Shape Preserving Least–Squares Approximation by Polynomial Parametric Spline Curves

Bert Jüttler

University of Technology
Department of Mathematics
Schloßgartenstraße 7
64289 Darmstadt, Germany
juettler@mathematik.th-darmstadt.de

Abstract. This article presents a method for shape preserving least–squares approximation. This method generalizes an algorithm of Dierckx (1980, 1993) to the case of planar parametric curves. Using a reference curve we generate linear sufficient conditions for the convexity of the approximant. This leads us to a quadratic programming problem which can be solved exactly, e.g., with an active set strategy.

Keywords. Least–squares approximation, parametric curve, B–spline curve, shape–preserving approximation.

Introduction

Whereas algorithms for shape preserving interpolation by planar parametric spline curves have been discussed by several authors (see the references in Hoschek & Lasser, 1993, p. 103, 105, 114), the analogous approximation problem has attracted much less attention so far. But for the construction of parametric curves from input data consisting of a relatively large number of points, an interpolation scheme leads to spline curves which consist of a huge number of segments. Moreover, if the input data is only approximately convex, then an interpolation scheme fails to produce the desired result.

In the case of shape preserving approximation by spline *functions*, the first algorithm for solving the problem was developed by Dierckx (1980, 1993). Other contributions to this topic include the articles by Elfving & Anderson (1988), Micchelli &

Utreras (1988), Schmidt & Scholz (1990), Elliott (1993), and Schwetlick & Kunert (1993). A method for shape preserving approximation by circular splines based on linear programming is outlined in a paper by Burchard, Ayers, Frey & Sapidis (1994).

In order to overcome the difficulties of shape preserving interpolation, the present article proposes a method for shape preserving least–squares approximation by planar polynomial parametric spline curves. The method generalizes Dierckx' algorithm to the case of parametric curves. Based on a reference curve (which is used in order to specify the desired shape of the approximant), we are able to generate linear sufficient constraints ensuring the convexity of the approximating spline curve. The control points of the approximant are found as the solution of a quadratic programming problem. According to theoretical results, this problem can be solved exactly in finite time (Fletcher, 1991). For the implementation of the method, however, it is of crucial importance to avoid degeneracies of the constraints as far as possible.

After some preliminary considerations in the first section, the details of the approximation problem are formulated in Section 2. Then we present a construction for the required reference curve. Section 5 describes the generation of the linear sufficient constraints ensuring the convexity of the approximant. Finally, the computation of the control points of the approximating spline curve is outlined in Section 6. The method is illustrated by an example.

1 Preliminaries

Consider a segment of a B–spline curve of degree d, see (Hoschek & Lasser, 1993, Chapter 4). This segment can be represented as a Bézier curve

$$\mathbf{y}(\rho) = \sum_{i=0}^{d} B_i^d(\rho)\,\mathbf{b}_i, \tag{1}$$

of degree d, where $\rho \in [0,1]$ is a local parameter. The blending functions $B_i^d(\rho) = \binom{d}{i}\rho^i(1-\rho)^{d-i}$ are the Bernstein polynomials, whereas the vector–valued coefficients $\mathbf{b}_0, \ldots, \mathbf{b}_d \in \mathbb{R}^2$ are the Bézier control points of the spline segment. The first and second differences of the control points are denoted by

$$\begin{aligned} \Delta^1\mathbf{b}_i &= \mathbf{b}_{i+1} - \mathbf{b}_i \quad (i = 0, \ldots, d-1) \quad \text{and} \\ \Delta^2\mathbf{b}_i &= \mathbf{b}_{i+2} - 2\,\mathbf{b}_{i+1} + \mathbf{b}_i \quad (i = 0, \ldots, d-2). \end{aligned} \tag{2}$$

They are related to the first and second derivative vectors

$$\begin{aligned} \dot{\mathbf{y}}(\rho) &= \frac{\mathrm{d}}{\mathrm{d}\rho}\,\mathbf{y}(\rho) = d \sum_{i=0}^{d-1} B_i^{d-1}(\rho)\,\Delta^1\mathbf{b}_i \quad \text{and} \\ \ddot{\mathbf{y}}(\rho) &= \frac{\mathrm{d}^2}{\mathrm{d}\rho^2}\,\mathbf{y}(\rho) = d\,(d-1) \sum_{i=0}^{d-2} B_i^{d-2}(\rho)\Delta^2\mathbf{b}_i \end{aligned} \tag{3}$$

of the spline segment. Let $[\vec{\mathbf{a}}, \vec{\mathbf{b}}] = a_1\, b_2 - a_2\, b_1$ be the exterior product of the vectors $\vec{\mathbf{a}} = (a_1\ a_2)^\top, \vec{\mathbf{b}} = (b_1\ b_2)^\top \in \mathbb{R}^2$. Recall that the curve segment (1) is said to be convex if the region bounded by it and by the line segment connecting the points $\mathbf{y}(0)$ and $\mathbf{y}(1)$ is convex. For instance, is it sufficient for convexity if the curvature is either non–positive or non–negative and the tangent vector is contained in a wedge of maximal angle π.

Lemma 1. *If two vectors $\vec{\mathbf{u}}, \vec{\mathbf{v}} \in \mathbb{R}^2 \setminus \{\vec{\mathbf{0}}\}$ with $[\vec{\mathbf{u}}, \vec{\mathbf{v}}] > 0$ exist such that the $4d-2$ linear inequalities*

$$[\vec{\mathbf{u}}, \Delta^1 \mathbf{b}_i] \geq 0,\ [\Delta^1 \mathbf{b}_i, \vec{\mathbf{v}}] \geq 0 \qquad (i{=}0,..,d{-}1) \quad (4)$$

$$\textit{and}\quad [\vec{\mathbf{v}}, \Delta^2 \mathbf{b}_j] \geq 0\ (\textit{resp.}\ \leq 0),\ [\Delta^2 \mathbf{b}_j, -\vec{\mathbf{u}}] \geq 0\ (\textit{resp.}\ \leq 0) \quad (j{=}0,..,d{-}2) \quad (5)$$

are satisfied, then the Bézier curve (1) is convex and it has nonnegative (resp. non-positive) curvature for $\rho \in [0,1]$.

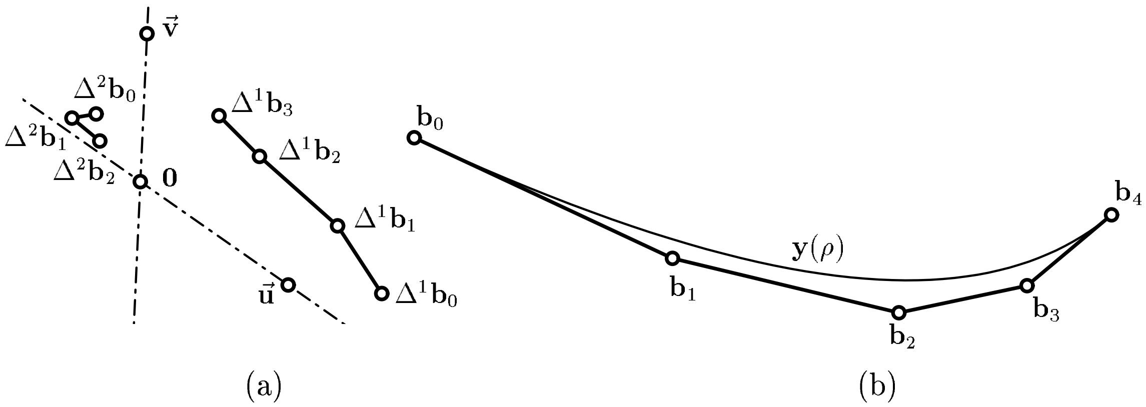

Figure 1: Linear sufficient convexity conditions (a) for a quartic Bézier curve (b).

