L_1 Spline Methods for Scattered Data Interpolation and Approximation

Ming-Jun Lai[1] and Paul Wenston [2]

Abstract. We generalize the L_1 spline methods proposed in [Lavery'00, 01] for scattered data interpolation and fitting using bivariate spline spaces of any degree d and any smoothness r (of course, $r < d$) over any triangulation. Some numerical experiments are presented to illustrate the better performance of the L_1 spline methods as compared to the minimal energy method. We include some extensions for dealing with other surface design problems.

§1. Introduction

Given $\{(x_i, y_i, f_i), i = 1, \cdots, V\}$, we wish to find a smooth surface which interpolates or approximates the given data and resembles the data as closely as possible. We will use bivariate splines. Usually, the well-known minimal energy method is used to construct interpolatory smooth surfaces (cf. e.g., [Fasshauer and Schumaker'96] for a survey of the minimal energy method). However, it may not give a desirable surface. See, e.g., Fig. 4.3. It sometimes produces too many oscillations. Recently, the research work in [Lavery'00] and [Lavery'01] suggests that the L_2 norms in the usual minimal quadratic energy

$$Q(s) = \sum_{t\in\triangle} \int_t \left(\left| \frac{\partial^2}{\partial x^2} s \right|^2 + 2 \left| \frac{\partial^2}{\partial x \partial y} s \right|^2 + \left| \frac{\partial^2}{\partial y^2} s \right|^2 \right) dxdy. \tag{1.1}$$

can be replaced by L_1 norms. In particular the L_1 spline method proposed in [Lavery'01] can be described as follows: Let S_3^1 be the space of Sibson elements (assuming the data locations are over a rectangular grid). Find $s_f \in S_3^1$ such that

$$E(s_f) = \min\{E(s), s \in S_3^1, s(x_i, y_i) = f_i, i = 1, \cdots, V\}, \tag{1.2}$$

where E is the L_1 norm of the second order derivatives of spline functions, i.e.

$$E(s) = \sum_{t\in\triangle} \int_t \left(\left| \frac{\partial^2}{\partial x^2} s \right| + 2 \left| \frac{\partial^2}{\partial x \partial y} s \right| + \left| \frac{\partial^2}{\partial y^2} s \right| \right) dxdy. \tag{1.3}$$

[1] Department of Mathematics, University of Georgia, Athens Georgia 30602, mjlai@math.uga.edu. Supported by the National Science Foundation under grant DMS-9870187.

[2] Department of Mathematics, University of Georgia, Athens, GA 30602 paul@math.uga.edu.

(Other similar L_1 norms were also considered in [Lavery'01].) Also, in their numerical experiments, the minimization was carried out by adding "regularization" terms. One of the regularization terms is

$$\epsilon \sum_{i=1}^{N} |\frac{\partial s}{\partial x}(x_i, y_i)| + |\frac{\partial s}{\partial y}(x_i, y_i)|.$$

(See [Lavery'01] for details.) The curves and surfaces produced by the L_1 spline methods in their papers have less wrinkles than those obtained by the minimal energy spline interpolation method.

Following the ideas in [Lavery'00] and [Lavery'01], we would like to extend their work by considering scattered data sets, using more general spline spaces, and presenting a detailed mathematical analysis for the L_1 spline methods. (We thank Dr. John Lavery for providing us with some references.) One of the advantages of our extension is that we are able to deal with arbitrary scattered data.

Let $\triangle$ be a triangulation of the data locations $\{(x_i, y_i), i = 1, \cdots, V\}$ and

$$S_d^r(\triangle) = \{s \in C^r(\Omega) : s|_t \in \mathbb{P}_d, \forall t \in \triangle\}$$

be the spline space of degree d and smoothness r with $r < d$, where Ω is the union of all triangles in $\triangle$ and $\mathbb{P}_d$ stands for the space of all polynomials of degree $\leq d$. The L_1 spline method can be in general defined by finding $s_f \in S_d^r(\triangle)$ such that

$$E(s_f) = \min\{E(s), s \in S_d^r(\triangle), s(x_i, y_i) = f_i, i = 1, \cdots, V\}. \tag{1.4}$$

We will call this the L_1 spline interpolation method. Similarly, we may also consider finding $s_f \in S_d^r(\triangle)$ such that

$$E(s_f) + \alpha \ell(s_f) = \min\{E(s) + \alpha \ell(s_f), s \in S_d^r(\triangle)\}, \tag{1.5}$$

where $\ell(s) = \sum_{i=1}^{V} |s(x_i, y_i) - f_i|$ and $\alpha > 0$ is a parameter. Finding s_f which minimizes (1.5) is called the L_1 spline smoothing method and finding s_f which minimizes

$$Q(s_f) + \alpha \ell_2(s_f) = \min\{Q(s) + \alpha \ell_2(s_f), s \in S_d^r(\triangle)\}, \tag{1.6}$$

where $\ell_2(s) = \sum_{i=1}^{V} |s(x_i, y_i) - f_i|^2$, is called the penalized least squares method.

In §2, we shall prove the existence of such interpolatory or smoothing L_1 spline functions for any given data. Certainly, we need to assume that the spline space $S_d^r(\triangle)$ has an interpolation spline function first before we apply the L_1 spline interpolation method. However, for the L_1 spline smoothing method, we do not need the assumption. In §3, we shall discuss how to compute approximates of such

minimizers by discretizing the integrals and converting the minimization problems into linear programming problems. We will also discuss how to use the well-known Karmarkar algorithm to compute a solution of these linear programming problems. Computational evidence will be presented in §4 to illustrate the advantage of the L_1 spline methods over the minimal energy method. We emphasize here that based on our numerical experiments, the minimal energy method often produces good interpolatory surfaces. However, the L_1 spline methods indeed provide an alternative method for surface designers. Finally, in §5, we discuss some extensions of the L_1 spline methods for other surface design problems.

§2. The L_1 Spline Interpolation and Smoothing Methods

In this section, we first discuss the L_1 spline interpolation method. The comparison of the L_1 spline interpolation method and the minimal energy method for data fitting will be given in §4. Let

$$\Lambda(f) := \{s \in S_d^r(\triangle) : s(x_\ell, y_\ell) = f_\ell, \ell = 1, \cdots, V\}. \tag{2.1}$$

We note that $\Lambda(f)$ will be nonempty whenever $d \geq 3r + 2$. However, $d \geq 3r + 2$ is not necessary. In our experiments, $\Lambda(f)$ is often nonempty for $d = 3$ and $r = 1$.

