Surface fitting using convex tensor–product splines

Bert Jüttler

Technische Universität Darmstadt, Fachbereich Mathematik
Schloßgartenstraße 7, 64289 Darmstadt, Germany
juettler@mathematik.th-darmstadt.de

Abstract. A construction of linear sufficient convexity conditions for polynomial tensor–product spline functions is presented. As the main new feature of this construction, the obtained conditions are *asymptotically necessary:* increasing the number of linear inequalities in a suitable manner adapts them to any finite set of strongly convex spline surfaces. Based on the linear constraints we formulate least–squares approximation of scattered data by spline surfaces as a quadratic programming problem.

Keywords. convexity conditions, surface fitting, tensor–product spline functions, quadratic programming.

AMS Subject Classification. 65D07, 65D15, 65D17, 68U05, 90C20.

Introduction

Convexity conditions for bivariate spline functions and the construction of convex spline functions from given data has been subject of a remarkable number of publications. Results on convexity conditions for multivariate polynomials, in particular for polynomials in Bernstein–Bézier representation over simplices, are summarized in the survey article by Dahmen [6]. In the case of convexity constraints for tensor–product spline surfaces, however, only relatively few related publications seem to exist. Very strong conditions (which are only fulfilled by convex translational surfaces) have been derived by Schelske in his Ph.D. thesis, see also [12]. They have been rediscovered by Cavaretta and Sharma in 1990 (cited in [6]). Some weaker constraints which lead to systems of quadratic inequalities for the Bézier coefficients have been developed by Floater [11]. A recent preprint of Carnicer, Floater and Peña [2] derives weakened linear conditions.

In Section 2 we present a construction of linear sufficient convexity conditions for polynomial tensor–product spline functions. For the sake of simplicity, the construction is described only for bicubic spline functions, but it applies to splines of arbitrary degree. As the main new feature of our construction, the obtained conditions can be adapted to any strongly convex spline function, simply by increasing the number of linear inequalities. Moreover we can even adapt the constraints to *any finite set of strongly convex spline functions.*

One may identify a spline surface with a point in $\mathbb{R}^d$, where d is the number of coefficients of the spline surface: the coefficients of the surface serve as coordinates of the point. Then, points in $\mathbb{R}^d$ which correspond to convex surfaces form a convex cone $\Omega \subset \mathbb{R}^d$, as non–negative linear combinations of convex functions are again convex. The linear convexity constraints of our construction describe a certain convex polyhedral cone $\Omega^* \subset \Omega$ of $\mathbb{R}^d$. We can weaken them such that this cone contains any other polyhedron $\hat{\Omega} \subset \Omega$ which has no points on the boundary $\partial\Omega$ of Ω.

In this sense, the linear constraints of Section 2 could be said to be *asymptotically necessary* (if the number if inequalities is increased in a suitable manner). These conditions can be made as weak as necessary for the specific application. In Section 2.3 we present a comparison with the linear conditions of Carnicer et al.

In the remainder of the paper we use the linearized constraints in order to construct a bicubic spline function which approximates given scattered data and fulfills additional convexity or concavity constraints. Constructions for surfaces from given data with various shape restrictions have been discussed by various authors (e.g., [3, 13, 15]), and it is virtually impossible to give a complete list here. Willemans and Dierckx [18] use piecewise quadratic functions over Powell–Sabin splits for bivariate least–squares approximation of scattered data with convexity constraints. Based on convexity conditions obtained by Chang and Feng [4] they are led to a quadratic optimization problem with linear and quadratic constraints.

We consider least–squares approximation by tensor–product spline functions subject to segment-wise convexity and concavity constraints. With the help of the previously constructed linear sufficient convexity conditions we are able to formulate this task as a quadratic programming problem (minimization of a quadratic function subject to linear constraints). This generalizes the method proposed by Dierckx [7] to the bivariate case. A strategy for adapting the linear constraints to the given data is presented. In order to keep notations relatively simple we consider only the bicubic case, but it would be possible to generalize the results to arbitrary degrees. The method is illustrated by two examples.

1 The approximation problem

In this article we develop a method for solving the following approximation problem. A set of $R+1$ data $\{(x_i, y_i, z_i) \mid i = 0, \ldots, R\} \subset [a, b] \times [c, d] \times \mathbb{R}$ is assumed to be given. These data are to be approximated by a bicubic tensor–product spline function which is required to possess a certain specified shape (see below),

$$f(x,y) = \sum_{i=0}^{P-4} \sum_{j=0}^{Q-4} M_i(x)\, N_j(y)\, d_{i,j}, \tag{1}$$

with the unknown coefficients $d_{i,j} \in \mathbb{R}$. The B–spline basis functions $(M_i(x))_{i=0}^{P-4}$ and $(N_j(y))_{j=0}^{Q-4}$ are defined over the given knot sequences

$$\Xi = (\xi_0, \xi_1, \ldots, \xi_P) \text{ and } \Theta = (\theta_0, \theta_1, \ldots, \theta_Q), \tag{2}$$

$(P, Q \geq 7)$ respectively, whose knots are to satisfy

$$\begin{aligned} a = \xi_0 = \xi_1 = \xi_2 = \xi_3 < \xi_4 < \ldots < \xi_{P-4} < \xi_{P-3} = \xi_{P-2} = \xi_{P-1} = \xi_P = b, \\ c = \theta_0 = \theta_1 = \theta_2 = \theta_3 < \theta_4 < \ldots < \theta_{Q-4} < \theta_{Q-3} = \theta_{Q-2} = \theta_{Q-1} = \theta_Q = d. \end{aligned} \tag{3}$$

For more information on spline functions we refer to the textbooks by Dierckx [8] or Hoschek and Lasser [12]. Choosing 4–fold boundary knots we obtain B–spline basis functions whose support is contained within the intervals $[a, b]$ and $[c, d]$. Moreover we get a C^2 spline surface, as all inner knots possess the multiplicity 1.

The choice of appropriate knots for bivariate spline fitting is a non–trivial problem. For more information the reader is referred to [8, Chapter 9], where an automatic and adaptive algorithm for locating the knots is described (see also the concluding remarks).

Restricting the spline function (1) to the segment

$$(x,y) \in D^{(i,j)} = [\xi_{i+3}, \xi_{i+4}] \times [\theta_{j+3}, \theta_{j+4}] \quad (0 \leq i \leq P-7, \;\; 0 \leq j \leq Q-7) \tag{4}$$

we get a bicubic polynomial $f^{(i,j)}(x, y)$ which may be represented in Bernstein/Bézier form,

$$f^{(i,j)}(x,y) = \sum_{r=0}^{3} \sum_{s=0}^{3} B_r^3(x^{(i)})\, B_s^3(y^{(j)})\, b_{r,s}^{(i,j)} \tag{5}$$

with the local parameters

$$x^{(i)} = \frac{x - \xi_{i+3}}{\xi_{i+4} - \xi_{i+3}}, \quad y^{(j)} = \frac{y - \theta_{j+3}}{\theta_{j+4} - \theta_{j+3}} \tag{6}$$

and the cubic Bernstein polynomials $B_k^3(z) = \binom{3}{k} z^k (1-z)^{3-k}$. The coefficients $b_{r,s}^{(i,j)}$ are affine combinations of the B–spline coefficients $d_{i,j}$. They can be constructed using the knot insertion algorithm. For the convenience of the reader, the conversion formulas are provided by Appendix A.

As the main feature of our scheme, the user has the possibility to specify the shape of the spline segments $D^{(i,j)}$. This is done by choosing a value $\sigma^{(i,j)} \in \{-1, 0, 1\}$ for each spline segment. The spline segment $D^{(i,j)}$ will be guaranteed to be convex (concave) for $\sigma^{(i,j)} = 1$ ($\sigma^{(i,j)} = -1$), whereas no restrictions are imposed if $\sigma^{(i,j)} = 0$

holds. We assume, however, that no neighbouring concave and convex spline segments exist,

$$\forall i_1, j_1, i_2, j_2 \text{ with } 0 \leq i_1, i_2 \leq P-7 \text{ and } 0 \leq j_1, j_2 \leq Q-7: \\ \sigma^{(i_1,j_1)}\,\sigma^{(i_2,j_2)} = -1 \Rightarrow \max\{|i_1-i_2|, |j_1-j_2|\} > 1. \tag{7}$$

In order to formulate constraints which guarantee the convexity of the spline segments $D^{(i,j)}$, we have to consider their partial second derivatives

$$f^{(i,j,k,2-k)}(x,y) = \frac{\partial^2}{\partial x^k\, \partial y^{2-k}} f^{(i,j)}(x,y) = \sum_{r=0}^{3}\sum_{s=0}^{3} B_r^3(x^{(i)})\, B_s^3(y^{(j)})\, b_{r,s}^{(i,j,k,2-k)} \tag{8}$$

($k = 0, 1, 2$). Note that these derivatives are polynomials of degree $(3,1)$, $(2,2)$ and $(1,3)$. We represent them by bicubic polynomials as we will have to compute certain linear combinations of them later. The formulas for their Bézier coefficients $b_{r,s}^{(i,j,k,2-k)}$ are stated in Appendix B.

2 Linear convexity conditions

In the first part of this section we examine some linear conditions which guarantee, that a symmetric 2×2–matrix is non–negative definite (also called positive semi–definite). Based on these results we present linear sufficient convexity conditions for the bicubic polynomial segments of the approximating spline function $f(x, y)$.

2.1 Non–negative definite 2×2–matrices

At first we consider a symmetric ($h_{2,1} = h_{1,2}$) real 2×2–matrix $H = (h_{i,j})_{i,j=1,2}$. In addition to the matrix, the two strictly increasing finite sequences $\Psi = (\psi_0, \psi_1, \ldots, \psi_R)$ and $\Phi = (\phi_0, \phi_1, \ldots, \phi_S)$ satisfying

$$0 = \psi_0 < \psi_1 < \psi_2 < \ldots < \psi_R = 1 \text{ and } 0 = \phi_0 < \phi_1 < \phi_2 < \ldots < \phi_S = 1 \\ \text{with } R, S > 1 \tag{9}$$

are assumed to be given. Let $p(u, v)$ and $q(u, v)$ denote the bilinear functions

$$p(u,v) = (1-u)\,(1-v)\,h_{1,1} + (u+v-2\,u\,v)\,h_{1,2} + u\,v\,h_{2,2}, \\ q(u,v) = (1-u)\,(1-v)\,h_{1,1} - (u+v-2\,u\,v)\,h_{1,2} + u\,v\,h_{2,2} \tag{10}$$

of the parameters u, v. These functions are symmetric: $p(u,v) = p(v,u)$, $q(u,v) = q(v,u)$. Moreover, restricting them to the diagonal leads to the values of the quadratic polynomials

$$p(u,u) = (\,u \;\; 1{-}u\,) \cdot H \cdot \begin{pmatrix} u \\ 1{-}u \end{pmatrix} \quad \text{and} \quad q(u,u) = (\,u \;\; u{-}1\,) \cdot H \cdot \begin{pmatrix} u \\ u{-}1 \end{pmatrix}. \tag{11}$$

Using these bilinear functions we associate with any matrix H and with any two sequences Ψ, Φ the set

$$\mathcal{I}(H, \Psi, \Phi) = \{p(\psi_{i-1}, \psi_i) \geq 0 \mid i = 1, .., R\} \cup \{q(\phi_{j-1}, \phi_j) \geq 0 \mid j = 1, .., S\} \quad (12)$$

of inequalities. If we assume that the sequences Ψ, Φ are known, then this set consists of $R + S$ linear inequalities for the components $h_{i,j}$ of the matrix H.

