Convex Surface Fitting with Parametric Bézier Surfaces

Bert Jüttler

Abstract. We present a method for approximating scattered data with a convex parametric Bézier surface patch. In the first step we construct a reference surface which roughly specifies the expected shape of the approximating surface. Based on this reference surface we generate linearized convexity conditions. The approximating surface is then found by solving a quadratic programming problem.

§1. Introduction

Convexity conditions for bivariate piecewise polynomial functions have been studied in a number of publications, see the survey articles by Goodman [7] and Dahmen [2]. In the case of bivariate polynomials in Bernstein–Bézier representation with respect to a basis triangle, Chang and Davis [1] observed that convexity of the control net implies convexity of the polynomial. This result was later generalized to the multivariate case by Dahmen and Micchelli [3]. For tensor–product spline functions, weak convexity conditions were developed by Floater [5]. These conditions lead to a system of quadratic inequalities for the spline coefficients, where each inequality involves only relatively few coefficients.

Willemans and Dierckx [13] developed a method for convex surface fitting with Powell–Sabin spline functions. Based on the quadratic convexity conditions by Chang and Feng (see [2]) they were able to formulate this task as a quadratic optimization problem with quadratic inequality constraints.

In the case of *parametric* surface patches, only a few related results seem to exist. A very strong sufficient convexity condition for parametric tensor–product Bézier surfaces has been formulated by Schelske in 1984, see [8]. This condition is fulfilled only by convex translational surfaces. Zhou [14] derived convexity conditions for parametric triangular Bézier surfaces. These conditions lead to a system of inequalities whose left–hand sides are polynomials

Mathematical Methods for Curves and Surfaces II
Morten Dæhlen, Tom Lyche, Larry L. Schumaker (eds.), pp. 1–8.

ISBN 1-xxxxx-xxx-x.

of degree 6 in the components of the control points. Recently, similar conditions for parametric tensor–product surfaces were developed by Koras and Kaklis [12]. An approximate method for removing shape flaws from tensor–product B–spline surfaces is described in [11]. This method is based on local modifications of the control net.

We present a new method for convex surface fitting with parametric tensor–product Bézier surfaces. Based on linearized convexity conditions we formulate this task as a quadratic optimization problem with linear inequality constraints. This problem can be solved with the help of standard algorithms from optimization theory. The method is illustrated by an example.

§2. Outline of the Method

We present a method for solving the following approximation problem. A cloud of data $\boldsymbol{p}_i \in \mathbb{R}^3$ $(i = 0, \ldots, P)$ is given. In addition to the data, we assume that associated parameter values $(u_i, v_i) \in [0,1]^2$ are given. If these parameters are unknown, then they can be estimated from the data, e.g. by projecting them into a suitably chosen plane. A more sophisticated scheme for assigning the parameter values has been developed by Floater [6].

The given data are to be approximated by the parametric tensor–product Bézier surface patch (see [8])

$$\boldsymbol{x}(u,v) = \sum_{r=0}^{m}\sum_{s=0}^{n} B_r^m(u)\, B_s^n(v)\, \boldsymbol{b}_{r,s}, \quad (u,v) \in [0,1]^2, \tag{1}$$

with the unknown control points $\boldsymbol{b}_{r,s} = (\ b_{r,s,1}\ \ b_{r,s,2}\ \ b_{r,s,3}\ )^\top \in \mathbb{R}^3$ and with the well–known Bernstein polynomials $B_q^p(t) = \binom{p}{q} t^q (1-t)^{p-q}$. The control points $\boldsymbol{b}_{r,s}$ are found by the following procedure.

1) Find a *reference surface.* This surface is used in order to specify the expected shape of the approximating surface (1). Its construction is described in Section 3.
2) Generate *linearized convexity conditions.* Based on the reference surface we generate a system of linear inequalities for the components of the control points $\boldsymbol{b}_{r,s}$ which guarantee the convexity of the surface patch (1), see Section 4.
3) Compute the control points $\boldsymbol{b}_{r,s} \in \mathbb{R}^3$. The control points are found by solving a quadratic programming problem as outlined in Section 5.

