order Taylor expansion of the desired branch of the intersection curve. Techniques from symbolic computation will facilitate the computation of such an expansion.
3. The orthogonalized distance function is not defined at  $p$ . Therefore, the integration in (25) should start at some small positive value  $\varepsilon \ll 1$ . For the segment  $[0, \varepsilon]$  one may use the (25), but with the original functions  $f$  and  $g$ .

*Example 4* We consider a rational curve segment on Viviani’s window, which is defined as the intersection of a cylinder and a sphere which touch each other. This curves possesses an exact rational parameterization of degree 4, and we generate an approximate parameterization of degree 3. Even for longer segments, the error remains quite small.

## 6 Conclusion

We presented an algorithm for the approximate parameterization of regular segments of the intersection curves of two implicitly defined surfaces by a rational curve. The algorithm optimizes both the numerator (control points) and the denominator (“weights”) of the rational parameterization. In order to obtain a good approximation, we introduced an orthogonalized distance function, which is used to measure the quality of the fit. The method can be



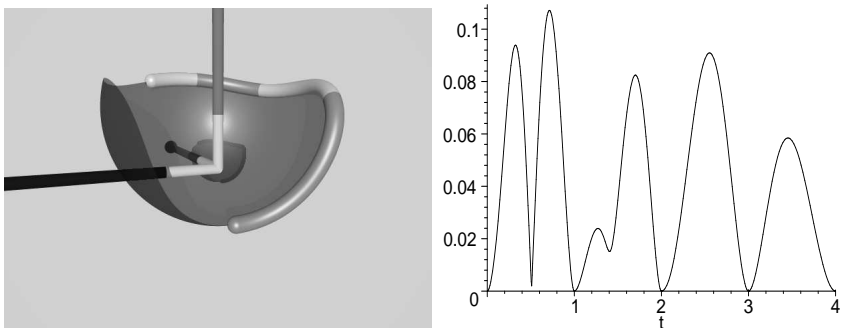
**Fig. 4** Rational curve approximating Viviani's window, starting at the singular point. Since the orthogonalized distance function is not defined at  $t = 0$ , the graphs show the plots starting at  $t = 0.1$ ,  $0.013$  and  $0.003$ , respectively.

used to generate spline curves (i.e., piecewise rational parameterizations), see Fig. 5. It can be modified so as to deal with curves starting at a singular point.

Open problems include the exploitation of techniques for analyzing the topology of algebraic space curves (cf. [AS05, GLMT05]) in connection with the parameterization algorithm, and the generation of distance bounds in the case of non-polynomial functions  $f$  and  $g$ . Also, it would be desirable to derive a priori distance bounds for the approximating curve.

*Acknowledgments* This research was supported by the Austrian Science Fund (FWF) through the Special Research Programme (SFB) F013 "Nu-





**Fig. 5** Rational approximation of intersection of a sphere (radius 2) and a cubic surface by a  $G^1$  rational spline curve (left) and the graph of the orthogonalized distance function (right).

merical and Symbolic Scientific Computing”, subproject 15. The second author would like to thank to Dalibor Lukáš for valuable discussions. The authors thank the anonymous referees for their valuable comments on an earlier version of this paper.

## References

- [ASJS05] M. Aigner, I. Szilágyi, B. Jüttler, and J. Schicho. Implicitization and distance bounds. In B. Mourrain, M. Elkadi, and R. Piene, editors, *Algebraic Geometry and Geometric Modelling*. Springer, 2006. pages 71–86.
- [ASS05] J. G. Alcazar, J. Schicho, and J. R. Sendra. Computation of the topological types of the level curves of an algebraic surface. 2005. submitted.
- [AS05] J. G. Alcazar and J. R. Sendra. Computation of the topology of real algebraic space curves. *J. Symb. Comp.*, 39(6):719–744, 2005.
- [AC04] D. A. Aruliah and R. M. Corless. Numerical parameterization of affine varieties using ODEs. In J. Gutierrez, editor, *Proc. ISSAC*, pages 12–18. ACM Press, New York, 2004.
- [BR00] C. L. Bajaj and A. V. Royappa. Parameterization in finite precision. *Algorithmica*, 27(1):100–114, 2000.
- [BX97] C. L. Bajaj and G. L. Xu. Piecewise rational approximations of real algebraic curves. *J. Comput. Math.*, 15(1):55–71, 1997.
- [Ber97] T. G. Berry. Parameterization of algebraic space curves. *J. Pure Appl. Algebra*, 117-118:81–95, 1997.
- [GL04] X.-S. Gao and M. Li. Rational quadratic approximation to real algebraic curves. *Comput. Aided Geom. Des.*, 21(8):805–828, 2004.
- [GLMT05] G. Gattelier, A. Labrouzy, B. Mourrain, and J. P. Tércourt. Computing the topology of three-dimensional algebraic curves. In T. Dokken and B. Jüttler, editors, *Computational Methods for Algebraic Spline Surfaces*, pages 27–43. Springer, 2005.
- [Har00] E. Hartmann. Numerical parameterization of curves and surfaces. *Comput. Aided Geom. Des.*, 17(3):251–266, 2000.

- [Har83] R. Hartshorne. *Algebraic geometry*. Springer, New York, 1983.
- [HL93] J. Hoschek and D. Lasser. *Fundamentals of computer aided geometric design*. A. K. Peters., Wellesley, Mass., 1993.
- [PM02] N. M. Patrikalakis and T. Maekawa. Chapter 25: Intersection problems. In G. Farin, J. Hoschek, and M.-S. Kim, editors, *Handbook of computer aided geometric design*. Elsevier, 2002.
- [Sam82] P. D. Sampson. Fitting conic sections to very scattered data: an iterative refinement of the Bookstein algorithm. *Computer Graphics and Image Processing*, 18:97–108, 1982.
- [Sch92] J. Schicho. On the choice of pencils in the parametrization of curves. *J. Symb. Comp.*, 14(6):557–576, 1992.
- [Sen02] J. R. Sendra. Normal parametrizations of algebraic plane curves. *J. Symb. Comp.*, 33(6):863–885, 2002.
- [SW97] J. R. Sendra and F. Winkler. Parametrization of algebraic curves over optimal field extensions. *J. Symb. Comp.*, 23(2/3):191–208, 1997.
- [SW01] J. R. Sendra and F. Winkler. Tracing index of rational curve parametrizations. *Comp. Aided Geom. Des.*, 18:771–795, 2001.
- [SB05] J. Stoer and R. Bulirsch. *Numerical mathematics 2*. Springer, Berlin, 2005. 5th ed.
- [WJG02] W. Wang, B. Joe, and R. Goldman. Computing quadric surface intersections based on an analysis of plane cubic curves. *Graph. Models*, 64(6):335–367, 2002.