The construction of the set of inequalities $\mathcal{I}(H, \Psi, \Phi)$ is based on the idea of *blossoming*, which is applied to the quadratic polynomials (11). Blossoming is one of the standard techniques in Computer Aided Geometric Design, see [12]. It is a compact way to generate the control points of subsegments of polynomials in Bernstein/Bézier form. For instance, representing $p(u, u)$ as polynomial over $u \in [\psi_{i-1}, \psi_i]$ leads to the Bézier coefficients $p(\psi_{i-1}, \psi_{i-1}), p(\psi_{i-1}, \psi_i)$ and $p(\psi_i, \psi_i)$.

The inequalities $\mathcal{I}(H, \Psi, \Phi)$ possess the following properties:

Lemma 1. *If the components $h_{i,j}$ of the symmetric real 2×2–matrix H fulfill the inequalities $\mathcal{I}(H, \Psi, \Phi)$ for two arbitrary but fixed finite sequences Ψ, Φ satisfying (9), then the matrix H is non–negative definite.*

Proof. It has to be shown that the conditions of the system $\mathcal{I}(H, \Psi, \Phi)$ imply

$$\bar{z}_1^2\, h_{1,1} + 2\, \bar{z}_1\, \bar{z}_2\, h_{1,2} + \bar{z}_2^2\, h_{2,2} \geq 0 \quad (13)$$

for any $(\bar{z}_1, \bar{z}_2) \in \mathbb{R}^2 \setminus \{(0, 0)\}$. Resulting from $\psi_0 = \phi_0 = 0$ we get

$$\begin{aligned} p(\psi_0, \psi_1) &= (1 - \psi_1)\, h_{1,1} + \psi_1\, h_{1,2} \geq 0 \text{ and} \\ q(\phi_0, \phi_1) &= (1 - \phi_1)\, h_{1,1} - \phi_1\, h_{1,2} \geq 0. \end{aligned} \quad (14)$$

Hence, as $R, S > 1$ (and therefore $\psi_1, \phi_1 < 1$) was assumed, we have $p(\psi_0, \psi_0) = q(\phi_0, \phi_0) = h_{1,1} \geq 0$. Analogously, the two inequalities $p(\psi_{R-1}, \psi_R) \geq 0$ and $q(\phi_{S-1}, \phi_S) \geq 0$ yield $p(\psi_R, \psi_R) = q(\phi_S, \phi_S) = h_{2,2} \geq 0$.

First case: $\bar{z}_1\, \bar{z}_2 \geq 0$. We project (z_1, z_2) onto a point $(z, 1 - z) \in \mathbb{R}^2$, see Fig. 1. Choosing $\alpha = \frac{\mathrm{sgn}(\bar{z}_1)}{|\bar{z}_1| + |\bar{z}_2|}$ we obtain $\bar{z}_1 = \alpha\, z$ and $\bar{z}_2 = \alpha\, (1 - z)$ for $z = \frac{1}{\alpha} \bar{z}_1$ (but $z = 0$ for $\bar{z}_1 = 0$). Note that $z \in [0, 1]$ holds. The above inequality (13) is therefore equivalent to

$$\alpha^2 \left(z^2 h_{1,1} + 2\, z\, (1 - z)\, h_{1,2} + (1 - z)^2 h_{2,2} \right) = \alpha^2\, p(z, z) \geq 0. \quad (15)$$

We have $z \in [\psi_{i-1}, \psi_i]$ for some fixed i, $1 \leq i \leq R$. Exploiting twice the linearity and symmetry of $p(., .)$ in each of its arguments we obtain

$$\begin{aligned} p(z, z) &= \tfrac{\psi_i - z}{\psi_i - \psi_{i-1}}\, p(\psi_{i-1}, z) + \tfrac{z - \psi_{i-1}}{\psi_i - \psi_{i-1}}\, p(\psi_i, z) \\ &= \underbrace{\tfrac{(\psi_i - z)^2}{(\psi_i - \psi_{i-1})^2}}_{(*)}\, \underbrace{p(\psi_{i-1}, \psi_{i-1})}_{(a)} + 2\, \underbrace{\tfrac{(\psi_i - z)(z - \psi_{i-1})}{(\psi_i - \psi_{i-1})^2}}_{(*)}\, \underbrace{p(\psi_{i-1}, \psi_i)}_{(b)} \\ &+ \underbrace{\tfrac{(z - \psi_{i-1})^2}{(\psi_i - \psi_{i-1})^2}}_{(*)}\, \underbrace{p(\psi_i, \psi_i)}_{(c)}. \end{aligned} \quad (16)$$

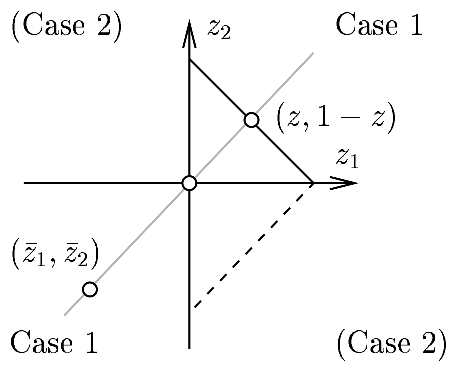


Figure 1: Normalizing the direction $(\bar z_1, \bar z_2)$.

The three terms $(*)$ are non–negative as $z \in [\psi_{i-1}, \psi_i]$ holds. According to one of the inequalities of the set $\mathcal{I}$, also the middle coefficient (b) is guaranteed to be non–negative. Consider the first coefficient (a). In case $i = 1$ it is nonnegative due to $h_{1,1} \geq 0$, as already observed earlier. Otherwise we have

$$p(\psi_{i-1}, \psi_{i-1}) = \underbrace{\tfrac{\psi_i - \psi_{i-1}}{\psi_i - \psi_{i-2}}}_{(*)} \underbrace{p(\psi_{i-2}, \psi_{i-1})}_{(d)} + \underbrace{\tfrac{\psi_{i-1} - \psi_{i-2}}{\psi_i - \psi_{i-2}}}_{(*)} \underbrace{p(\psi_{i-1}, \psi_i)}_{(b)} \geq 0. \tag{17}$$

The terms $(*)$ are non–negative due to (9), whereas two inequalities from $\mathcal{I}$ guarantee the non–negativity of the coefficients $(b), (d)$. Similarly we conclude that also the last coefficient (c) is guaranteed to be non–negative. Therefore we have $p(z, z) \geq 0$ in this case.

Second case: $\bar z_1 \, \bar z_2 < 0$. We project (z_1, z_2) onto a point $(z, -(1-z)) \in \mathbb{R}^2$, see again Fig. 1. Choosing once more $\alpha = \frac{\operatorname{sgn}(\bar z_1)}{|\bar z_1| + |\bar z_2|}$ we have $\bar z_1 = \alpha\, z$ and $\bar z_2 = -\alpha\,(1-z)$ for $z = \frac{1}{\alpha} \bar z_1$ $(z \in [0,1])$. Inequality (13) is therefore equivalent to

$$\alpha^2 \left(z^2 h_{1,1} - 2\, z\,(1-z)\, h_{1,2} + (1-z)^2 h_{2,2} \right) = \alpha^2\, q(z,z) \geq 0. \tag{18}$$

Using similar arguments as in the first part of the proof (now applied to the polynomial $q(z, z)$) shows that this inequality is again implied by the conditions of the set $\mathcal{I}$. This completes the proof. □

The lemma is based on conditions which guarantee the non–negativity of a polynomial on an interval. Such conditions have been studied thoroughly in the literature, see e.g. [16]. As the basic new idea behind the convexity conditions of the present paper, we show that convexity of bivariate functions can be guaranteed by non–negativity of polynomials on intervals, and that this leads to arbitrarily weak linear conditions.

The inequalities $\mathcal{I}(H, \Psi, \Phi)$ imply that the matrix H is non–negative definite. The finite sequences Ψ, Φ control how necessary these conditions are. Refining these sequences leads to weaker conditions:

Lemma 2. *Assume that the finite sequences $\bar{\Psi}, \bar{\Phi}$ result by inserting additional knots into the finite sequences Ψ, Φ satisfying (9), i.e., the relations $\{\Psi\} \subseteq \{\bar{\Psi}\}$ and $\{\Phi\} \subseteq \{\bar{\Phi}\}$ hold. Then the inequalities $\mathcal{I}(H, \bar{\Psi}, \bar{\Phi})$ imply the inequalities $\mathcal{I}(H, \Psi, \Phi)$.*

The proof of this observation is again a direct consequence from the blossoming principle. For the sake of brevity it is omitted here.

It can be shown, that the inequalities $\mathcal{I}(H, \Psi, \Phi)$ can be applied to any positive definite matrix:

Lemma 3. *Consider an arbitrary but positive definite symmetric real 2×2 matrix H, i.e., we have $\mathbf{z}^\top H\,\mathbf{z} > 0$ for all $\mathbf{z} \in \mathbb{R}^2 \setminus \{\mathbf{0}\}$. Then two finite sequences Ψ, Φ satisfying (9) exist such that the inequalities $\mathcal{I}(H, \Psi, \Phi)$ are fulfilled.*

Proof. Consider the finite sequences $\Psi^{(l)} = \Phi^{(l)} = \{\frac{k}{2^l} \mid k = 0, \ldots, 2^l\}$ for $l = 1, 2, \ldots$ We assume the above assertion is not satisfied for all $l \in \mathbb{N}$. This assumption will be shown to lead to a contradiction.