Proof. Resulting from (4) and from the convex hull property of Bézier curves, the first derivative vector $\dot{\mathbf{y}}(\rho)$ (see (3)) is always contained in the wedge bounded by the vectors $\vec{\mathbf{u}}$ and $\vec{\mathbf{v}}$ as its control polygon is contained in this wedge, see Figure 1. Thus, we can represent it as a nonnegative linear combination of the vectors $\vec{\mathbf{u}}, \vec{\mathbf{v}}$:

$$\dot{\mathbf{y}}(\rho) = \alpha_1(\rho)\,\vec{\mathbf{u}} + \alpha_2(\rho)\,\vec{\mathbf{v}}. \qquad (6)$$

Similarly the second derivative vector $\ddot{\mathbf{y}}(\rho)$ is always contained in the wedge bounded by the vectors $\vec{\mathbf{v}}$ and $-\vec{\mathbf{u}}$ (resp. by the vectors $\vec{\mathbf{u}}$ and $-\vec{\mathbf{v}}$), see (5). Hence,

$$\ddot{\mathbf{y}}(\rho) = \beta_1(\rho)\,\vec{\mathbf{v}} - \beta_2(\rho)\,\vec{\mathbf{u}} \qquad (7)$$

where $\beta_1(\rho), \beta_2(\rho) \geq 0$ (resp. $\beta_1(\rho), \beta_2(\rho) \leq 0$) for $\rho \in [0,1]$. The sign of the curvature of the curve is equal to the sign of the exterior product $[\dot{\mathbf{y}}(\rho), \ddot{\mathbf{y}}(\rho)]$. From (6) and (7) we find that

$$[\,\dot{\mathbf{y}}(\rho), \ddot{\mathbf{y}}(\rho)\,] = \Big(\underbrace{\alpha_1(\rho)}_{\geq 0}\quad\underbrace{\beta_1(\rho)}_{\geq 0 \text{ (resp. } \leq 0)} \quad + \quad \underbrace{\alpha_2(\rho)}_{\geq 0}\quad\underbrace{\beta_2(\rho)}_{\geq 0 \text{ (resp. } \leq 0)}\Big)\,\underbrace{[\vec{\mathbf{u}}, \vec{\mathbf{v}}]}_{>0} \tag{8}$$

and so $\mathbf{y}(\rho)$ has nonnegative (resp. nonpositive) curvature for $\rho \in [0,1]$. Moreover, the first derivative vector of the curve is always contained in the wedge bounded by the vectors $\vec{\mathbf{u}}$ and $\vec{\mathbf{v}}$. Thus, the curve (1) is convex. □

As a very popular criterion, convexity of the control polygon implies convexity of a planar Bézier curve (see e.g. Hoschek and Lasser, 1993). The conditions of the Lemma are stronger than this criterion. However, it is possible to derive tighter conditions by applying Lemma (1) to subsegments of the curve, see below. For more information on convexity conditions see also (Goodman, 1991).

If the Bézier curve is restricted to the parameter interval $\rho \in [\rho_a, \rho_b] \subseteq [0,1]$, then we can represent this segment again as a Bézier curve

$$\mathbf{y}(\rho) = \sum_{i=0}^{d} B_i^d\Big(\frac{\rho - \rho_a}{\rho_b - \rho_a}\Big)\, \mathbf{b}_i(\rho_a, \rho_b), \tag{9}$$

where the new control points $\mathbf{b}_i(\rho_a, \rho_b)$ result from the de Casteljau scheme. The obtained curve segment is denoted by $\mathbf{y} \mid [\rho_a, \rho_b]$. Exploiting the blossoming principle one can derive a formula for the new control points, see (Hoschek & Lasser, 1993). For instance, in the cubic case ($d = 3$) one gets

$$\mathbf{b}_i(\rho_a, \rho_b) = \mathbf{y}_{\text{blossom}}(\underbrace{\rho_a, \ldots, \rho_a}_{3-i \text{ times}}, \underbrace{\rho_b, \ldots, \rho_b}_{i \text{ times}}) \qquad (i = 0,1,2,3) \tag{10}$$

with $\mathbf{y}_{\text{blossom}}(\rho_0, \rho_1, \rho_2) =$

$$\begin{aligned}((1-\rho_0)(1-\rho_1)(1-\rho_2))\, \mathbf{b}_0 + (\rho_0+\rho_1+\rho_2-2\rho_0\rho_1-2\rho_1\rho_2-2\rho_0\rho_2+3\rho_0\rho_1\rho_2)\, \mathbf{b}_1 \\ + (\rho_0\rho_1+\rho_1\rho_2+\rho_0\rho_2-3\rho_0\rho_1\rho_2)\, \mathbf{b}_2 + \rho_0\rho_1\rho_2\, \mathbf{b}_3.\end{aligned} \tag{11}$$

If the length of the parameter interval tends to zero, then the control points converge to the curve points,

$$\lim_{\rho_a \to \rho_0,\, \rho_b \to \rho_0} \mathbf{b}_i(\rho_a, \rho_b) = \mathbf{y}(\rho_0) \qquad (i = 0, \ldots, d). \tag{12}$$

Analogously one gets

$$\begin{aligned}\lim_{\substack{\rho_a \to \rho_0,\, \rho_b \to \rho_0 \\ \rho_a \neq \rho_b}} \frac{d}{\rho_b - \rho_a}\, \Delta^1 \mathbf{b}_i(\rho_a, \rho_b) = \dot{\mathbf{y}}(\rho_0) \quad (i = 0, \ldots, d-1) \quad \text{and} \\ \lim_{\substack{\rho_a \to \rho_0,\, \rho_b \to \rho_0 \\ \rho_a \neq \rho_b}} \frac{d\,(d-1)}{(\rho_b - \rho_a)^2}\, \Delta^2 \mathbf{b}_i(\rho_a, \rho_b) = \ddot{\mathbf{y}}(\rho_0) \quad (i = 0, \ldots, d-2)\end{aligned} \tag{13}$$

because subdivision applies to the control vectors of the derivatives too.

Lemma 2. *Any Bézier curve (1) with non-vanishing curvature for* $\rho \in [0,1]$ *can be subdivided into a finite number* N *of segments* $\mathbf{y}\,|\,[\rho_j, \rho_{j+1}]$ $(j = 0,..,N-1)$ *with* $0 = \rho_0 < \rho_1 \ldots < \rho_N = 1$*, such that all segments satisfy the convexity conditions of Lemma 1 for certain vectors* $\vec{\mathbf{u}}_j, \vec{\mathbf{v}}_j$*.*

Proof. We prove the Lemma in the case of positive curvature, i.e.

$$[\,\dot{\mathbf{y}}(\rho), \ddot{\mathbf{y}}(\rho)\,] > 0 \tag{14}$$

holds for $\rho \in [0,1]$. Consider the sequence of intervals $\rho \in I_{i,n}$ where $I_{i,n} = [\frac{i}{2^n}, \frac{i+1}{2^n}]$, $i = 0, \ldots, 2^n - 1$, for $n = 1, 2, \ldots$. We suppose the assertion is wrong for all $n \in \mathbb{N}$. Thus, for each n a number i_n exists $(0 \leq i_n < 2^n)$, such that the convexity condition of Lemma 1 is not satisfied for $\mathbf{y}|I_{i_n,n}$, whereas it is fulfilled for all segments $\mathbf{y}|I_{i,n}$ with $i < i_n$.

The control points of any subsegment of a Bézier curve are contained in the convex hull of the control points of the whole curve. This also applies to the control polygons of the derivatives. Thus, if the condition of Lemma 1 is satisfied for the segment $\mathbf{y}|I_{i,n}$, then it is also true for all segments contained in it. Therefore the sequence $(\frac{i_n}{2^n})_{n\in\mathbb{N}}$ is monotonically increasing. Moreover it is bounded by 1, hence the limit

$$\rho_0 = \lim_{n\to\infty} \frac{i_n}{2^n} \tag{15}$$

exists. Let $\dot{\mathbf{y}}_0 = \dot{\mathbf{y}}(\rho_0)$ and $\ddot{\mathbf{y}}_0 = \ddot{\mathbf{y}}(\rho_0)$. From Equations (13) and (14) we obtain

