Predictor–Corrector Technique for Approximate Parameterization of Intersection Curves

Pavel Chalmovianský[1] and Bert Jüttler[2]

[1] Radon Institute of Computational and Applied Mathematics
Austrian Academy of Science
pavel.chalmoviansky@ricam.oeaw.ac.at,
homepage: http://www.ricam.oeaw.ac.at/

[2] Institute of Applied Geometry, Johannes Kepler University, Linz, Austria
Bert.juettler@jku.at,
homepage: http://www.ag.jku.at/

Abstract. We describe a method for computing a rational curve which approximates the intersection curves of two implicitly defined surfaces. Based on a preconditioning of the two given surfaces, the problem can be 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. Special attention is paid to the generation of an initial predictor, which is found by a region–growing–type technique. We use the predictor of de Casteljau type and Runge-Kutta type. We briefly discuss how to treat singularities of the intersection curve.

1 Introduction

Algebraic space curves, which can be defined as the intersection of two algebraic surfaces, arise in various ways in geometric modeling, e.g., as the intersections of two natural quadrics. Recently, powerful computational techniques for analyzing the topology of such curves have been formulated [1]. Techniques for converting them into parametric form are needed for various application in geometric modeling and processing, such as the generation of blending surfaces.

Some algebraic space curves, such as truly spatial cubics, admit an exact representation as rational parametric curves, and various parameterization techniques are known from classical algebraic geometry [2–4]. However, such a representation does not exist in most cases.

The particular case of the intersections of two quadric surfaces has been studied thoroughly (see [5] and the references cited therein). It was observed that it is possible to parameterize them using square root functions. Certain special types of curves (such as curves with polynomial parameterization) were considered in [6].

In the general case, no such results can be expected, and the use of approximate techniques is unavoidable (see [7]). Numerical parameterization of the intersection curves via polynomial approximation was considered in [8]. Certain numerical aspects of such parameterizations were discussed in [9]. Ideally, approximate techniques would be able to reproduce exact rational parameterizations, if those are available.

The paper is organized as follows. Section 2 formulates the problem, recalls several facts on singularities and describes a preconditioning step which can be applied to the given implicitly defined surfaces, in order to approximate the scalar field of the squared distance to the intersection curve. After these preparations, Section 3 describes the predictor–corrector method. Examples are given in section 4. Finally we conclude the paper. Some additional details of the algorithm are presented in appendices.

2 Preliminaries

2.1 Outline

For any function $f\colon \mathbb{R}^3 \to \mathbb{R}$, let

$$Z(f) = \{(x, y, z)^\top \in \mathbb{R}^3 : f(x, y, z) = 0\} \tag{1}$$

be the *zero set*. In the vicinity of a point $\mathbf{p} \in Z(f)$, the zero set of a function is a surface if $\nabla f(\mathbf{p}) \neq \mathbf{0}$. Such a point is called *regular point of surface* $Z(f)$.

We consider the intersection curve of two zero sets of functions f and g,

$$C(f, g) = Z(f) \cap Z(g), \tag{2}$$

where the functions are assumed to be C^3. If both f and g are polynomials, the curve C is called an *algebraic* curve.

A point $\mathbf{p}$ of 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

$$\mathbf{u} = \pm \frac{\nabla f(\mathbf{p}) \times \nabla g(\mathbf{p})}{\|\nabla f(\mathbf{p}) \times \nabla g(\mathbf{p})\|}. \tag{3}$$

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

2.2 Singular Points

There can be singular points of the intersection curve C which are or are not singular points of the surfaces $Z(f)$ and $Z(g)$.

The singularities, which are not singular points of any of the surface, occur iff the tangent planes of both surfaces defined by non-zero gradient vector are identical.

Let $\mathbf{p}$ be such a point and $\frac{1}{k_f}\nabla f(\mathbf{p}) = (0, 0, 1)^\top = \frac{1}{k_g}\nabla g(\mathbf{p})$, where $k_f, k_g \in \mathbb{R}$ are appropriate coefficients scaling the gradient vectors. This can be achieved via suitable Euclidean transformation. Then, using the implicit function theorem, the surfaces $Z(f)$ and $Z(g)$ can locally be written as

$$z_f = \tilde{f}(x,y) \quad \text{and} \quad z_g = \tilde{g}(x,y), \tag{4}$$

where x, y are Cartesian coordinates in the common tangent plane $T_{\mathbf{p}}(Z(f)) = T_{\mathbf{p}}(Z(g))$. Clearly, $\nabla \tilde{f}(\mathbf{p}) = \nabla \tilde{g}(\mathbf{p}) = \mathbf{0}$. Consequently, using the Taylor expansions of $\tilde{f}$ and $\tilde{g}$ at $\mathbf{p} = (0,0)$ in local coordinates, we get

$$z_f(x,y) - z_g(x,y) = \tilde{f}(x,y) - \tilde{g}(x,y) = \sum_{i=2}^{\infty} \tilde{f}_i(x,y) - \tilde{g}_i(x,y), \tag{5}$$

where $\tilde{f}_i$ and $\tilde{g}_i$ are homogeneous polynomials of degree i. Note that in case of lower differentiability of f or g, the sum on the right of (5) has to be replaced with the finite sum and an appropriate Taylor's remainder.

The intersection curve may have several branches. The possible tangent directions of the branches at the point $\mathbf{p}$ are the solutions of the homogeneous polynomial equation

$$\tilde{f}_k(x,y) - \tilde{g}_k(x,y) = 0 \tag{6}$$

for $k = \min\{i \colon \tilde{f}_i(x,y) - \tilde{g}_i(x,y) \neq 0\}$. Consequently, k is an upper bound for the number of possible curve branches at the point $\mathbf{p}$.

For the case $k = 2$, $\tilde{f}_2(x,y)$ and $\tilde{g}_2(x,y)$ are polynomials representing (up to a constant) the Dupin's indicatrices of the given surfaces at $\mathbf{p}$. Then,

$$D_f(\mathbf{p}) := \{(x,y) \colon \tilde{f}_2(x,y) = \pm 1\} \quad \text{and} \quad D_g(\mathbf{p}) := \{(x,y) \colon \tilde{g}_2(x,y) = \pm 1\} \tag{7}$$

with an appropriate choice of the signs. The tangential directions of the curve C at point $\mathbf{p}$ must be determined by the points, where $D_f(\mathbf{p})$ intersects $D_g(\mathbf{p})$ (see Figure 1). This connects two concepts from differential geometry and algebraic geometry.

If a singular point of the intersection curve is also a singular point of the surface $Z(f)$ or $Z(g)$, we can detect it by computing zero set of the system of equations

$$f = 0, \quad g = 0, \quad \text{and} \quad \nabla f = \mathbf{0} \quad \text{or} \quad \nabla g = \mathbf{0}. \tag{8}$$

Generically, it is a 0-dimensional set (a finite number of points). Consequently, we can use Bézier clipping algorithm (see [10]), once we have a bounding volume containing a unique singular point. Clearly, if the set of the singular points has a non-zero dimension (an algebraic curve provided f and g are polynomials), we need to use an additional constraints to compute the singular points (e.g. a moving hyperplane).

2.3 Orthogonalization

An important issue in our algorithm is the approximation of the Euclidean distance of the point $\mathbf{x}$ to the intersection curve C. In order to compute it efficiently, one may preprocess the two given functions f and g.

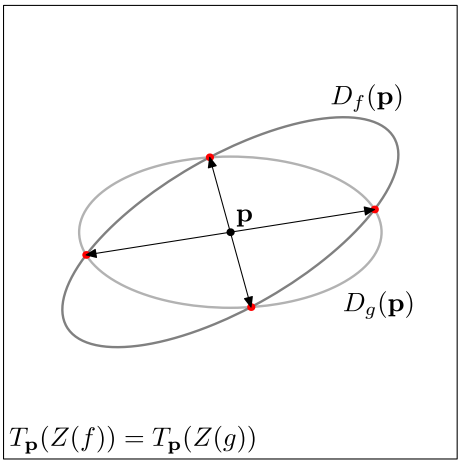

Fig. 1: The tangential directions at the point $\mathbf{p}$ of intersection with the coincident tangent planes – the simplest case.