Theorem 2.1. *Suppose that $\Lambda(f)$ is not empty. Then there exists an $s_f \in S_d^r(\triangle)$ minimizing (1.4).*

Proof: Since $\Lambda(f)$ is not empty, let $s_0 \in \Lambda(f)$. Let

$$K := \{s \in \Lambda(f) : E(s) \leq E(s_0)\}.$$

Note that on any triangle $t \in \triangle$ the spline s can be expressed in B-form by

$$s(x, y) = \sum_{i+j+k=d} c_{i,j,k}^t B_{i,j,k}^t(x, y),$$

(cf. [Farin'86]). Let $\mathbf{c} = (c_{i,j,k}^t, i+j+k = d, t \in \triangle)$ denote the coefficient vector of s. Then the smoothness condition that $s \in C^r(\Omega)$ is equivalent to the condition that $\mathbf{c}$ belongs to the nullspace of a matrix H constructed from the smoothness conditions, which are linear equations relating B-form coefficients in adjacent triangles. (For the smoothness conditions across a common edge of two triangles, see [Farin'86].) Thus, K is a set in the nullspace of H which in turn is a subspace of the finite dimensional space $\mathbb{R}^{\hat{d} \times T}$ with $\hat{d} = (d + 1)(d + 2)/2$ and T being the number of triangles in $\triangle$.

We claim that K is a bounded set. Indeed, we first note that

$$\int_t |D_x^2 s(x, y)| dxdy \leq E(s_0)$$

for any triangle $t \in \triangle$. Since any norm in the finite dimension space $D_x^2 \mathbb{P}_d$ is equivalent, we have

$$\max_{(x,y)\in t} |D_x^2 s(x,y)| \leq CE(s_0)$$

for some positive constant C independent of s. Similarly, defining

$$|s|_{2,\infty,t} = \max_{(x,y)\in t} \left(|D_x^2 s(x,y)| + 2|D_x D_y s(x,y)| + |D_y^2 s(x,y)| \right),$$

we have

$$|s|_{2,\infty,t} \leq 4CE(s_0). \tag{2.2}$$

We need to show that $s(x,y)$ is uniformly bounded. To this end, we show that $s(x,y)|_t$ is bounded for any t. For convenience, we denote $f(v_i)$ by f_i for all vertices $v_i, i = 1, \cdots, V$ of the triangulation $\triangle$. Fix a point $v = (x,y) \in t = \langle v_1, v_2, v_3 \rangle$. By the Taylor expansion of s at v,

$$f(v_i) = s(v_i) = s(v) + \nabla s(v) \cdot (v_i - v) + O(|s|_{2,\infty,t}|t|^2) \tag{2.3}$$

for $i = 1, 2, 3$ with $|t|$ the diameter of triangle t. It follows that

$$\begin{aligned} f(v_2) - f(v_1) =& \nabla s(v) \cdot (v_2 - v_1) + O(|s|_{2,\infty,t}|t|^2) \\ f(v_3) - f(v_1) =& \nabla s(v) \cdot (v_3 - v_1) + O(|s|_{2,\infty,t}|t|^2). \end{aligned}$$

Solving this linear system for $\nabla s(v) = (s_x(v), s_y(v))$ gives

$$\begin{aligned} s_x(v) &= O(|t|^3 |s|_{2,\infty,t}/A_t) + O((|f(v_2) - f(v_1)| + |f(v_3) - f(v_1)|)|t|/A_t) \\ s_y(v) &= O(|t|^3 |s|_{2,\infty,t}/A_t) + O((|f(v_3) - f(v_1)| + |f(v_2) - f(v_1)|)|t|/A_t), \end{aligned}$$

where A_t stands for the area of triangle t. Inserting these estimates for $\nabla s(v)$ in (2.3), we immediately get

$$|s(v)| \leq C \left((1 + |t|^2/A_t)\|f\|_\infty + |t|^4 |s|_{2,\infty,t}/A_t \right),$$

where $\|f\|_\infty = \max\{|f_\ell|, \ell = 1, \cdots, V\}$. Hence,

$$|s(v)| \leq C \left(\|f\|_\infty + E(s_0) \right)$$

for some positive constant C dependent on the geometry of t. It follows that the Beziér coefficients of $s|_t$ are uniformly bounded and hence $\mathbf{c}$ is bounded for any $s \in K$.

Let $\delta = \inf\{E(s), s \in K\}$. There exists a sequence $\{s_n \in K\}$ such that $\delta_n = E(s_n) \longrightarrow \delta$. Since s_n is uniformly bounded, the corresponding sequence of coefficient vectors $\mathbf{c}_n$ is uniformly bounded and hence has a convergent subsequence. Without loss of generality, we may assume that $\mathbf{c}_n$ converges to $\mathbf{c}^*$. Since $H\mathbf{c}_n = 0$,

we also have $H\mathbf{c}^* = 0$. Thus, the spline function s^* with coefficients $\mathbf{c}^*$ is in $S_d^r(\triangle)$. In addition, since s_n satisfies the interpolation condition, so does s^*. Therefore, we have

$$\begin{aligned}
\delta &= \lim_{n\to\infty} E(s_n) \\
&= \sum_{t\in\triangle} \lim_{n\to\infty} \int_t \left(|D_x^2 s_n| + 2|D_x D_y s_n| + |D_y^2 s_n|\right) \\
&= \sum_{t\in\triangle} \int_t \left(|D_x^2 s^*| + 2|D_x D_y s^*| + |D_y^2 s^*|\right) \\
&= E(s^*).
\end{aligned}$$

We have thus completed the proof. □

Next we consider the L_1 spline smoothing method (1.5). Similar to the above proof, we can show the following

Theorem 2.2. *Suppose that $\triangle$ is a triangulation whose vertices are a subset of the given data locations. Fix $r \geq 0$ and $d > r$. For any $\alpha > 0$, there exists an $s_f \in S_d^r(\triangle)$ minimizing (1.5).*

Proof: Let

$$K := \{s \in S_d^r(\triangle) : E(s) + \alpha\ell(s) \leq \alpha \sum_{i=1}^{V} |f_i|\}.$$

Noting that $E(0) + \alpha\ell(0) = \alpha \sum_{i=1}^{V} |f_i|$, it follows that K is a nonempty set and any minimizer if it exists will be in K. We now show that K is a bounded set, and hence is pre-compact since $S_d^r(\triangle)$ is a finite dimensional space. As in the previous theorem, we can show (2.2). By using Taylor's expansion, we have

$$s(v_i) = s(v) + \nabla s(v) \cdot (v_i - v) + O(|s|_{2,\infty,t}|t|^2) \tag{2.4}$$

for $i = 1, 2, 3$. It follows that

$$\begin{aligned}
s(v_2) - s(v_1) =& \nabla s(v) \cdot (v_2 - v_1) + O(|s|_{2,\infty,t}|t|^2) \\
s(v_3) - s(v_1) =& \nabla s(v) \cdot (v_3 - v_1) + O(|s|_{2,\infty,t}|t|^2).
\end{aligned}$$

