

Convexity preserving splines over triangulations

Larry L. Schumaker *Hendrik Speleers*

Report TW 564, April 2010



Katholieke Universiteit Leuven
Department of Computer Science
Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

Convexity preserving splines over triangulations

Larry L. Schumaker *Hendrik Speleers*

Report TW 564, April 2010

Department of Computer Science, K.U.Leuven

Abstract

A general method is given for constructing sets of sufficient linear conditions that ensure convexity of a polynomial in Bernstein–Bézier form on a triangle. Using the linear conditions, computational methods based on macro-element spline spaces are developed to construct convexity preserving splines over triangulations that interpolate or approximate given scattered data.

Keywords : spline interpolation, shape preservation, convex surfaces
MSC : Primary : 65D07, Secondary : 41A15, 41A29

Convexity preserving splines over triangulations

Larry L. Schumaker¹⁾ and Hendrik Speleers²⁾

Abstract. A general method is given for constructing sets of sufficient linear conditions that ensure convexity of a polynomial in Bernstein–Bézier form on a triangle. Using the linear conditions, computational methods based on macro-element spline spaces are developed to construct convexity preserving splines over triangulations that interpolate or approximate given scattered data.

Keywords: spline interpolation, shape preservation, convex surfaces

§1. Introduction

The problem of constructing convex surfaces interpolating or approximating given scattered data has been investigated in a string of papers. Convexity preserving methods have been developed for a wide range of function classes, e.g., macro-element splines defined on triangulations [1,4,7,17,30,32,34,39] and quadrangulations [27,33], tensor-product splines [16,23], and certain other smooth functions [3,29].

Polynomial spline functions are often represented in Bernstein–Bézier form. Therefore, an important subproblem is to find sets of (sufficient) convexity conditions for polynomials in Bernstein–Bézier representation. Particular sets of linear convexity conditions have been derived in [6,8,26,31]; sets of nonlinear conditions have been described in [9,38], and methods based on convexity of the control net and subdivision can be found in [15,18,20,21].

The purpose of this paper is twofold. Our first aim is to describe a general method for constructing sets of sufficient linear conditions that ensure convexity of a polynomial written in Bernstein–Bézier form on a triangle. The method, which is based on blossoming, can be used to create nested sequences of weaker and weaker conditions, including all of the known sets of linear conditions. Our second aim is to develop explicit convexity preserving interpolation and approximation methods using the linear convexity conditions together with certain macro-element spline spaces defined on triangulations.

The paper is organized as follows. In Sect. 2 we discuss conditions on the B-coefficients of a polynomial to ensure its convexity. This section contains our main

¹⁾ Department of Mathematics, Vanderbilt University, Nashville, TN 37240, larry.schumaker@vanderbilt.edu.

²⁾ Department of Computer Science, Katholieke Universiteit Leuven, B-3001 Leuven, Belgium, hendrik.speleers@cs.kuleuven.be

theoretical results. Sect. 3 is devoted to convexity of splines defined on triangulations and the description of macro-element based methods involving minimizing an appropriate quadratic functional subject to side conditions. Numerical examples are presented in Sect. 4. We conclude the paper with remarks and references.

§2. Convexity of polynomials in Bernstein–Bézier form

As is well known (cf. Sect. 2.3 of [28]), any polynomial p of degree d defined on a triangle $T := \langle v_1, v_2, v_3 \rangle$ can be written in Bernstein–Bézier (B-) form as

$$p(x, y) = \sum_{i+j+k=d} c_{ijk} B_{ijk}^d(x, y), \quad (2.1)$$

where

$$B_{ijk}^d(x, y) = \frac{d!}{i!j!k!} \beta_1^i \beta_2^j \beta_3^k$$

are the Bernstein polynomials of degree d , and $\beta := (\beta_1, \beta_2, \beta_3)$ are the barycentric coordinates of (x, y) relative to the triangle T . Let u be any direction vector in \mathbb{R}^2 , say

$$u = \eta_2(v_2 - v_1) + \eta_3(v_3 - v_1).$$

Then (see Sect. 2.6 of [28]),

$$D_u^2 p(x, y) = d(d-1) \sum_{i+j+k=d-2} \left(\sum_{\nu=2}^3 \sum_{\mu=2}^3 \eta_\nu \eta_\mu \Delta_{\nu 1} \Delta_{\mu 1} c_{ijk} \right) B_{ijk}^{d-2}(x, y),$$

where $\Delta_{\nu\mu}$ is the difference operator along direction $v_\nu - v_\mu$. Thus, for example,

$$\begin{aligned} \Delta_{21} c_{ijk} &= c_{i,j+1,k} - c_{i+1,j,k}, \\ \Delta_{21}^2 c_{ijk} &= c_{i,j+2,k} - 2c_{i+1,j+1,k} + c_{i+2,j,k}, \\ \Delta_{21} \Delta_{31} c_{ijk} &= c_{i,j+1,k+1} - c_{i+1,j+1,k} - c_{i+1,j,k+1} + c_{i+2,j,k}. \end{aligned}$$

The polynomial p is convex on T if and only if $D_u^2 p(x, y) \geq 0$ for all $(x, y) \in T$ and for any direction u . Thus, a sufficient condition to ensure convexity is that the matrix

$$A_{ijk} = \begin{bmatrix} \Delta_{21}^2 c_{ijk} & \Delta_{21} \Delta_{31} c_{ijk} \\ \Delta_{21} \Delta_{31} c_{ijk} & \Delta_{31}^2 c_{ijk} \end{bmatrix} \quad (2.2)$$

be nonnegative definite for each $i + j + k = d - 2$, see Theorems 3.14 and 3.15 in [28]. This condition is equivalent to

$$\Delta_{21}^2 c_{ijk} \geq 0, \quad \Delta_{31}^2 c_{ijk} \geq 0, \quad (\Delta_{21}^2 c_{ijk})(\Delta_{31}^2 c_{ijk}) - (\Delta_{21} \Delta_{31} c_{ijk})^2 \geq 0, \quad (2.3)$$

for $i + j + k = d - 2$. This condition is also a necessary condition for quadratic and cubic polynomials.

2.1. Linear conditions for convexity

The conditions in (2.3) for a polynomial p to be convex are nonlinear. In this subsection we derive some alternative sufficient conditions that are linear. First we need a technical lemma concerning the matrices A_{ijk} appearing in (2.2). Fix $i + j + k = d - 2$, and let

$$q_m(t) := a_m t^2 + 2b_m t(1-t) + c_m(1-t)^2, \quad m = 1, 2, 3, \quad (2.4)$$

with

$$a_1 := \Delta_{21}^2 c_{ijk}, \quad b_1 := \Delta_{21} \Delta_{31} c_{ijk}, \quad c_1 := \Delta_{31}^2 c_{ijk}, \quad (2.5)$$

$$a_2 := \Delta_{32}^2 c_{ijk}, \quad b_2 := \Delta_{32} \Delta_{12} c_{ijk}, \quad c_2 := \Delta_{12}^2 c_{ijk}, \quad (2.6)$$

$$a_3 := \Delta_{13}^2 c_{ijk}, \quad b_3 := \Delta_{13} \Delta_{23} c_{ijk}, \quad c_3 := \Delta_{23}^2 c_{ijk}. \quad (2.7)$$

Note that the a_m , b_m , and c_m all depend on i, j, k , but for simplicity, we have not incorporated this into the notation. For later use, we observe that

$$a_1 = c_2, \quad a_2 = c_3, \quad a_3 = c_1, \quad (2.8)$$

and

$$a_1 - b_1 = b_2, \quad a_2 - b_2 = b_3, \quad a_3 - b_3 = b_1. \quad (2.9)$$

Lemma 2.1. *The matrix A_{ijk} in (2.2) is nonnegative definite if and only if $q_m(t) \geq 0$ for $m = 1, 2, 3$ and all $0 \leq t \leq 1$.*