$$\begin{aligned}
0 &< [\,\dot{\mathbf{y}}_0\,,\ \dot{\mathbf{y}}_0 + \ddot{\mathbf{y}}_0\,] = \lim_{n\to\infty}\ 2^n\, d\,[\,\Delta^1\mathbf{b}_i(\tfrac{i_n}{2^n}, \tfrac{i_n+1}{2^n})\ ,\ \dot{\mathbf{y}}_0 + \ddot{\mathbf{y}}_0\,]\,, \\
0 &< [\,\dot{\mathbf{y}}_0 - \ddot{\mathbf{y}}_0\,,\ \dot{\mathbf{y}}_0\,] = \lim_{n\to\infty}\ 2^n\, d\,[\,\dot{\mathbf{y}}_0 - \ddot{\mathbf{y}}_0\ ,\ \Delta^1\mathbf{b}_i(\tfrac{i_n}{2^n}, \tfrac{i_n+1}{2^n})\,]\,, \\
0 &< [\,\dot{\mathbf{y}}_0 + \ddot{\mathbf{y}}_0\,,\ \ddot{\mathbf{y}}_0\,] = \lim_{n\to\infty}\ (2^n)^2\, d\,(d-1)\,[\,\dot{\mathbf{y}}_0 + \ddot{\mathbf{y}}_0\ ,\ \Delta^2\mathbf{b}_j(\tfrac{i_n}{2^n}, \tfrac{i_n+1}{2^n})\,]\ \text{ and} \\
0 &< [\,\ddot{\mathbf{y}}_0\,,\ \ddot{\mathbf{y}}_0 - \dot{\mathbf{y}}_0\,] = \lim_{n\to\infty}\ (2^n)^2\, d\,(d-1)\,[\,\Delta^2\mathbf{b}_j(\tfrac{i_n}{2^n}, \tfrac{i_n+1}{2^n})\ ,\ \ddot{\mathbf{y}}_0 - \dot{\mathbf{y}}_0\,]
\end{aligned} \tag{16}$$

for all $i = 0, \ldots, d-1$ and $j = 0, \ldots, d-2$. As the right–hand sides of all inequalities are strictly positive, a number n_0 exists such that all exterior products $[.,.]$ in the limits on the right–hand sides are nonnegative for $n > n_0$. Thus, the convexity criterion of Lemma 1 is satisfied for the segment $\mathbf{y}|I_{i_n,n}$ $(n > n_0)$ with $\vec{\mathbf{u}} = \dot{\mathbf{y}}_0 - \ddot{\mathbf{y}}_0$ and $\vec{\mathbf{v}} = \dot{\mathbf{y}}_0 + \ddot{\mathbf{y}}_0$. This is a contradiction. □

It is possible to generalize Lemma 1 to curves with vanishing second derivative vectors at the segment end points. This will be useful for the construction of linear sufficient conditions for convexity from a reference curve in the presence of inflection points at the segment boundaries.

Proposition 3. *Consider a Bézier curve (1) with nonzero curvature for all interior points, i.e., for* $\rho \in (0,1)$*. If for both segment end points* $a = 0$*,* $a = 1$ *either the* *conditions* $\ddot{\mathbf{y}}(a) = \vec{\mathbf{0}}$ *and* $[\dot{\mathbf{y}}(a), \dddot{\mathbf{y}}(a)] \neq 0$ *hold or the curvature at* $\rho = a$ *is nonzero* $([\dot{\mathbf{y}}(a), \ddot{\mathbf{y}}(a)] \neq 0)$, *then the curve can be subdivided into a finite number* N *of segments* $\mathbf{y} \,|\, [\rho_j, \rho_{j+1}]$ $(j = 0, .., N-1)$ *with* $0 = \rho_0 < \rho_1 \ldots < \rho_N = 1$, *such that all segments satisfy the convexity conditions of Lemma 1 for certain vectors* $\vec{\mathbf{u}}_j, \vec{\mathbf{v}}_j$.

Proof. We consider a Bézier curve with nonnegative curvature, thus (14) holds for $\rho \in (0, 1)$. Consider the case $\ddot{\mathbf{y}}(0) = \vec{\mathbf{0}}$. Let $\dot{\mathbf{y}}_0 = \dot{\mathbf{y}}(0)$ and $\dddot{\mathbf{y}}_0 = \dddot{\mathbf{y}}(0)$. From the Taylor expansion of the curve in a right neighbourhood of $\rho_0 = 0$ we obtain

$$0 < [\,\dot{\mathbf{y}}_0 + \frac{1}{2}\rho^2\,\dddot{\mathbf{y}}_0 + \ldots\,,\; \rho\,\dddot{\mathbf{y}}_0 + \ldots\,] = \rho\,[\,\dot{\mathbf{y}}_0, \dddot{\mathbf{y}}_0\,] + \ldots \quad (0 < \rho < \epsilon), \tag{17}$$

hence $[\,\dot{\mathbf{y}}_0, \dddot{\mathbf{y}}_0\,] > 0$. Similar to Equation (13), one gets for the control polygon of the third derivative of a Bézier curve

$$\lim_{\delta \to 0+0} \frac{d\,(d-1)\,(d-2)}{\delta^3}\,\Delta^3 \mathbf{b}_i(0, \delta) = \dddot{\mathbf{y}}_0 \quad (i = 0, \ldots, d-3). \tag{18}$$

Analogously to the proof of the previous lemma we may conclude from (13), (18) and $[\,\dot{\mathbf{y}}_0, \dddot{\mathbf{y}}_0\,] > 0$, that for a sufficiently small $\delta > 0$ the inequalities

$$0 \leq [\,\Delta^1 \mathbf{b}_i(0, \delta)\;,\; \dot{\mathbf{y}}_0 + \dddot{\mathbf{y}}_0\,]\,, \qquad 0 \leq [\,\dot{\mathbf{y}}_0 - \dddot{\mathbf{y}}_0\;,\; \Delta^1 \mathbf{b}_i(0, \delta)\,]\,, \tag{19}$$

$$0 \leq [\,\dot{\mathbf{y}}_0 + \dddot{\mathbf{y}}_0\;,\; \Delta^3 \mathbf{b}_j(0, \delta)\,] \quad \text{and} \quad 0 \leq [\,\Delta^3 \mathbf{b}_j(0, \delta)\;,\; \dddot{\mathbf{y}}_0 - \dot{\mathbf{y}}_0\,] \tag{20}$$

are satisfied $(i = 0, \ldots, d-1;\; j = 0, \ldots, d-3)$. On the other hand, $\ddot{\mathbf{y}}(0) = \vec{\mathbf{0}}$ implies $\Delta^2 \mathbf{b}_0(0, \delta) = \vec{\mathbf{0}}$, hence

$$\Delta^2 \mathbf{b}_k(0, \delta) = \sum_{j=0}^{k-1} \Delta^3 \mathbf{b}_j(0, \delta) \quad (k = 0, \ldots, d-2). \tag{21}$$

Thus, the inequalities (20) are even true for the second differences of the curve control points. Therefore the curve segment $\mathbf{y} \,|\, [0, \delta]$ satisfies the convexity criterion of Lemma (1) with $\vec{\mathbf{u}} = \dot{\mathbf{y}}_0 - \dddot{\mathbf{y}}_0$ and $\vec{\mathbf{v}} = \dot{\mathbf{y}}_0 + \dddot{\mathbf{y}}_0$. An analogous construction yields convexity constraints for the curve segment $\mathbf{y} \,|\, [1 - \delta, 1]$ in the case $\ddot{\mathbf{y}}(1) = \vec{\mathbf{0}}$. The existence of convexity conditions for the remaining segment of the curve is guaranteed by the previous lemma. □

2 The approximation problem

We will discuss the following problem. Let a sequence of $N + 1$ points $\mathbf{p}_i \in \mathbb{R}^2$ with an associated strictly increasing sequence of parameter values $t_i \in \mathbb{R}$ be given

($i = 0, \ldots, N$). If the parameters t_i are unknown, then they can be estimated from the given data, see (Hoschek & Lasser, 1993, Chapter 4.4.1). Additionally, a strictly increasing sequence of knots τ_k ($k = 0, \ldots, K$) is assumed to be known. For instance, one can choose an appropriate subsequence of the parameters t_i.

The given data is to be approximated by a polynomial spline curve $\mathbf{x}(t)$ with parameter domain $t \in [\tau_0, \tau_K]$ which is defined over the knots τ_k. Some of the knots are assumed to be marked as the desired inflections of the approximating spline curve. We denote them by $\tau_{w(j)}$, their indices form the sequence $0 < w(1) < w(2) < \ldots < w(I-1) < K$. In addition we set $w(0) = 0$ and $w(I) = K$. For example one may estimate the inflections from the lines of regression of adjacent r-tuples ($r = 3, 4, ..$) of data points. Finally, the signs $\sigma_j \in \{-1, +1\}$ of the curvature for each curve segment $\mathbf{x} \mid [\tau_{w(j-1)}, \tau_{w(j)}]$ connecting two inflections have to be specified ($j = 1, \ldots, I$). Of course, adjacent segments should possess different curvature signs, $\sigma_j\, \sigma_{j+1} = -1$.

In order to be as brief as possible we describe the method only for the case of cubic C^2 spline curves. This will sufficiently illustrate the key ideas of our approach. The reader who is interested in a more general presentation of the scheme (for polynomial C^l spline curves of degree $d > l$ with $l = 1, 2$) should consult the report (Jüttler, 1996).