Solving this linear system for $\nabla s(v) = (s_x(v), s_y(v))$ gives

$$\begin{aligned}
s_x(v) &= O(|t|^3|s|_{2,\infty,t}/A_t) + (|s(v_2) - s(v_1)| + |s(v_3) - s(v_1)|)|t|/A_t \\
s_y(v) &= O(|t|^3|s|_{2,\infty,t}/A_t) + (|s(v_3) - s(v_1)| + |s(v_2) - s(v_1)|)|t|/A_t.
\end{aligned}$$

Next we note that

$$|s(v_2) - s(v_1)| \leq |s(v_2) - f(v_2)| + |s(v_1) - f(v_1)| \leq \sum_{i=1}^{V} |f_i|$$

since $s \in K$. Similarly for $|s(v_3) - s(v_1)|$. We also note that

$$|s(v_i)| \le |s(v_i) - f(v_i)| + |f(v_i)| \le 2 \sum_{i=1}^{V} |f_i|$$

Thus, it follows from (2.4) and (2.2) that

$$|s(v)| \le C \left(1 + \sum_{i=1}^{V} |f_i| \right)$$

for some positive constant C dependent on the geometry of t. Thus, s is bounded over t and hence bounded over Ω. It follows that the B-coefficients of s are bounded and the rest of the proof is very similiar to the final steps in the proof of the previous theorem. □

It is easy to see that the functional $E(s)$ is convex and so is $\ell(s)$. Hence, any local minimum is a global minimum. In general, such minimization problems may have many solutions since the functionals are not strictly convex.

§3. Computation of the L_1 Spline Fits

Note that the computation of the minimizers of (1.4) and (1.5) is not so obvious. We first concentrate on deriving a numerical method for computing the minimizer of (1.4). The computational method for the minimizer of (1.5) will be similar.

First of all, we use a numerical quadrature to approximate the integral over triangle t. For example, we may use Gauss-Legendre quadrature over a triangular region. That is, assuming $t = \langle (0,0), (1,0), (0,1) \rangle$, we let $x_{n,i}, i = 0, \cdots, n$, be the nodes of the nth Gauss-Legendre quadrature over $[0,1]$, and let $y_{n,i,j}$ be the nodes of the nth Gauss-Legendre quadrature over $[0, 1 - x_{n,i}]$. Iterating these quadratures we have

$$\int_t f(x,y)dxdy \approx \sum_{i=0}^{n} \sum_{j=0}^{n} a_{i,j}^t f(\eta_{i,j}^t).$$

We note that $a_{i,j}^t \ge 0$. Then the functional $E(s)$ may be approximated by

$$\tilde{E}(s) := \sum_{t \in \triangle} \sum_{i=0}^{n} \sum_{j=0}^{n} a_{i,j}^t \left(|D_x^2 s(\eta_{i,j}^t)| + 2|D_x D_y s(\eta_{i,j}^t)| + |D_y^2 s(\eta_{i,j}^t)| \right), \qquad (3.1)$$

which can be viewed as a weighted ℓ_1 norm of the following vector

$$(D_x^2 s(\eta_{i,j}^t), 2D_x D_y s(\eta_{i,j}^t), D_y^2 s(\eta_{i,j}^t), i, j = 0, \cdots, n, t \in \triangle).$$

Furthermore, since

$$a_{i,j}^t D_x^2 s(\eta_{i,j}^t) = \sum_{p+q+r=d} c_{p,q,r}^t a_{i,j}^{(n)} D_x^2 B_{p,q,r}^t(\eta_{i,j}^t) := A_{xx}^t \mathbf{c}^t,$$

where A^t_{xx} is the matrix and $\mathbf{c}^t$ is the vector defined by

$$A^t_{xx} = \left[a^t_{i,j} D^2_x B^t_{p,q,r}(\eta^t_{i,j})\right]_{\substack{p+q+r=d \\ i,j=0,\cdots,n}} \text{ and } \mathbf{c}^t = (c^t_{p,q,r}, p+q+r=d).$$

Let $A = \begin{bmatrix} A_{xx} \\ 2A_{xy} \\ A_{yy} \end{bmatrix}$ with $A_{xx} = \text{diag}(A^t_{xx}, t \in \triangle)$ and similar definitions for A_{xy} and A_{yy}. Let $\mathbf{c} = (\mathbf{c}^t, t \in \triangle)$. Then our numerical method for computing (1.4) is to find

$$\min\{\|A\mathbf{c}\|_1 : \text{ subject to } H\mathbf{c} = 0, I\mathbf{c} = \mathbf{f}\} \tag{3.2}$$

where $\|\mathbf{b}\|_1$ denotes the usual ℓ_1 norm for a vector $\mathbf{b}$, $\mathbf{f} = (f_i, i = 1, \cdots, V)$, $H\mathbf{c} = 0$ denotes the smoothness conditions and $I\mathbf{c} = \mathbf{f}$ denotes the the linear system associated with the interpolation conditions. We can derive a similiar numerical method for computing (1.4).

Such a minimization problem can be converted into a linear programming problem (cf. [Bloomingfield and Steiger'83]) and thus can be solved using the simplex algorithm or numerical methods based on the Karmarkar algorithm (cf. [Vanderbei, Meketon and Freedman'86]). In this paper, we introduce the following algorithm which is a variant of the Karmarkar algorithm. Although many algorithms are available for solving linear programming problems, the algorithm we use is particularly easy to implement.

Theorem 3.1. *Let A be a matrix of size $m \times n$ and $\mathbf{y}$ be a given vector in $\mathbb{R}^m$. Then the following two minimization problems are equivalent:*
(1) Find $\mathbf{c}^* \in \mathbb{R}^n$ such that

$$\|\mathbf{y} - A\mathbf{c}^*\|_1 = \min\{\|\mathbf{y} - A\mathbf{c}\|_1 : H\mathbf{c} = 0, I\mathbf{c} = \mathbf{f}.\}$$

(2) Solve the following linear programming problem:

$$\begin{aligned} &\textit{minimize } \sum_{i=1}^{n} r_i^+ + r_i^- \\ &\textit{subject to} \\ &\qquad [\, A \quad I_n \quad -I_n \,] \begin{bmatrix} \mathbf{c} \\ \mathbf{r}^+ \\ \mathbf{r}^- \end{bmatrix} = \mathbf{y} \\ &\qquad \mathbf{r}^+ \geq 0, \mathbf{r}^- \geq 0, H\mathbf{c} = 0, I\mathbf{c} = \mathbf{f}, \end{aligned}$$

where $\mathbf{r}^+ = (r_1^+, \cdots, r_m^+)$ and $\mathbf{r}^- = (r_1^-, \cdots, r_m^-)^T$ are the positive and negative parts of the residual vector $\mathbf{r} = \mathbf{y} - A\mathbf{c}$.