This procedure can be iterated several times; one may use the first result as a new reference surface. The new reference surface leads to linear convexity constraints which are better suited for approximating the given data in the third step.

In addition, one may use the idea of *parameter correction* in each cycle: the parameter values (u_i, v_i) are replaced by new values $(\bar{u}_i, \bar{v}_i) \in [0,1]^2$ such that the new error vectors $\boldsymbol{x}(\bar{u}_i, \bar{v}_i) - \boldsymbol{p}_i$ are perpendicular to the surface, see [8] for details. By using the new parameters we can improve the result of the approximation procedure.

§3. The Reference Surface

An approximating parametric Bézier surface for scattered data can be found by minimizing the least–squares sum

$$L = \sum_{i=0}^{P} \|\boldsymbol{x}(u_i, v_i) - \boldsymbol{p}_i\|^2. \tag{2}$$

The minimum can easily be computed by solving the system of normal equations, see [8]. In general, however, minimizing the sum (2) leads to a non–convex surface.

In order to find a suitable convex reference surface, we modify the least squares sum by adding a *tension term*, $F = L + w\,T$, with

$$T = \int_0^1 \int_0^1 \|\boldsymbol{x}_{uuu}\|^2 + \|\boldsymbol{x}_{vvv}\|^2 + \|\boldsymbol{x}_{uv}\|^2 \,\mathrm{d}u\,\mathrm{d}v. \tag{3}$$

The subscripts denote the partial derivatives of the surface $\boldsymbol{x}(u, v)$. The tension term is introduced in order to increase the "stiffness" of the reference surface. Its influence is controlled by the weight w. The value of the tension term is zero if and only if the surface is a biquadratic translational surface, i.e. the parameter lines of both systems u = constant and v = constant are translated copies of the curves $\boldsymbol{x}(u, 0)$ and $\boldsymbol{x}(0, v)$.

Figure 1 shows a biquadratic translational surface and its Bézier control net (dashed lines). Three samples of the congruent parameter lines u = constant have been drawn as solid black curves. All faces of the control net are parallelograms; this property characterizes translational Bézier surfaces.

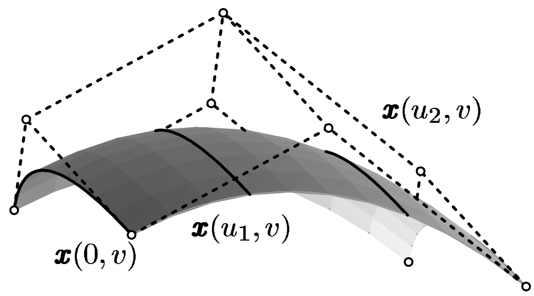

Fig. 1. A translational surface.

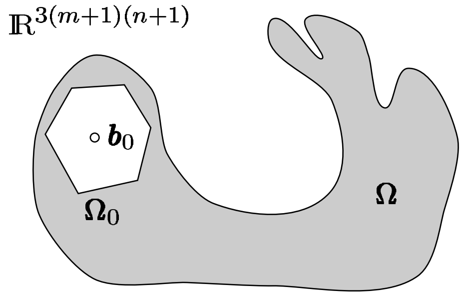

Fig. 2. Linearized convexity conditions.

If the weight w is increased, then the surface which minimizes the modified functional $F + w\,T$ converges to a biquadratic translational surface. Note that this limit translational surface is not guaranteed to be convex. In all practical examples, however, one can expect that this surface is convex, provided that the data stem from a roughly convex surface. Otherwise the given scattered data is unsuitable for a convex approximation, see also the comments at the end of this section.