We use the B–spline representation of the spline curve. The $P + 1$ (where $P = K+2$) cubic B–spline basis functions $N_0^3(t), N_1^3(t), \ldots, N_P^3(t) \in C^2[\tau_0, \tau_K]$ are defined over the knot vector

$$\mathcal{T} = (\tau_0, \tau_0, \tau_0, \tau_1, \tau_2, \ldots \tau_{K-1}, \tau_K, \tau_K, \tau_K). \tag{22}$$

For the definition of the B–spline functions the reader is referred to (Hoschek & Lasser, 1993; Farin, 1993) or to similar textbooks on spline functions. The approximating spline curve possesses the parametric representation

$$\mathbf{x}(t) = \sum_{i=0}^{P} N_i^3(t)\, \mathbf{d}_i\,, \qquad t \in [\tau_0, \tau_K], \tag{23}$$

where the de Boor points $\mathbf{d}_i \in \mathbb{R}^2$ are unknown yet ($i = 0, \ldots, P$). We assume $w(j+1) \geq w(j) + 2$ ($j = 1, \ldots, I-2$). This will ensure that the spline curve between two inflections possesses at least one free control point.

Alternatively, each spline segment $\mathbf{x} \mid [\tau_{k-1}, \tau_k]$ may be described by a Bézier curve,

$$\mathbf{x}(t) = \sum_{i=0}^{3} B_i^3\left(\frac{t - \tau_{k-1}}{\tau_k - \tau_{k-1}}\right) \mathbf{b}_{k,i} \quad \text{for} \quad t \in [\tau_{k-1}, \tau_k] \tag{24}$$

($k = 1, \ldots, K$). The Bézier control points $\mathbf{b}_{k,0}, \ldots, \mathbf{b}_{k,3}$ are certain affine combinations of the de Boor points which result from Böhm's algorithm, see (Farin, 1993).

The conversion formulas can also be gathered from Figure 4.14 of (Hoschek & Lasser, 1993).

Analogous to (9) we may even restrict these Bézier curves to subsegments with the parameter interval

$$t \in [(1-\rho_a)\tau_{k-1} + \rho_a \tau_k,\ (1-\rho_b)\tau_{k-1} + \rho_b \tau_k] \text{ with } 0 \leq \rho_a < \rho_b \leq 1. \tag{25}$$

The resulting control points will be denoted by $\mathbf{b}_{k,i}(\rho_a, \rho_b)$; they are certain affine combinations of the de Boor points as well.

The spline curve will be constructed from the following two requirements:

- If $\sigma_j = +1$ (resp. $\sigma_j = -1$) holds, then the curvature of $\mathbf{x}(t)$ is nonnegative (resp. nonpositive) for $t \in [\tau_{w(j-1)}, \tau_{w(j)}]$, i.e., we have

$$[\dot{\mathbf{x}}(t), \ddot{\mathbf{x}}(t)] \geq 0 \ \ (\text{resp. } \leq 0) \quad (j = 1, \ldots, I). \tag{26}$$

- The curve approximates the given data. This will be achieved by minimizing the least–squares sum

$$\sum_{i=0}^{N} \|\mathbf{x}(t_i) - \mathbf{p}_i\|^2 . \tag{27}$$

Of course, the computation of the *exact* solution to both requirements is a very hard task. In order to find an *approximate* solution of the problem we will replace the non–linear convexity conditions (26) by appropriate linear constraints for the de Boor points. The approximate solution is then found by minimizing the least–squares sum (27) subject to the linearized constraints.

In the remainder of the paper we assume that the quadratic part of the least–squares sum (27) is a positive definite quadratic functional of the components of the unknown de Boor points. This will ensure the existence and uniqueness of the solution of the quadratic programming problem. In most cases of practical interest, the number of data points is much bigger than the number of de Boor points, and also the distribution of the data points over the spline segments should be almost uniform. Thus, the above assumption will be automatically true. If the assumption is violated, then one can modify the least squares sum by adding a weighted "tension term", cf. (Hoschek & Lasser, 1991, Chapter 3.6), in order to achieve regularity.

3 Construction of the reference curve

With the help of a reference curve we will generate linear sufficient conditions for the convexity of the approximating spline curve. This section presents a heuristic construction for this curve which guarantees that no undesired inflections occur. This does not automatically imply that the curve segments possess the desired signs of the curvature. However, in all cases of practical interest it can be expected, that the curve also has the desired curvature signs. Otherwise the specified shape might be unsuitable for approximating the given data and should be checked. The obtained curve can be expected to be a reasonable starting point for the generation of the linearized convexity constraints and for the subsequent optimization.

Based on B–spline curves with prescribed directions of the legs of the control polygon, the report (Jüttler, 1996) derives another construction for the reference curve which guarantees that the reference curve possesses the desired shape. As an important disadvantage of that construction, one generally gets curves which are relatively bad approximations of the data points, and this causes problems for the subsequent optimization. For this reason we omit the details of that construction; it should not be used in practice.

The reference curve is chosen as a cubic B–spline curve

$$\mathbf{y}(t) = \sum_{i=0}^{\hat{P}} \hat{N}_i^3(t)\, \hat{\mathbf{c}}_i \qquad t \in [\tau_0, \tau_K], \tag{28}$$

with control points $\hat{\mathbf{c}}_i \in \mathbb{R}^2$, where the B–spline basis functions $\hat{N}_i^3(t)$ are defined over an appropriate subset $\hat{\mathcal{T}}$ of the original knot vector $\mathcal{T}$,

$$\hat{\mathcal{T}} = (\hat{\tau}_0, \hat{\tau}_0, \hat{\tau}_0, \hat{\tau}_1, \hat{\tau}_2, \ldots \hat{\tau}_{\hat{K}-1}, \hat{\tau}_{\hat{K}}, \hat{\tau}_{\hat{K}}, \hat{\tau}_{\hat{K}}). \tag{29}$$

For the moment we choose $\hat{\mathcal{T}} = \mathcal{T}$, thus $\hat{\tau}_i = \tau_i$ $(i = 0, \ldots, K)$, $\hat{P} = P$ and $\hat{K} = K$. Later we will remove some knots from $\hat{\mathcal{T}}$, but the parameter values $\tau_{w(j)} = \hat{\tau}_{\hat{w}(j)}$ (which are now identified by the sequence $\hat{w}(j)$ of indices) of the desired inflection points must remain knots of the spline curve (28) $(j = 1, \ldots, I-1)$.

In order to benefit from Proposition 3 we construct a reference curve with vanishing second derivative

$$\ddot{\mathbf{y}}(\tau_{w(j)}) = \ddot{\mathbf{y}}(\hat{\tau}_{\hat{w}(j)}) = \vec{\mathbf{0}} \tag{30}$$

at the desired inflections. This is equivalent to the conditions

$$\hat{\mathbf{c}}_{\hat{w}(j)+1} = \frac{(\Delta\hat{\tau}_{\hat{w}(j)-2}+\Delta\hat{\tau}_{\hat{w}(j)-1}+\Delta\hat{\tau}_{\hat{w}(j)})\, \hat{\mathbf{c}}_{\hat{w}(j)+2}+(\Delta\hat{\tau}_{\hat{w}(j)-1}+\Delta\hat{\tau}_{\hat{w}(j)}+\Delta\hat{\tau}_{\hat{w}(j)+1})\, \hat{\mathbf{c}}_{\hat{w}(j)}}{\Delta\hat{\tau}_{\hat{w}(j)-2}+2\Delta\hat{\tau}_{\hat{w}(j)-1}+2\Delta\hat{\tau}_{\hat{w}(j)}+\Delta\hat{\tau}_{\hat{w}(j)+1}} \tag{31}$$

where $\Delta\hat{\tau}_i = \hat{\tau}_{i+1} - \hat{\tau}_i$, but $\Delta\hat{\tau}_{-1} = \Delta\hat{\tau}_{\hat{K}} = 0$, see e.g. (Hoschek & Lasser, 1993, p. 176). The control points $\hat{\mathbf{c}}_i$ are constructed by solving the least–squares problem

$$\sum_{i=0}^{N} \|\mathbf{y}(t_i) - \mathbf{p}_i\|^2 \to \min \tag{32}$$

subject to the constraints (31). Using the constraints we eliminate the control points $\hat{\mathbf{c}}_{\hat{w}(j)+1}$, from the set of unknowns of the problem (32). Alternatively one may use Lagrangian multipliers. The normal equations of the least–squares problem (32) yield a system of linear equations for the unknown control points $\hat{\mathbf{c}}_j$ which is solved using an appropriate method, cf. (Hoschek & Lasser, 1993, Section 4.4.3).