Proof: Let $\mathbf{c}^*$ be a minimizer of (1). Then $\mathbf{r}^* = \mathbf{y} - A\mathbf{c}^*$ is a fixed residual vector. Then letting $R_i^+ = \max(0, r_i^*), R_i^- = \max(0, -r_i^*)$ for $i = 1, \cdots, m$, we know that

$\mathbf{R}^+ = (R_1^+, \cdots, R_m^+)^T$, $\mathbf{R}^- = (R_1^-, \cdots, R_m^-)^T$ and $\mathbf{c}^*$ satisfy the side conditions of (2). Thus, the optimal solution of (2) must satisfy

$$\min \sum_{i=1}^{n} r_i^+ + r_i^- \leq \sum_{i=1}^{n} R_i^+ + R_i^- = \|\mathbf{y} - A\mathbf{c}^*\|_1.$$

On the other hand, if $\mathbf{r}^+, \mathbf{r}^-$, and $\mathbf{c}$ is the optimal solution of (2), then we have

$$r_i^+ - r_i^- = y_i - \sum_{j=1}^{n} a_{ij} c_j, \qquad \text{for } i = 1, \cdots, m$$

with $A = (a_{ij})_{1 \leq i \leq m, 1 \leq j \leq n}$, $\mathbf{y} = (y_1, \cdots, y_m)^T$ and $\mathbf{c} = (c_1, \cdots, c_n)^T$. For each i, if $y_i - \sum_{j=1}^{n} a_{ij} c_j > 0$, then $r_i^+ = y_i - \sum_{j=1}^{n} a_{ij} c_j$ and $r_i^- = 0$. (Otherwise, $r_i^- > 0$ which implies that $\mathbf{r}^+, \mathbf{r}^-$, and $\mathbf{c}$ can not be the minimizer.) Similarly, if $y_i - \sum_{j=1}^{n} a_{ij} c_j < 0$, $r_i^+ = 0$ and $r_i^- = -(y_i - \sum_{j=1}^{n} a_{ij} c_j)$. If $y_i - \sum_{j=1}^{n} a_{ij} c_j = 0$, then $r_i^+ = 0$ and $r_i^- = 0$. Thus we have

$$\min \sum_{i=1}^{n} r_i^+ + r_i^- = \|\mathbf{y} - A\mathbf{c}\|_1 \geq \|\mathbf{y} - A\mathbf{c}^*\|_1$$

and it follows that problems (1) and (2) are equivalent. □

Note that for a sufficiently large M the minimization problem (3.2) can be approximated by the following problem:

$$\min\{\|A\mathbf{c}\|_1 + M(\|H\mathbf{c}\|_1 + \|I\mathbf{c} - \mathbf{f}\|_1)\}$$

which in turn can be converted into the linear programming problem below.

$$\text{minimize } \sum_{i=1}^{n} r_i^+ + r_i^- + M \left(\sum_{i=1}^{N_H} s_i^+ + s_i^- + \sum_{i=1}^{N_I} u_i^+ + u_i^- \right)$$

subject to

$$\begin{bmatrix} A & I_n & -I_n & 0 & 0 & 0 & 0 & 0 \\ H & 0 & 0 & I_{N_H} & -I_{N_H} & 0 & 0 & 0 \\ I & 0 & 0 & 0 & 0 & 0 & I_{N_I} & -I_{N_I} \end{bmatrix} \begin{bmatrix} \mathbf{c} \\ \mathbf{r}^+ \\ \mathbf{r}^- \\ \mathbf{s}^+ \\ \mathbf{s}^- \\ \mathbf{u}^+ \\ \mathbf{u}^- \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \mathbf{f} \end{bmatrix}$$

$$\mathbf{r}^+ \geq 0, \mathbf{r}^- \geq 0, \mathbf{s}^+ \geq 0, \mathbf{s}^- \geq 0, \mathbf{u}^+ \geq 0, \mathbf{u}^- \geq 0,$$

where, N_H and N_I stand for the numbers of rows in matrices H and I and I_N denotes the identity matrix of size N.

Next we write out the dual problem:

$$\begin{aligned}
&\max \mathbf{f}^T\mathbf{y} \\
&\text{subject to} \\
&\quad A^T\mathbf{v} + H^T\mathbf{w} + I^T\mathbf{y} = 0 \\
&\quad -\mathbf{1} \le \mathbf{v} \le \mathbf{1}, -\mathbf{M} \le \mathbf{w} \le \mathbf{M}, -\mathbf{M} \le \mathbf{y} \le \mathbf{M}.
\end{aligned}$$

Here, $\mathbf{1}$ denotes a vector the same size as $\mathbf{v}$ and with all entries equal to 1, while $\mathbf{M} = M\mathbf{1}$ denotes a vector whose entries all equal M and has the same size as $\mathbf{w}$ and $\mathbf{y}$, respectively.

It is clear that with appropriate choice of vector $\mathbf{Y}$ and matrix X the above linear programming problem can be simplified to the following problem.

$$\begin{aligned}
&\max \mathbf{Y}^T\mathbf{W} \\
&\text{subject to} \\
&\quad X^T\mathbf{W} = 0, -\mathbf{1} \le \mathbf{W} \le \mathbf{1}
\end{aligned}$$

with $\mathbf{W} = [\mathbf{v}\ \mathbf{w}/M\ \mathbf{y}/M]^T$ which involves M (=1000 in our numerical experiments), $\mathbf{Y} = [0\ 0\ \mathbf{f}]^T$ and $X = [A\ MH\ MI]^T$.

We now present the primal affine algorithm which is a variant of the Karmarkar algorithm for linear programming problems. This algorithm simultaneously solves both the primal and the dual problem. Although this may not be the optimal algorithm, it is easy to implement and we use it to illustrate the L_1 spline method.

Algorithm 3.2. *Let n be the size of $\mathbf{W}$.*

Step 1. Start with $k = 0$ and a zero vector $\mathbf{W}^0$.

Step 2. While $k = 1, 2, ...$, let D_k be the diagonal matrix whose i^{th} entry is $\min\{w_i + 1, 1 - w_i\}$ where $\mathbf{W}^{k-1} = (w_1, \cdots, w_n)$.

2.1 Compute $\mathbf{c}_k = (X^T D_k^2 X)^{-1} X^T D_k^2 \mathbf{Y}$ and $\mathbf{r}_k = \mathbf{Y} - X\mathbf{c}_k$.

2.2 Compute $\mathbf{p}_k = D_k^2 \mathbf{r}_k$. Writing $\mathbf{p}_k = (p_1, \cdots, p_n)$, let

$$\alpha_k = \max_i \{\max\left(\frac{p_i}{1 - w_i}, -\frac{p_i}{1 + w_i}\right)\}$$

2.3 Compute $\mathbf{W}^k = \mathbf{W}^{k-1} + 2/(3\alpha_k)\mathbf{p}_k$.

Step 3. If $\|\mathbf{r}_k\|_1 - \mathbf{Y}^T\mathbf{W}^k \le \epsilon$, then stop and $\mathbf{c}_k$ is an approximation of the L_1 norm minimizer.