Proof: By definition, A_{ijk} is nonnegative definite if and only if

$$q(\eta_1, \eta_2) := a_1 \eta_1^2 + 2b_1 \eta_1 \eta_2 + c_1 \eta_2^2 \geq 0, \quad \text{all } (\eta_1, \eta_2) \in \mathbb{R}^2. \quad (2.10)$$

The inequality is trivial for $\eta_1 = \eta_2 = 0$. For convenience, we now divide $\mathbb{R}^2 \setminus \{(0, 0)\}$ into the following three subsets:

$$\mathcal{R}_1 := \{(\eta_1, \eta_2) \in \mathbb{R}^2 \setminus \{(0, 0)\} : \eta_1 \eta_2 \geq 0\},$$

$$\mathcal{R}_2 := \{(\eta_1, \eta_2) \in \mathbb{R}^2 \setminus \mathcal{R}_1 : \eta_2 \geq -\eta_1 \ \& \ \eta_1 > 0, \text{ or } \eta_2 \leq -\eta_1 \ \& \ \eta_1 < 0\},$$

$$\mathcal{R}_3 := \{(\eta_1, \eta_2) \in \mathbb{R}^2 \setminus \mathcal{R}_1 : \eta_2 < -\eta_1 \ \& \ \eta_1 > 0, \text{ or } \eta_2 > -\eta_1 \ \& \ \eta_1 < 0\}.$$

Then with $s_1 := \eta_1 / (\eta_1 + \eta_2)$, we have

$$q(\eta_1, \eta_2) = (\eta_1 + \eta_2)^2 (a_1 s_1^2 + 2b_1 s_1(1-s_1) + c_1(1-s_1)^2) = (\eta_1 + \eta_2)^2 q_1(s_1),$$

and so (2.10) holds for all $(\eta_1, \eta_2) \in \mathcal{R}_1$ if and only if $q_1(t)$ is nonnegative for $0 \leq t \leq 1$. Similarly, setting $s_2 := (\eta_1 + \eta_2) / \eta_1$ and using (2.8)–(2.9), we have

$$\begin{aligned} q(\eta_1, \eta_2) &= \eta_1^2 (a_1 - 2b_1(1-s_2) + c_1(1-s_2)^2) \\ &= \eta_1^2 (a_1 s_2^2 + 2(a_1 - b_1)s_2(1-s_2) + (a_1 - 2b_1 + c_1)(1-s_2)^2) \\ &= \eta_1^2 (c_2 s_2^2 + 2b_2 s_2(1-s_2) + a_2(1-s_2)^2) = \eta_1^2 q_2(1-s_2). \end{aligned}$$

Thus, (2.10) holds for all $(\eta_1, \eta_2) \in \mathcal{R}_2$ if and only if $q_2(t)$ is nonnegative for $0 < t \leq 1$. Finally, with $s_3 := (\eta_1 + \eta_2)/\eta_2$, we get

$$\begin{aligned} q(\eta_1, \eta_2) &= \eta_2^2(a_1(1-s_3)^2 - 2b_1(1-s_3) + c_1) \\ &= \eta_2^2(c_1s_3^2 + 2(c_1 - b_1)s_3(1-s_3) + (a_1 - 2b_1 + c_1)(1-s_3)^2) \\ &= \eta_2^2(a_3s_3^2 + 2b_3s_3(1-s_3) + c_3(1-s_3)^2) = \eta_2^2q_3(s_3). \end{aligned}$$

It follows that (2.10) holds for all $(\eta_1, \eta_2) \in \mathcal{R}_3$ if and only if $q_3(t)$ is nonnegative for $0 < t < 1$. To complete the necessity part of the proof, we need to show that if A_{ijk} is nonnegative definite, then $q_2(0) \geq 0$, $q_3(0) \geq 0$ and $q_3(1) \geq 0$. This follows from $q_2(0) = q_1(1)$, $q_3(0) = q_2(1)$, and $q_3(1) = q_1(0)$. \square

We are now ready to derive rather general linear conditions that imply the convexity of p . The construction is inspired by ideas in [23], and is based on blossoming. In particular, we consider the blossoms r_m of the quadratic polynomials q_m in (2.4). Recall that these are the unique bivariate symmetric multi-affine polynomials such that $r_m(t, t) = q_m(t)$ for all t , see [37]. In particular, for $m = 1, 2, 3$,

$$r_m(t_1, t_2) := a_mt_1t_2 + b_m(t_1(1-t_2) + t_2(1-t_1)) + c_m(1-t_1)(1-t_2), \quad (2.11)$$

where the coefficients a_m , b_m and c_m are as in (2.5)–(2.7).

Theorem 2.2. *Given arbitrary strictly positive integers n_1 , n_2 , and n_3 , let $\mathcal{X} := \{x_l\}_{l=0}^{n_1}$, $\mathcal{Y} := \{y_l\}_{l=0}^{n_2}$, and $\mathcal{Z} := \{z_l\}_{l=0}^{n_3}$, where*

$$\begin{aligned} 0 &= x_0 < x_1 < \cdots < x_{n_1} = 1, \\ 0 &= y_0 < y_1 < \cdots < y_{n_2} = 1, \\ 0 &= z_0 < z_1 < \cdots < z_{n_3} = 1. \end{aligned}$$

Then the conditions

$$\begin{aligned} r_1(x_{l-1}, x_l) &\geq 0, & l &= 1, \dots, n_1, \\ r_2(y_{l-1}, y_l) &\geq 0, & l &= 1, \dots, n_2, \\ r_3(z_{l-1}, z_l) &\geq 0, & l &= 1, \dots, n_3, \end{aligned} \quad (2.12)$$

imply that the matrix A_{ijk} in (2.2) is nonnegative definite.

Proof: We shall apply Lemma 2.1. We need to show that the quadratic polynomials q_1 , q_2 , and q_3 appearing in the lemma are nonnegative on $[0, 1]$. We begin by showing that the conditions (2.12) imply

$$\begin{aligned} r_1(x_l, x_l) &\geq 0, & l &= 0, \dots, n_1, \\ r_2(y_l, y_l) &\geq 0, & l &= 0, \dots, n_2, \\ r_3(z_l, z_l) &\geq 0, & l &= 0, \dots, n_3. \end{aligned} \quad (2.13)$$

We first deal with the extreme values of l . To this end, note that

$$\begin{aligned} r_1(x_{n_1}, x_{n_1}) &= r_2(y_0, y_0) = a_1, \\ r_2(y_{n_2}, y_{n_2}) &= r_3(z_0, z_0) = a_2, \\ r_3(z_{n_3}, z_{n_3}) &= r_1(x_0, x_0) = a_3. \end{aligned}$$

We claim that the a_m , $m = 1, 2, 3$, are nonnegative. Using (2.9), we have

$$r_1(x_{n_1-1}, x_{n_1}) = a_1 x_{n_1-1} + b_1(1 - x_{n_1-1}) = a_1 - b_2(1 - x_{n_1-1}).$$

Then by (2.12),

$$a_1 \geq b_2(1 - x_{n_1-1}).$$

If $n_2 = 1$, then $b_2 = r_2(y_0, y_1) \geq 0$, and it follows that $a_1 \geq 0$. If $n_2 > 1$, then $0 \leq r_2(y_0, y_1) = b_2 y_1 + a_1(1 - y_1)$, which implies that

$$a_1 \geq |b_2| \min \left\{ 1 - x_{n_1-1}, \frac{y_1}{1 - y_1} \right\} \geq 0.$$

A similar proof shows that $a_2 \geq 0$ and $a_3 \geq 0$.