For each l we consider the (possibly empty) sets

$$\begin{aligned} \mathcal{P}^{(l)} &= \{\tfrac{k}{2^l} \,|\, p(\tfrac{k-1}{2^l}, \tfrac{k}{2^l}) < 0 \text{ and } 1 \le k \le 2^l, k \in \mathbb{N}\}, \\ \mathcal{Q}^{(l)} &= \{\tfrac{k}{2^l} \,|\, q(\tfrac{k-1}{2^l}, \tfrac{k}{2^l}) < 0 \text{ and } 1 \le k \le 2^l, k \in \mathbb{N}\}. \end{aligned} \tag{19}$$

According to our assumption, for each l at least one of these sets is non–empty. Moreover we assume that no number l_0 exists such that $\mathcal{P}^{(l)} = \emptyset$ holds for all $l > l_0$. (If this is not true, then we may consider the sets $\mathcal{Q}^{(l)}$ instead.) As all sets $\mathcal{P}^{(l)}$ are contained in the unit interval $[0, 1]$ we can choose a convergent sequence $\left(\frac{k(j)}{2^{l(j)}}\right)_{j=1,2,\ldots}$ with $\frac{k(j)}{2^{l(j)}} \in \mathcal{P}^{(l(j))}$ and strictly increasing refinement levels $l(j)$: $l(j) < l(j+1)$. Let $\bar{\psi} = \lim\limits_{j\to\infty} \frac{k(j)}{2^{l(j)}}$ be the limit of this sequence. According to our construction, we have

$$\forall j : p(\,\frac{k(j)-1}{2^{l(j)}}, \frac{k(j)}{2^{l(j)}}\,) < 0, \text{ hence } p(\bar{\psi}, \bar{\psi}) = \lim_{j\to\infty} p(\,\frac{k(j)-1}{2^{l(j)}}, \frac{k(j)}{2^{l(j)}}\,) \le 0. \tag{20}$$

On the other hand we get

$$p(\bar{\psi}, \bar{\psi}) = (\bar{\psi} \;\; 1-\bar{\psi})^\top \cdot H \cdot \begin{pmatrix} \bar{\psi} \\ 1-\bar{\psi} \end{pmatrix} > 0 \tag{21}$$

as the matrix H is positive definite. This is a contradiction. □

The proof of the lemma uses a simple uniform refinement of the sequences Ψ and Φ, independent on the given positive definite matrix H. As a consequence of Lemma 2 we therefore have:

Corollary 4. *For any finite set of positive definite matrices two finite sequences Ψ, Φ satisfying (9) exist, such that the inequalities $\mathcal{I}(H, \Psi, \Phi)$ are fulfilled for all matrices. For instance, choosing $\Psi = \Psi^{(L)} = \Phi = \Phi^{(L)}$ (see the proof of Lemma 3) with some (big enough) refinement level L yields such a set of inequalities.*

Proof. Applying the previous Lemma yields finite sequences Ψ, Φ for each matrix. Taking the union of these finite sequences leads to the desired set of inequalities. □

One could say that the inequalities $\mathcal{I}(H, \Psi, \Phi)$ are *asymptotically necessary:* they can be made as weak as desired.

2.2 Convex bicubic Bézier patches

Now we consider one of the bicubic polynomial segments $f^{(i,j)}(x, y)$ with parameter domain $(x, y) \in D^{(i,j)}$. The coefficients of the patch $b_{r,s}^{(i,j)}$ and those of the second partial derivatives $b_{r,s}^{(i,j,2,0)}$, $b_{r,s}^{(i,j,1,1)}$, $b_{r,s}^{(i,j,0,2)}$ $(r, s = 0, 1, 2)$ are certain constant linear combinations of the B–spline coefficients $(d_{k,l})_{k=i,..,i+3;\, l=j,..,j+3}$.

In addition to the sequences Ψ and Φ from the previous section, we assume that the two strictly increasing sequences $\Pi = (\pi_0, \pi_1, \ldots, \pi_T)$ and $\Lambda = (\lambda_0, \lambda_1, \ldots, \lambda_U)$ satisfying

$$0 = \pi_0 < \pi_1 < \pi_2 < \ldots < \pi_T = 1 \text{ and } 0 = \lambda_0 < \lambda_1 < \lambda_2 < \ldots < \lambda_U = 1 \\ \text{with } T, U > 0 \tag{22}$$

are given. These sequences are to be used in order to subdivide the patches $f^{(i,j)}(x, y)$. In order to simplify notations we introduce the following abbreviation: We denote by $H(u_1, v_1, w_1, u_2, v_2, w_2) = (h_{i,j})_{i,j=1,2}$ the real symmetric 2×2–matrix with the components

$$\begin{aligned} h_{1,1} &= \sum_{r=0}^{3}\sum_{s=0}^{3} \rho_r(u_1, v_1, w_1)\, \rho_s(u_2, v_2, w_2)\, b_{r,s}^{(i,j,2,0)}, \\ h_{1,2} = h_{2,1} &= \sum_{r=0}^{3}\sum_{s=0}^{3} \rho_r(u_1, v_1, w_1)\, \rho_s(u_2, v_2, w_2)\, b_{r,s}^{(i,j,1,1)}, \\ h_{2,2} &= \sum_{r=0}^{3}\sum_{s=0}^{3} \rho_r(u_1, v_1, w_1)\, \rho_s(u_2, v_2, w_2)\, b_{r,s}^{(i,j,0,2)}, \end{aligned} \tag{23}$$

with

$$\begin{aligned} \rho_0(u, v, w) &= (1-u)(1-v)(1-w), \\ \rho_1(u, v, w) &= u + v + w - 2uv - 2vw - 2uw + 3uvw, \\ \rho_2(u, v, w) &= uv + vw + uw - 3uvw, \\ \rho_3(u, v, w) &= uvw. \end{aligned} \tag{24}$$

Similar to the functions $p(u, v)$ and $q(u, v)$ of the previous section, the matrix components are linear functions in $u_1, v_1, w_1, u_2, v_2, w_2$. The components are invariant with respect to permutations of the arguments u_1, v_1, w_1 and u_2, v_2, w_2. Moreover, restricting the components to the diagonal $x^{(i)} = u_1 = v_1 = w_1$ and $y^{(j)} = u_2 = v_2 = w_2$ yields the values of the partial second derivatives, i.e., the Hessian matrix:

$$\begin{aligned} h_{1,1}(x^{(i)}, x^{(i)}, x^{(i)}, y^{(j)}, y^{(j)}, y^{(j)}) &= f^{(i,j,2,0)}(x, y), \\ h_{1,2} = h_{2,1} = (x^{(i)}, x^{(i)}, x^{(i)}, y^{(j)}, y^{(j)}, y^{(j)}) &= f^{(i,j,1,1)}(x, y), \\ h_{2,2}(x^{(i)}, x^{(i)}, x^{(i)}, y^{(j)}, y^{(j)}, y^{(j)}) &= f^{(i,j,0,2)}(x, y). \end{aligned} \tag{25}$$

Analogously to the previous section, the matrix $H(u_1, v_1, w_1, u_2, v_2, w_2)$ has been constructed by blossoming the Hessian matrix. If we restrict the Hessian matrix to the sub–patch $(x^{(i)}, y^{(j)}) \in [\pi_{k-1}, \pi_k] \times [\lambda_{l-1}, \lambda_l]$, then the corresponding 16 Bézier coefficients are obtained from the blossom $H(u_1, v_1, w_1, u_2, v_2, w_2)$ for the values

$$\begin{aligned} &(u_1, v_1, w_1) \in \{\, (\pi_{k-1}, \pi_{k-1}, \pi_{k-1}), \underbrace{(\pi_{k-1}, \pi_{k-1}, \pi_k), (\pi_{k-1}, \pi_k, \pi_k)}_{(*)}, (\pi_k, \pi_k, \pi_k) \,\} \\ &\text{and} \\ &(u_2, v_2, w_2) \in \{\, (\lambda_{l-1}, \lambda_{l-1}, \lambda_{l-1}), \underbrace{(\lambda_{l-1}, \lambda_{l-1}, \lambda_l), (\lambda_{l-1}, \lambda_l, \lambda_l)}_{(*)}, (\lambda_l, \lambda_l, \lambda_l) \,\}. \end{aligned} \tag{26}$$

We consider the following set of inequalities:

$$\mathcal{J}^{(i,j)}(\Psi, \Phi, \Pi, \Lambda) = \bigcup_{\substack{(u_1,v_1,w_1) \in \Pi^* \\ (u_2,v_2,w_2) \in \Lambda^*}} \mathcal{I}(H(u_1, v_1, w_1, u_2, v_2, w_2), \Psi, \Phi) \tag{27}$$

with

$$\begin{aligned} \Pi^* &= \bigcup_{i=1}^{T} \{(\pi_{i-1}, \pi_{i-1}, \pi_i),\ (\pi_{i-1}, \pi_i, \pi_i)\} \cup \{(\pi_0, \pi_0, \pi_0), (\pi_T, \pi_T, \pi_T)\}, \\ \Lambda^* &= \bigcup_{i=1}^{U} \{(\lambda_{i-1}, \lambda_{i-1}, \lambda_i),\ (\lambda_{i-1}, \lambda_i, \lambda_i)\} \cup \{(\lambda_0, \lambda_0, \lambda_0), (\lambda_U, \lambda_U, \lambda_U)\}. \end{aligned} \tag{28}$$

If we assume that the finite sequences Ψ, Φ, Π and Λ are known, then this set consists of $(2T+2)(2U+2)(R+S)$ linear inequalities for the coefficients $b_{r,s}^{(i,j,k,2-k)}$, hence for the unknown B–spline coefficients $d_{i,j}$. It possesses the following properties:

Theorem 5. *If the inequalities $\mathcal{J}^{(i,j)}$ are fulfilled for four arbitrary but fixed finite sequences Ψ, Φ, Π, Λ satisfying (9) and (22), then the bicubic spline surface segment $f^{(i,j)}(x, y)$, $(x, y) \in D^{(i,j)}$, is convex.*