The knot vector $\hat{\mathcal{T}}$ is chosen such that the approximating spline curve obtained from (32) does not possess any undesired inflections. This can be achieved with the help of the following algorithm:

1.) Compute the control points $\hat{\mathbf{c}}_0, \ldots, \hat{\mathbf{c}}_{\hat{P}}$ by solving the least–squares problem (32) subject to the constraints (31).

2.) Find the first undesired inflection or flat point $\mathbf{y}(t_0)$ in all curve segments $\mathbf{y} \,|\, [\hat{\tau}_{\hat{w}(j)}, \hat{\tau}_{\hat{w}(j+1)}]$ with $\hat{w}(j+1) - \hat{w}(j) \geq 2$ (i.e., we consider only those spline segments between two desired inflections which possess at least one inner knot), for details see below ($j = 0, \ldots, I-1$). If no such points exists, then continue with Step 4. Otherwise

3.) Let $t_0 \in [\hat{\tau}_{\hat{w}(j)}, \hat{\tau}_{\hat{w}(j+1)}]$. Delete the knot $\hat{\tau}_i$ with $\hat{w}(j) < i < \hat{w}(j+1)$ from $\hat{\mathcal{T}}$ which is as close as possible to t_0. This implies $\hat{P} = \hat{P} - 1$ and $\hat{K} = \hat{K} - 1$. Continue with Step 1.

4.) Reinsert all knots which have been deleted from $\hat{\mathcal{T}}$ in order to find a representation of the spline curve defined over the original knot vector (22). Its control points are denoted by $\mathbf{c}_0, \ldots, \mathbf{c}_P$; they result from the knot insertion algorithm, see (Hoschek & Lasser, 1993).

The algorithm always produces a reference curve without undesired inflections, provided that at least one desired inflection point has been specified. This is due to the fact, that a segment of the spline curve with only one free inner control point cannot have inflections. In almost all cases of practical interest, the reference curve will then possess the desired curvature signs too.

If no desired inflections have been specified, then the algorithm may lead to a single cubic curve with inflections. In this case one should approximate with a quadratic curve in order to get a suitable reference curve.

A slight modification of the algorithm would allow to delete more than one knot from $\hat{\mathcal{T}}$ in the third step. This is advantageous in the case of relatively large knot vectors.

For detecting undesired inflections or flat points in the second step of the algorithm we firstly split the curve into its polynomial segments. Each segment $\mathbf{y} \,|\, [\hat{\tau}_{k-1}, \hat{\tau}_k]$ can be represented as a cubic Bézier curve

$$\mathbf{y}(t) = B_0^3(\rho)\, \hat{\mathbf{a}}_{k,0} + B_1^3(\rho)\, \hat{\mathbf{a}}_{k,1} + B_2^3(\rho)\, \hat{\mathbf{a}}_{k,2} + B_3^3(\rho)\, \hat{\mathbf{a}}_{k,3} \tag{33}$$

with the local parameter $\rho = (t - \hat{\tau}_{k-1})/(\hat{\tau}_k - \hat{\tau}_{k-1})$, where the control points $\hat{\mathbf{a}}_{k,0}$, $\ldots, \hat{\mathbf{a}}_{k,3}$ can be generated with the help of Böhms algorithm. A short calculation yields

$$[\dot{\mathbf{y}}(t), \ddot{\mathbf{y}}(t)] = \frac{18}{(\hat{\tau}_k - \hat{\tau}_{k-1})^3} \Big(B_0^2(\rho)\,[\Delta^1 \hat{\mathbf{a}}_{k,0}, \Delta^1 \hat{\mathbf{a}}_{k,1}] + \frac{1}{2} B_1^2(\rho)\,[\Delta^1 \hat{\mathbf{a}}_{k,0}, \Delta^1 \hat{\mathbf{a}}_{k,2}] + B_2^2(\rho)\,[\Delta^1 \hat{\mathbf{a}}_{k,1}, \Delta^1 \hat{\mathbf{a}}_{k,2}] \Big). \tag{34}$$

If this quadratic polynomial possesses a root ρ_0 in $[0, 1]$, then the point $\mathbf{y}(t_0)$ with $t_0 = (1 - \rho_0)\hat{\tau}_{k-1} + \rho_0\, \hat{\tau}_k$ is an inflection or a flat point of the spline curve $\mathbf{y}(t)$.

4 An example

As an example we consider the approximation of 63 data points which are depicted by crosses and (partially filled) boxes in Figure 2a. Their parameters have been chosen according to $t_i = \frac{i}{10}$ $(i = 0, \ldots, 62)$. The knot vector $\mathcal{T}$ is chosen as a subsequence of the parameters t_i. The data points which correspond to knots are marked by boxes. Two desired inflections have been specified, they are indicated by black boxes.

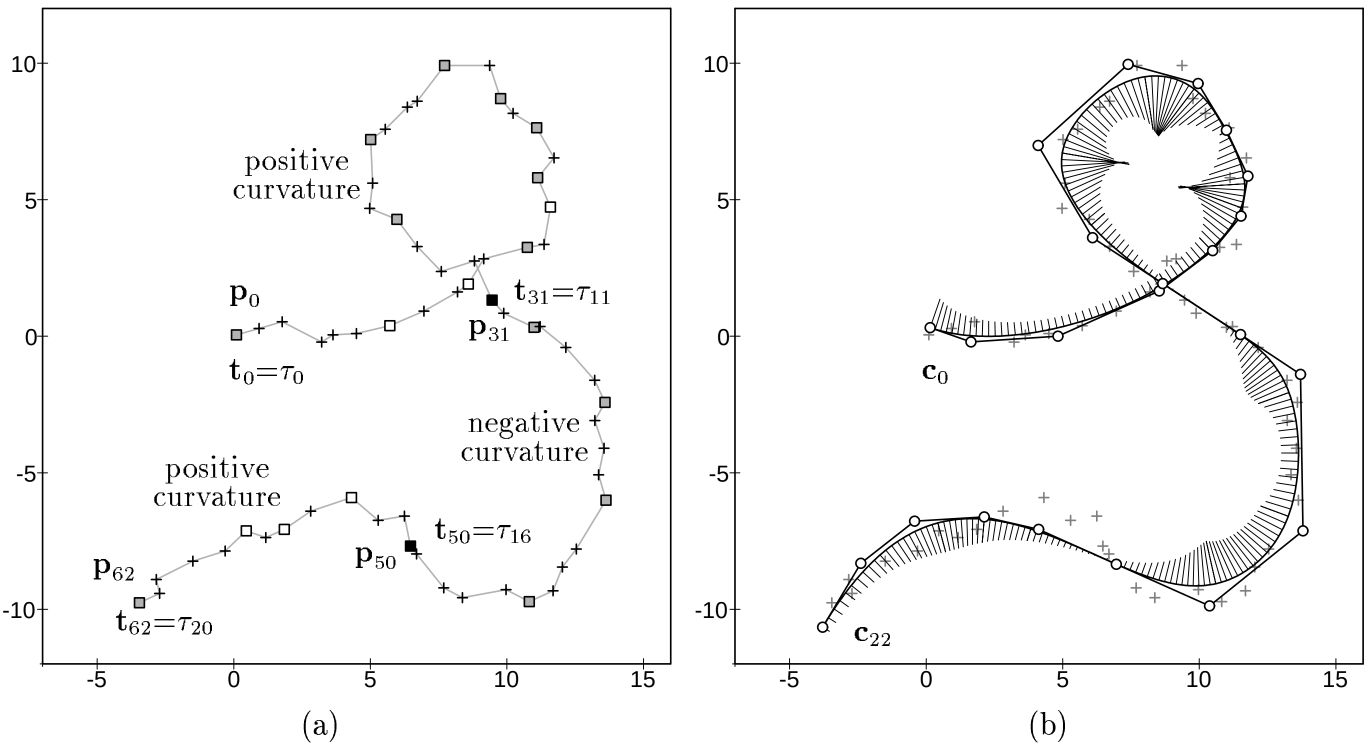

Figure 2: The 63 data points (a) and the reference curve (b).

The above algorithm led to the reference curve in Figure 2b. The de Boor points with respect to the knot vector $\mathcal{T}$ have been drawn. The desired inflections correspond to collinear triples of adjacent control points. The curvature is visualized by scaled curve normals ("porcupines"). Six knots had to be deleted from $\mathcal{T}$ in order to avoid undesired inflections, the remaining knots (forming the knot vector $\hat{\mathcal{T}}$) correspond to those data points in Figure 2a which are marked by grey or black squares.