To see that r_1 satisfies (2.13) for $l = 1, \dots, n_1 - 1$, we use (2.12) along with the fact that r_1 is symmetric and multi-affine, i.e.,

$$r_1(x_l, x_l) = \frac{x_{l+1} - x_l}{x_{l+1} - x_{l-1}} r_1(x_{l-1}, x_l) + \frac{x_l - x_{l-1}}{x_{l+1} - x_{l-1}} r_1(x_l, x_{l+1}) \geq 0,$$

for $l = 1, \dots, n_1 - 1$. A similar argument shows that r_m satisfies (2.13) for $l = 1, \dots, n_m - 1$ for $m = 2, 3$.

We are now in a position to show that the polynomial q_1 is nonnegative on $[0, 1]$. Fix $1 \leq l \leq n_1$, and let $q_{1,l}$ be the restriction of q_1 to the interval $[x_{l-1}, x_l]$. Since r_1 is the blossom of q_1 , it follows that $q_{1,l}$ can be written in B-form with B-coefficients $r_1(x_{l-1}, x_{l-1})$, $r_1(x_{l-1}, x_l)$ and $r_1(x_l, x_l)$. But by (2.12) and (2.13), these B-coefficients are nonnegative, and we conclude that q_1 is nonnegative on $[x_{l-1}, x_l]$. Since this holds for $l = 1, \dots, n_1$, we conclude that q_1 is nonnegative on $[0, 1]$. A similar argument applies to q_2 and q_3 . \square

It is easy to see that all conditions in (2.12) depend linearly on c_{ijk} .

Corollary 2.3. *Suppose that in Theorem 2.2 we choose $\mathcal{X} = \mathcal{Y} = \mathcal{Z}$ to be symmetric around $1/2$. Then the set of conditions (2.12) is symmetric with respect to b_m , $m = 1, 2, 3$.*

Proof: By (2.8)–(2.9) and the definition of r_m , the symmetry of the conditions follows. \square

The symmetry in the corollary implies that the set of conditions will not change if the indices $(1, 2, 3)$ in the difference operators are replaced by any permutation of $(1, 2, 3)$.

Corollary 2.4. *Suppose $\mathcal{X} \subset \tilde{\mathcal{X}}$, $\mathcal{Y} \subset \tilde{\mathcal{Y}}$ and $\mathcal{Z} \subset \tilde{\mathcal{Z}}$. Then the convexity conditions (2.12) generated by the sets $\tilde{\mathcal{X}}$, $\tilde{\mathcal{Y}}$ and $\tilde{\mathcal{Z}}$ are weaker than the conditions generated by the sets \mathcal{X} , \mathcal{Y} and \mathcal{Z} .*

Proof: The control polygons generated by $\tilde{\mathcal{X}}$, $\tilde{\mathcal{Y}}$ and $\tilde{\mathcal{Z}}$ are closer to their corresponding polynomials $q_m(t)$, $m = 1, 2, 3$, than the control polygons generated by \mathcal{X} , \mathcal{Y} and \mathcal{Z} . \square

2.2. Some particular sets of sufficient conditions for convexity

In this section we present several corollaries of Theorem 2.2 which show that our conditions for the convexity of a polynomial patch subsume the known linear conditions in the literature.

Corollary 2.5. *Suppose p is a polynomial in B-form as in (2.1) with coefficients satisfying*

$$\Delta_{21}\Delta_{31}c_{ijk} \geq 0, \quad \Delta_{32}\Delta_{12}c_{ijk} \geq 0, \quad \Delta_{13}\Delta_{23}c_{ijk} \geq 0, \quad (2.14)$$

for all $i + j + k = d - 2$. Then the corresponding surface patch is convex on T .

Proof: We apply Theorem 2.2 with $\mathcal{X} = \mathcal{Y} = \mathcal{Z} = \{0, 1\}$. Then $r_m(0, 1) = b_m$ for $m = 1, 2, 3$, and the claim follows. \square

These conditions were first observed in [8]. They can be interpreted as requiring the convexity of the control net associated with p .

Corollary 2.6. *Suppose p is a polynomial in B-form as in (2.1) with coefficients satisfying*

$$\begin{aligned} (\Delta_{21} + \Delta_{31})\Delta_{31}c_{ijk} &\geq 0, & \Delta_{21}(\Delta_{21} + \Delta_{31})c_{ijk} &\geq 0, \\ \Delta_{32}\Delta_{12}c_{ijk} &\geq 0, & \Delta_{13}\Delta_{23}c_{ijk} &\geq 0, \end{aligned} \quad (2.15)$$

for all $i + j + k = d - 2$. Then the corresponding surface patch is convex on T .

Proof: We apply Theorem 2.2 with $\mathcal{X} = \{0, 1/2, 1\}$ and $\mathcal{Y} = \mathcal{Z} = \{0, 1\}$. Then we get $r_1(0, 1/2) = (b_1 + c_1)/2 \geq 0$, $r_1(1/2, 1) = (a_1 + b_1)/2 \geq 0$, and $r_m(0, 1) = b_m \geq 0$ for $m = 2, 3$. \square

By (2.9) it follows that (2.15) is equivalent to

$$\Delta_{21}^2 c_{ijk} \geq |\Delta_{21}\Delta_{31}c_{ijk}|, \quad \Delta_{31}^2 c_{ijk} \geq |\Delta_{21}\Delta_{31}c_{ijk}|.$$

These conditions (first given in [26]) imply that the matrices A_{ijk} in (2.2) are weakly diagonally dominant.

Corollary 2.7. *Suppose p is a polynomial in B-form as in (2.1) with coefficients satisfying*

$$\begin{aligned}(\Delta_{21} + \Delta_{31})\Delta_{31}c_{ijk} &\geq 0, & \Delta_{21}(\Delta_{21} + \Delta_{31})c_{ijk} &\geq 0, \\(\Delta_{32} + \Delta_{12})\Delta_{32}c_{ijk} &\geq 0, & \Delta_{32}(\Delta_{32} + \Delta_{12})c_{ijk} &\geq 0, \\(\Delta_{13} + \Delta_{23})\Delta_{23}c_{ijk} &\geq 0, & \Delta_{13}(\Delta_{13} + \Delta_{23})c_{ijk} &\geq 0,\end{aligned}\tag{2.16}$$

for all $i + j + k = d - 2$. Then the surface patch associated with p is convex on T .

Proof: We apply Theorem 2.2 with $\mathcal{X} = \mathcal{Y} = \mathcal{Z} = \{0, 1/2, 1\}$. Then we get $r_m(0, 1/2) = (b_m + c_m)/2 \geq 0$ and $r_m(1/2, 1) = (a_m + b_m)/2 \geq 0$ for $m = 1, 2, 3$. \square

Using (2.9) the inequalities in (2.16) can be reformulated as

$$\begin{aligned}(2\Delta_{21}\Delta_{31} + \Delta_{13}\Delta_{23})c_{ijk} &\geq 0, & (2\Delta_{21}\Delta_{31} + \Delta_{32}\Delta_{12})c_{ijk} &\geq 0, \\(2\Delta_{32}\Delta_{12} + \Delta_{21}\Delta_{31})c_{ijk} &\geq 0, & (2\Delta_{32}\Delta_{12} + \Delta_{13}\Delta_{23})c_{ijk} &\geq 0, \\(2\Delta_{13}\Delta_{23} + \Delta_{32}\Delta_{12})c_{ijk} &\geq 0, & (2\Delta_{13}\Delta_{23} + \Delta_{21}\Delta_{31})c_{ijk} &\geq 0,\end{aligned}$$

which are the conditions found in [6]. From Corollary 2.4 it directly follows that this set of linear conditions is weaker than the ones in Corollaries 2.5 and 2.6.