Lemma 1. *At any regular point* $\mathbf{x}$ *of the intersection curve* $C(f, g)$*, the two functions*

$$\bar{F}(\mathbf{x}) = f(\mathbf{x})\ \|\nabla g(\mathbf{x})\| + g(\mathbf{x})\ \|\nabla f(\mathbf{x})\| \tag{9}$$

$$\bar{G}(\mathbf{x}) = f(\mathbf{x})\ \|\nabla g(\mathbf{x})\| - g(\mathbf{x})\ \|\nabla f(\mathbf{x})\| \tag{10}$$

have mutually perpendicular gradients, $\nabla \bar{F}(\mathbf{x}) \cdot \nabla \bar{G}(\mathbf{x}) = 0$*, and the two functions*

$$F(\mathbf{x}) = \frac{\bar{F}(\mathbf{x})}{\|\nabla \bar{F}(\mathbf{x})\|}, \quad G(\mathbf{x}) = \frac{\bar{G}(\mathbf{x})}{\|\nabla \bar{G}(\mathbf{x})\|} \tag{11}$$

even have unit and mutually perpendicular gradients.

Proof. These observations can be verified by direct computations. □

Figure 2 is an illustration of the orthogonalization for two quadrics. The tubular surface is an isosurface of the Sampson distance to the intersection curve of the quadrics. After the orthogonalization, the shape of the isosurface is locally more cylindric tube.

Clearly, even if the function f and g are polynomials, neither $\bar{F}$ and $\bar{G}$ nor F and G are polynomials in general. It could be of some interest to study pairs of algebraic surfaces which intersect each other orthogonally.

The orthonormal gradient frame defined by ∇F and ∇G may still rotate around the curve. Here one may ask for similar ways of modifying them, which leads to particular frames of the curve, such as the Frenet frame or a rotation-minimizing frame.

Now we consider a well–known first order approximation [11] of the oriented distance function $\text{dist}_{\text{or}}(\mathbf{x}, Z(f))$ of a point $\mathbf{x}$ from a given implicitly defined surface $Z(f)$.

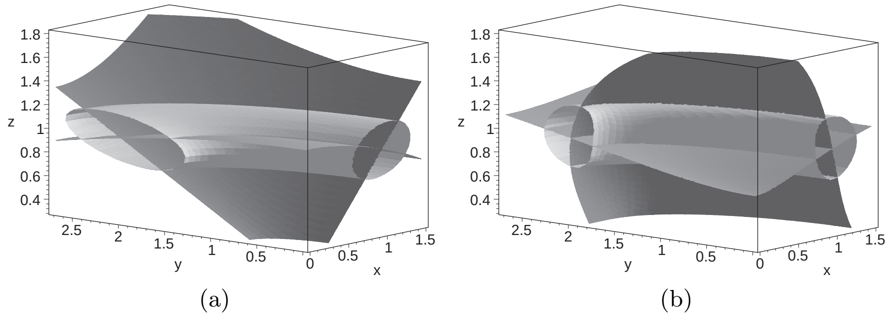

Fig. 2: Orthogonalization: (a) original quadric; (b) orthogonalized surfaces. The tubular-like surface is an isosurface of the squared Sampson distance to the intersection curve of the other two surfaces.

Lemma 2. *Consider a C^3 function f, whose gradients satisfy $0 < m < ||\nabla f(\mathbf{x})|| < M$ for all points within a certain neighborhood $N(\mathbf{p})$ of a point $\mathbf{p} \in Z(f)$, where m and M are constants. Let $\operatorname{Hess} f(\mathbf{x})$ be the Hessian matrix of f at $\mathbf{x}$. We assume the eigenvalues of $\operatorname{Hess} f(\mathbf{x})$ are bounded in the neighborhood of $N(\mathbf{p})$. Moreover, let any $\mathbf{x} \in N(\mathbf{p})$ have a unique footpoint[3] $\mathbf{x}_0$ on $Z(f)$. The oriented distance function*

$$\operatorname{dist}_{\mathrm{or}}(\mathbf{x}, Z(f)) = \operatorname{sgn}(f(\mathbf{x}))\, ||\mathbf{x} - \mathbf{x}_0|| \tag{12}$$

satisfies

$$\operatorname{dist}_{\mathrm{or}}(\mathbf{x}, Z(f)) = \frac{f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|} + \mathcal{O}(d^2) \tag{13}$$

with $d = ||\mathbf{x} - \mathbf{x}_0||$.

Proof. Consider the function

$$L(t) = \frac{f(\mathbf{x}(t))}{\|\nabla f(\mathbf{x}(t))\|} \tag{14}$$

where

$$\mathbf{x}(t) = \mathbf{x}_0 + t\nabla f(\mathbf{x}_0) \tag{15}$$

where $\mathbf{x}_0 \in N(\mathbf{p}) \cap Z(f) \subset E^3$.

The Taylor expansion of the function $L(t)$ of order 2 at $t = 0$ is

$$L(t) = t\|\nabla f(\mathbf{x}_0)\| - \frac{t^2}{2}\nabla f(\mathbf{x}_0)^\top \operatorname{Hess}(f(\mathbf{x}_0))\nabla f(\mathbf{x}_0)\frac{1}{\|\nabla f(\mathbf{x}_0)\|} + \mathcal{O}(t^3). \tag{16}$$

The oriented distance equals

$$\operatorname{dist}_{\mathrm{or}}(\mathbf{x}(t), Z(f)) = t\|\nabla f(\mathbf{x}_0)\| \tag{17}$$

[3] The point $\mathbf{x}^* \in Z(f)$ is said to be a footpoint of a point $\mathbf{x} \in \mathbb{R}^3$, if $\mathbf{x} - \mathbf{x}^*$ is perpendicular to the surface and $\|\mathbf{x} - \mathbf{x}^*\|$ is minimized. The footpoint need not be unique.

for $\mathbf{x}(t) \in N(\mathbf{p})$. Since $\nabla f(\mathbf{x}_0)^\top \mathrm{Hess}\,(f(\mathbf{x}_0))\nabla f(\mathbf{x}_0)$ is bounded in $N(\mathbf{p})$ and $t \in \mathcal{O}(d)$, we get

$$\mathrm{dist}_{\,\mathrm{or}}(\mathbf{x}, Z(f)) = \frac{f(\mathbf{x})}{\|\nabla f(\mathbf{x})\|} + \mathcal{O}(d^2). \tag{18}$$

for each point $\mathbf{x}$ in $N(\mathbf{p})$. $\square$

This observation can be generalized to intersection curves C, as follows.

Lemma 3. *For any two real functions f and g, consider the functions F and G defined in Lemma 1. We assume that the intersection curve C is regular in a certain neighborhood $N(\mathbf{p})$ of a point $\mathbf{p} \in C$, and that any point $\mathbf{y} \in N(\mathbf{p})$ has a unique footpoint $\mathbf{y}_0$ on C. Then*

$$||\mathbf{y} - \mathbf{y}_0||^2 = F(\mathbf{y})^2 + G(\mathbf{y})^2 + \mathcal{O}(||\mathbf{y} - \mathbf{y}_0||^3). \tag{19}$$

Proof. For any parameterization $\mathbf{x}(s)$ of the intersection curve, we may parameterize a tubular neighborhood by

$$\mathbf{h}(s, t, u) = \mathbf{x}(s) + t\,\nabla F(\mathbf{x}(s)) + u\,\nabla G(\mathbf{x}(s)), \tag{20}$$

and its points satisfy

$$||\mathbf{h}(s, t, u) - \mathbf{x}(s)||^2 = t^2 + u^2, \tag{21}$$

due to the properties of F and G. In addition, $f(\mathbf{x}(s))$ is also the footpoint of $\mathbf{h}(s, t, u)$. The result (19) follows by analyzing the Taylor expansion of the function $F(\mathbf{h}(s, t, u))^2 + G(\mathbf{h}(s, t, u))^2$ in a similar way as in the proof of lemma 2. $\square$

3 Approximate Parameterization of Intersection Curve

3.1 Tracing of the Curve

For the sake of completeness, we outline here a tracing algorithm of Runge-Kutta type. The curve tracing is used in several contexts during the approximation algorithm. We need it to construct an initial solution of the optimization algorithm, for finding suitable boundary conditions for the system of ordinary differential equations (ODE) which are satisfied by predictor and for the computation of the predictor itself.