5 Generating the convexity conditions

In order to simplify notations, the Bézier spline representation of the spline curves is used throughout this section. Similar to (33) the polynomial segments of the reference curve are described by Bézier curves with control points $\mathbf{a}_{k,0}, \ldots, \mathbf{a}_{k,3}$ (now without "ˆ"). Again we consider subsegments (25); the resulting control points are denoted by $\mathbf{a}_{k,i}(\rho_a, \rho_b)$. The Bézier spline representation of the approximating curve is given by (24).

We denote by lwb $(\vec{\mathbf{v}}_1, \ldots, \vec{\mathbf{v}}_s)$ and rwb $(\vec{\mathbf{v}}_1, \ldots, \vec{\mathbf{v}}_s)$ the left and the right bound of the wedge spanned by the vectors $\vec{\mathbf{v}}_1, \ldots, \vec{\mathbf{v}}_s$, respectively, provided that these vectors are contained in a wedge of maximal angle π, and at least one vector is not the null vector. More precisely, we define

$$\begin{gathered}\mathrm{lwb}\,(\vec{\mathbf{v}}_1, \ldots, \vec{\mathbf{v}}_s)\ (\text{resp. } \mathrm{rwb}\,(\vec{\mathbf{v}}_1, \ldots, \vec{\mathbf{v}}_s)\ ) = \vec{\mathbf{u}}, \quad \text{where } \vec{\mathbf{u}} \text{ satisfies} \\ \exists i_0 \in \{1, \ldots, s\} : \ \vec{\mathbf{u}} = \frac{1}{\|\vec{\mathbf{v}}_{i_0}\|}\, \vec{\mathbf{v}}_{i_0} \ \text{and} \ \forall i \in \{1, \ldots, s\} : \ [\vec{\mathbf{v}}_i, \vec{\mathbf{u}}] \geq 0 \ (\text{resp. } \leq 0).\end{gathered} \tag{35}$$

The algorithm described below yields a set of linear sufficient conditions for the convexity of the spline curve. It my be generalized to curves of higher degree. The algorithm generates appropriate bounds for the wedges spanned by the first and second difference vectors of the Bézier control points $\mathbf{a}_{k,i}$ of the reference curve, according to Lemma 1. Based on some heuristics it tries to ensure, that

(*i*) most of the constraints are non–active for the reference curve. Therefore the subsequent optimization finds enough degrees of freedom at the initial point.

(*ii*) a certain maximal level of refinement $\Lambda_{\max}$ is not exceeded except if otherwise no linear sufficient conditions can be found. The number $\Lambda_{\max}$ has to be specified by the user. The example in this paper have been computed with $\Lambda_{\max} = 3$.

Moreover, the algorithm ensures that

(*iii*) if the reference curve is a segment of linear curve (i.e., of a straight line) for $t \in [\tau_{k-1}, \tau_k]$ ($\ddot{\mathbf{y}}(t) = \vec{\mathbf{0}}$ holds for this segment), then also the approximating curve possesses a linear segment there. (The first reference curve should not possess such segments. But linear segments may occur if the result of the optimization described in the next section is used as a new reference curve in order to obtain better results.)

The symbols $\mathcal{I}$ and $\mathcal{E}$ denote the sets of the inequality and equality constraints for the unknown control points of the approximating spline curve. The algorithm is illustrated by Figure 3, which shows the construction of the constraints in the case of a curve segment possessing non–negative curvature. We present a recursive description

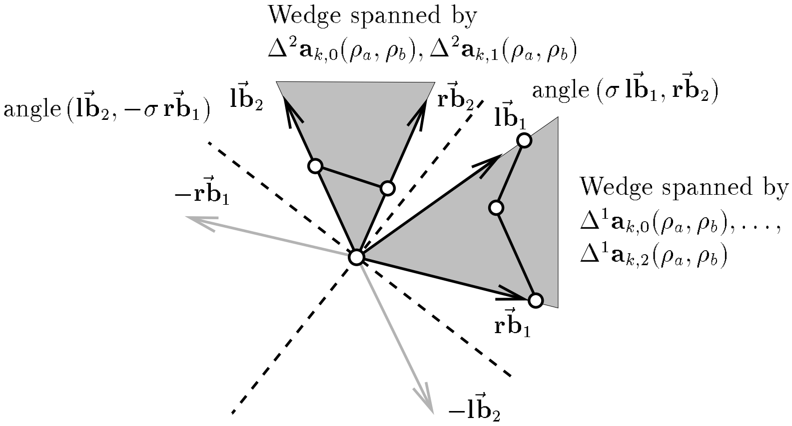

Figure 3: Construction of linear sufficient convexity conditions ($\sigma = 1$).

which is to be applied to the k–th spline segment with the initial parameters $(\rho_a, \rho_b) = (0, 1), \Lambda = 0$ and $\sigma \in \{+1, -1\}$ (depending on the specified curvature sign):

`generate_lscc` $(\rho_a, \rho_b, \Lambda,\ \sigma,\ k)$

The knots ρ_a, ρ_b specify the current subsegment of the k–th spline segment, see (25), whereas Λ and σ denote the refinement level and the desired sign of the curvature, respectively.

1.) If $\Delta^2\mathbf{a}_{k,i}(\rho_a, \rho_b) = \vec{\mathbf{0}}$ holds for $i = 0, 1$, then add the equations $\Delta^2\mathbf{b}_{k,i}(\rho_a, \rho_b) = \vec{\mathbf{0}}$ to the set $\mathcal{E}$ of equality constraints and return. This guarantees the property (*iii*) of the algorithm. (In practice one should check whether the magnitudes of these vectors are smaller than an appropriate constant ϵ.)

2.) Compute the left and the right bounds of the wedges spanned by the first and second derivatives,

$$\begin{aligned}
\vec{\mathbf{lb}}_1 &= \mathrm{lwb}\,(\Delta^1\mathbf{a}_{k,0}(\rho_a,\rho_b), \Delta^1\mathbf{a}_{k,1}(\rho_a,\rho_b), \Delta^1\mathbf{a}_{k,2}(\rho_a,\rho_b)),\\
\vec{\mathbf{rb}}_1 &= \mathrm{rwb}\,(\Delta^1\mathbf{a}_{k,0}(\rho_a,\rho_b), \Delta^1\mathbf{a}_{k,1}(\rho_a,\rho_b), \Delta^1\mathbf{a}_{k,2}(\rho_a,\rho_b)),\\
\vec{\mathbf{lb}}_2 &= \mathrm{lwb}\,(\Delta^2\mathbf{a}_{k,0}(\rho_a,\rho_b), \Delta^2\mathbf{a}_{k,1}(\rho_a,\rho_b)),\\
\vec{\mathbf{rb}}_2 &= \mathrm{rwb}\,(\Delta^2\mathbf{a}_{k,0}(\rho_a,\rho_b), \Delta^2\mathbf{a}_{k,1}(\rho_a,\rho_b)).
\end{aligned} \tag{36}$$

If one of these bounds does not exist, then continue with Step 5.

3.) If $\Lambda < \Lambda_{\max}$ holds and at least one of the first difference vectors $\Delta^1 \mathbf{a}_{k,0}(\rho_a, \rho_b)$, $\ldots, \Delta^1 \mathbf{a}_2(\rho_a, \rho_b)$ vanishes (cf. the remark on Step 1), then continue with Step 5. Similarly, if one of the second difference vectors $\Delta^2 \mathbf{a}_{k,0}(\rho_a, \rho_b), \Delta^2 \mathbf{a}_{k,1}(\rho_a, \rho_b)$ vanishes, then continue with Step 5, provided that this is not implied by one of the desired inflections (30). These subdivisions are necessary due to (i). Let

$$\phi = \begin{cases} 0 \text{ if } \Lambda \geq \Lambda_{\max} \\ \gamma \operatorname{angle}(\vec{\mathbf{rb}}_1, \vec{\mathbf{lb}}_1) \end{cases} \tag{37}$$

where angle $(.,.)$ is the oriented angle between two vectors. If the angles between the two wedges spanned by the first and second differences of the control points and also between the corresponding "negative wedges" is less than ϕ, then subdivide the curve segment. More precisely, if the inequalities

$$\operatorname{angle}(\sigma\, \vec{\mathbf{lb}}_1, \vec{\mathbf{rb}}_2) < \phi \text{ or } \operatorname{angle}(\vec{\mathbf{lb}}_2, -\sigma\, \vec{\mathbf{rb}}_1) < \phi \tag{38}$$

(cf. Figure 3) hold, then continue with Step 5. The choice of the angle ϕ is motivated by the desired property (ii) for the algorithm: if $\Lambda \geq \Lambda_{\max}$ holds, then only unavoidable subdivisions are performed. The constant γ has to be specified by the user. In our examples we chose $\gamma = 0.3$.