The above tension term (3) is just one possible choice. One may choose any other functional which increases the stiffness of the resulting surface. The kernel of the tension term, however, should not only contain the linear or bilinear Bézier surfaces (this would happen if we choose the tension term as an integral of squared second derivative vectors), as these surfaces can never be strongly convex. (A surface is said to be strongly convex if it is convex and the Gaussian curvature is positive everywhere.) The tension term (3) seems to be a reasonable choice as its kernel (biquadratic translational surfaces) offers enough degrees of freedom for providing a reasonable approximation.

The control points of the reference surface are found by solving the system of normal equations

$$\frac{\partial F}{\partial b_{r,s,k}} = 0 \qquad (r = 0,\dots,m;\ s = 0,\dots,n;\ k = 1,2,3). \tag{4}$$

They form a system of linear equations for the components of the control points.

Still, the weight w has to be chosen. On the one hand, the influence of the tension term should so great that solving (4) leads to a convex reference surface. On the other hand, the weight w should be as small as possible in order to get linearized convexity conditions which are well adapted to the specific data.

We compute an appropriate value for the weight w with the help of simple binary search. Let $w = 1.0$ be the initial weight and compute the resulting reference surface. If this surface is strongly convex but it does not fit very well to the data, then we decrease the weight, $w_{\text{new}} = \frac{1}{2} w_{\text{old}}$, until the approximation gets good enough. If the reference surface becomes non–convex, however, than we go back to the previously used value of w. It may happen that the reference surface never gets non–convex. In this case the reference surface is already the final approximating surface; no convexity constraints are required.

If the initial reference surface is non–convex, then we increase the weight, $w_{\text{new}} = 2w_{\text{old}}$, until a strongly convex surface is obtained. If this procedure fails (this may happen if the data stem from a non–convex surface), then the data is probably unsuitable for convex approximation. In this case one may use the best fitting plane as a convex approximation.

§4. Linearized Convexity Conditions

With the help of the reference surface it is now possible to find linear constraints which guarantee the convexity of the surface (1). The situation is illustrated by the schematic Figure 2. Each surface patch (1) is associated with the point $\boldsymbol{b} \in \mathbb{R}^{3(m+1)(n+1)}$ with the coordinates

$$\boldsymbol{b} = (\ \boldsymbol{b}_{0,0}^\top\ \ \boldsymbol{b}_{0,1}^\top\ \ \boldsymbol{b}_{0,n}^\top\ \ \boldsymbol{b}_{1,0}^\top\ \ \dots\ \ \boldsymbol{b}_{m,n}^\top\ )^\top. \tag{5}$$

The convex surfaces correspond to a certain subset Ω of this space. The reference surface is associated with an interior point $\boldsymbol{b}_0$ of this subset. By generating linear sufficient convexity conditions we construct a circumscribed polyhedron $\Omega_0 \subset \Omega$ for the point $\boldsymbol{b}_0$.

The linearized convexity conditions are found with the help of the following procedure.