Proof. It has to be shown that the real 2×2–matrix of the second partial derivatives $H(x^{(i)}, x^{(i)}, x^{(i)}, y^{(j)}, y^{(j)}, y^{(j)})$ (the Hessian matrix) is non–negative definite for all $(x^{(i)}, y^{(j)}) \in [0,1]^2$, i.e., for all $(x, y) \in D^{(i,j)}$ (cf. (6)). We consider an arbitrary but fixed point $(x^{(i)}, y^{(j)}) \in [0,1]^2$. We have $x^{(i)} \in [\pi_{k-1}, \pi_k]$ and $y^{(j)} \in [\lambda_{l-1}, \lambda_l]$ for some fixed k, $1 \le k \le T$, and l, $1 \le l \le U$. Resulting from the inequalities $\mathcal{J}^{(i,j)}$, the 16 matrices $H(u_1, v_1, w_1, u_2, v_2, w_2)$ which are obtained for all possible pairs from (26) are guaranteed to be non–negative definite. This can be seen as follows. For matrices obtained from the triples (u_1, v_1, w_1), (u_2, v_2, w_2) marked by $(*)$, the inequalities $\mathcal{I}(H(...), \Psi, \Phi)$ are contained in the set. It is therefore possible to apply Lemma 1. As an example for the remaining matrices we consider the matrix $H(\pi_{k-1}, \pi_{k-1}, \pi_{k-1}, \lambda_{l-1}, \lambda_{l-1}, \lambda_l)$. If $k = 1$ holds, then the inequalities $\mathcal{I}(H(...), \Psi, \Phi)$ are again contained in the set $\mathcal{J}^{(i,j)}$ and we use Lemma 1. Otherwise we may exploit the linearity and symmetry of the components $h_{i,j}(u_1, v_1, w_1, u_2, v_2, w_2)$ and obtain

$$\begin{aligned} H(\pi_{k-1}, \pi_{k-1}, \pi_{k-1}, \lambda_{l-1}, \lambda_{l-1}, \lambda_l) \\ = \underbrace{\tfrac{\pi_k - \pi_{k-1}}{\pi_k - \pi_{k-2}}}_{(a)} \underbrace{H(\pi_{k-1}, \pi_{k-1}, \pi_{k-2}, \lambda_{l-1}, \lambda_{l-1}, \lambda_l)}_{(b)} \\ + \underbrace{\tfrac{\pi_{k-1} - \pi_{k-2}}{\pi_k - \pi_{k-2}}}_{(a)} \underbrace{H(\pi_{k-1}, \pi_{k-1}, \pi_k, \lambda_{l-1}, \lambda_{l-1}, \lambda_l)}_{(b)}. \end{aligned} \tag{29}$$

The coefficients (a) are positive due to (22). The two matrices (b) are non–negative definite, as the corresponding inequalities $\mathcal{I}(...)$ are contained in the set $\mathcal{J}^{(i,j)}$. Hence, also $H(\pi_{k-1}, \pi_{k-1}, \pi_{k-1}, \lambda_{l-1}, \lambda_{l-1}, \lambda_l)$ is non–negative definite, as it is a non–negative linear combination of two non–negative definite matrices. Analogous considerations apply to the remaining 11 matrices obtained from (26).

Exploiting the symmetry and linearity of the components $h_{i,j}(u_1, v_1, w_1, u_2, v_2, w_2)$ in u_1, v_1, w_1 and u_2, v_2, w_2 three more times (for each triple!) we get

$$\begin{aligned} H(x^{(i)}, x^{(i)}, x^{(i)}, y^{(j)}, y^{(j)}, y^{(j)}) \\ = \sum_{r=0}^{3} \sum_{s=0}^{3} \underbrace{\binom{3}{r}\binom{3}{s} \frac{(x^{(i)} - \pi_{k-1})^r (\pi_k - x^{(i)})^{3-r}}{(\pi_k - \pi_{k-1})^3} \frac{(y^{(j)} - \lambda_{l-1})^s (\lambda_l - y^{(j)})^{3-s}}{(\lambda_l - \lambda_{l-1})^3}}_{(*)} H_{r,s} \end{aligned} \tag{30}$$

where $H_{r,s}$ denotes the matrix obtained by choosing the r–th element of the first and the s–th element of the second set in (26), $r, s = 0, 1, 2, 3$. All these matrices are non–negative definite, and also the coefficients $(*)$ are non–negative. Hence, the Hessian matrix is non–negative definite. □

The inequalities of the set $\mathcal{J}^{(i,j)}$ are linear sufficient convexity conditions for the bicubic patch $f^{(i,j)}(x, y)$. The finite sequences Ψ, Φ, Π, Λ control how necessary these conditions are. Refining these sequences again weakens these conditions:

Proposition 6. *Assume that the finite sequences $\bar{\Psi}, \bar{\Phi}, \bar{\Pi}, \bar{\Lambda}$ result by inserting additional knots into the finite sequences Ψ, Φ, Π, Λ satisfying (9) and (22), i.e., the relations $\{\Psi\} \subseteq \{\bar{\Psi}\}$, $\{\Phi\} \subseteq \{\bar{\Phi}\}$, $\{\Pi\} \subseteq \{\bar{\Pi}\}$ and $\{\Lambda\} \subseteq \{\bar{\Lambda}\}$ hold. Then the inequalities $\mathcal{J}^{(i,j)}(\bar{\Psi}, \bar{\Phi}, \bar{\Pi}, \bar{\Lambda})$ imply the inequalities $\mathcal{J}^{(i,j)}(\Psi, \Phi, \Pi, \Lambda)$*

Similar to Lemma 2, the proof of this observation is a direct consequence of the blossoming principle.

The conditions $\mathcal{J}^{(i,j)}$ can be adapted to any strongly convex bicubic patch:

Theorem 7. *Consider an arbitrary strongly convex surface patch, i.e., the 2×2 matrix of the second partial derivatives is positive definite for all points. Then four finite sequences Ψ, Φ, Π, Λ satisfying (9) and (22) exist such that the control points of the patch fulfill the linear inequalities of the system $\mathcal{J}^{(i,j)}$.*

Proof. Consider the finite sequences $\Pi^{(l)} = \Lambda^{(l)} = \{\frac{k}{2^l} \mid k = 0, \ldots, 2^l\}$ for $l = 1, 2, \ldots$ We consider the matrices $H(u_1, v_1, w_1, u_2, v_2, w_2)$ which are obtained for $(u_1, v_1, w_1) \in \Pi^{(l),*}$ and $(u_2, v_2, w_2) \in \Lambda^{(l),*}$, see (28). We assume the above assertion is not satisfied for all $l \in \mathbb{N}$. Then, according to Lemma 3, for each l at least one of these matrices is not positive definite. Otherwise we could find finite sequences Ψ, Φ for each matrix $H(..)$, such that the inequalities $\mathcal{I}(H(..), \Psi, \Phi)$ are fulfilled. Taking the union of all these finite sequences would then lead to appropriate global sequences Ψ, Φ for the set $\mathcal{J}^{(i,j)}$.

For each l we consider the set

$$\begin{aligned} \mathcal{R}^{(l)} = \{(u_1, v_1, w_1, u_2, v_2, w_2) \mid (u_1, v_1, w_1) \in \Pi^{(l),*} \text{ and } (u_2, v_2, w_2) \in \Lambda^{(l),*} \\ \text{and } H(u_1, v_1, w_1, u_2, v_2, w_2) \text{ is not positive definite } \}. \end{aligned} \tag{31}$$

According to our assumption, this set is non–empty for all l, and it is moreover contained in the six–dimensional unit cube $[0, 1]^6$. We can therefore choose a convergent sequence $\big((u_1(r), ..., w_2(r))\big)_{r=1,2,..}$ with $(u_1(r), ..., w_2(r)) \in \mathcal{R}^{(l(r))}$ and strictly increasing refinement levels $l(r)$: $l(r) < l(r + 1)$. Let

$$(\bar{u}_1, \bar{v}_1, \bar{w}_1, \bar{u}_2, \bar{v}_2, \bar{w}_2) = \lim_{r\to\infty} (u_1(r), ..., w_2(r)) \tag{32}$$

be the limit of this sequence. Note that $\bar{u}_1 = \bar{v}_1 = \bar{w}_1 = \bar{x} \in [0, 1]$ and $\bar{u}_2 = \bar{v}_2 = \bar{w}_2 = \bar{y} \in [0, 1]$ hold, because the refinement level $l(r)$ is strictly increasing for $r \to \infty$ (cf. (28) and the definition of $\Pi^{(l)}$ and $\Lambda^{(l)}$). As all matrices $H(u_1(r), ..., w_2(r))$ are assumed to be not positive definite, the limit matrix $H(\bar{x}, \bar{x}, \bar{x}, \bar{y}, \bar{y}, \bar{y})$ is not positive definite too. On the other hand, the surface patch $f^{(i,j)}(x, y)$ was assumed to be strongly convex, therefore the matrix $H(\bar{x}, \bar{x}, \bar{x}, \bar{y}, \bar{y}, \bar{y})$ is positive definite, as it is the Hessian matrix of the surface patch at $(\bar{x}, \bar{y})$. This is a contradiction. □

Resulting from this theorem, the sufficient convexity conditions $\mathcal{J}^{(i,j)}$ can be made as weak as desired, they are again *asymptotically necessary*:

Corollary 8. *For any finite set of strongly convex surface patches $f^{(i,j)}$ four finite sequences Ψ, Φ, Π, Λ satisfying (9) and* (22) *exist, such that the resulting inequalities $\mathcal{J}^{(i,j)}(\Psi, \Phi, \Pi, \Lambda)$ are fulfilled for all patches. For instance, choosing $\Psi = \Psi^{(L)} = \Phi = \Phi^{(L)}$ (see the proof of Lemma 3) and $\Pi = \Pi^{(K)} = \Lambda = \Lambda^{(K)}$ (see the proof of Lemma 7) with some (big enough) refinement levels K, L yields such a set of inequalities.*

The proof is analogous to that of Corollary 4. In this section we used a simple uniform refinement strategy for the finite sequences Ψ, Φ, Π, Λ in order to prove Lemma 3 and Theorem 7. However, for applications as described in the next section it is more appropriate to use an adaptive refinement strategy, because otherwise the number of inequalities will be too large. Such a strategy will be presented in Section 4.

Of course we can also apply the conditions $\mathcal{J}^{(i,j)}$ to concave surface patches $f^{(i,j)}(x, y)$: we have simply to consider the functions $-f^{(i,j)}(x, y)$ instead.