If we are to solve numerically the ODE

$$\dot{\mathbf{y}} = \phi(x, \mathbf{y}) \tag{22}$$

with boundary conditions

$$\mathbf{y}(0) = \mathbf{y}_0, \tag{23}$$

we can use the following sequence of approximations

$$\mathbf{y}(h) \approx \mathbf{y}(0) + h\frac{1}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4) \tag{24}$$

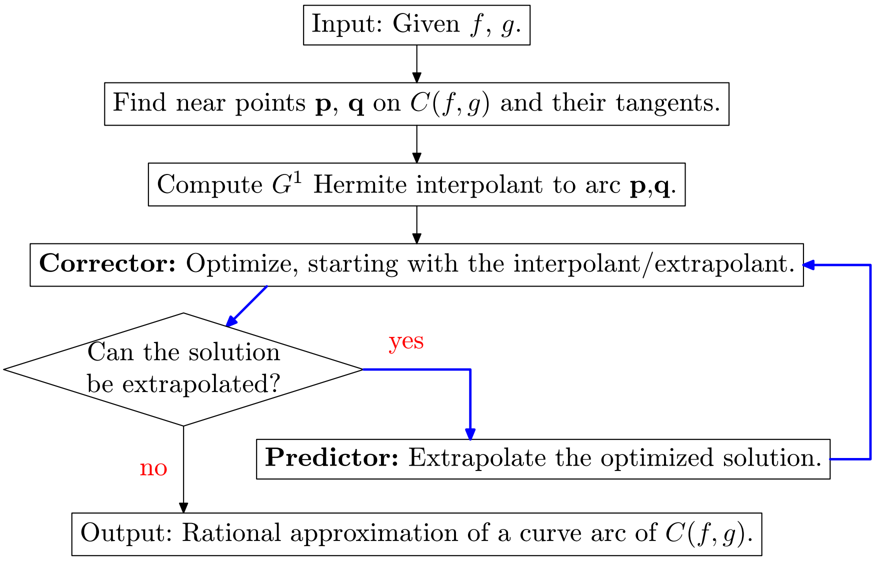

Fig. 3: Approximation algorithm for an intersection curve.

where

$$\mathbf{k}_1 = \phi(x, \mathbf{y}(x)) \tag{25}$$
$$\mathbf{k}_2 = \phi(x + 0.5h, \mathbf{y}(x) + 0.5h\mathbf{k}_1) \tag{26}$$
$$\mathbf{k}_3 = \phi(x + 0.5h, \mathbf{y}(s) + 0.5h\mathbf{k}_2) \tag{27}$$
$$\mathbf{k}_4 = \phi(x + h, \mathbf{y}(s) + h\mathbf{k}_3). \tag{28}$$

The method is known to have approximation of order 4 (see e.g. [12]).

3.2 Outline of the Method

The approximation algorithm consists of the following three steps (see Figure 3):

1. **Initial predictor**. Find two close regular points $\mathbf{p}, \mathbf{q}$ with their tangent vectors, and compute the cubic Hermite interpolant to these data.
2. **Corrector.** The curve segment $\widehat{\mathbf{pq}}$ of C is approximated by a rational Bézier segment satisfying G^1 endpoints conditions.
3. **Generic predictor.** The solution is extrapolated, in order to get a new initial solution for the corrector step.

Steps 2 and 3 are iterated until certain termination conditions are satisfied. They are discussed in section 3.5.

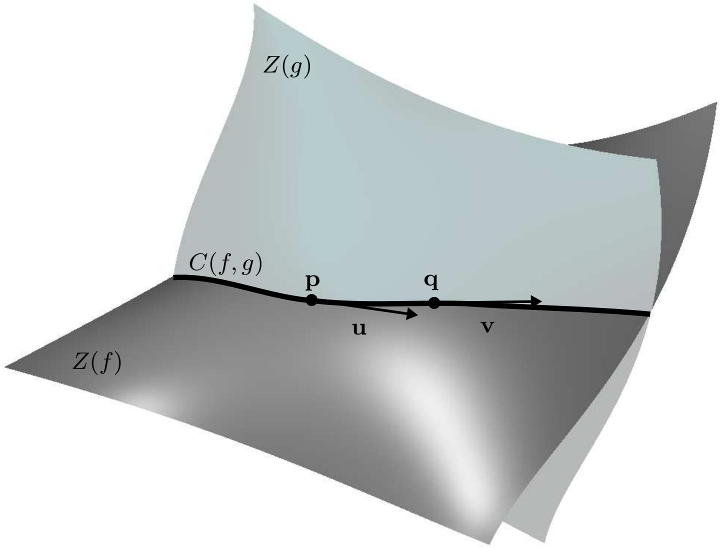

Fig. 4: Initial set of data for a cubic Hermite interpolation

3.3 Initial predictor

The approximate curve is represented as a rational Bézier curve of degree n in the form $(x_1(t)/x_0(t), x_2(t)/x_0(t), x_3(t)/x_0(t))$, where

$$(x_1(t), x_2(t), x_3(t), x_0(t))^\top = \tilde{\mathbf{x}}(t) = \sum_{i=0}^{n} B_i^n(t)\tilde{\mathbf{p}}_i, \tag{29}$$

with the control points

$$\tilde{\mathbf{p}}_i = (b_{4i}, b_{4i+1}, b_{4i+2}, b_{4i+3})^\top \in \mathbb{R}^4 - \{\mathbf{0}\} \tag{30}$$

(in homogeneous coordinates) and the Bernstein polynomials

$$B_i^n(t) = \binom{n}{i}(1-t)^{n-i}t^i, \quad i = 0, 1, \ldots, n. \tag{31}$$

For later reference, we collect all control points into the vector

$$\mathbf{b} = (b_0, \ldots, b_{4n+3})^\top \in \mathbb{R}^{d_n} \tag{32}$$

with the dimension $d_n = 4n + 4$.

The initial curve starts from the given point $\mathbf{p}$ in the direction of unit tangent vector $\mathbf{u}$. We trace a small segment of the intersection curve, in order to get a second point $\mathbf{q}$ with unit tangent vector $\mathbf{v}$, see Figure 4. The two tangent vectors should satisfy

$$\mathbf{u} \cdot \mathbf{v} \approx 1. \tag{33}$$

These G^1 boundary data are used to define an integral cubic Bézier curve, where the length of the boundary derivative vector is chosen to be $||\mathbf{p} - \mathbf{q}||$.

Note that a parameterization of a rational curve is not unique. During the approximation algorithm, we always normalize the curve to the standard form (for more information see [13]), where the boundary weights are equal to

$$b_3 = b_{4n+3} = 1. \tag{34}$$

This defines a map

$$sf \colon \mathbb{R}^{d_n} \to \mathbb{R}^{d_n}. \tag{35}$$

In the case of the initial solution, which is an integral curve,

$$b_{4i+3} = 1 \quad \text{for} \quad i = 0, \ldots, n. \tag{36}$$

3.4 Corrector: Optimization of segment

We consider all normalized rational Bézier curves of degree n, which satisfy the G^1 boundary conditions

$$\mathbf{x}(0) = \mathbf{p} \tag{37}$$

$$\mathbf{x}(1) = \mathbf{q} \tag{38}$$

$$\dot{\mathbf{x}}(0) = \lambda \mathbf{u} \quad \text{for} \quad \lambda > 0 \tag{39}$$

$$\dot{\mathbf{x}}(1) = \mu \mathbf{v} \quad \text{for} \quad \mu > 0. \tag{40}$$

The set of all feasible curves is denoted Φ. It can be shown to be a convex set. Among all curves in Φ, we minimize the distance of the curve $\mathbf{x}(t)$ to C on the parameter domain $[0, 1]$, using a variational approach.