Corollary 2.8. *Suppose p is a polynomial in B-form as in (2.1) with coefficients satisfying*

$$\begin{aligned}(\Delta_{21} + 2\Delta_{31})\Delta_{31}c_{ijk} &\geq 0, & (\Delta_{21}^2 + 3\Delta_{21}\Delta_{31} + 2\Delta_{31}^2)c_{ijk} &\geq 0, \\ \Delta_{21}(2\Delta_{21} + \Delta_{31})c_{ijk} &\geq 0, & (2\Delta_{21}^2 + 3\Delta_{21}\Delta_{31} + \Delta_{31}^2)c_{ijk} &\geq 0, \\(\Delta_{32} + 2\Delta_{12})\Delta_{12}c_{ijk} &\geq 0, & (\Delta_{32}^2 + 3\Delta_{32}\Delta_{12} + 2\Delta_{12}^2)c_{ijk} &\geq 0, \\ \Delta_{32}(2\Delta_{32} + \Delta_{12})c_{ijk} &\geq 0, & (2\Delta_{32}^2 + 3\Delta_{32}\Delta_{12} + \Delta_{12}^2)c_{ijk} &\geq 0, \\(\Delta_{13} + 2\Delta_{23})\Delta_{23}c_{ijk} &\geq 0, & (\Delta_{13}^2 + 3\Delta_{13}\Delta_{23} + 2\Delta_{23}^2)c_{ijk} &\geq 0, \\ \Delta_{13}(2\Delta_{13} + \Delta_{23})c_{ijk} &\geq 0, & (2\Delta_{13}^2 + 3\Delta_{13}\Delta_{23} + \Delta_{23}^2)c_{ijk} &\geq 0,\end{aligned}\tag{2.17}$$

for all $i + j + k = d - 2$. Then the corresponding surface patch is convex on T .

Proof: We apply Theorem 2.2 with $\mathcal{X} = \mathcal{Y} = \mathcal{Z} = \{0, 1/3, 1/2, 2/3, 1\}$. Then we get $r_m(0, 1/3) = (b_m + 2c_m)/3 \geq 0$, $r_m(1/3, 1/2) = (a_m + 3b_m + 2c_m)/6 \geq 0$, $r_m(1/2, 2/3) = (2a_m + 3b_m + c_m)/6 \geq 0$ and $r_m(2/3, 1) = (2a_m + b_m)/3 \geq 0$ for $m = 1, 2, 3$. \square

By (2.9) the first two inequalities in (2.17) are equivalent to

$$(3\Delta_{21}\Delta_{31} + 2\Delta_{13}\Delta_{23})c_{ijk} \geq 0, \quad (6\Delta_{21}\Delta_{31} + 2\Delta_{13}\Delta_{23} + \Delta_{32}\Delta_{12})c_{ijk} \geq 0.$$

The remaining conditions can be reformulated in a similar way, and we obtain the conditions derived in [31]. By Corollary 2.4 this set of conditions is weaker than (2.16). The symmetry is implied by Corollary 2.3.

2.3. A recurrence relation

Suppose the set $\mathcal{X} = \{x_l\}_{l=0}^{n_1}$ in Theorem 2.2 is replaced by $\tilde{\mathcal{X}} = \mathcal{X} \cup \{\tilde{x}_k\}$ with $x_{k-1} < \tilde{x}_k < x_k$. Then all linear conditions $r_1(x_{l-1}, x_l) \geq 0$ will be retained, except for $l = k$. The condition $r_1(x_{k-1}, x_k) \geq 0$ will be replaced by the pair of weaker conditions

$$r_1(x_{k-1}, \tilde{x}_k) \geq 0, \quad r_1(\tilde{x}_k, x_k) \geq 0.$$

Both conditions can be written in terms of the original ones. Since r_1 is symmetric and multi-affine, it follows that for any $2 \leq k \leq n_1$,

$$r_1(x_{k-1}, \tilde{x}_k) = \frac{x_k - \tilde{x}_k}{x_k - x_{k-2}} r_1(x_{k-2}, x_{k-1}) + \frac{\tilde{x}_k - x_{k-2}}{x_k - x_{k-2}} r_1(x_{k-1}, x_k), \quad (2.18)$$

and for $k = 1$,

$$\begin{aligned} r_1(x_0, \tilde{x}_1) &= \frac{x_1 - \tilde{x}_1}{x_1 - x_0} r_1(x_0, x_0) + \frac{\tilde{x}_1 - x_0}{x_1 - x_0} r_1(x_0, x_1) \\ &= \frac{x_1 - \tilde{x}_1}{x_1} c_1 + \frac{\tilde{x}_1}{x_1} r_1(x_0, x_1). \end{aligned} \quad (2.19)$$

Similarly, for any $1 \leq k \leq n_1 - 1$,

$$r_1(\tilde{x}_k, x_k) = \frac{x_{k+1} - \tilde{x}_k}{x_{k+1} - x_{k-1}} r_1(x_{k-1}, x_k) + \frac{\tilde{x}_k - x_{k-1}}{x_{k+1} - x_{k-1}} r_1(x_k, x_{k+1}), \quad (2.20)$$

and for $k = n_1$,

$$\begin{aligned} r_1(\tilde{x}_{n_1}, x_{n_1}) &= \frac{x_{n_1} - \tilde{x}_{n_1}}{x_{n_1} - x_{n_1-1}} r_1(x_{n_1-1}, x_{n_1}) + \frac{\tilde{x}_{n_1} - x_{n_1-1}}{x_{n_1} - x_{n_1-1}} r_1(x_{n_1}, x_{n_1}) \\ &= \frac{1 - \tilde{x}_{n_1}}{1 - x_{n_1-1}} r_1(x_{n_1-1}, x_{n_1}) + \frac{\tilde{x}_{n_1} - x_{n_1-1}}{1 - x_{n_1-1}} a_1. \end{aligned} \quad (2.21)$$

Formulae (2.18)–(2.21) can also be used when multiple knots \tilde{x}_{k_i} , $i = 1, \dots, \tilde{n}$, are inserted simultaneously, on condition that there is at most one new knot per interval (x_{l-1}, x_l) . Analogous expressions can be obtained if the sets \mathcal{Y} and \mathcal{Z} are extended.

2.4. A geometric interpretation

In this section we give a geometric interpretation of the linear convexity conditions of Theorem 2.2. Let $\xi_{ijk} := (iv_1 + jv_2 + kv_3)/(i + j + k)$ be the domain points on the triangle $T := \langle v_1, v_2, v_3 \rangle$, and let $Q_{ijk} := (\xi_{ijk}, c_{ijk})$ be the control points of a polynomial defined on T . Fix $i + j + k = d - 2$, and for $m = 1, 2, 3$, define

$$R_m(t_1, t_2) := A_m t_1 t_2 + B_m (t_1(1 - t_2) + t_2(1 - t_1)) + C_m (1 - t_1)(1 - t_2), \quad (2.22)$$

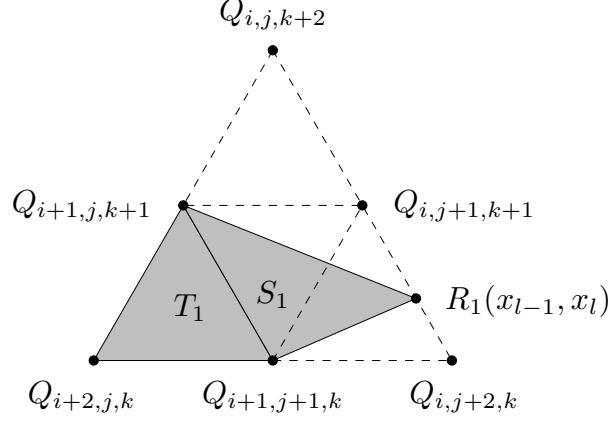


Fig. 1. The graph of $T_1 \cup S_1$ (shaded) must be convex.

with

$$A_1 := Q_{i,j+2,k}, \quad B_1 := Q_{i,j+1,k+1}, \quad C_1 := Q_{i,j,k+2}, \quad (2.23)$$

$$A_2 := Q_{i,j,k+2}, \quad B_2 := Q_{i+1,j,k+1}, \quad C_2 := Q_{i+2,j,k}, \quad (2.24)$$

$$A_3 := Q_{i+2,j,k}, \quad B_3 := Q_{i+1,j+1,k}, \quad C_3 := Q_{i,j+2,k}. \quad (2.25)$$

These quantities all depend on i, j, k , but for simplicity, we have suppressed this in the notation.