1) Use the reference surface for specifying bounding polyhedral cones for the first derivative vectors of the approximating surface. Let $(\vec{\boldsymbol{r}}_i)_{i=1,\ldots,R}$ and $(\vec{\boldsymbol{s}}_j)_{j=1,\ldots,S}$ be the spanning vectors of these cones. We generate linear inequalities for the unknown control points $\boldsymbol{b}_{r,s}$ which guarantee that the first derivative vectors are contained within these cones.
2) Find a bounding polyhedral cone for the cross product $\boldsymbol{x}_u \times \boldsymbol{x}_v$. Let $(\vec{\boldsymbol{t}}_k)_{k=1,\ldots,T}$ be the spanning vectors of this cone. They are a certain subset of $\{\vec{\boldsymbol{r}}_i \times \vec{\boldsymbol{s}}_j \mid i = 1, \ldots, R;\ j = 1, \ldots, S\}$. Then,
$$\boldsymbol{x}_u(u,v) \times \boldsymbol{x}_v(u,v) = \sum_{k=1}^{T} \tau_k(u,v)\,\vec{\boldsymbol{t}}_k, \quad (u,v) \in [0,1]^2, \tag{6}$$
holds with some non–negative functions $\tau_k(u,v)$.
3) Generate linear constraints which guarantee that the second fundamental form of the surface is either non–negative or non–positive definite for all $(u,v) \in [0,1]^2$. As a sufficient condition, we guarantee that the T matrices
$$H_k = \begin{pmatrix} \boldsymbol{x}_{uu} \cdot \vec{\boldsymbol{t}}_k & \boldsymbol{x}_{uv} \cdot \vec{\boldsymbol{t}}_k \\ \boldsymbol{x}_{uv} \cdot \vec{\boldsymbol{t}}_k & \boldsymbol{x}_{vv} \cdot \vec{\boldsymbol{t}}_k \end{pmatrix} \qquad (k = 1, \ldots, T) \tag{7}$$
are either all non–negative or non–positive definite. This is sufficient for (local) convexity as the second fundamental form
$$\frac{1}{\|\boldsymbol{x}_u \times \boldsymbol{x}_v\|} \begin{pmatrix} \boldsymbol{x}_{uu} \cdot (\boldsymbol{x}_u \times \boldsymbol{x}_v) & \boldsymbol{x}_{uv} \cdot (\boldsymbol{x}_u \times \boldsymbol{x}_v) \\ \boldsymbol{x}_{uv} \cdot (\boldsymbol{x}_u \times \boldsymbol{x}_v) & \boldsymbol{x}_{vv} \cdot (\boldsymbol{x}_u \times \boldsymbol{x}_v) \end{pmatrix} \tag{8}$$
is a non–negative linear combination of these matrices, cf. (6). The matrices (7) are non–negative (non–positive) definite if and only if the $2T$ quadratic polynomials
$$\begin{pmatrix} \xi & \pm(1-\xi) \end{pmatrix} H_k \begin{pmatrix} \xi \\ \pm(1-\xi) \end{pmatrix} \qquad (k = 1, \ldots, T) \tag{9}$$
are non–negative (non–positive) for $\xi \in [0,1]$. These polynomials are obtained if we restrict the quadratic form $\boldsymbol{z}^\top H_k \boldsymbol{z}$ ($\boldsymbol{z} \in \mathbb{R}^2$) to the linearly parameterized edges of the square with the vertices $(\pm 1, 0)$ and $(0, \pm 1)$. Their coefficients depend linearly on the control points $\boldsymbol{b}_{r,s}$. Based on this fact we are able to generate linear inequalities for the control points which imply convexity.

This procedure leads to a system $\mathcal{I}$ of linear inequalities for the components of the control points $\boldsymbol{b}_{r,s}$. It may be necessary to subdivide the reference surface in order to find a suitable bounding cone for the cross product $\boldsymbol{x}_u \times \boldsymbol{x}_v$

(for instance, if the reference surface cannot be considered as the graph of a function). Due to space limitations we cannot present any details of the procedure. See [9] for more information. As observed in [9], the linearized convexity conditions can be adapted to any strongly convex reference surface. This is also obvious from Figure 2; each inner point of Ω possesses a circumscribed polyhedron.

§5. Quadratic Programming

The control points $\boldsymbol{b}_{r,s}$ of the approximating surface patch (1) are found by minimizing the least–squares sum (2) subject to the linearized convexity conditions $\mathcal{I}$. This is a quadratic programming (qp) problem; a quadratic objective function is to be minimized subject to linear equality and inequality constraints.

A number of fast and efficient solvers for solving problems of this type have been developed in optimization theory. For instance, an approximate solution can be found with the help of the LOQO package by Vanderbei (available from `http://www.princeton.edu/~rvdb/`). Alternatively one may use an active set strategy (which works similarly to the simplex algorithm) as described in the textbook by Fletcher [4]. The example in the next section has been computed by using LOQO.