Step 4. Otherwise, go back to Step 2.

Note that in Step 2.1 we in fact compute a least squares solution $\mathbf{c}_k$ of the system of equations. To give a heuristic reason why the above algorithm works, we borrow some arguments from [Meketon'85]. We begin with

Lemma 3.3. $\mathbf{W}^k$ *is feasible for all* k.

Proof: Clearly, $\mathbf{W}^0 = 0$ is feasible. Assume that $\mathbf{W}^{k-1}$ is feasible. That is, $X^T\mathbf{W}^{k-1} = 0$ and $-\mathbf{1} \le \mathbf{W}^{k-1} \le \mathbf{1}$. Since

$$\mathbf{W}^k = \mathbf{W}^{k-1} + \frac{2}{3\alpha_k} D_k^2 \left(\mathbf{Y} - X\mathbf{c}_k\right),$$

and

$$\begin{aligned} X^T D_k^2 \left(\mathbf{Y} - X\mathbf{c}_k\right) &= X^T D_k^2 \left(\mathbf{Y} - X(X^T D_k^2 X)^{-1} X^T D_k^2 \mathbf{Y}\right) \\ &= \left(X^T D_k^2 - X^T D_k^2 X (X^T D_k^2 X)^{-1} X^T D_k^2\right) \mathbf{Y} \\ &= 0, \end{aligned}$$

we have $X^T\mathbf{W}^k = 0$. The choice of α_k ensures that $-\mathbf{1} \le \mathbf{W}^k \le \mathbf{1}$. □

We next show that α_k is bounded above.

Lemma 3.4. *For all* k,

$$\alpha_k \le \|D_k(\mathbf{Y} - X\mathbf{c}_k)\|_2.$$

Proof: *Writing* $\mathbf{g} = (g_1, \cdots, g_n)^T = D_k(\mathbf{Y} - X\mathbf{c}_k)$, *we have* $p_i = \min(1 + w_i, 1 - w_i)g_i$ *and*

$$\begin{aligned} \alpha_k &= \max_i \left[\max\left(\frac{\min(1 + w_i, 1 - w_i)g_i}{1 - w_i}, -\frac{\min(1 + w_i, 1 - w_i)g_i}{1 + w_i}\right)\right] \\ &= \max_i \left[|g_i| \max\left(\frac{\min(1 + w_i, 1 - w_i)}{1 - w_i}, \frac{\min(1 + w_i, 1 - w_i)}{1 + w_i}\right)\right] \\ &= \max_i |g_i| \le \|\mathbf{g}\|_2 = \|D_k(\mathbf{Y} - X\mathbf{c}_k)\|_2. \end{aligned}$$

This completes the proof. □

Lemma 3.5. *The objective value* $\mathbf{Y}^T\mathbf{W}^k$ *is increasing and* $\|D_k(\mathbf{Y} - X\mathbf{c}_k)\|_2 \longrightarrow 0$ *as* k *increases.*

Proof: Indeed, we have

$$\begin{aligned} \mathbf{Y}^T\mathbf{W}^k &= \mathbf{Y}^T\mathbf{W}^{k-1} + \frac{2}{3\alpha_k}\mathbf{Y}^T D_k^2 \left(\mathbf{Y} - X\mathbf{c}_k\right) \\ &= \mathbf{Y}^T\mathbf{W}^{k-1} + \frac{2}{3\alpha_k}\mathbf{Y}^T D_k \left(I - D_k X (X^T D_k^2 X)^{-1} D_k\right) D_k \mathbf{Y} \\ &= \mathbf{Y}^T\mathbf{W}^{k-1} + \frac{2}{3\alpha_k}\| \left(I - D_k X (X^T D_k^2 X)^{-1} D_k\right) D_k \mathbf{Y}\|_2^2 \\ &= \mathbf{Y}^T\mathbf{W}^{k-1} + \frac{2}{3\alpha_k}\|D_k(\mathbf{Y} - X\mathbf{c}_k)\|_2^2. \end{aligned}$$

Since the feasible set is bounded and the sequence $\{\mathbf{Y}^T\mathbf{W}^k, k = 1, 2, \cdots,\}$ is increasing, the sequence must converge. This implies that

$$\frac{1}{\alpha_k}\|D_k(\mathbf{Y} - X\mathbf{c}_k)\|_2^2 \longrightarrow 0.$$

By Lemma 3.4, we conclude that $\|D_k(\mathbf{Y} - X\mathbf{c}_k)\|_2 \longrightarrow 0$. $\square$

Since the feasible set is bounded, there is a subsequence in $\{\mathbf{W}^k, k = 0, 1, \cdots,\}$ which converges. Let us assume that $\mathbf{W}^k$ converges to $\mathbf{W}^*$ for convenience. Then D_k converges and so does $\mathbf{c}_k$. Let $\mathbf{c}^*$ be the limit vector. Since $D_k(\mathbf{Y} - X\mathbf{c}_k) \longrightarrow 0$, we have that if $-1 < (\mathbf{W}^*)_i < 1$, then $y_i - (X\mathbf{c}_k)_i \to 0$. If $(\mathbf{W}^*)_i = -1$, we claim that $y_i - (X\mathbf{c}^*)_i < 0$. Otherwise, for $(\mathbf{W}^k)_i$ close to -1 and $y_i - (X\mathbf{c}_k)_i \geq 0$ which implies that

$$(\mathbf{W}^{k+1})_i - (\mathbf{W}^k)_i = \frac{2}{3\alpha_{k+1}}(1 + (\mathbf{W}^k)_i)^2(y_i - (X\mathbf{c}_k)_i) \geq 0.$$

That is, $(\mathbf{W}^{k+1})_i > (\mathbf{W}^k)_i$ for all k sufficiently large while $(\mathbf{W}^k)_i$ converges to $(\mathbf{W}^*)_i = -1$. This is a contradiction. Similarly, we have that if $(\mathbf{W}^*)_i = 1$, then $y_i - (X\mathbf{c}^*)_i > 0$. Therefore, we conclude

$$\begin{aligned}
\mathbf{Y}^T\mathbf{W}^* &= (\mathbf{Y} - X\mathbf{c}^* + X\mathbf{c}^*)^T\mathbf{W}^* \\
&= (\mathbf{Y} - X\mathbf{c}^*)^T\mathbf{W}^* + (\mathbf{c}^*)^T X^T\mathbf{W}^* \\
&= (\mathbf{Y} - X\mathbf{c}^*)^T\mathbf{W}^* \\
&= \sum_{(\mathbf{W}^*)_i=1} (y_i - (X\mathbf{c}^*)_i) + \sum_{(\mathbf{W}^*)_i=-1} -(y_i - (X\mathbf{c}^*)_i) \\
&= \sum_i |y_i - X\mathbf{c}^*| = \|\mathbf{Y} - X\mathbf{c}^*\|_1.
\end{aligned}$$

Based on the duality theorem for linear programming problems, we know that

$$\max_{\mathbf{W}} \mathbf{Y}^T\mathbf{W} \leq \min_{\mathbf{c}} \|\mathbf{Y} - X\mathbf{c}\|_1$$

with equality if optimal. Thus, $\mathbf{W}^*$ and $\mathbf{c}^*$ are optimal solutions. This is the rationale for the stopping criterion in Algorithm 3.2 above.