Theorem 2.9. *Let*

$$T_1 := \langle Q_{i+2,j,k}; Q_{i+1,j+1,k}; Q_{i+1,j,k+1} \rangle, \quad (2.26)$$

$$S_1 := \langle R_1(x_{l-1}, x_l); Q_{i+1,j,k+1}; Q_{i+1,j+1,k} \rangle. \quad (2.27)$$

The condition $r_1(x_{l-1}, x_l) \geq 0$ is equivalent to requiring that the graph of $T_1 \cup S_1$ be convex (see Fig. 1).

Proof: Let (α, β, γ) such that

$$R_1(x_{l-1}, x_l) = \alpha A_1 + \beta B_1 + \gamma C_1.$$

Requiring that the graph of $T_1 \cup S_1$ be convex is equivalent to

$$\frac{Q_{i+2,j,k} + R_1(x_{l-1}, x_l)}{2} \geq \alpha Q_{i+1,j+1,k} + \beta \frac{Q_{i+1,j+1,k} + Q_{i+1,j,k+1}}{2} + \gamma Q_{i+1,j,k+1}.$$

Since $\alpha + \beta + \gamma = 1$, we have

$$\begin{aligned} Q_{i+2,j,k} + R_1(x_{l-1}, x_l) &= \alpha(Q_{i+2,j,k} + Q_{i,j+2,k}) + \beta(Q_{i+2,j,k} + Q_{i,j+1,k+1}) \\ &\quad + \gamma(Q_{i+2,j,k} + Q_{i,j,k+2}). \end{aligned}$$

Thus, we get the condition

$$\begin{aligned} & \alpha(Q_{i+2,j,k} + Q_{i,j+2,k} - 2Q_{i+1,j+1,k}) \\ & + \beta(Q_{i+2,j,k} + Q_{i,j+1,k+1} - Q_{i+1,j+1,k} - Q_{i+1,j,k+1}) \\ & + \gamma(Q_{i+2,j,k} + Q_{i,j,k+2} - 2Q_{i+1,j,k+1}) \geq 0, \end{aligned}$$

or,

$$\alpha\Delta_{21}^2 Q_{ijk} + \beta\Delta_{21}\Delta_{31} Q_{ijk} + \gamma\Delta_{31}^2 Q_{ijk} \geq 0,$$

which is equivalent to $r_1(x_{l-1}, x_l) \geq 0$. \square

The next two theorems can be proven in a similar way.

Theorem 2.10. *Let*

$$T_2 := \langle Q_{i,j+2,k}; Q_{i,j+1,k+1}; Q_{i+1,j+1,k} \rangle, \quad (2.28)$$

$$S_2 := \langle R_2(y_{l-1}, y_l); Q_{i+1,j+1,k}; Q_{i,j+1,k+1} \rangle. \quad (2.29)$$

The condition $r_2(y_{l-1}, y_l) \geq 0$ is equivalent to requiring that the graph of $T_2 \cup S_2$ be convex.

Theorem 2.11. *Let*

$$T_3 := \langle Q_{i,j,k+2}; Q_{i+1,j,k+1}; Q_{i,j+1,k+1} \rangle, \quad (2.30)$$

$$S_3 := \langle R_3(z_{l-1}, z_l); Q_{i,j+1,k+1}; Q_{i+1,j,k+1} \rangle. \quad (2.31)$$

The condition $r_3(z_{l-1}, z_l) \geq 0$ is equivalent to requiring that the graph of $T_3 \cup S_3$ be convex.

§3. Constructing convex splines

In this section we discuss some methods for constructing convex splines which make use of the above convexity conditions. Given a triangulation Δ of a domain Ω , let $\mathcal{S}_d^r(\Delta)$ be the space of C^r splines of degree d defined on Δ . It is well known that if $s \in \mathcal{S}_d^1(\Delta)$ is such that $s|_T$ is convex on T for every triangle $T \in \Delta$, then s is convex on Ω , see e.g. [11,20]. This immediately gives us the following useful result.

Theorem 3.1. *Suppose $s \in \mathcal{S}_d^1(\Delta)$ is a C^1 spline defined on Δ such that for every triangle $T \in \Delta$, $s|_T$ satisfies a set of sufficient convexity conditions as given in Theorem 2.2. Then s is convex on Ω .*

3.1. A quadratic programming problem

To construct a convex spline that fits given data, it is natural to formulate the problem as a quadratic programming program where convexity is enforced by including appropriate side conditions on the coefficients of the spline, see e.g. [23,27]. We discuss two approaches. The first, to be discussed in this subsection, does not require an explicit basis for the spline space being used. The second approach, to be discussed in the following subsection, applies to spline spaces with stable bases.

Given a triangulation Δ , let $\mathcal{S}_d^0(\Delta)$ be the space of C^0 splines of degree d defined on Δ . Then as shown in Theorem 5.1 of [28], every $s \in \mathcal{S}_d^0(\Delta)$ is uniquely defined by its set of B-coefficients $\{c_\xi\}_{\xi \in \mathcal{D}_{d,\Delta}}$, where $\mathcal{D}_{d,\Delta}$ is the set of domain points associated with Δ . Thus, $n_0 := \dim \mathcal{S}_d^0(\Delta) = \#\mathcal{D}_{d,\Delta}$. Since we want to apply Theorem 3.1, we need our splines to be at least C^1 continuous. We can achieve this by enforcing standard smoothness conditions across the edges of the triangulation. These conditions can be described by a linear system of equations of the form

$$Ac = 0, \tag{3.1}$$

where A is a given $n_s \times n_0$ matrix and c is the n_0 -vector of B-coefficients in some agreed-upon order, see Sect. 5.5.4 of [28]. In some cases we may also require our spline to satisfy some additional interpolation conditions. Such conditions are also linear, i.e.,

$$Bc = z, \tag{3.2}$$

where B is a given $n_I \times n_0$ matrix and z is a vector of length n_I . We write

$$\mathcal{S} := \{s \in \mathcal{S}_d^0(\Delta) : Ac = 0 \text{ and } Bc = z\}.$$

Now suppose we want to compute a spline $s \in \mathcal{S}$ that is convex. Let C be an $n_c \times n_0$ matrix such that

$$Cc \geq 0 \tag{3.3}$$

is a sufficient set of linear conditions for s to be convex. Let $J(c) := \frac{1}{2}c^T Hc + g^T c + h$ be a convenient quadratic form defined by the $n_0 \times n_0$ matrix H , the vector g of length n_0 and the scalar h . We give some examples of possible choices for J in Sect. 3.3 below. Then we pose the following quadratic programming problem which can be solved using standard software packages.

Problem 3.2. *Find c that minimizes $J(c)$ subject to the linear side conditions $Ac = 0$, $Bc = z$, and the convexity constraints $Cc \geq 0$.*

Assuming the matrix H is symmetric positive definite, this problem has a unique solution provided that

- a) there exists a feasible solution, i.e., a set of coefficients satisfying all of the constraints,
- b) $J(c)$ is bounded below on the feasible region.

This is not always the case, particularly if the set of convexity and interpolation conditions is too restrictive. In this case it may be possible to replace the given set of linear convexity conditions by weaker ones to produce a feasible problem, see Remarks 3–6.