§6. An Example

We sampled 51 points from an ellipsoid and perturbed them by using random numbers. These data are to be approximated by a bicubic Bézier surface patch (1). Figure 3 compares the unconstrained approximating surface (a) and the convex approximating surface (b). Whereas the unconstrained surface possesses a huge number of oscillations, the constrained approximation possesses a convex shape. The least squares sums of both surfaces are equal to 0.153 and 0.244, respectively (reference surface: 0.422). The convex approximation has been obtained after 2 iterations (with the use of parameter correction) of the procedure from Section 2. In the first (second) iteration we had to solve a qp problem with 1761 (9674) linear inequalities for 48 (48) unknowns. In addition to the surfaces, both Figures 3 a,b show the Bézier control nets and some level curves $x_3 =$ constant. The ellipses in Figure 3b visualize the curvature distribution. Their principal axes are the principal curvature directions, the diameters are proportional to the principal curvatures.

The two plots in Figure 4 visualize the distribution of the Gaussian curvature $K(u,v)$ for both approximating surfaces. The Gaussian curvature of the unconstrained approximation indicates huge hyperbolic surface regions, $-63.33 \leq K_{\mathrm{unc}}(u,v) \leq 12.44$. In contrast to this, the convex surface has non–negative Gaussian curvature values, $0.01 \leq K_{\mathrm{conv}}(u,v) \leq 2.32$.

§7. Final Remark

In this article we described a method for convexity–preserving surface fitting with parametric tensor–product Bézier surface patches. An analogous method

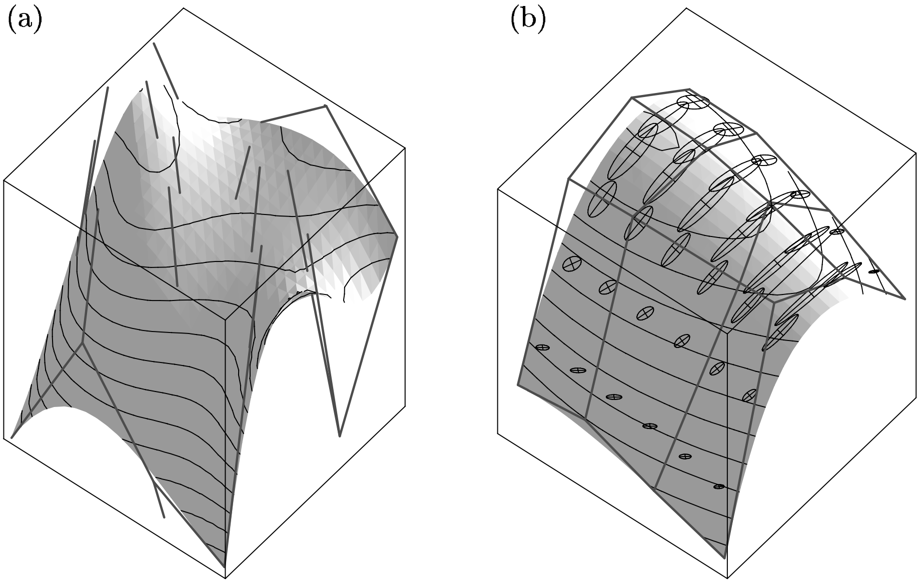

Fig. 3. The unconstrained approximation (a) and the convex approximation (b).

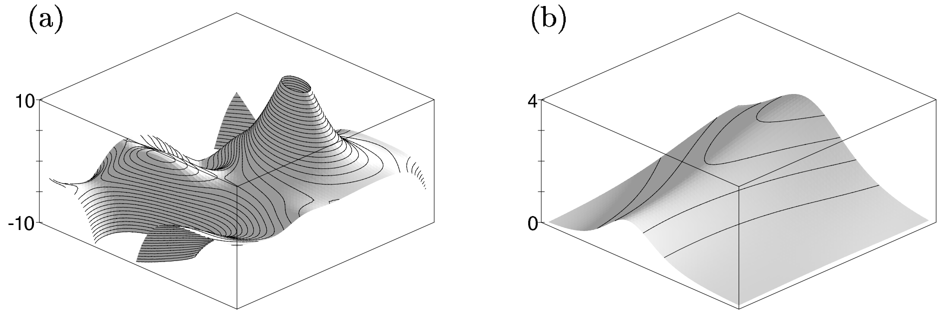

Fig. 4. The Gaussian curvature plots of the surfaces shown in Fig. 3.