§4. Computational Experiments for L_1 Spline

In this section we present five sets of numerical experiments using the L_1 interpolation method (1.4) and the minimal energy method (including the penalized least squares method). Our observation is that the L_1 spline method produces interpolatory or fitting surfaces with less wrinkles than those from the minimal energy method. The observation is consistent with the features of the surfaces in [Lavery'00] and [Lavery'01]. This gives more evidence that the L_1 spline methods are useful for surface designers.

Example 4.1. *Consider a set of data obtained from a dome as shown in Fig. 4.1. We used* 81 *equally spaced data points over the square domain* $[-1.5, 1.5] \times [-1.5, 1.5]$. *We first use* C^1 *cubic bivariate splines to find the* L_1 *spline interpolation. The* L_1 *spline interpolant satisfies the interpolation conditions as shown in Fig. 4.2.*

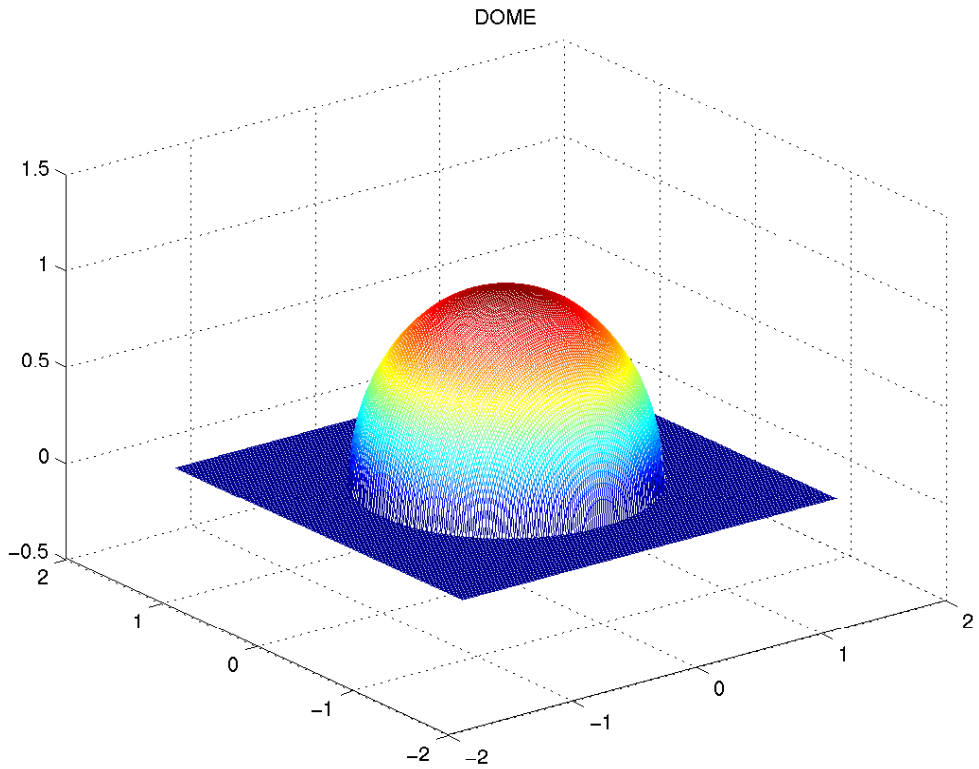

Fig. 4.1. Dome

For comparison, we also used the minimal energy method to construct a C^1 cubic spline surface interpolating these 81 data values as shown in Fig. 4.3. The surface in Fig. 4.3 using the minimal energy method clearly shows a lot of ripples.

Example 4.2. *Similarly we use 289 data points equally spaced over $[-1.5, 1.5] \times [-1.5, 1.5]$ and compute the L_1 spline interpolant and the minimal energy interpolant as shown in Figs. 4.4 and 4.5. Again the surface in Fig. 4.5 using the minimal energy method shows a lot of wrinkles.*

Example 4.3. *We use a set of data points which were manually sampled from a car model as shown in Fig. 4.6. The triangulation of the data locations is shown in Fig. 4.7. We use C^1 quintic spline functions to find the L_1 spline interpolation and the minimal energy interpolate. The surfaces constructed are shown in Fig. 4.8 and Fig. 4.9, respectively. From the figures we can see that the L_1 spline method yields a better surface than that from the minimal energy method which shows ripples along the connection of carhood and fender.*

Example 4.4. *In this example, we shall see how splines of higher degrees can be beneficial for surface design. Consider the following data set and triangulation. We use C^1 cubic and quintic spline spaces and the L_1 spline interpolation method to generate the surfaces in Figures 4.11 and 4.12. Next we use these two spline spaces to fit the given data by the minimal energy interpolation method. This produces the surfaces in Fig. 4.13 and 4.14. We observe that the graph in Fig. 4.12 is closer to the piecewise linear interpolant of the given data than the graph in Fig. 4.11.*