3.2. Using a stable spline basis

For applications, the most convenient spline spaces \mathcal{S} have a stable basis $\mathcal{B} := \{B_i\}_{i=1}^n$. Such bases can be constructed using stable local minimal determining sets, see Sect. 5.6 of [28], or stable local nodal determining sets, see Sect. 5.9 of [28]. A spline $s \in \mathcal{S}$ can then be uniquely represented as

$$s(x, y) = \sum_{i=1}^n \bar{c}_i B_i(x, y). \quad (3.4)$$

The coefficients $\bar{c} = \{\bar{c}_i\}_{i=1}^n$ are referred to as the degrees of freedom of \mathcal{S} . If we use the basis corresponding to a stable local minimal determining set \mathcal{M} , then $\bar{c} = \{c_\xi\}_{\xi \in \mathcal{M}}$.

When \mathcal{S} is a spline space with a stable local basis, we can generally solve Problem 3.2 by dealing with a significantly smaller equivalent problem. For such spline spaces, we can construct an $n_0 \times n$ matrix G such that for any spline $s \in \mathcal{S}$, the B-coefficients of s are given by

$$c = G\bar{c}. \quad (3.5)$$

Then it is clear that Problem 3.2 is equivalent to the following one.

Problem 3.3. Find \bar{c} that minimizes $\bar{J}(\bar{c}) := \frac{1}{2}\bar{c}^T G^T H G \bar{c} + g^T G \bar{c} + h$ subject to the linear side conditions $B G \bar{c} = z$ and the convexity constraints $C G \bar{c} \geq 0$.

Once we have a solution \bar{c} to this problem, we then get the solution of Problem 3.2 by setting $c = G\bar{c}$. One advantage of this approach is that we no longer have to explicitly enforce the smoothness conditions. In addition, the number of unknowns n here is generally much smaller than the number n_0 in Problem 3.2. Moreover, in case a local minimal determining set \mathcal{M} is used, if we want to interpolate at the vertices of Δ , then we can usually choose \mathcal{M} to include these points, thus reducing the number of unknowns even further.

3.3. Some explicit methods

In this section we describe three explicit choices of J that lead to useful algorithms.

Method 1. (A two-stage method) Let \mathcal{S} be a given spline space defined on a triangulation Δ . Suppose \tilde{s} is a spline in \mathcal{S} that has been constructed by any convenient interpolation or approximation method, and let \tilde{c} be its set of B-coefficients. Then

we construct a convex spline $s \in \mathcal{S}$ approximating \tilde{s} by solving the above quadratic programming problem with

$$J_1(c) := \frac{1}{2} \sum_{\xi \in \mathcal{D}_{d,\Delta}} [c_\xi - \tilde{c}_\xi]^2. \quad (3.6)$$

This can be regarded as a two-stage method. In the first stage we compute \tilde{s} , and in the second stage we approximate it with a convex spline. As a first stage we can use any interpolation, discrete least squares, penalized least squares, or minimal energy method, see [36].

Method 2. (Convex least-squares) Suppose we are given data $\{(x_i, y_i, f_i)\}_{i=1}^{n_d}$. Typically, $f_i = f(x_i, y_i)$ for some (possibly unknown) convex function f . Let \mathcal{S} be a spline space defined on a triangulation Δ . Then we construct a convex spline $s \in \mathcal{S}$ approximating f by solving the above quadratic programming problem with

$$J_2(c) := \frac{1}{2} \sum_{i=1}^{n_d} [s(x_i, y_i) - f_i]^2. \quad (3.7)$$

We emphasize that in this problem, the data values are not necessarily at the vertices of the triangulation Δ , and in fact usually the number of data values n_d is much larger than the dimension n of the spline space \mathcal{S} . Although the computation of the quadratic form $J_2(c)$ would appear to require an explicit basis for \mathcal{S} , as shown in [36], for spline spaces with a stable local minimal determining set, the construction of an explicit basis can be avoided as long as we know the matrix G in (3.5) connecting the B-coefficients of a spline in \mathcal{S} to its degrees of freedom.

Method 3. (Convex minimal energy splines). Let Δ be a triangulation with vertices $\{x_i, y_i\}_{i=1}^{n_v}$, and let \mathcal{S} be a space of splines defined on Δ . Suppose we are given values $\{z_i\}_{i=1}^{n_v}$ at the vertices. Then we construct a convex spline s by solving Problem 3.2 while enforcing the interpolation conditions at the vertices, where now

$$J_3(c) := \int_{\Omega} [(D_{xx}s)^2 + 2(D_{xy}s)^2 + (D_{yy}s)^2] dx dy. \quad (3.8)$$

Note that for some spline spaces, the derivatives required in (3.8) might not be defined at every point in the domain Ω , in which case $J_3(c)$ is computed by summing over the triangles. As shown in [36], as with Method 2, it is not necessary to have an explicit basis for \mathcal{S} to compute $J_3(c)$.

3.4. Some C^1 macro-element spaces

In this section we briefly describe three macro-element spline spaces that are suitable for convex data fitting. For a comprehensive treatment of macro-element spline spaces, see Chapter 6–8 of [28].

- 1) $\mathcal{S}_5^{1,2}(\Delta)$. This space has dimension $6n_v + n_e$, where n_v and n_e are the number of vertices and edges in the triangulation Δ , respectively. As shown in Theorem 6.1 of [28], a stable minimal determining set can be chosen consisting of six B-coefficients in the 2-disk around each vertex, along with one B-coefficient near each edge. The basis functions in this case have been called Argyris elements.
- 2) $\mathcal{S}_3^1(\Delta_{CT})$, where Δ_{CT} is the corresponding Clough–Tocher refinement of Δ . This space has dimension $3n_v + n_e$, and a stable minimal determining set can be chosen consisting of three B-coefficients in the 1-disk around each vertex of Δ , along with one B-coefficient near each edge, see Theorem 6.5 of [28].
- 3) $\mathcal{S}_2^1(\Delta_{PS})$, where Δ_{PS} is the corresponding Powell–Sabin refinement of Δ . This space has dimension $3n_v$, and a stable minimal determining set can be chosen consisting of three B-coefficients in the 1-disk around each vertex of Δ , see Theorem 6.9 of [28].

§4. Numerical examples

In this section we describe some computational experiments with the above methods. We work with known test functions f on the unit square $\Omega := [0, 1] \times [0, 1]$. To check that a C^1 spline is convex, it suffices to show that for every direction u , the directional derivative $D_u^2 s$ is nonnegative on every triangle. This in turn can be checked by showing that $D_{xx}s$, $D_{yy}s$ and

$$D_{det}s = (D_{xx}s)(D_{yy}s) - (D_{xy}s)^2 \quad (4.1)$$

are nonnegative on every triangle. We cannot check these quantities at every point in every triangle, but we can compute their minimum values on a fairly dense set of points P in Ω . Here we choose P to be a 400×400 uniform grid on Ω . We also use the grid P to measure the accuracy of any spline fit s . In particular, in the experiments below we report the maximum error on P :

$$\text{errmax} := \max_{(x,y) \in P} |f(x,y) - s(x,y)|.$$

In each of the experiments, we suppose that we are given measurements of f on a uniform grid of 5×5 sample points. We report results only for the two-stage Method 1 described in Sect. 3.3. For each experiment we carry out the following steps:

- 1) choose a triangulation Δ with vertices at the sample points;

- 2) choose a macro-element space \mathcal{S} defined on an appropriate refinement Δ_R of Δ ;
- 3) compute a Hermite interpolating spline $\tilde{s} \in \mathcal{S}$ using the true derivatives of f where needed;
- 4) choose a set of convexity conditions to apply to every triangle in Δ_R ;
- 5) solve Problem 3.2 or 3.3 using the quadratic form $J_1(c)$ defined in (3.6), where \tilde{c} are the coefficients of the spline \tilde{s} constructed in step 3);
- 6) compute the minimum of D_{xx} , D_{yy} and D_{det} on the grid P to test convexity of both \tilde{s} and s ;
- 7) compute the maximum error on P for both \tilde{s} and s .