2.3 Comparison of different linear convexity conditions

In this section we compare our linear convexity conditions with those derived by Carnicer et al. [2]. It is sufficient to examine the conditions which guarantee that the Hessian matrix $H = (h_{i,j})_{i,j=1,2}$ is non–negative definite. Carnicer et al. propose to use the sufficient conditions

$$\begin{aligned}\mathcal{C}_K = \{K\,h_{1,1} \geq |h_{1,2}|,\ K\,h_{2,2} \geq |h_{1,2}|,\ h_{1,1} + K\,h_{2,2} \geq (K+1)|h_{1,2}|, \\ K\,h_{1,1} + h_{2,2} \geq (K+1)|h_{1,2}|\}\end{aligned} \tag{33}$$

for some constant $K \geq 1$, and in particular the choice $K = 2$ is suggested (without further motivation). These conditions lead to a system of 8 linear inequalities. Choosing $K = 1$ yields conditions which are equivalent to the four linear inequalities which describe diagonal dominance of the Hessian matrix,

$$\mathcal{D} = \{|h_{1,1}| \geq h_{1,2},\ |h_{2,2}| \geq h_{1,2}\}. \tag{34}$$

Various other linear convexity conditions have been developed in the case of polynomials defined over triangles, see [6]. According to [2], these conditions (especially those proposed by Chang and Davis [5] and by Lai, see [2, 6]) are stronger than the conditions $\mathcal{C}_2$.

We compare the sufficient conditions of Carnicer et al. with the inequalities obtained from (12) by choosing equidistant finite sequences Ψ, Φ with stepsize $\frac{1}{n}$:

$$\mathcal{I}_n = \mathcal{I}(H, \Psi_n, \Phi_n) \text{ with } \Psi_n = \Phi_n = (0, \tfrac{1}{n}, \tfrac{2}{n}, \ldots, \tfrac{n-1}{n}, 1), \quad n = 2, \ldots, 20. \tag{35}$$

The set $\mathcal{I}_n$ contains $2n$ linear inequalities. The conditions $\mathcal{I}_2$ are equivalent to diagonal dominance $\mathcal{D}$ of the Hessian matrix. In addition, the criteria $\mathcal{I}_\triangle$ and $\mathcal{C}_\ni$ are equivalent.

In order to compare the conditions we performed the following numerical experiment. Using pseudo–random numbers we generated 2,438 symmetric non–negative definite 2×2 matrices $H = (h_{i,j})_{i,j=1,2}$ with components from

$$\{\ (h_{1,1}, h_{1,2}, h_{2,2}) \in \mathbb{R}^3 \mid \mathrm{h}_{1,1}^2 + \mathrm{h}_{1,2}^2 + \mathrm{h}_{2,2}^2 \leq 1,\ \mathrm{h}_{1,1} \geq 0,\ \mathrm{h}_{2,2} \geq 0\ \}. \tag{36}$$

As non–negative definiteness is invariant with respect to scaling of the matrix by non–negative factors, we chose the randomly generated components from the unit ball. Matrices with negative diagonal elements were excluded because they are never non–negative definite. For the experiment we generated 10,000 matrices with components $(h_{1,1}, h_{1,2}, h_{2,2}) \in [0,1]\times[-1,1]\times[0,1]$ and considered only those 2,438 matrices which were non–negative definite and whose components were contained in the set (36).

We applied the conditions $\mathcal{D}, \mathcal{C}_{1.5}, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4$ and $\mathcal{I}_2, \ldots, \mathcal{I}_{\in\prime}$ to these matrices. The following table shows the number of non–negative definite matrices which have been detected by the various conditions. The second and third lines of the table give the percentage of all (2,438) generated matrices and the number of inequalities, respectively:

	$\mathcal{D}=\mathcal{C}_1=\mathcal{I}_2$	$\mathcal{C}_{1.5}$	$\mathcal{C}_2$	$\mathcal{C}_3$	$\mathcal{C}_4$	$\mathcal{I}_3$	$\mathcal{I}_4$	$\mathcal{I}_5$	$\mathcal{I}_6$	...	$\mathcal{I}_{19}$	$\mathcal{I}_{20}$	all
detected:	1764	2149	2250	2217	2170	2088	2217	2297	2347	...	2427	2428	2438
percentage:	72.5	88.1	92.2	90.9	89.0	85.6	90.9	94.2	96.3	...	99.5	99.6	100
# inequs:	4	8	8	8	8	6	8	10	12	...	38	40	–

The relation between the number of inequalities and the percentage of detected matrices has been plotted in Figure 2. The circles indicate the percentages of detected

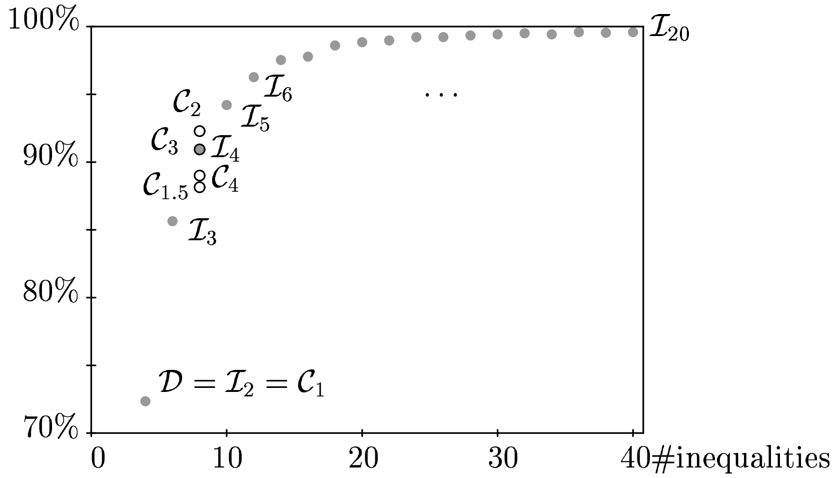


Figure 2: Comparison of convexity conditions.

constraints obtained for $K = 1.5, 2, 3, 4$ by the conditions of Carnicer et al. [2], whereas the grey dots indicate the corresponding percentages for the constraints $\mathcal{I}_2, \mathcal{I}_3, \dots, \mathcal{I}_{20}$. The figure illustrates that the percentage of detected constraints for the conditions $\mathcal{I}_n$ tends to 100% if the number of inequalities increases. It can also be seen, that the eight linear inequalities of $\mathcal{C}_2$ perform slightly better than the eight inequalities of $\mathcal{I}_4$, but worse than $\mathcal{I}_5$. In this sense, these conditions could be said to be slightly "more efficient" than $\mathcal{I}_4$. However, the construction of Carnicer et al. does not offer the asymptotic necessity of the constraints $\mathcal{I}_n$.

3 Computing the approximating surface

Based on the linear convexity conditions of the preceding section we are able to formulate our approximation task as a quadratic programming problem. After generating the linear shape constraints (according to the specified shape of the approximating surface) and choosing an appropriate objective function we construct an initial solution. The coefficients $d_{i,j}$ of the approximating function are then found with the help of appropriate tools from optimization theory.

3.1 Generating the shape constraints

We consider the bicubic polynomial segments $f^{(i,j)}(x,y)$ with $(x,y) \in D^{(i,j)}$ of the approximating bivariate spline function (1), see (5) ($0 \le i \le P-7$, $0 \le j \le Q-7$). If a convex spline segment has been specified, $\sigma^{(i,j)} = 1$, then we generate the linear convexity constraints $\mathcal{J}^{(i,j)}$, see (27). Similarly, if a concave spline segment has been specified, $\sigma^{(i,j)} = -1$, then we generate the corresponding linear concavity constraints $\mathcal{J}^{(i,j)}_{(-)}$. This is easily achieved by applying the construction of the previous section to the function $-f^{(i,j)}(x,y)$.

For generating the inequalities for each patch $f^{(i,j)}(x,y)$ we have to choose the finite sequences Ψ, Φ, Π and Λ. Of course, one may choose different sequences for each patch: $\Psi^{(i,j)}, \Phi^{(i,j)}, \Pi^{(i,j)}, \Lambda^{(i,j)}$. We choose identical initial values for all patches,

$$\begin{aligned} \Psi = \Psi^{(i,j)} = \Phi = \Phi^{(i,j)} &= (0.0, 0.5, 1.0) \\ \text{and} \quad \Pi = \Pi^{(i,j)} = \Lambda = \Lambda^{(i,j)} &= (0.0, 1.0). \end{aligned} \tag{37}$$

After performing the refinement steps which are proposed in the next section, however, we will get generally different sequences, depending on the given data.

We collect all inequality constraints to a set $\mathcal{J}$,

$$\mathcal{J} = \bigcup_{\substack{i=0,\ldots,P-7 \\ j=0,\ldots,Q-7 \\ \sigma^{(i,j)}=1}} \mathcal{J}^{(i,j)} \quad \cup \bigcup_{\substack{i=0,\ldots,P-7 \\ j=0,\ldots,Q-7 \\ \sigma^{(i,j)}=-1}} \mathcal{J}^{(i,j)}_{(-)}. \tag{38}$$

For solving the quadratic programming problems as described below, it is very important to remove redundant constraints from this set as far as possible. Such redundancies may occur

- at segment boundaries and at corners of the patches $f^{(i,j)}$. The inequalities $\mathcal{I}(H(u_1, v_1, w_1, u_2, v_2, w_2), \Psi, \Phi)$ which are obtained for

$$(u_1, v_1, w_1) \in \{(\pi_0, \pi_0, \pi_0), (\pi_T, \pi_T, \pi_T)\} = \{(0,0,0), (1,1,1)\} \tag{39}$$

  and / or

$$(u_2, v_2, w_2) \in \{(\lambda_0, \lambda_0, \lambda_0), (\lambda_U, \lambda_U, \lambda_U)\} = \{(0,0,0), (1,1,1)\} \tag{40}$$

  (cf. (27)) may be contained twice in $\mathcal{J}$, depending on the specified shape of the approximating spline function. For instance, the inequalities obtained for $(u_1, v_1, w_1) = (1,1,1)$ in $\mathcal{J}^{(i,j)}$ and those obtained for $(u_1, v_1, w_1) = (0,0,0)$ in $\mathcal{J}^{(i+1,j)}$ will be identical (if the sequences Ψ, Φ chosen for these patches are identical) or at least dependent. One should therefore remove some of these constraints from $\mathcal{J}$.