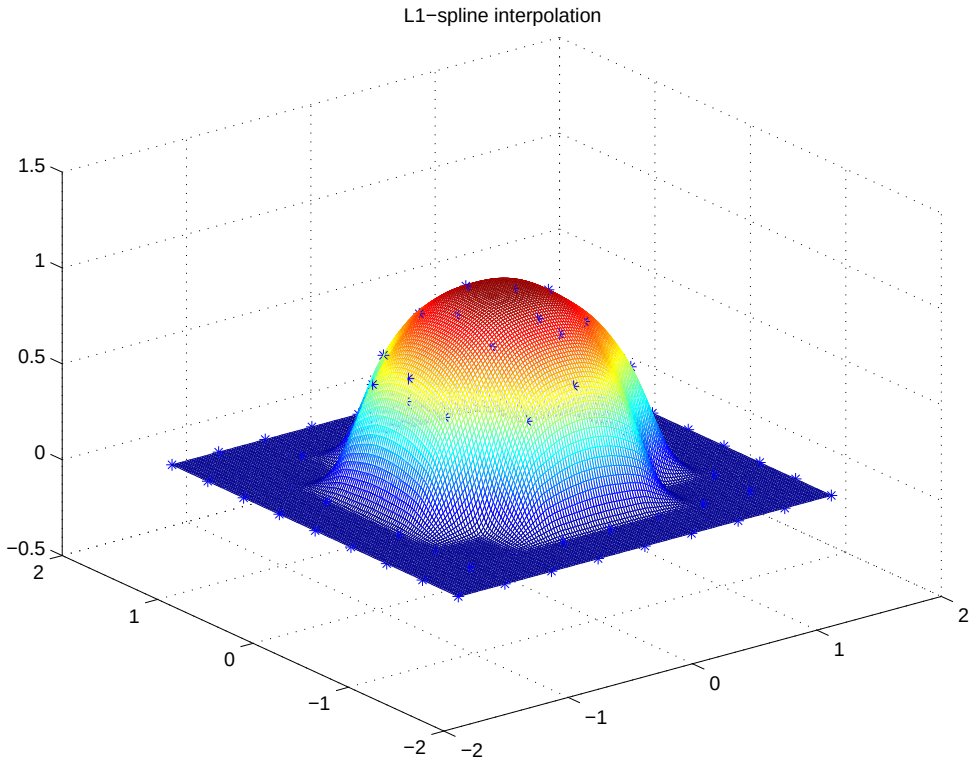

Fig. 4.2. The L_1 spline interpolant of the dome

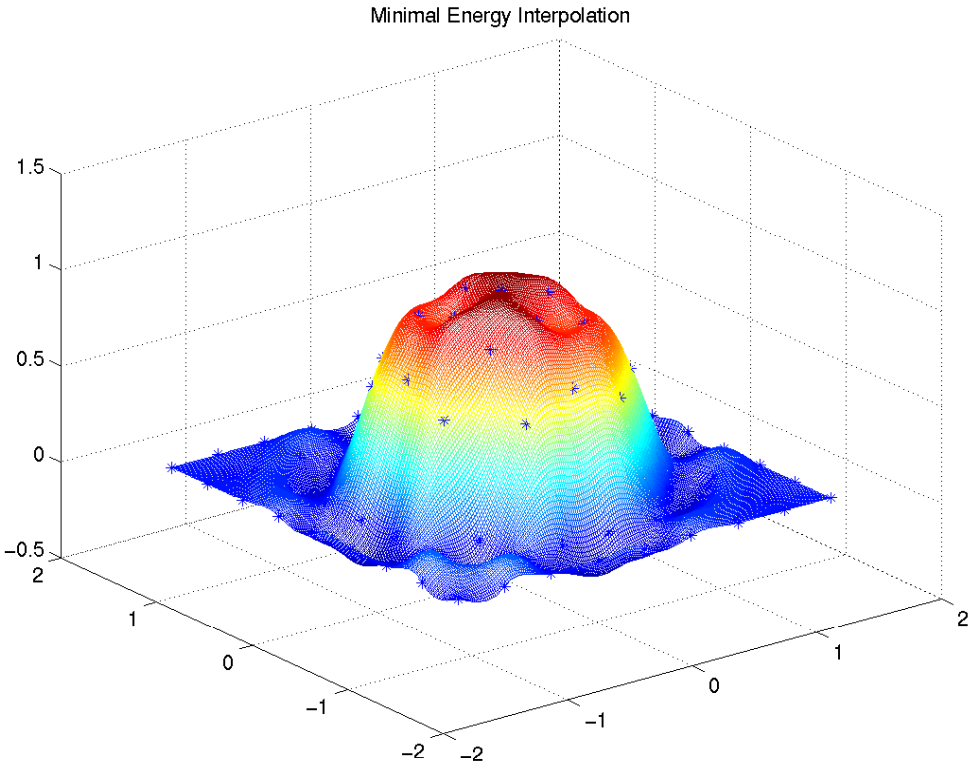

Fig. 4.3. The minimal energy interpolant of the dome

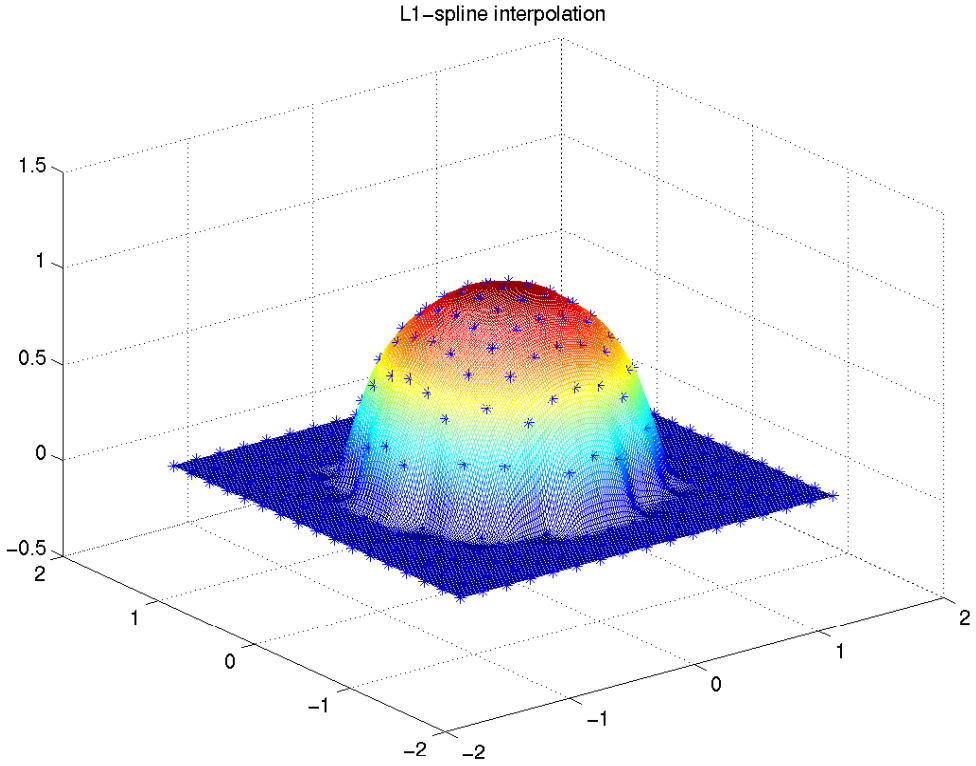

Fig. 4.4. The L_1 spline interpolant of the dome

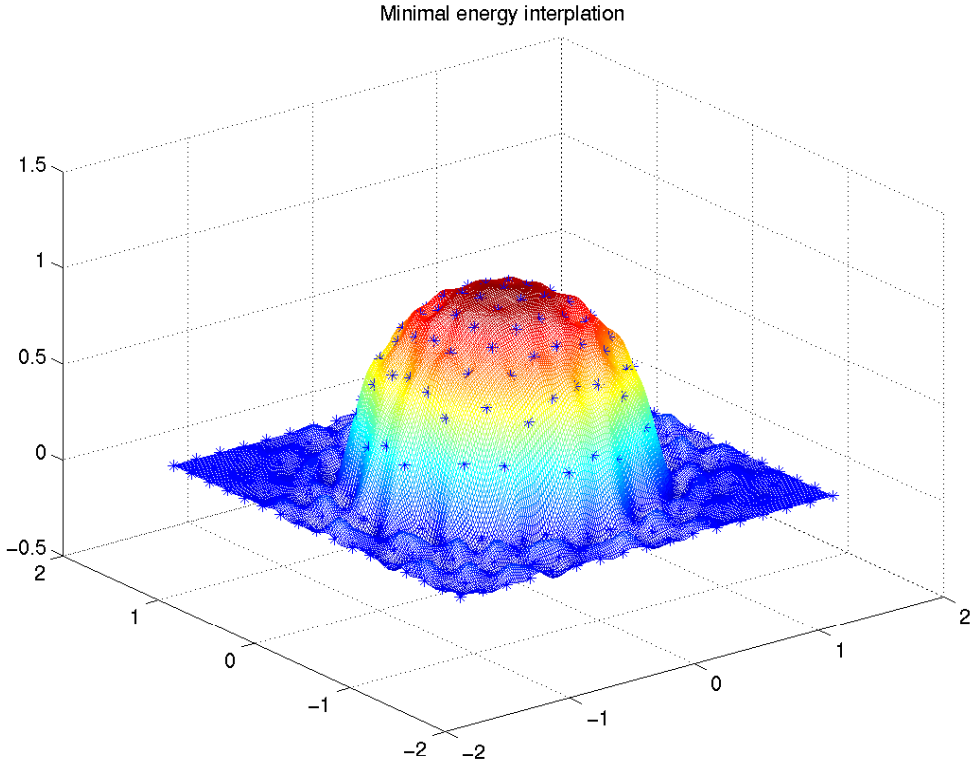

Fig. 4.5. The minimal energy interpolant of the dome

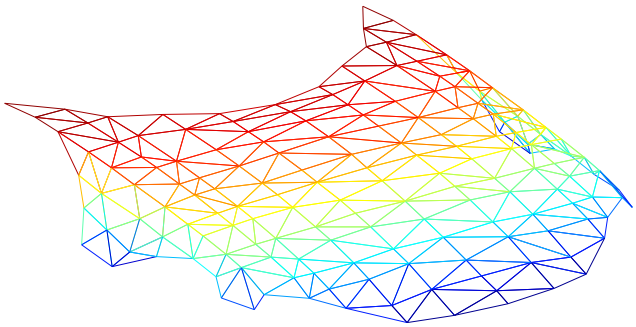

Fig. 4.6. A set of scattered data

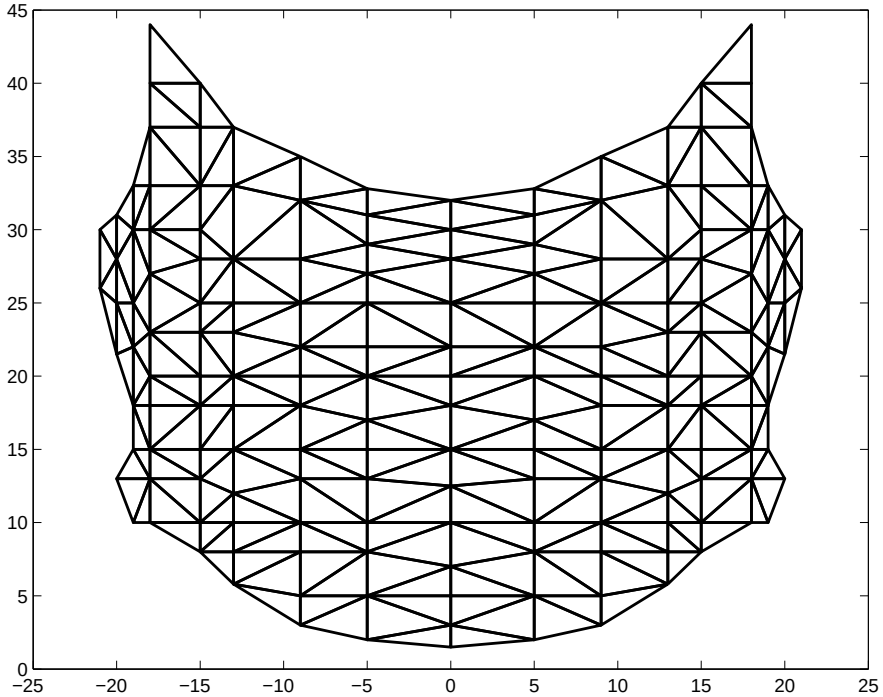

Fig. 4.7. A triangulation of the scattered data