Example 4.1. Consider the convex function

$$f(x, y) = \begin{cases} 3 \left(\sqrt{\left(x - \frac{1}{2}\right)^2 + \left(y - \frac{1}{2}\right)^2} - \frac{1}{4} \right)^2, & \left(x - \frac{1}{2}\right)^2 + \left(y - \frac{1}{2}\right)^2 > \frac{1}{16}, \\ 0, & \text{otherwise,} \end{cases} \quad (4.2)$$

on the domain $\Omega := [0, 1] \times [0, 1]$.

Discussion: For this example we choose Δ to be the triangulation shown in Fig. 2. It is the triangulation obtained after applying the swapping algorithm described in [5] to the type-1 triangulation associated with vertices at the 25 grid points. It is known that with this triangulation, the piecewise linear spline interpolating the data at these sample points is convex on Ω . For step 3), we fit these data with the three macro-element spline spaces described in Sect. 3.4. In all three cases, we enforced the interpolation conditions. We used the convexity conditions of Corollary 2.7 for the Powell–Sabin and Argyris elements. For the Clough–Tocher elements we used the convexity conditions of Corollary 2.8 since there is no feasible solution when using those in (2.16). Tab. 1 shows the minimum values of D_{xx} and D_{det} for each of the Hermite interpolants along with the maximum error on P . It also shows the number n_c of convexity conditions enforced, and the maximum error for the resulting convex splines.

The table shows similar results for the three different spline spaces. In all three cases, the maximum errors for the convex spline fits were only slightly larger than those for the original (non-convex) Hermite interpolants. \square

spline type	Hermite interpolant			convex interpolant	
	D_{xx}	D_{det}	error	n_c	error
Powell–Sabin	-0.43	-0.31	2.62e-3	1152	3.55e-3
Clough–Tocher	-1.60	-2.45	1.67e-3	3456	3.71e-3
Argyris	-1.62	-3.59	6.48e-3	1920	6.98e-3

Tab. 1. The results for Example 4.1.

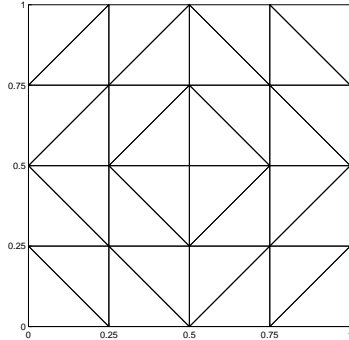


Fig. 2. The triangulation for Example 4.1.

Example 4.2. Consider the convex function

$$f(x, y) = x^3 + 5(y - 0.6)^2 + 1 \quad (4.3)$$

on the domain $\Omega := [0, 1] \times [0, 1]$.

Discussion: This test function is taken from [30,39]. We fit this function with the C^1 quadratic Powell–Sabin macro element, with and without interpolation conditions at the 25 data points. For this example we take Δ to be the triangulation shown in Fig. 3. We take $\mathcal{X} = \mathcal{Y} = \mathcal{Z}$ with equally-spaced points in $[0, 1]$. We do a sequence of tests where we start with one interval, and then double the number $n_{\mathcal{X}}$ of intervals at each step. This gives nested feasible regions spanned by the corresponding convexity conditions. In Tab. 2 we show the maximum errors for the convex interpolating and non-interpolating splines, along with the number n_c of convexity constraints for each fit. For $n_{\mathcal{X}} = 1$ and $n_{\mathcal{X}} = 2$, there is no feasible convex Powell–Sabin interpolant. For these $n_{\mathcal{X}}$ there is a non-interpolating convex fit, but the error is much worse than for the cases where $n_{\mathcal{X}}$ is larger. For the interpolating splines, the errors improve with an increasing number of convexity constraints. \square

convex Powell–Sabin fit			
$n_{\mathcal{X}}$	n_c	interp. error	approx. error
1	576	–	8.78e-1
2	1152	–	2.39e-1
4	2304	1.04e-3	9.77e-4
8	4608	9.90e-4	9.77e-4
16	9216	9.80e-4	9.77e-4

Tab. 2. The results for Example 4.2.

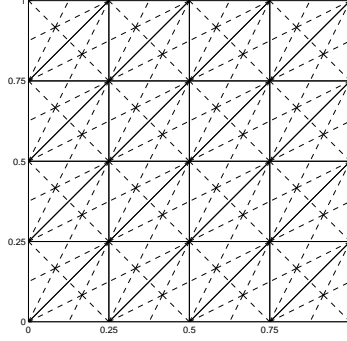


Fig. 3. The triangulation and its PS-refinement for Example 4.2.

§5. Remarks

Remark 1. If we define $J_4(c) := J_2(c) + \lambda J_3(c)$ for fixed λ , then we get a variant of Method 2 which leads to a convex penalized least squares fit. This may be useful in the case where the data is noisy. The parameter λ controls the trade-off between closeness of fit and smoothness of the surface, and is usually taken to be quite small.

Remark 2. The energy expression used in Method 3 is the natural way to measure the energy of a spline surface. However, the method works also with other energy expressions. For example, one could use a mix of higher derivatives.

Remark 3. The solutions of Problems 3.2 and 3.3 usually improve as we weaken the convexity conditions. Given a set of linear convexity conditions generated by the knot sets \mathcal{X} , \mathcal{Y} and \mathcal{Z} , one can easily construct a weaker set of linear conditions by globally refining the partitions \mathcal{X} , \mathcal{Y} and \mathcal{Z} , e.g., with a dyadic split. The number of conditions will grow exponentially with the number of refinement steps.

Remark 4. In weakening convexity conditions with the aim of improving the fit, the number of convexity conditions obtained can be reduced by only weakening the set of conditions locally. Let c be the solution of Problem 3.2 for a given set of convexity conditions generated by the knot sets \mathcal{X} , \mathcal{Y} and \mathcal{Z} . A natural adaptive strategy is to refine only the intervals in \mathcal{X} , \mathcal{Y} and \mathcal{Z} that correspond to active convexity conditions, i.e., the rows \tilde{C} in matrix C where $\tilde{C}c = 0$.

Remark 5. When solving the new optimization problem with weaker convexity constraints (see Remarks 3 and 4), it is recommended to use the solution of the previous problem as the initial solution for the new problem.

Remark 6. An adaptive strategy similar to the one in Remark 4 can be applied in case the original problem is not feasible. It consists of two steps. In the first step, we construct a spline that satisfies the convexity conditions as well as possible. For instance, one can solve the following linear programming problem: find c that minimizes the value of the (nonnegative) parameter γ subject to the linear side

conditions $Ac = 0$, $Bc = z$, and the constraints $Cc \geq -\gamma$. In the second step, we refine the intervals in \mathcal{X} , \mathcal{Y} and \mathcal{Z} that correspond to violated convexity conditions, i.e., the rows \tilde{C} in matrix C where $\tilde{C}c > 0$. Note that this procedure does not guarantee that the new problem is feasible, it only increases the chance of finding a feasible convex solution.

Remark 7. In [23] an approach based on blossoming is presented to construct sets of linear convexity conditions for tensor-product splines. The construction there can also be applied in case of triangular Bernstein–Bézier polynomials. Our approach, however, is more natural in the triangular case. It naturally subsumes the particular sets of linear convexity conditions found in the literature with simple choices of knot sets \mathcal{X} , \mathcal{Y} and \mathcal{Z} . In addition, it allows us to give a geometric interpretation, and we can easily construct sets of linear conditions that are symmetric.

Remark 8. In [6] another technique is described to generate weaker sufficient convexity conditions. It is based on the construction of polygons inscribed in a particular parabola. This paper also mentions the three univariate polynomials in (2.4).

Remark 9. The method in Sect. 3.2 works with any stable spline basis. Some other appropriate bases for the Powell-Sabin macro-element space can be found in [13,14].

Remark 10. The approach described in Sect. 3 can of course be used with any of the other macro-element spaces described in [28], including those based on quadrangulations instead of triangulations. For an example of the use of a quadrangulation-based macro-element space, see [27].