- due to built–in continuity of the terms $p(u,v)$ and $q(u,v)$, cf. (10), which are used for generating the constraints. Consider the piecewise bicubic polynomials $p(u,v)$ and $q(u,v)$ which are obtained for $h_{1,1} = f^{(i,j,2,0)}$, $h_{1,2} = h_{2,1} = f^{(i,j,1,1)}$ and $h_{2,2} = f^{(i,j,0,2)}$. Generally, for $0 < u, v < 1$, $p(u,v)$ and $q(u,v)$ are then only guaranteed to be C^0 functions for $(x, y) \in [0,1]^2$. But, for instance, for $u = 0$ we get functions which are guaranteed to be C^1 with respect to y. It is therefore not necessary to create constraints which guarantee the non–negativity of the Bézier coefficients of these functions at all patch boundaries (knot lines) with $y =$ constant, provided that the corresponding inequalities for the inner coefficients of the neighbouring patches are included in the set $\mathcal{J}$. Such redundant constraints may be generated (depending on the specified shape for the approximating spline function) when computing $p(\psi_0, \psi_1)$ and $q(\phi_0, \phi_1)$ for matrices $H(u_1, v_1, w_1, u_2, v_2, w_2)$ with

$$(u_2, v_2, w_2) \in \{(\lambda_0, \lambda_0, \lambda_0), (\lambda_U, \lambda_U, \lambda_U)\} = \{(0,0,0), (1,1,1)\}. \tag{41}$$

Similarly, we may get redundant constraints when computing $p(\psi_{R-1}, \psi_R)$ and $q(\phi_{S-1}, \phi_S)$ for matrices $H(u_1, v_1, w_1, u_2, v_2, w_2)$ with

$$(u_1, v_1, w_1) \in \{(\pi_0, \pi_0, \pi_0), (\pi_T, \pi_T, \pi_T)\} = \{(0,0,0), (1,1,1)\}. \tag{42}$$

Of course, these redundant constraints should be eliminated from $\mathcal{J}$.

The set of inequalities which is obtained after removing the above–mentioned redundant inequalities from $\mathcal{J}$ is denoted by $\mathcal{J}^*$.

3.2 The objective function

The objective function of our problem is obtained by combining the least–squares sum for the given data (cf. (1)) with a weighted "tension–term" (also called smoothing term),

$$\begin{aligned} \mathcal{F} &= \mathcal{F}((d_{i,j})_{i=0..P-4, j=0..Q-4}) \\ &= \sum_{k=0}^{R} \left(\sum_{i=0}^{P-4} \sum_{j=0}^{Q-4} M_i(x_k)\, N_j(y_k)\, d_{i,j} - z_k \right)^2 \\ &\quad + w \left[\sum_{i=1}^{P-4} \sum_{j=0}^{Q-4} \left(\frac{d_{i,j} - d_{i-1,j}}{\bar{x}_i - \bar{x}_{i-1}} \right)^2 + \sum_{i=0}^{P-4} \sum_{j=1}^{Q-4} \left(\frac{d_{i,j} - d_{i,j-1}}{\bar{y}_j - \bar{y}_{j-1}} \right)^2 \right] \end{aligned} \tag{43}$$

with the weight $w > 0$. The objective function depends quadratically on the unknown coefficients $d_{i,j}$. The tension term [...] is introduced in order to guarantee that the minimization of (43) leads to a full rank system of linear equations for the coefficients $d_{i,j}$. The tension term represents the sum of the squared lengths of the legs of the B–spline control net; the constants $\bar{x}_i$ and $\bar{y}_j$ are the Greville–abscissae

$$\bar{x}_i = \frac{1}{3}(\xi_{i+1} + \xi_{i+2} + \xi_{i+3}) \text{ and } \bar{y}_j = \frac{1}{3}(\eta_{j+1} + \eta_{j+2} + \eta_{j+3}), \tag{44}$$

see [9]. The weight w will be chosen such that the influence of the tension term is rather small, compared with the least–squares sum, see below. Resulting from this, the particular choice of the tension term is not so important. Instead of [...] one may also use certain integrals involving squared derivatives or not-a-knot-type conditions. See [8, Chapter 9] for a detailed discussion of rank deficiencies and the use of tension (smoothing) terms.

3.3 Initial solution

The approximating tensor–product spline function is to be found by minimizing the objective function (43) subject to the linear shape constraints of the set $\mathcal{J}^*$ of inequalities. This quadratic programming problem is to be solved with the help of an active set strategy which requires an initial solution. We construct the initial solution in the following way: Solving the unconstrained problem (i.e., minimizing the objective function $\mathcal{F}$ without constraints whereby the weight w is set to 0.01) leads to coefficients $(d^*_{i,j})_{i=0..P-4,\,j=0..Q-4}$ which generally do not fulfill the inequalities $\mathcal{J}^*$. With the help of the substitution

$$d_{i,j} = d^*_{i,j} + d^{(+)}_{i,j} - d^{(-)}_{i,j} \qquad (i = 0..P-4,\ j = 0..Q-4) \tag{45}$$

we introduce new variables $d^{(+)}_{i,j}$ and $d^{(-)}_{i,j}$. The inequalities $\mathcal{J}^*$ are transformed into inequalities for these variables. The initial solution is found by minimizing the linear objective function

$$\sum_{i=0}^{P-4} \sum_{j=0}^{Q-4} d^{(+)}_{i,j} + d^{(-)}_{i,j} \tag{46}$$

subject to the linear constraints

$$\mathcal{J}^* \ \cup\ \{d^{(+)}_{i,j} \geq 0, d^{(-)}_{i,j} \geq 0 \mid i = 0, \ldots, P-4;\ j = 0, \ldots, Q-4\} \tag{47}$$

This linear programming problem can be solved with the help of the simplex algorithm. In our implementation we use a public–domain optimization code provided by Berkelaar [1].

The existence of solutions is automatically guaranteed, as at least constant functions ($d_{i,j} = d$ for all i, j) satisfy all constraints $\mathcal{J}^*$. The choice of a constant function as initial solution, however, would cause problems for the quadratic programming described below, as the optimization would start in a highly degenerate situation (all constraints would be active).

Due to the substitution (45), the objective function (46) measures the l^1 distance (of the coefficients $d_{i,j}$) to the solution of the unconstrained problem. Thus, minimizing this function can be expected to lead to an initial solution which is relatively close to the solution of the quadratic programming problem.

3.4 Quadratic programming

The initial solution is used in order to adapt the weight w of the tension term [...] in (43). We choose the weight such that the value of the weighted tension term is equal to $\frac{1}{100}$ times the value of the least–squares sum for the initial solution. After adjusting the value of w we compute the minimum of the quadratic objective function (43) subject to the linear inequality constraints $\mathcal{J}^*$, i.e., by quadratic programming. Due to the regularity of the objective function (which is guaranteed by adding the tension term [...]), a unique solution for this constrained optimization problem exists.

Our implementation is based on the active set strategy as described in the textbook by Fletcher [10], and it uses pseudo–random numbers to avoid cycling at degenerate situations. The computing times of our optimization code are not completely satisfactory yet; perhaps a commercial package might lead to faster results.

Alternatively one may try to solve the quadratic programming problem with the help of the LOQO package by Vanderbei [17] (an efficient implementation of an interior-point method for large-scale linear or quadratic programming problems). In our examples the active set strategy gave better results.

Solving the quadratic programming problem leads to the coefficients $d_{i,j}$ of the approximating tensor–product spline functions. We will illustrate the method by some examples in Section 5.

4 Adapting the constraints

After computing the solution of the first quadratic programming problem it is possible to adapt the constraints by choosing more appropriate sequences $\Psi^{(i,j)}, \Phi^{(i,j)}, \Pi^{(i,j)}$ and $\Lambda^{(i,j)}$. This will lead to linear shape constraints which are better suited for the approximating surface to the given data. We present two refinement strategies:

A: Adapting the subdivision of the directions. For each surface segment $f^{(i,j)}(x,y)$ we consider the system (12) of inequalities which is obtained for $H = H(u_1, v_1, w_1, u_2, v_2, w_2)$ and $\Psi = \Psi^{(i,j)}$, $\Phi = \Phi^{(i,j)}$ with $(u_1, v_1, w_1) \in \Pi^{(i,j)*}$ and $(u_2, v_2, w_2) \in \Lambda^{(i,j)*}$, see (28). Collecting these inequalities would yield the set $\mathcal{J}^{(i,j)}$, cf. (27). If one of the inequalities $p(\psi_{k-1}^{(i,j)}, \psi_k^{(i,j)}) \geq 0$ is active (i.e., $p(..) = 0$ holds), then we add the new knot $\frac{1}{2}(\psi_{k-1}^{(i,j)} + \psi_k^{(i,j)})$ to the finite sequence $\Psi^{(i,j)}$ $(k = 1...R^{(i,j)})$. Analogously, if one of the inequalities $q(\phi_{k-1}^{(i,j)}, \phi_k^{(i,j)}) \geq 0$ is active, then we add the new knot $\frac{1}{2}(\phi_{k-1}^{(i,j)} + \phi_k^{(i,j)})$ to the finite sequence $\Phi^{(i,j)}$ $(k = 1...S^{(i,j)})$.

B: Adapting the subdivision of the patches. We consider the inequalities (27) which result for $\Pi = \Pi^{(i,j)}$ and $\Lambda = \Lambda^{(i,j)}$. If at least one of the inequalities $\mathcal{I}(\mathcal{H}(..), \ominus^{(),|)}, \oplus^{(),|)})$ obtained for

$$(u_1, v_1, w_1) \in \{(\pi_{k-1}^{(i,j)}, \pi_{k-1}^{(i,j)}, \pi_k^{(i,j)}), (\pi_{k-1}^{(i,j)}, \pi_k^{(i,j)}, \pi_k^{(i,j)})\} \text{ and } (u_2, v_2, w_2) \in \Lambda^{(i,j)*} \quad (48)$$

is active, then we add the new knot $\frac{1}{2}(\pi_{k-1}^{(i,j)} + \pi_k^{(i,j)})$ to the finite sequence $\Pi^{(i,j)}$ $(k = 1...T^{(i,j)})$. Analogously, if one of the inequalities $\mathcal{I}(\mathcal{H}(..), \ominus^{(),|)}, \oplus^{(),|)})$ obtained for

$$(u_2, v_2, w_2) \in \{(\lambda_{k-1}^{(i,j)}, \lambda_{k-1}^{(i,j)}, \lambda_k^{(i,j)}), (\lambda_{k-1}^{(i,j)}, \lambda_k^{(i,j)}, \lambda_k^{(i,j)})\} \text{ and } (u_1, v_1, w_1) \in \Pi^{(i,j)*} \quad (49)$$

is active, then we add the new knot $\frac{1}{2}(\lambda_{k-1}^{(i,j)} + \lambda_k^{(i,j)})$ to the finite sequence $\Lambda^{(i,j)}$ $(k = 1...U^{(i,j)})$. As neither $\Pi^{(i,j)}$ nor $\Lambda^{(i,j)}$ is to be preferred, the refinement for $\Lambda^{(i,j)}$ is computed with the original sequence $\Pi^{(i,j)}$. If both refinement criteria (48) and (49) are fulfilled, then both sequences $\Pi^{(i,j)}$ and $\Lambda^{(i,j)}$ are refined.