The objective function has the form

$$H(\mathbf{b}) = H_0(\mathbf{b}) + w_1 H_1(\mathbf{b}) + w_2 H_2(\mathbf{b}) \tag{41}$$

with

$$H_0(\mathbf{b}) = \int_0^1 F(\mathbf{x}(t))^2 + G(\mathbf{x}(t))^2 \mathrm{d}t, \tag{42}$$

$$H_1(\mathbf{b}) = \int_0^1 h(x_0(t)) \mathrm{d}t, \tag{43}$$

where

$$h(t) = (t-1)^8, \tag{44}$$

and the regularization term

$$H_2(\mathbf{b}) = \sum_{i=0}^{n-1} ||\tilde{\mathbf{p}}_{i+1} - \tilde{\mathbf{p}}_i||^2. \tag{45}$$

As an alternative term for the regularization, we can use the term

$$H_{2'}(\mathbf{b}) = \sum_{i=0}^{n-2} \|\tilde{\mathbf{p}}_{i+2} - 2\tilde{\mathbf{p}}_{i+1} + \tilde{\mathbf{p}}_i\|^2. \tag{46}$$

Currently, the weights are chosen by the user, such that they satisfy

$$0 < w_1 \ll w_2 \ll 1. \tag{47}$$

They are semi-automatically adjusted during the optimization.

The second term H_1 measures and penalizes the deviation of the rational Bézier curve $\mathbf{x}(t)$ from a polynomial one. The last term of the objective function penalizes the length of the control polygon in homogeneous coordinates of the rational Bézier curve and provides a regularization of the objective function.

The objective function is non-linear. In order to find a minimum, we use an SQP method. The technical details are described in Appendix A.

An example obtained by the approximate parameterization is shown in Figure 5. Two quadrics intersect in an algebraic curve of order 4. We have approximated it using a rational cubic curve. The thick white part of the curve is the image of the interval $[0, 1]$.

3.5 Predictor of the extrapolation

We use two types of predictor methods. The first one uses de Casteljau algorithm expansion of the approximation at the endpoint. The second approach uses Runge-Kutta method for evolution of the system of ODE.

Casteljau extension predictor. Let $\boldsymbol{\rho}$ be the vector of control variables for an optimal segment on $[0, 1]$ for $\widehat{\mathbf{pq}}$. We compute the control points of the same curve over the interval $[0, 1 + \epsilon]$ using de Casteljau's algorithm. After transformation into standard form, we get a new rational Bézier curve $\boldsymbol{\rho}_\epsilon$. Then, we project its new endpoint onto the intersection curve C and modify the control polygon so as to match the G^1 boundary conditions at the new projected endpoint. We denote this curve as $\boldsymbol{\rho}^*_\epsilon$. If the boundary conditions of $sf(\boldsymbol{\rho}^*_\epsilon)$ are satisfied, we proceed with corrector step.

However, the above extension method may lead to an infeasible curve with respect to the boundary conditions (39) and (40). Hence, the extension parameter ϵ has to be decreased. We then use bisection towards 0.

Additionally, we require to find a convenient extension with respect to the value of the objective function. Since the projection on the intersection curve C and its standardization is continuous with respect to ϵ for certain $0 < \epsilon < \epsilon_0(\mathbf{p}_n)$, we can find such a value of ϵ that

$$sf(\boldsymbol{\rho}^*_\epsilon) \in N(\boldsymbol{\rho}) \cap \Phi \subset \mathbb{R}^{d_n} \tag{48}$$

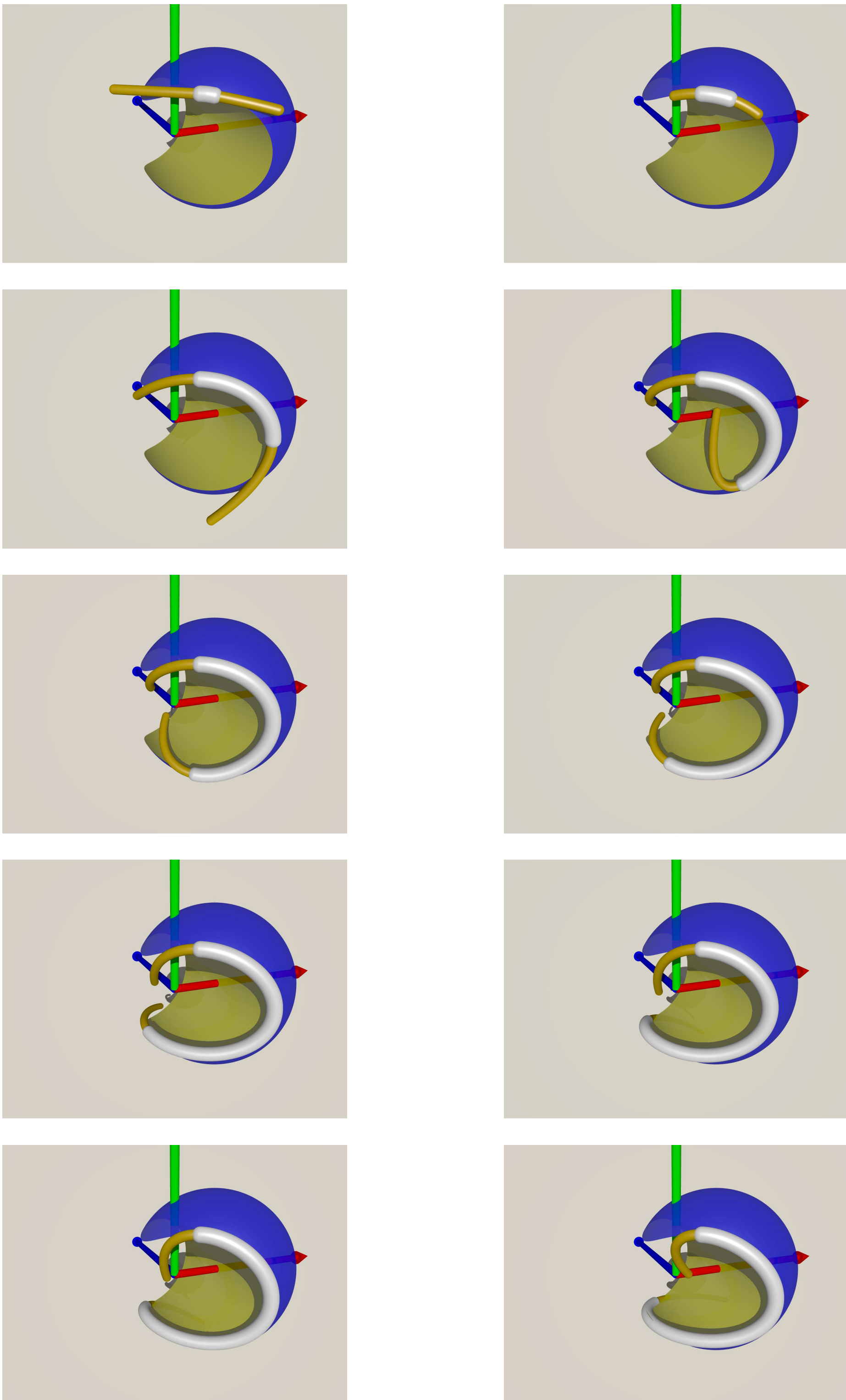

Fig. 5: Approximation of the intersection of sphere and cylinder by a rational cubic. The white part of the curve is parameterized over the interval $I = [0, 1]$. The right endpoint extends via predictor-corrector algorithm along the intersection curve.