4.) Add the linear inequalities

$$\left.\begin{array}{l} [\vec{\mathbf{rb}}_1 - \sigma\, \vec{\mathbf{lb}}_2, \Delta^1 \mathbf{b}_{k,i}(\rho_a, \rho_b)] \geq 0 \\ [\vec{\mathbf{lb}}_1 + \sigma\, \vec{\mathbf{rb}}_2, \Delta^1 \mathbf{b}_{k,i}(\rho_a, \rho_b)] \leq 0 \end{array}\right\} i = 0, 1, 2 \quad \text{and}$$
$$\left.\begin{array}{l} [\vec{\mathbf{rb}}_2 + \sigma\, \vec{\mathbf{lb}}_1, \Delta^2 \mathbf{b}_{k,j}(\rho_a, \rho_b)] \geq 0 \\ [\vec{\mathbf{lb}}_2 - \sigma\, \vec{\mathbf{rb}}_1, \Delta^2 \mathbf{b}_{k,j}(\rho_a, \rho_b)] \leq 0 \end{array}\right\} j = 0, 1 \tag{39}$$

to the set $\mathcal{I}$ of inequality constraints and return. In order to avoid active inequalities (cf. (i)), the bounds for the differences of the control points $\mathbf{b}_{k,i}$ are chosen as the bisectors of the angles between the wedges spanned by the differences of the control points $\mathbf{a}_{k,i}$ (indicated by the dashed lines in Figure 3).

5.) Subdivide the curve segment at its midpoint and apply the algorithm to the obtained segments:

$$\begin{array}{l} \texttt{generate\_lscc}(\rho_a,\ \tfrac{1}{2}(\rho_a + \rho_b),\ \Lambda{+}1,\ \sigma,\ k), \\ \texttt{generate\_lscc}(\tfrac{1}{2}(\rho_a + \rho_b),\ \rho_b,\ \Lambda{+}1,\ \sigma,\ k). \end{array} \tag{40}$$

The algorithm is applied to all segments $t \in [\tau_{k-1}, \tau_k]$ $(k = 1, \ldots, K)$ of the approximating spline curve (23),

$$\texttt{generate\_lscc}(0,\ 1,\ 0,\ \sigma_j,\ k) \quad \text{for} \quad j = 1, \ldots, I;\ k = w(j{-}1) + 1, \ldots, w(j). \tag{41}$$

Due to Proposition 3 it is guaranteed to be successful, provided that the reference curve possesses the desired shape and it satisfies the conditions (30).

6 Computing the approximating spline curve

With the help of the algorithm `generate_lscc(...)` we obtain two sets $\mathcal{E}$ and $\mathcal{I}$ of equality and inequality constraints for the unknown control points $\mathbf{d}_0, \ldots, \mathbf{d}_P$ which are sufficient conditions for the desired shape of the approximating curve. (Note that the Bézier control points used in the previous section are to be replaced by certain constant affine combinations of the de Boor points of the approximating spline curve.) Now we are able to compute the unknown control points by minimizing the least–squares sum

$$\sum_{i=0}^{N} \|\mathbf{x}(t_i) - \mathbf{p}_i\|^2 = \sum_{i=0}^{N} \| \sum_{j=0}^{P} N_j^d(t_i)\, \mathbf{d}_j - \mathbf{p}_i\|^2 \rightarrow \min \tag{42}$$

subject to the constraints $\mathcal{E}$ and $\mathcal{I}$.

The convexity constraints imply, that the second derivative of the approximating curve vanishes at the desired interior inflections,

$$\ddot{\mathbf{x}}(\tau_{w(j)}) = 0 \qquad (j = 1, \ldots, I-1). \tag{43}$$

This is equivalent to linear equations which are analogous to (31). Using the resulting equations we eliminate the control points $\mathbf{d}_{w(j)+1}$ from the problem.

The objective function (42) is a quadratic expression in the components of the control points $\mathbf{d}_0, \ldots, \mathbf{d}_P$. Therefore, its minimization subject to the constraints $\mathcal{I}$ and $\mathcal{E}$ is a quadratic programming (QP) problem, see (Boot, 1964; Fletcher, 1991). For solving the analogous problem in the functional case, Dierckx (1993) uses the Theil–van de Panne algorithm. This method is restricted to relatively small QP problems, otherwise the required computing time becomes too large. For our problem we use an active set strategy as described in the textbook by Fletcher (1991, pp. 240–245). Basically, this method works in the following way:

1.) Choose the reference curve as initial point $\mathbf{D} = (\mathbf{d}_0, \ldots, \mathbf{d}_P) = (\mathbf{c}_0, \ldots, \mathbf{c}_P)$.

2.) Determine the set $\mathcal{A}$ of inequality constraints which are active for the current point $\mathbf{D}$. In order to decide whether some of them must be removed from $\mathcal{A}$, one has to compute certain Lagrangian multipliers, see (Fletcher, 1991, p. 241). If the set $\mathcal{A}$ is the same as in the previous step, then stop.

3.) Compute the point $\mathbf{D}_{\min}$ which minimizes the objective function (42) subject to $\mathcal{E}$ and to the active inequality constraints with the help of Lagrangian multipliers. The new current point is $(1-\lambda)\,\mathbf{D} + \lambda\,\mathbf{D}_{\min}$, where the stepsize $\lambda \in [0,1]$ is chosen as big as possible, but such that the new point satisfies all inequality constraints. Continue with step 2.).

According to (Fletcher, 1991, p. 242), most solvers for quadratic programming problems have difficulties with degenerate situations. Such situations occur if some of the active inequality constraints are redundant. This causes problems for the choice of the active constraints. It is therefore of crucial importance to avoid redundancies as far as possible. For instance, if we linearize the convexity constraints as described in the previous section, then the tangent vectors at the interior knots

$$\dot{\mathbf{x}}(\tau_k) = 3\,\Delta^1 \mathbf{b}_{k-1,2} = 3\,\Delta^1 \mathbf{b}_{k,0} \qquad (k = 1, \ldots, K-1) \tag{44}$$

are generally subject to four linear inequality constraints. These constraints result from the bounds for the wedges of the first derivative vectors for the left and for the right curve segment. Two of them are redundant, they should be deleted from the set of inequality constraints. Similarly one should check for redundant constraints at the segment boundaries generated by the subdivision step of the algorithm `generate_lscc`(...), and also for redundant constraints involving second difference vectors of the control points.

The quadratic programming yields the control points $\mathbf{d}_0, \ldots, \mathbf{d}_P$ of the approximating spline curve which possesses the desired shape. It is now possible to iterate the construction: the approximating curve can be used as a reference curve to generate new sets of equality and inequality constraints. These sets can be expected to be more appropriate for the approximation of the data than the original constraints. Then, an improved approximating curve can be found by minimizing the least squares sum (42) subject to the new constraints.

As an example, Figure 4 shows the shape–preserving least–squares approximation of the 63 data points from Section 4, see Figure 2. Figure 4a depicts the shape preserving least–squares approximation over the knot vector indicated in Figure 2a. In order to compare the results, the corresponding unconstrained least–squares approximation is shown in Figure 4b. For both curves, the curves and their B–spline control polygons have been drawn. Additionally, the curvature distribution of both spline curves is visualized by “porcupines” (cf. Section 4). The approximation without constraints shown in Figure 4b possesses 11 inflections and very high curvatures at its end point, whereas the curve in Figure 4a clearly has the desired shape.