Both refinement strategies leads to sets of inequalities which are weaker than the original ones. After the first approximation (with uniform finite sequences) we apply one of both refinement strategies. Based on the new finite sequences we get weaker convexity conditions which lead to a better approximating spline function (1). The first solution may serve as the initial solution for the second quadratic programming. The obtained solution can be used once more in order to adapt the finite sequences. Iterating this procedure a few times yields the final result.

In our examples, the first refinement strategy often led to a bigger improvement of the objective function than the second one. Thus, one should use the first strategy more often than the second one. For instance, one may use the first strategy a few times and then apply the second strategy once.

Whereas the number of variables for the approximation problem remains unchanged, adapting the finite sequences increases the number of inequalities. Of course, refining the knot vectors Ξ and Θ of the approximating spline function simultaneously would be possible.

5 Examples

For the first example we sampled randomly 121 data points with $(x, y) \in [0, 1]^2$ from the function $x^3 + 5(y - 0.6)^2 + 1$. The z–values of the data have been perturbed with the help of pseudo–random numbers. The same function has been used as an example in [18]. These data have been approximated by a bicubic spline function with 6×8 patches (99 coefficients). The unconstrained and the convex approximation are compared in Figure 3, where the obtained approximating functions have been plotted. The plots show the level curves of the approximating spline functions (thin black lines), the given data and the error vectors (short black lines) and the control nets (thick grey lines).

The unconstrained approximation (a) possesses a lot of oscillations and non–convex regions, in particular along the boundaries. This can also be seen from the determinant of the second partial derivatives, Figure 3c: negative values of the determinant correspond to non–convex regions of the approximating spline function. (Only regions with positive values of the determinant have been drawn.)

The convex approximating surface (b) has been obtained after 3 iterations of our method. All patches of the spline function are restricted to be convex. At first we solved an quadratic programming (qp) problem with 1385 inequalities and 99 unknowns. 32 inequalities were active for the solution. Applying the refinement step A twice led to qp problems with 1853 and 2436 inequalities with 41 and 54 active

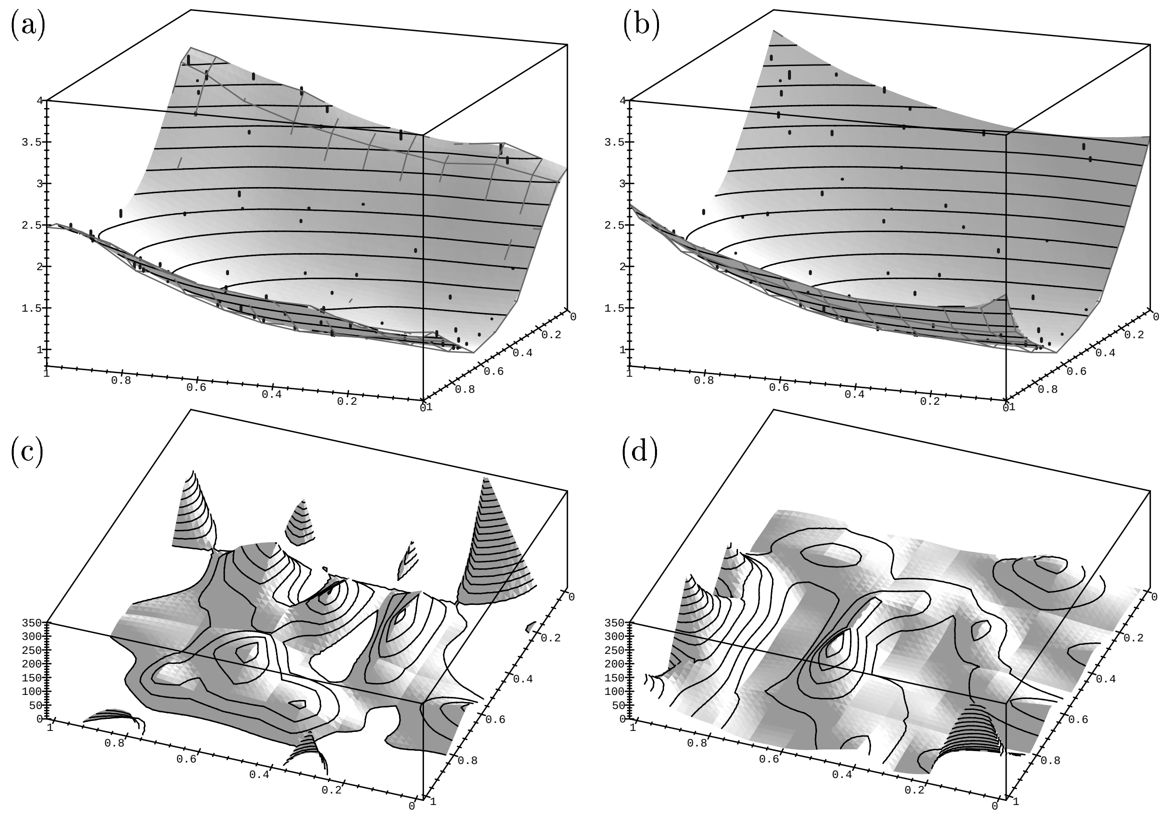
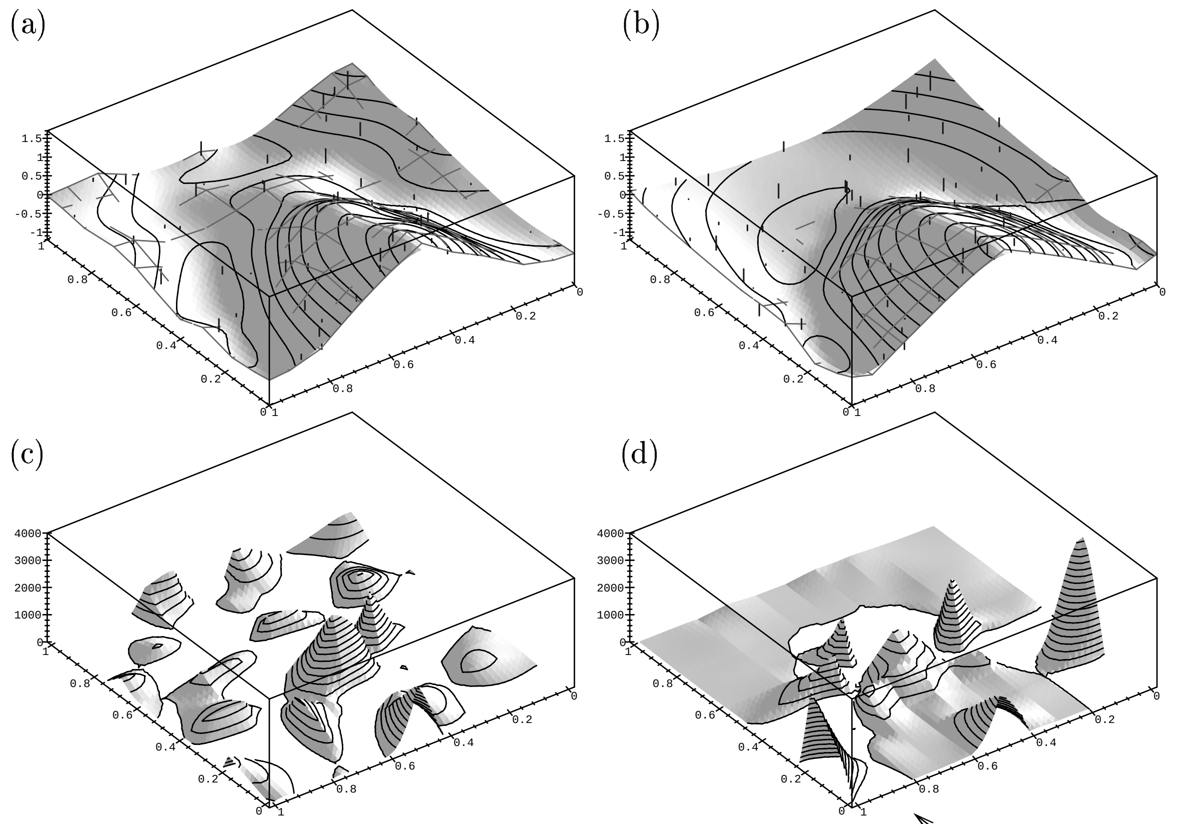


Figure 3: Example 1. Unconstrained approximation (a), convex approximation (b) and the determinants $f_{xx}f_{yy} - f_{xy}^2$ of the Hessian matrices (c,d).

inequalities for the solutions. The final value of the least–squares sum (0.221 after the first quadratic programming) was 0.095. The determinant of the Hessian matrix (d) is now nonnegative on $[0,1]^2$.

For the second example we sampled 121 data points with $(x,y) \in [0,1]^2$ from the function $\cos(7\sqrt{\frac{1}{4}x^2 + (y-\frac{1}{2})^2})$. Again, the z–values of the data have been perturbed by pseudo–random numbers. These data are approximated by a bicubic spline function with 8×8 patches (121 coefficients). Analogous to the first example, the unconstrained and the convex approximation are compared in Figure 4. For computing the constrained approximation we specified the desired curvature signs of the spline segments. These signs are indicated in Figure 4e. For the constrained approximation, the determinant of the second partial derivatives has non–negative values in regions which are specified to be either convex or concave. In contrast to this, the corresponding plot (c) for the unconstrained approximation indicates giant hyperbolic regions.



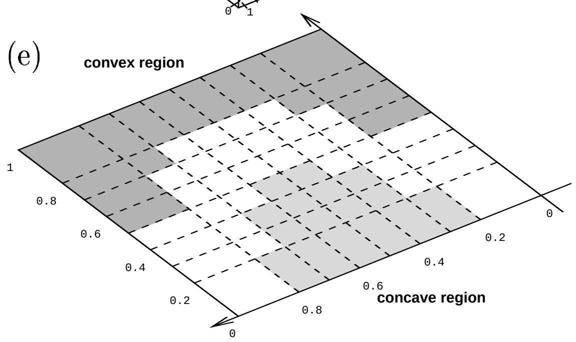


Figure 4: Example 2. The unconstrained approximation (a), the constrained approximation (b) and the determinants $f_{xx}f_{yy} - f_{xy}^2$ of the Hessian matrices (c,d). The knot lines and the specified curvature signs are shown in (e).