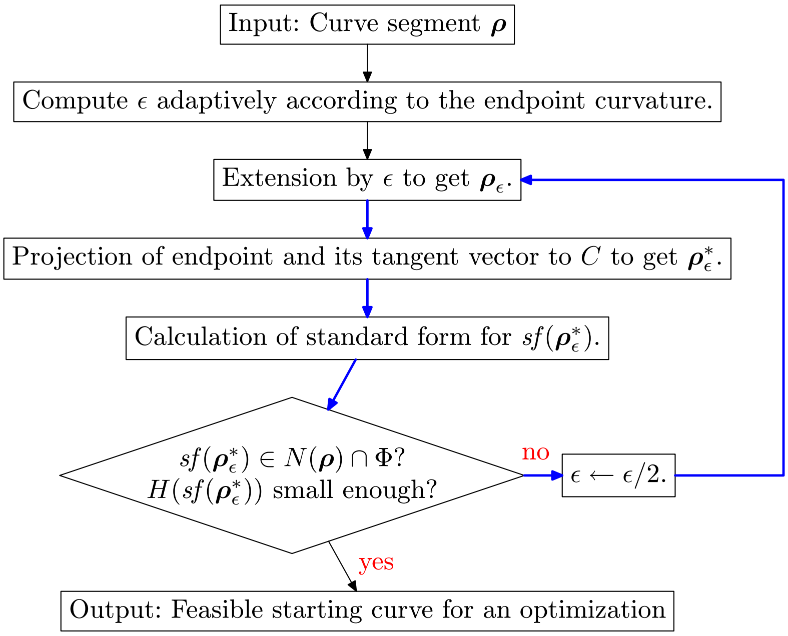

Fig. 6: The scheme of the de Casteljau predictor algorithm

with $N(\boldsymbol{\rho}) \subset \mathbb{R}^{d_n}$ a suitable neighborhood of the vector $\boldsymbol{\rho}$ and moreover the value of the value of the objective function $H(sf(\boldsymbol{\rho}^*_\epsilon))$ does not increase over the prescribed threshold.

If these conditions are not satisfied or ϵ is under the minimal threshold, we terminate the extension process. The next piece of the approximation curve can be started here.

The ϵ in the first step is chosen according to the curvature of the curve $\boldsymbol{\rho}$ at the extended endpoint. A suitable choice is e.g.

$$\epsilon = \min\{\epsilon_{\max}, \frac{\sqrt{3}}{3\kappa_1}\} \tag{49}$$

where $\epsilon_{\max}$ is a user-defined maximum step and κ_1 is curvature of $\boldsymbol{\rho}$ at the extrapolated endpoint.

The algorithm is depicted on Figure 6.

ODE extension predictor. The extension of the approximate parameterization is done by an evolution process. Let $\mathbf{b}(s)$ be a minimum of H for each $s \in [s_0, s_1]$, that is $H(\mathbf{b}(s)) \to \min$. The optimality condition leads to a system of ODE with respect to a certain minimum set of parameters $\mathbf{r}(s)$. We get a system of ordinary differential equations

$$C(\mathbf{r}, s)\dot{\mathbf{r}}(s) + D(\mathbf{r}, s)\mathbf{r}(s) + E(\mathbf{r}, s) = \mathbf{0}, \tag{50}$$

where $C(\mathbf{r}, s)$, $D(\mathbf{r}, s)$ and $E(\mathbf{r}, s)$ are certain matrices dependent on the boundary conditions. The whole computation can be found in Appendix B.

The approximation of the solution for the parameter $s+h$ is found using the Runge-Kutta method described in Section 3.1, where

$$\phi(\mathbf{r}, s) = -C(\mathbf{r}, s)^{-1} D(\mathbf{r}, s)\mathbf{r} - C(\mathbf{r}, s)^{-1} E(\mathbf{r}, s) \tag{51}$$

provided that the matrix $C(r, s)$ is regular. The regularity is guaranteed under certain assumptions.

Lemma 4. *If w_2 is sufficiently large, the matrix $C(r, s)$ in* (51) *is regular for the approximation with cubic rational curves.*

The proof of the lemma 4 is somewhat long and technical and can be found in appendix.

The computation of the terms (25)–(28) require the knowledge of the boundary conditions for the parameters $s + 0.5h$ and $s + h$. They can be gained via solution of the another system of differential equations, which describes the arc length parameterization of the intersection curve. We use again a Runge-Kutta method for the approximation of the solution.

Summing up, the de Casteljau's predictor method needs projection of the boundary data back to the curve and it is a heuristic approach. The ODE approach is more complex and maintains the boundary conditions.

The stability of the singular points computation strongly depends on the floating point arithmetic used as well as the structure of the singularity. Hence, the symbolic methods to compute common tangents at the point $\mathbf{p}$ are an option in this case.

4 Examples

See Figure 7 for an example of the approximation starting at the singular point. The sphere and cylinder in special position intersect at a singular curve (Viviani's window).

The approximation error of the curve is visualized using the porcupine plot in Figure 8. The porcupine represents the distance of the corresponding point of the approximation curve toward the given curve C multiplied by a large constant.

5 Conclusion

We have presented an algorithm for the approximate parameterization of the intersection curves of two implicitly defined surfaces. If the surfaces intersect transversally, we can define locally cylindrical coordinates, which approximate the square of the Euclidean distance. The modified functions can then be used to compute approximate parameterization as an optimization problem, where the objective function uses the approximation of the Euclidean distance as one of its terms.