can be formulated for approximating scattered data with a tensor–product spline *function* subject to piecewise convexity/concavity constraints, see [10]. Similar to the parametric case, the convexity of the approximating surface can be guaranteed by linear inequalities for the spline coefficients. In the functional case, however, the approximation scheme is much simpler as no reference surface is required. The third step of the algorithm from Section 4 can be applied directly to the Hessian matrix of the spline surface. Moreover, the linear inequalities can be shown to be *asymptotically necessary:* if the number of inequalities is increased in a suitable manner, the feasible set of spline functions approximates the set of all convex spline functions as accurately as desired. This property can be achieved as the convex spline functions form a convex set.

The set Ω of all convex *parametric* surface patches, by contrast, is non–convex, see Section 4. Hence, no asymptotically necessary linearized convexity conditions for parametric Bézier surfaces can be found. As outlined in Section 4, is possible to find linear sufficient convexity conditions for each strongly convex surface, i.e. for each point from the interior of Ω.

References

1. Chang, G.-Z. and P. J. Davis, The convexity of Bernstein polynomials over triangles, J. Approx. Theory **40** (1984), 11–28.
2. Dahmen, W., Convexity and Bernstein–Bézier polynomials, in *Curves and Surfaces,* P.-J. Laurent, A. Le Méhauté, and L. L. Schumaker (eds), Academic Press, New York, 1991, 107–134.
3. Dahmen, W. and C. A. Micchelli, Convexity of multivariate Bernstein polynomials and box spline surfaces, Stud. Sci. Math. Hung. **23** (1988), 265–287.
4. Fletcher, R., *Practical Methods of Optimization,* Second Edition, John Wiley & Sons, Chichester, 1987.
5. 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.
6. Floater, M. S., Parametrization and smooth approximation of surface triangulations, Comput. Aided Geom. Design **14** (1997), 231–250.
7. Goodman, T. N. T., Shape preserving representations, in *Mathematical Methods in Computer Aided Geometric Design,* T. Lyche and L. Schumaker (eds), Academic Press, New York, 1989, 333–351.
8. Hoschek, J. and D. Lasser, *Fundamentals of Computer Aided Geometric Design,* A. K. Peters, Boston MA, 1993.
9. Jüttler, B., Linear convexity conditions for parametric tensor–product Bézier surface patches, in *Mathematics of Surfaces VII,* T. N. T. Goodman (ed), Information Geometers, Winchester, 1997, in press.
10. Jüttler, B., Surface fitting using convex tensor–product splines, J. Comput. Appl. Math. **84** (1997), 23–44.
11. Kaklis, P. D. and G. D. Koras, A quadratic–programming method for removing shape-failures from tensor-product B-spline surfaces, NTU Athens, Dept. of Naval Architecture, Report TR–SDL–CAGD–1/97.
12. Koras, G. D. and P. D. Kaklis, Convexity conditions for parametric tensor–product B-spline surfaces, NTU Athens (Greece), Dept. of Naval Architecture, manuscript, 1997.
13. Willemans, K. and P. Dierckx, Surface fitting using convex Powell–Sabin Splines, J. Comput. Appl. Math. **56** (1994), 263–282.
14. Zhou, C.-Z., On the convexity of parametric Bézier triangular surfaces, Comput. Aided Geom. Design **7** (1990), 459–463.

Bert Jüttler
Technische Universität Darmstadt
Fachbereich Mathematik
Schloßgartenstraße 7
64289 Darmstadt, GERMANY
`juettler@mathematik.tu-darmstadt.de`