Concluding remarks

In this paper we presented a new construction for linear sufficient convexity conditions for polynomial tensor–product spline functions. The obtained constraints have been used in order to formulate convex least–square approximation as a quadratic programming problem. Our present implementation of the method is not completely satisfying yet: adapting the constraints sometimes leads to high numbers of inequalities. Each of these inequalities involves only relatively few coefficients $d_{i,j}$ (16 at most), so it would be advantageous to exploit this sparsity, both for the storage of the constraints and for the quadratic programming.

We prescribe only regions with convex or concave shape, but no constraints for the remaining spline segments are imposed yet. The resulting unconstrained spline segments often possess a wavy shape (see the second example). By generalizing the linear shape constraints one could try to specify directions with positive and negative second directional derivatives for each spline segment. Of course, one would need to estimate these directions a priori from the given data.

As pointed out by the referee, one could combine the methods of this paper with an strategy for adaptively adding knots ξ_i, θ_j until the sum of the squared residuals is less than an upper bound (cf. [7] for the curve case). This nice idea would automatically produce appropriate knots for the splines. In addition to the refinement strategies of Section 4 one would have to apply refinement strategies for the spline knots. Moreover, in many cases this idea would also lead to a suitable initial solution: one might use an unconstrained least–squares approximation over sparser knot vectors. On the other hand, the use this idea would result in a higher number of quadratic programming problems which need to be solved. In our implementation we therefore keep the numbers of knots (which are chosen with the help of some heuristics) always fixed.

Appendix A: Polynomial segments of bicubic tensor–product splines

The coefficients of polynomial segments (5) of the spline function (1) result from

$$b_{r,s}^{(i,j)} = \sum_{k=0}^{3} \sum_{l=0}^{3} \beta_{r,k}^{(i)} \, \gamma_{s,l}^{(j)} \, d_{i+k,j+l} \tag{50}$$

with the coefficients $\beta_{r,k}^{(i)}$,

$$\begin{gathered} \beta_{0,0}^{(i)} = \frac{\xi_{i+4}-\xi_{i+3}}{\xi_{i+4}-\xi_{i+2}} \, \frac{\xi_{i+4}-\xi_{i+3}}{\xi_{i+4}-\xi_{i+1}}, \qquad \beta_{0,2}^{(i)} = \frac{\xi_{i+3}-\xi_{i+2}}{\xi_{i+4}-\xi_{i+2}} \, \frac{\xi_{i+3}-\xi_{i+2}}{\xi_{i+5}-\xi_{i+2}}, \\ \beta_{0,1}^{(i)} = \frac{\xi_{i+4}-\xi_{i+3}}{\xi_{i+4}-\xi_{i+2}} \, \frac{\xi_{i+3}-\xi_{i+1}}{\xi_{i+4}-\xi_{i+1}} + \frac{\xi_{i+3}-\xi_{i+2}}{\xi_{i+4}-\xi_{i+2}} \, \frac{\xi_{i+5}-\xi_{i+3}}{\xi_{i+5}-\xi_{i+2}}, \end{gathered} \tag{51}$$

$$\begin{aligned} \beta_{1,1}^{(i)} &= \frac{\xi_{i+5}-\xi_{i+3}}{\xi_{i+5}-\xi_{i+2}}, & \beta_{1,2}^{(i)} &= \frac{\xi_{i+3}-\xi_{i+2}}{\xi_{i+5}-\xi_{i+2}}, \\ \beta_{2,1}^{(i)} &= \frac{\xi_{i+5}-\xi_{i+4}}{\xi_{i+5}-\xi_{i+2}}, & \beta_{2,2}^{(i)} &= \frac{\xi_{i+4}-\xi_{i+2}}{\xi_{i+5}-\xi_{i+2}}, \end{aligned} \tag{52}$$

$$\begin{gathered} \beta_{3,2}^{(i)} = \frac{\xi_{i+5}-\xi_{i+4}}{\xi_{i+5}-\xi_{i+3}} \, \frac{\xi_{i+4}-\xi_{i+2}}{\xi_{i+5}-\xi_{i+2}} + \frac{\xi_{i+4}-\xi_{i+3}}{\xi_{i+5}-\xi_{i+3}} \, \frac{\xi_{i+6}-\xi_{i+4}}{\xi_{i+6}-\xi_{i+3}}, \\ \beta_{3,1}^{(i)} = \frac{\xi_{i+5}-\xi_{i+4}}{\xi_{i+5}-\xi_{i+3}} \, \frac{\xi_{i+5}-\xi_{i+4}}{\xi_{i+5}-\xi_{i+2}}, \qquad \beta_{3,3}^{(i)} = \frac{\xi_{i+4}-\xi_{i+3}}{\xi_{i+5}-\xi_{i+3}} \, \frac{\xi_{i+4}-\xi_{i+3}}{\xi_{i+6}-\xi_{i+3}}, \end{gathered} \tag{53}$$

and

$$\beta_{0,3}^{(i)} = \beta_{1,0}^{(i)} = \beta_{1,3}^{(i)} = \beta_{2,0}^{(i)} = \beta_{2,3}^{(i)} = \beta_{3,0}^{(i)} = 0. \tag{54}$$

The formulas for remaining coefficients $\gamma_{s,l}^{(j)}$ are obtained from (51)–(54) after replacing ξ by θ, i by j and β by γ.

Appendix B: Second partial derivatives of bicubic patches

The coefficients $b_{r,s}^{(i,j,k,2-k)}$ $(k = 0, 1, 2)$ of the second partial derivatives (8) are obtained from

$$\begin{aligned}
\left(b_{r,s}^{(i,j,2,0)}\right)_{r=0..3,s=0..3} &= \frac{1}{(\xi_{i+4}-\xi_{i+3})^2} D_1^\top \cdot \left(\Delta_{[1]}^2 b_{r,s}^{(i,j)}\right)_{r=0..1,s=0..3}, \\
\left(b_{r,s}^{(i,j,1,1)}\right)_{r=0..3,s=0..3} &= \frac{1}{(\xi_{i+4}-\xi_{i+3})(\theta_{j+4}-\theta_{j+3})} D_2^\top \cdot \left(\Delta_{[1]}\Delta_{[2]} b_{r,s}^{(i,j)}\right)_{r=0..2,s=0..2} \cdot D_2, \\
\left(b_{r,s}^{(i,j,0,2)}\right)_{r=0..3,s=0..3} &= \frac{1}{(\theta_{j+4}-\theta_{j+3})^2} \left(\Delta_{[2]}^2 b_{r,s}^{(i,j)}\right)_{r=0..3,s=0..1} \cdot D_1.
\end{aligned} \tag{55}$$

For computing these control points we need the degree elevation matrices

$$D_1 = \begin{pmatrix} 1 & \frac{2}{3} & \frac{1}{3} & 0 \\ 0 & \frac{1}{3} & \frac{2}{3} & 1 \end{pmatrix} \text{ and } D_2 = \begin{pmatrix} 1 & \frac{1}{3} & 0 & 0 \\ 0 & \frac{2}{3} & \frac{2}{3} & 0 \\ 0 & 0 & \frac{1}{3} & 1 \end{pmatrix} \tag{56}$$

and the difference operators $\Delta_{[1]}x_{r,s} = x_{r+1,s} - x_{r,s}$ and $\Delta_{[2]}x_{r,s} = x_{r,s+1} - x_{r,s}$. With the help of the formulas from Appendix A we can express these control points as linear combinations of the spline coefficients $d_{i,j}$.

References

[1] M. Berkelaar, `lp_solve`, available via anonymous ftp from `ftp.es.ele.tue.nl` at `/pub/lp_solve`.

[2] J. M. Carnicer, M. S. Floater and J. M. Peña, Linear convexity conditions for triangular and rectangular Bernstein–Bézier surfaces, submitted to *Computer Aided Geometric Design*, 1996.

[3] J. M. Carnicer and W. Dahmen, Convexity preserving interpolation and Powell–Sabin elements, *Computer Aided Geometric Design* **9** (1992), 279–289.

[4] G. Chang and Y. Feng, An improved condition for the convexity of Bernstein–Bézier polynomials over triangles, *Computer Aided Geometric Design* **1** (1984), 279–283.

[5] G. Chang and P. J. Davis, The convexity of Bernstein polynomials over triangles, *Journal of Approximation Theory* **40** (1984), 11–28.

[6] W. Dahmen, Convexity and Bernstein–Bézier polynomials, in: P. J. Laurent, A. Le Méhauté and L. L. Schumaker, Eds., *Curves and Surfaces* (Academic Press, Boston, 1991), pp. 107–134.

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

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

[9] G. Farin, *Curves and surfaces for computer aided geometric design: a practical guide* (Academic Press, Boston, 1990, 2nd ed.).

[10] R. Fletcher, *Practical methods of optimization* (Wiley–Interscience, Chichester, 1990).

[11] M. Floater, A weak condition for the convexity of tensor–product Bézier and B–spline surfaces., *Advances in Computational Mathematics* **2** (1994), 67-80.

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

[13] B. Mulansky, J. W. Schmidt and M. Walther, Tensor product spline interpolation subject to piecewise bilinear lower and upper bounds, in: J. Hoschek and P. Kaklis, Eds., *Advanced Course on FAIRSHAPE* (Teubner, Stuttgart, 1996), 201–216.

[14] H. J. Schelske, *Lokale Glättung segmentierter Bézierkurven und Bézierflächen* (Dissertation, TH Darmstadt, 1984).

[15] J. W. Schmidt, Results and problems in shape preserving interpolation and approximation with polynomial splines, in: J. W. Schmidt and H. Späth, Eds., *Splines in Numerical Analysis* (Mathematical Research Vol. 52, Akademie–Verlag, Berlin, 1989), 159–170.

[16] J. W. Schmidt and W. Heß, Positivity of cubic polynomials on intervals and positive spline interpolation, *BIT* **28** (1988), 340–352.

[17] R. Vanderbei, `LOQO`, available via anonymous ftp from `elib.zib-berlin.de` at `/pub/opt-net/software/loqo/1.08`.

[18] K. Willemans and P. Dierckx, Surface fitting using convex Powell–Sabin splines, *Journal of Computational and Applied Mathematics* **56** (1994), 263–282.

Author's address

Dr. B. Jüttler
Technische Universität Darmstadt
Fachbereich Mathematik, AG 3
Schloßgartenstraße 7
64289 Darmstadt, Germany
Phone: +49 6151 16 2789, Fax: +49 6151 16 2131
Electronic mail: juettler@mathematik.th-darmstadt.de