Remark 11. Convexity conditions for parametric triangular Bézier surfaces can be found in [2,10,40,41]. Conditions for parametric tensor-product surfaces are given in [24,25].

Remark 12. Convexity of multivariate Bernstein–Bézier polynomials has been addressed in [12,19,22,26,35].

Acknowledgments. This work was done while Hendrik Speleers was a postdoctoral fellow at Vanderbilt University with support from the Research Foundation Flanders (Belgium).

References

1. Bangert, C. and H. Prautzsch, A geometric criterion for the convexity of Powell–Sabin interpolants and its multivariate generalization, *Comput. Aided Geom. Design* **16** (1999), 529–538.
2. Cao, Y. and X. J. Hua, The convexity of quadratic parametric triangular Bernstein–Bézier surfaces, *Comput. Aided Geom. Design* **8** (1991), 1–6.

3. Carnicer, J. M., Multivariate convexity preserving interpolation by smooth functions, *Advances in Comp. Math.* **3** (1995), 395–404.
4. Carnicer, J. M. and W. Dahmen, Convexity preserving interpolation and Powell–Sabin elements, *Comput. Aided Geom. Design* **9** (1992), 279–289.
5. Carnicer, J. M. and M. S. Floater, Piecewise linear interpolants to Lagrange and Hermite convex scattered data, *Numer. Algorithms* **13** (1996), 345–364.
6. Carnicer, J. M., M. S. Floater, and J. M. Peña, Linear convexity conditions for rectangular and triangular Bernstein–Bézier surfaces, *Comput. Aided Geom. Design* **15** (1997), 27–38.
7. Carnicer, J. M., T. N. T. Goodman, and J. M. Peña, Convexity preserving scattered data interpolation using Powell–Sabin elements, *Comput. Aided Geom. Design* **26** (2009), 779–796.
8. Chang, G. and P. J. Davis, The convexity of Bernstein polynomials over triangles, *J. Approx. Theory* **40** (1984), 11–28.
9. Chang, G. and Y. Y. Feng, An improved condition for the convexity of Bernstein–Bézier surfaces over triangles, *Comput. Aided Geom. Design* **1** (1984), 279–283.
10. Cheng, Z. and C. K. Chui, Convexity of parametric Bézier surfaces in terms of Gaussian curvature signatures, *Advances in Comp. Math.* **2** (1994), 437–459.
11. Dahmen, W., Convexity and Bernstein–Bézier polynomials, in *Curves and Surfaces*, P.-J. Laurent, A. LeMéhauté, and L. L. Schumaker (eds), Academic Press, New York, 1991, 107–134.
12. Dahmen, W. and C. A. Micchelli, Convexity of multivariate Bernstein polynomials and box spline surfaces, *Stud. Sci. Math. Hung.* **23** (1988), 265–287.
13. Dierckx, P., S. Van Leemput, and T. Vermeire, Algorithms for surface fitting using Powell-Sabin splines, *IMA J. Numer. Anal.* **12** (1992), 271–299.
14. Dierckx, P., On calculating normalized Powell-Sabin B-splines, *Comput. Aided Geom. Design* **15** (1997), 61–78.
15. Feng, Y. Y., F. L. Chen, and H. L. Zhou, The invariance of weak convexity conditions of B-nets with respect to subdivision, *Comput. Aided Geom. Design* **11** (1994), 97–107.
16. Floater, M. S., A weak condition for the convexity of tensor-product Bézier and B-spline surfaces, *Advances in Comp. Math.* **2** (1994), 67–80.
17. Floater, M. S., A counterexample to a theorem about the convexity of Powell–Sabin elements, *Comput. Aided Geom. Design* **14** (1997), 383–385.
18. Goodman, T. N. T., Convexity of Bézier nets on triangulations, *Comput. Aided Geom. Design* **8** (1991), 175–180.
19. Goodman, T. N. T. and J. Peters, Bézier nets, convexity and subdivision on higher-dimensional simplices, *Comput. Aided Geom. Design* **12** (1995), 53–65.
20. Grandine, T. A., On convexity of piecewise polynomial functions on triangulations, *Comput. Aided Geom. Design* **6** (1989), 181–187.

21. Gregory, J. A. and J. W. Zhou, Convexity of Bézier nets on sub-triangles, *Comput. Aided Geom. Design* **8** (1991), 207–211.
22. He, T. X., Shape criteria of Bernstein–Bézier polynomials over simplexes, *Comput. Math. Appl.* **30** (1995), 317–333.
23. Jüttler, B., Surface fitting using convex tensor-product splines, *J. Comput. Appl. Math.* **84** (1997), 23–44.
24. Jüttler, B., Convex surface fitting with parametric Bézier surfaces, in *Mathematical Methods for Curves and Surfaces II*, M. Dæhlen, T. Lyche, and L. L. Schumaker (eds), Vanderbilt University Press, Nashville, 1998, 263–270.
25. Koras, G. D. and P. D. Kaklis, Convexity conditions for parametric tensor-product B-spline surfaces, *Advances in Comp. Math.* **10** (1999), 291–309.
26. Lai, M. J., Some sufficient conditions for convexity of multivariate Bernstein–Bézier polynomials and box spline surfaces, *Stud. Sci. Math. Hung.* **28** (1993), 363–374.
27. Lai, M. J., Convex preserving scattered data interpolation using bivariate C^1 cubic splines, *J. Comput. Appl. Math.* **119** (2000), 249–258.
28. Lai, M. J. and L. L. Schumaker, *Spline Functions on Triangulations*, Cambridge University Press, Cambridge, 2007.
29. Leung, N. K. and R. J. Renka, C^1 convexity-preserving interpolation of scattered data, *SIAM J. Scient. Computing* **20** (1999), 1732–1752.
30. Li, A., Convexity preserving interpolation, *Comput. Aided Geom. Design* **16** (1999), 127–147.
31. Liu, Z., J. Tan, X. Chen, and L. Zhang, The conditions of convexity for Bernstein–Bézier surfaces over triangles, manuscript, 2010.
32. Lorente-Pardo, J., P. Sablonnière, and M. C. Serrano-Pérez, On the convexity of Powell–Sabin finite elements, in *Advances in Computational Mathematics*, Z. Chen, Y. Li, C. Micchelli, and Y. Xu (eds), Marcel Dekker, New York, 1998, 395–404.
33. Lorente-Pardo, J., P. Sablonnière, and M. C. Serrano-Pérez, On the convexity of C^1 surfaces associated with some quadrilateral finite elements, *Advances in Comp. Math.* **13** (2000), 271–292.
34. Lorente-Pardo, J., P. Sablonnière, and M. C. Serrano-Pérez, On the convexity of Bézier nets of quadratic Powell–Sabin splines on 12-fold refined triangulations, *J. Comput. Appl. Math.* **115** (2000), 383–396.
35. Sauer, T., Multivariate Bernstein polynomials and convexity, *Comput. Aided Geom. Design* **8** (1991), 465–478.
36. Schumaker, L. L., Computing bivariate splines in scattered data fitting and the finite-element method, *Numerical Algorithms* **48** (2008), 237–260.
37. Seidel, H. P., An introduction to polar forms, *IEEE Comp. Graph. Appl.* **13** (1993), 38–46.

38. Wang, Z. B. and Q. M. Liu, An improved condition for the convexity and positivity of Bernstein–Bézier surfaces over triangles, *Comput. Aided Geom. Design* **5** (1988), 269–275.
39. Willemans, K. and P. Dierckx, Surface fitting using convex Powell–Sabin splines, *J. Comput. Appl. Math.* **56** (1994), 263–282.
40. Zheng, J. J., The convexity of parametric Bézier triangular patches of degree 2, *Comput. Aided Geom. Design* **10** (1993), 521–530.
41. Zhou, C. Z., On the convexity of parametric Bézier triangular surfaces, *Comput. Aided Geom. Design* **7** (1990), 459–463.