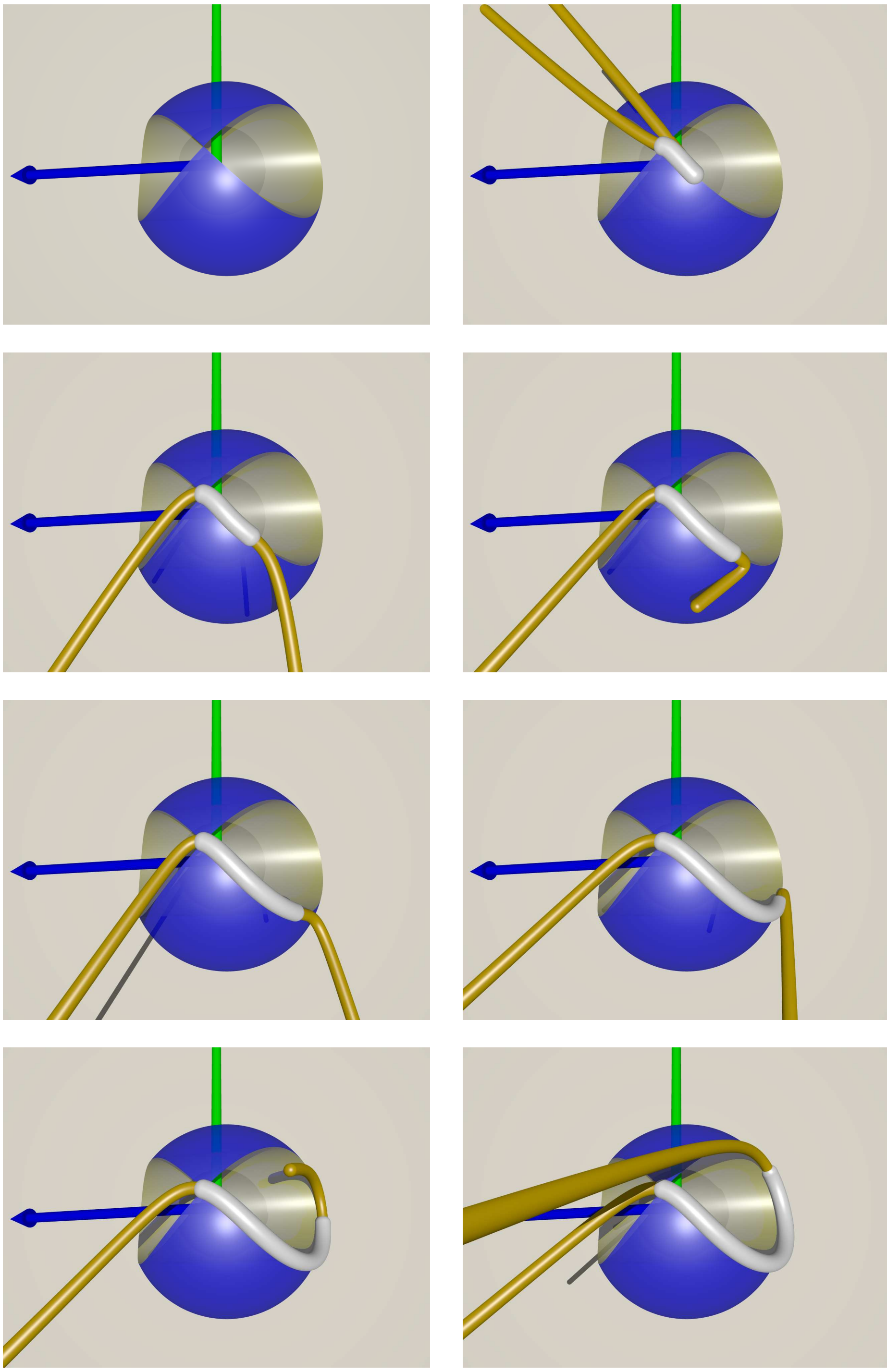

Fig. 7: Approximation of the intersection of sphere and cylinder starting from a singularity.

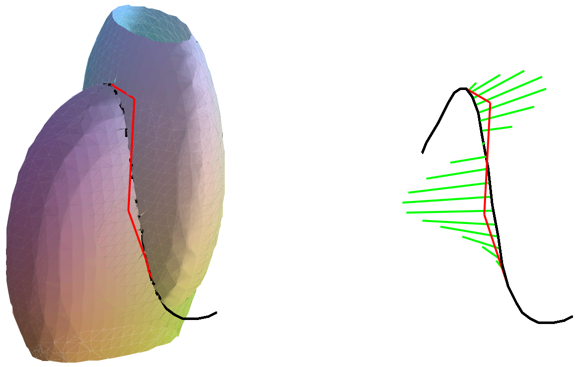

Fig. 8: Porcupines of the cubic curve approximating the quartic intersection of two quadrics. The error is multiplied by 0.4×10^4 for visualization purpose.

We provided examples demonstrating the usefulness of the method even in the case of occurrence of simple singularities along the intersection curve.

According to our numerical experiments, the reproduction of the polynomial/rational parameterization of the intersection curve is expected, but no theoretical results are known. This may be a subject of further investigations.

A SQP Optimization of Objective Function

We present technical details of the optimization of the objective function (41).

We look for the curve segment represented by a vector $\boldsymbol{\rho}$ satisfying

$$\boldsymbol{\rho} = \min_{\mathbf{b} \in \mathbb{R}^{d_n}} H(\mathbf{b}), \tag{52}$$

where the boundary conditions (34), (37)–(40) are satisfied.

Since the objective function (41) is non-linear, we use an SQP method for a numerical solution of the system. The stationary point of the objective function satisfies the equations

$$\frac{\partial H(\mathbf{b})}{\partial b_i} = 0 \quad \text{for} \quad i = 0, \ldots, d_n \tag{53}$$

and the boundary conditions (34), (37)–(40). Using them, we get

$$\tilde{\mathbf{p}}_0 = \begin{pmatrix} \mathbf{p} \\ 1 \end{pmatrix}, \tag{54}$$

$$\tilde{\mathbf{p}}_1 = b_7 \begin{pmatrix} \mathbf{p} \\ 1 \end{pmatrix} + \lambda \frac{1}{n} \begin{pmatrix} \mathbf{u} \\ 0 \end{pmatrix} \tag{55}$$

and

$$\tilde{\mathbf{p}}_n = \begin{pmatrix} \mathbf{q} \\ 1 \end{pmatrix}, \tag{56}$$

$$\tilde{\mathbf{p}}_{n-1} = b_{4n-1} \begin{pmatrix} \mathbf{q} \\ 1 \end{pmatrix} - \mu \frac{1}{n} \begin{pmatrix} \mathbf{v} \\ 0 \end{pmatrix}. \tag{57}$$

Hence, the vector $\mathbf{b}$ depends linearly on variables

$$\mathbf{r} = (r_0, \ldots, r_{4n-9})^\top \tag{58}$$

with

$$r_0 = \lambda, \quad r_1 = b_7, \quad r_{4n-10} = b_{4n-1}, \quad r_{4n-9} = \mu$$

and

$$r_i = b_{i+6} \quad \text{for} \quad i = 2, \ldots, 4n-11.$$

The set of feasible vectors is restricted by inequalities

$$\lambda > 0 \quad \text{and} \quad \mu > 0 \tag{59}$$

in order to preserve the initial orientation of the endpoint tangent vectors. Alternatively, we can use λ^2 and μ^2 in the boundary conditions (39) and (40) respectively with an additional assumption $\lambda, \mu \neq 0$.

Since the values of $\tilde{\mathbf{p}}_0$ and $\tilde{\mathbf{p}}_n$ are constant during the optimization, we have

$$\mathbf{b}(\mathbf{r}) = A\mathbf{r} + \mathbf{b}_{\mathrm{ep}}, \tag{60}$$

i.e. a linear change of variables given by the matrix A of type $(4n+3) \times (4n-8)$, where

$$A(s) = \begin{pmatrix} A_{11} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & A_{22} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & A_{33} \end{pmatrix} \tag{61}$$

with the blocks

$$A_{11} = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \tilde{\mathbf{u}} & \tilde{\mathbf{p}}_0 \end{pmatrix}, \quad A_{33} = \begin{pmatrix} \tilde{\mathbf{p}}_n & -\tilde{\mathbf{v}} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \quad A_{22} = I, \tag{62}$$

$$\mathbf{b}_{\mathrm{ep}} = (\tilde{\mathbf{p}}^\top, 0, \ldots, 0, \tilde{\mathbf{q}}^\top)^\top. \tag{63}$$

Then

$$\nabla_{\mathbf{r}} H(\mathbf{b}(\mathbf{r})) = A^\top \nabla_{\mathbf{b}} H(\mathbf{b}) \tag{64}$$

and

$$\nabla^2_{\mathbf{r}} H(\mathbf{b}(\mathbf{r})) = A^\top \nabla^2_{\mathbf{b}} H(\mathbf{b}) A. \tag{65}$$

We use a SQP method for the numerical solution of (52). Let $\mathbf{r}^0$ be an initial curve represented as a vector of the type (58).

We construct a sequence of feasible curves

$$\{\mathbf{r}^i\}_{i=0}^{\infty} \tag{66}$$

such that

$$\mathbf{r}^{i+1} = \mathbf{r}^i + \gamma_i (\mathbf{r}^i - \operatorname{sgn}(\nabla_r H(\mathbf{b}(\mathbf{r}^i)) \cdot \mathbf{q}^i) \mathbf{q}^i, \tag{67}$$

where

$$\mathbf{q}^i = \nabla^2_{\mathbf{r}} H(\mathbf{b}(\mathbf{r}^i))^{-1} \nabla H(\mathbf{b}(\mathbf{r}^i)) \tag{68}$$

with the values $\gamma_i \in (0, 1]$ so that

$$H(\mathbf{b}(\mathbf{r}^i)) > H(\mathbf{b}(\mathbf{r}^{i+1})) \tag{69}$$

for $i = 0, 1, \ldots$. The values γ_i are computed by bisection in each step.

Starting with a value close to the minimum of H, this sequence converges to the minimum. Then

$$\boldsymbol{\rho} = \lim_{i\to\infty} \mathbf{b}(\mathbf{r}^i). \tag{70}$$

Since the set Φ of feasible solutions has a boundary, we need to address the case when λ or μ converge to zero. In our case, we terminated the optimization if their values were under a prescribed threshold.