The least–squares sums of the reference curve, the shape–preserving approximation, and the approximation without constraints are equal to 17.4, 9.5, and 7.4, respectively. The shape–preserving approximating curve possesses 42 degrees of freedom. The final result was obtained after four iterations of the above–described method. The first QP problem had 140 inequality, but no equality constraints, and the resulting least–squares sum was equal to 9.84 (4 seconds needed on a HP 715/64 workstation, 12 binding inequalities). For the second (third) iteration we got 144 (162) inequality and 4 (4) equality constraints, and the least–squares sum was now

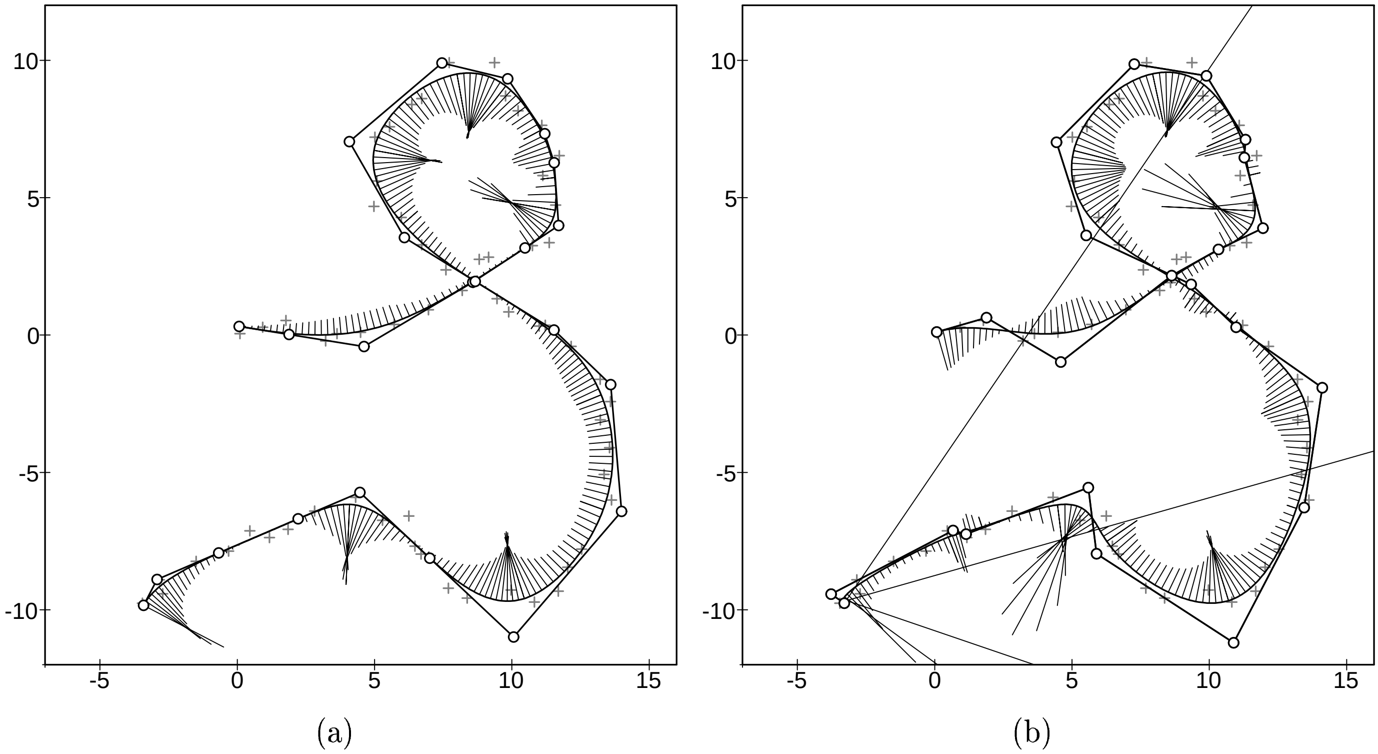

Figure 4: The shape preserving approximation (a) and the corresponding unconstrained approximation (b).

equal to 9.63 (9.57) (2 (2) seconds needed, 6 (6) binding inequalities). Finally, for the fourth iteration we solved a QP problem with 180 inequality and 4 equality constraints and got the value 9.53 (2 seconds needed, 6 binding inequalities).

Concluding remarks

The method described in the previous sections yields an approximate solution of the problem of shape–preserving least–squares approximation by polynomial parametric spline curves. The desired shape of the approximant is specified with the help of a reference curve, which is then used in order to generate an appropriate set of linear sufficient constraints ensuring the convexity of the approximating curve.

A slight modification of the conditions obtained from Lemma 1 could guarantee, that the curvature of the approximating spline curve is nonzero at all inner points of the spline segments, and that the first derivative possesses a certain minimal magnitude. This would be achieved by introducing additional constraints

$$[\epsilon\,(\vec{\mathbf{u}}+\vec{\mathbf{v}}), \Delta^2\mathbf{b}_i] \quad (i=0,\ldots,d-2), \quad \text{and} \quad [\epsilon\,(\vec{\mathbf{u}}-\vec{\mathbf{v}}), \Delta^1\mathbf{b}_i] \quad (i=0,\ldots,d-1), \quad (45)$$

(in the case of positive curvature) respectively, where $\epsilon > 0$ is a small constant.

The generated inequality and equality constraints and also the gradient of the objective function (42) possess a band structure which is not exploited in our implementation yet. Using this sparsity might lead to a more efficient approximation scheme.

As pointed out by one of the referees, a possible modification of the scheme could compute a sequence of approximating curves by adaptively adding knots, starting from the knots of the reference curve, until the approximation is as good as desired. Of course this would require an appropriate refinement strategy for the knot vectors. Using this idea may lead to more appropriate knot sequences for the approximation. However, it would result in a higher number of QP problems which need to be solved. In addition, adapting the linearized constraints simultaneously restricts the shape of the approximant as the feasible regions for the first derivatives may get smaller in each step. So it might be better to start with the full set of knots, because then the approximating spline curve has all available degrees of freedom.

As another possibility one could firstly create the knot sequence by adaptively adding knots for the least–squares approximation without constraints. The shape–preserving approximation over the obtained knot sequence would then result in a second step which iteratively adapts the linearized constraints (as in the preceding section).

The linear sufficient convexity constraints can be generalized to piecewise polynomial parametric surfaces. This is described by the forthcoming article (Jüttler, 1997), where also an application to the so–called lifting problem is discussed.

Acknowledgements

This research was performed while the author was visiting the University of Dundee (Scotland). The financial support of this visit by the European Union through the network FAIRSHAPE is gratefully acknowledged. The author wishes to thank Professor T.N.T. Goodman for many helpful discussions concerning this paper. In addition, the author would like to thank the referees for their very useful comments which have helped to improve the paper.

Bibliography

Boot, J. C. G. (1964), *Quadratic Programming.* Rand McNally, Chicago.

Burchard, H. G., Ayers, J. A., Frey, W. H., and Sapidis, N. S. (1994), Approximation with Aesthetic Constraints. in: Sapidis, N. S. (ed.), *Designing Fair Curves and Surfaces.* SIAM, Philadelphia, 3–28.

Dierckx, P. (1980), An algorithm for cubic spline fitting with convexity constraints. Computing 24, 349–371.

Dierckx, P. (1993), *Curve and Surface Fitting with Splines.* Clarendon Press, Oxford.

Elfving, T., and Anderson, L.–E. (1988), An algorithm for computing constrained smoothing spline functions. Numer. Math. 52, 583–595.

Elliott, G. H. (1993), Least squares data fitting using shape preserving piecewise approximations. Numer. Algorithms 5, 365–371.

Farin, G. (1993), *Curves and Surfaces for Computer Aided Geometric Design.* Academic Press, Boston.

Fletcher, R. (1991), *Practical Methods of Optimization.* John Wiley & Sons, Chichester (2nd ed.).

Goodman, T.N.T. (1991), Inflections on curves in two and three dimensions. Comp. Aided Geom. Design 8, 37–50.

Hoschek, J., and Lasser, D. (1993), *Fundamentals of Computer Aided Geometric Design.* AK Peters, Wellesley MA.

Jüttler, B. (1996), Shape preserving least–squares approximation by polynomial parametric spline curves. University of Dundee, Applied Analysis Report 965.

Jüttler, B. (1997), Linear sufficient convexity conditions for parametric tensor–product Bézier– and B–spline surfaces. submitted to: Goodman, T.N.T. (ed.), *The Mathematics of Surfaces VII.* Oxford University Press.

Micchelli, C. A., and Utreras, F. I. (1988), Smoothing and interpolation in a convex subset of a Hilbert space. SIAM J. Sci. Stat. Comput. 9, 728–746.

Schmidt, J. W., and Scholz, I. (1990), A dual algorithm for convex–concave data smoothing by cubic C^2–splines. Numer. Math. 57, 333–350.

Schwetlick, H., and Kunert, V. (1993), Spline smoothing under constraints on derivatives. BIT 33, 512–528.