B ODE derivation

Let $H(\mathbf{b})$ be the objective function. We define the approximation of the segment of the curve C as

$$\mathbf{b}(s) = (b_0(s), \ldots, b_{4n+3}(s))^\top = (\tilde{\mathbf{p}}_0(s)^\top, \ldots, \tilde{\mathbf{p}}_n(s)^\top)^\top, \tag{71}$$

where $\tilde{\mathbf{p}}_i(s) = (b_{4i}(s), b_{4i+1}(s), b_{4i+2}(s), b_{4i+3}(s))^\top$ for $i = 0, 1, \ldots, n$.

We compute the minimum of $H(\mathbf{b}(s))$ for each each $s \in [s_0, s_1]$ such that $\mathbf{b}(s)$ satisfies the boundary conditions

$$\tilde{\mathbf{p}}_0(s) = (\mathbf{p}(s)^\top, 1)^\top \tag{72}$$

$$\tilde{\mathbf{p}}_n(s) = (\mathbf{q}(s)^\top, 1)^\top \tag{73}$$

$$\tilde{\mathbf{p}}_1(s) = b_7(s)\tilde{\mathbf{p}}_0(s) + \lambda(s)\tilde{\mathbf{u}}(s) \tag{74}$$

$$\tilde{\mathbf{p}}_{n-1}(s) = b_{4n-1}(s)\tilde{\mathbf{p}}_n(s) - \mu(s)\tilde{\mathbf{v}}(s) \tag{75}$$

with

$$\tilde{\mathbf{u}}(s) = (\mathbf{u}(s)^\top, 0)^\top, \quad \tilde{\mathbf{v}}(s) = (\mathbf{v}(s)^\top, 0)^\top, \quad \lambda(s) > 0 \quad \text{and} \quad \mu(s) > 0 \tag{76}$$

where $\mathbf{u}(s)$, $\mathbf{v}(s)$ are unit tangent vectors of the curve C at point $\mathbf{u}(s)$, $\mathbf{v}(s)$ respectively for $s \in [s_0, s_1]$.

Let $\mathbf{q}(s)$ be the arc length parameterization of certain arc of C starting at $\mathbf{q}(s_0)$ and $\mathbf{v}(s_0) = \mathbf{q}'(s_0)$ is a unit tangent vector of the curve C (these are boundary conditions). The differential equation for these functions is

$$\mathbf{v}(s) = \mathbf{q}'(s) \tag{77}$$

where $\mathbf{v}(s)$ is given by (3) substituting $\mathbf{q}(s)$ for $\mathbf{p}$.

Since (77) is non-linear in general, the boundary conditions for the evolution can be computed only numerically. We use again a Runge-Kutta method from Section 3.1 with $\Phi(\mathbf{q}, s) = \mathbf{v}(s)$.

The boundary conditions (72)–(75) determine a linear change of coordinates given by

$$\mathbf{b}(s) = A(s)\mathbf{r}(s) + \mathbf{b}_{\text{ep}}(s) \tag{78}$$

where

$$A(s) = \begin{pmatrix} A_{11}(s) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & A_{22}(s) & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & A_{33}(s) \end{pmatrix} \tag{79}$$

with the blocks

$$A_{11}(s) = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \tilde{\mathbf{u}}(s) & \tilde{\mathbf{p}}_0(s) \end{pmatrix}, \quad A_{33}(s) = \begin{pmatrix} \tilde{\mathbf{p}}_n(s) & -\tilde{\mathbf{v}}(s) \\ \mathbf{0} & \mathbf{0} \end{pmatrix}, \quad A_{22}(s) = I, \tag{80}$$

and

$$\mathbf{b}_{\mathrm{ep}}(s) = (\tilde{\mathbf{p}}(s)^\top, 0, \ldots, 0, \tilde{\mathbf{q}}(s)^\top)^\top. \tag{81}$$

where $\mathbf{r}(s) = (r_0(s), \ldots, r_{4n-9}(s))^\top$ is given by

$$r_0(s) = \lambda(s), \quad r_1(s) = b_7(s), \quad r_{4n-10}(s) = b_{4n-1}(s), \quad r_{4n-9}(s) = \mu(s)$$

and

$$r_i(s) = b_{i+6}(s) \quad \text{for} \quad i = 2, \ldots, 4n-11$$

After the substitution into the objective function H, we get a constrained objective function

$$\bar{H}(\mathbf{r}(s), s) = H(\mathbf{b}(s)). \tag{82}$$

The stationary points of the objective function (82) for fixed s are the solutions of the system

$$\frac{\partial \bar{H}(\mathbf{r}(s), s)}{\partial r_i} = 0 \quad \text{for} \quad i = 0, \ldots, 4n-9. \tag{83}$$

Using (82), we have

$$\nabla_{\mathbf{r}} \bar{H}(\mathbf{r}(s), s) = A(s)^\top \nabla_{\mathbf{b}} H(A(s)\mathbf{r}(s) + \mathbf{b}_{\mathrm{ep}}(s)) \tag{84}$$

Since these equations should hold for every s in the considered interval, we have the equations for the evolution of the vector of control functions $\mathbf{r}(s) = (r_0(s), \ldots, r_{4n-9}(s))^\top$ given as

$$\frac{\partial^2 \bar{H}(\mathbf{r}(s), s)}{\partial r_i \partial s} = 0. \tag{85}$$

Let $\partial_s = \frac{\partial}{\partial s}$ and $\dot{\mathbf{r}}(s) = \partial_s \mathbf{r}(s)$. The straightforward calculation gives

$$\begin{aligned} \mathbf{0} = \partial_s \nabla_{\mathbf{r}} \bar{H}(\mathbf{r}(s), s) &= \partial_s A(s)^\top \nabla H(\mathbf{b}(s)) + A(s)^\top \nabla^2_{\mathbf{b}} H(\mathbf{b}(s)) \partial_s A(s) \mathbf{r}(s) + \\ &\quad A(s)^\top \nabla^2_{\mathbf{b}} H(\mathbf{b}(s)) A(s) \partial_s \mathbf{r}(s) + A(s)^\top \nabla^2_{\mathbf{b}} H(\mathbf{b}(s)) \partial_s \mathbf{b}_{\mathrm{ep}}(s) \\ &= C(s)\dot{\mathbf{r}}(s) + D(s)\mathbf{r}(s) + E(s) \end{aligned}$$

Such a system of ODE can be solved via Runge-Kutta method, provided $C(s)$ is regular.

C Local Regularity of the ODE System

If the initial solution is such that the matrix $C(s_0)$ is regular then in certain neighborhood of s_0 the regularity holds and we can use a Runge-Kutta method.

However, we show that the matrix

$$C(s) = A(s)^\top \nabla^2_{\mathbf{b}} H(\mathbf{b}(s)) A(s) \tag{86}$$

is always regularized for the term H_2 of the objective function if cubics are used for the approximation.

Using (41), we have

$$A(s)^\top \nabla^2_{\mathbf{b}} H(\mathbf{b}(s)) A(s) = A(s)^\top \nabla^2_{\mathbf{b}} H_0(\mathbf{b}(s)) A(s) + \tag{87}$$
$$w_1 A(s)^\top \nabla^2_{\mathbf{b}} H_1(\mathbf{b}(s)) A(s) + \tag{88}$$
$$w_2 \underbrace{A(s)^\top \nabla^2_{\mathbf{b}} H_2(\mathbf{b}(s)) A(s)}_{C_2(s)} \tag{89}$$

Clearly,

$$C_2(s) = \begin{pmatrix} C_{11} & C_{12} & \mathbf{0} \\ C_{21} & C_{22} & C_{23} \\ \mathbf{0} & C_{32} & C_{33} \end{pmatrix} \tag{90}$$

where the blocks are given by

$$C_{11} = \begin{pmatrix} I & -I \\ -I & 2I \end{pmatrix}, \quad C_{33} = \begin{pmatrix} 2I & -I \\ -I & I \end{pmatrix}, \quad C_{22} = \begin{pmatrix} 2I & -I & \mathbf{0} & \cdots & \mathbf{0} \\ -I & 2I & -I & \cdots & \mathbf{0} \\ \mathbf{0} & -I & 2I & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & 2I \end{pmatrix}, \tag{91}$$

$$C_{21} = C_{12}^\top = C_{32} = C_{23}^\top = \begin{pmatrix} \mathbf{0} & \cdots & \mathbf{0} & -I \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \end{pmatrix}. \tag{92}$$

and they are independent of s.

Hence,

$$A(s)^\top C_2(s) A(s) = \begin{pmatrix} R_{11} & R_{12} & \mathbf{0} \\ R_{21} & R_{22} & R_{23} \\ \mathbf{0} & R_{32} & R_{33} \end{pmatrix}, \tag{93}$$

where

$$R_{11} = 2 \begin{pmatrix} \tilde{\mathbf{u}}(s)\tilde{\mathbf{u}}(s) & \tilde{\mathbf{u}}(s)\tilde{\mathbf{p}}_0(s) \\ \tilde{\mathbf{u}}(s)\tilde{\mathbf{p}}_0(s) & \tilde{\mathbf{p}}_0(s)\tilde{\mathbf{p}}_0(s) \end{pmatrix}, \tag{94}$$

$$R_{33} = 2 \begin{pmatrix} \tilde{\mathbf{p}}_n(s)\tilde{\mathbf{p}}_n(s) & -\tilde{\mathbf{v}}(s)\tilde{\mathbf{p}}_n(s) \\ -\tilde{\mathbf{v}}(s)\tilde{\mathbf{p}}_n(s) & \tilde{\mathbf{v}}(s)\tilde{\mathbf{v}}(s) \end{pmatrix}, \tag{95}$$

$$R_{22} = C_{22}, \tag{96}$$

$$R_{21} = R_{12}^\top = \begin{pmatrix} -\tilde{\mathbf{u}}(s) & -\tilde{\mathbf{p}}_0(s) \\ 0 & 0 \\ \vdots & \vdots \end{pmatrix}, \quad R_{23} = R_{32}^\top = \begin{pmatrix} -\tilde{\mathbf{p}}_n(s) & \tilde{\mathbf{v}}(s) \\ 0 & 0 \\ \vdots & \vdots \end{pmatrix}. \tag{97}$$

Now, let $n = 3$. Then,

$$R(s) = A(s)^\top C_2(s) A(s) = \begin{pmatrix} 2\Gamma(\mathbf{a}, \mathbf{a}; \mathbf{b}, \mathbf{b}) & -\Gamma(\mathbf{a}, \mathbf{b}; \mathbf{c}, \mathbf{d}) \\ -\Gamma(\mathbf{a}, \mathbf{b}; \mathbf{c}, \mathbf{d})^\top & 2\Gamma(\mathbf{c}, \mathbf{c}; \mathbf{d}, \mathbf{d}) \end{pmatrix}, \tag{98}$$

with

$$\Gamma(\mathbf{x}_1, \mathbf{x}_2; \mathbf{y}_1, \mathbf{y}_2) = \begin{pmatrix} \mathbf{x}_1 \cdot \mathbf{y}_1 & \mathbf{x}_1 \cdot \mathbf{y}_2 \\ \mathbf{x}_2 \cdot \mathbf{y}_1 & \mathbf{x}_2 \cdot \mathbf{y}_2 \end{pmatrix}. \tag{99}$$

Clearly, the diagonal blocks are Gram matrices of the corresponding vectors up to a multiple.

Using a suitable choice of orthonormal coordinates, we have $\mathbf{p}_0(s) = (0, 0, 0)^\top$, $\mathbf{u}(s) = (1, 0, 0)^\top$, $\mathbf{p}_3(s) = (x(s), y(s), z(s))^\top$ and $\mathbf{v}(s) = (v_x(s), v_y(s), v_z(s))^\top$ with $v_x(s)^2 + v_y(s)^2 + v_z(s)^2 = 1$. Hence (omitting the parameter s),

$$R = \begin{pmatrix} 2 & 0 & -x & -v_x \\ 0 & 2 & -1 & -1 \\ -x & -1 & 2\,x^2 + 2\,y^2 + 2\,z^2 + 2 & 0 \\ -v_y & -1 & 0 & 2 \end{pmatrix} \tag{100}$$

and using the inequalities

$$1 - v_x^2 \geq 0 \quad \text{and} \quad (x - v_x)^2 \geq 0, \tag{101}$$

the determinant

$$\begin{aligned} \det R &= 8 + 12z^2 + 9x^2 + 12y^2 + (-3 - 4x^2 - 4y^2 - 4z^2)v_x^2 - 2v_x x & (102) \\ &\geq 8 - 4v_x^2 + 4x^2 + 9y^2 + 9z^2 + (4x^2 + 4y^2 + 4z^2)(1 - v_x^2) \geq 4 & (103) \end{aligned}$$

is a positive polynomial. Hence, the original system of ODE is regularized by the term (89). Summing up, we get Lemma 4.

Geometrically, this means the uniqueness of the control polygon of a rational cubic in the standard form satisfying given G^1 boundary conditions.

However, the determinant of the matrix $R(s)$ can have an arbitrary sign already for the case $n = 4$. Hence, the regularity is guaranteed only for a certain neighborhood starting with boundary conditions which induce a regular matrix $R(0)$.

Acknowledgments

This research was supported by the Austrian Science Fund (FWF) through the SFB F013 “Numerical and Symbolic Scientific Computing” at Linz, project 15. We would like to thank to Dalibor Lukáš for valuable discussions.

References

1. Gatellier, G., Labrouzy, A., Mourrain, B., Técourt, J.: Computing the topology of three-dimensional algebraic curves. In Dokken, T., Jüttler, B., eds.: Computational Methods for Algebraic Spline Surfaces, Springer (2004) 27–43
2. Hartshorne, R.: Algebraic geometry. Graduate Texts in Mathematics, 52. New York-Heidelberg-Berlin: Springer- Verlag. (1983)
3. Shafarevich, I.: Basic algebraic geometry. Translated from the Russian by K. A. Hirsch. 2nd ed. Springer Study Edition. Berlin-Heidelberg-New York: Springer-Verlag. (1977)
4. Abhyankar, S.S., Bajaj, C.L.: Automatic parameterization of rational curves and surfaces. IV: Algebraic space curves. ACM Trans. Graph. **8** (1989) 325–334
5. Wang, W., Joe, B., Goldman, R.: Computing quadric surface intersections based on an analysis of plane cubic curves. Graph. Models **64** (2002) 335–367
6. Manocha, D., Canny, J.: Rational curves with polynomial parameterization. (1991)
7. Owen, J., Rockwood, A.: Intersection of general implicit surfaces. In Farin, G.E., ed.: Geometric Modelling: Algorithms and new Trends, Philadelphia, Society for Industrial and Applied Mathematics (1987) 335–345
8. Hartmann, E.: Numerical parameterization of curves and surfaces. Comput. Aided Geom. Des. **17** (2000) 251–266
9. Bajaj, C., Royappa, A.: Parameterization in finite precision. Algorithmica **27** (2000) 100–114
10. Sederberg, T., Nishita, T.: Curve intersection using Bézier clipping. (1990)
11. Sampson, P.: Fitting conic sections to very scattered data: an iterative refinement of the Bookstein algorithm. In: Computer Graphics and Image Processing. Volume 18. (1982) 97–108
12. Stoer, J., Bulirsch, R.: Numerical mathematics 2. An introduction – under consideration of lectures by F. L. Bauer. 5th ed. Springer-Lehrbuch. Berlin: Springer. (2005)
13. Hoschek, J., Lasser, D.: Fundamentals of computer aided geometric design. Wellesley, MA: A. K. Peters. (1993)