Example 4.5. *In Figures 4.15–4.17 we now present an example comparing the* L_1 *spline smoothing method (minimizing (1.5)) and the penalized least squares method (minimizing (1.6)). In Fig. 4.15 the data and the underlaying triangulation are*

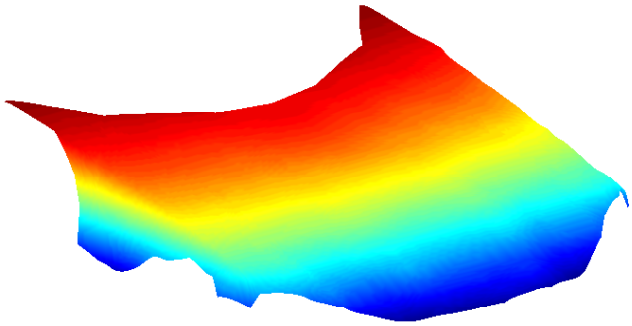

Fig. 4.8. The L_1 spline interpolant of the car hood

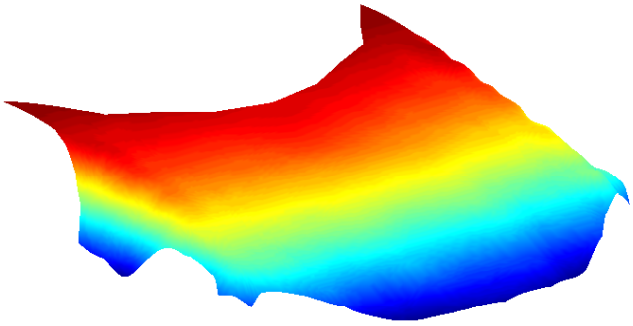

Fig. 4.9. The minimal energy interpolant of the car hood

ploted. In Fig. 4.16 the L_1 smothing spline is ploted and the penalized least squares spline in Fig. 4.17. Clearly the L_1 smothing spline gives a tighter approximation

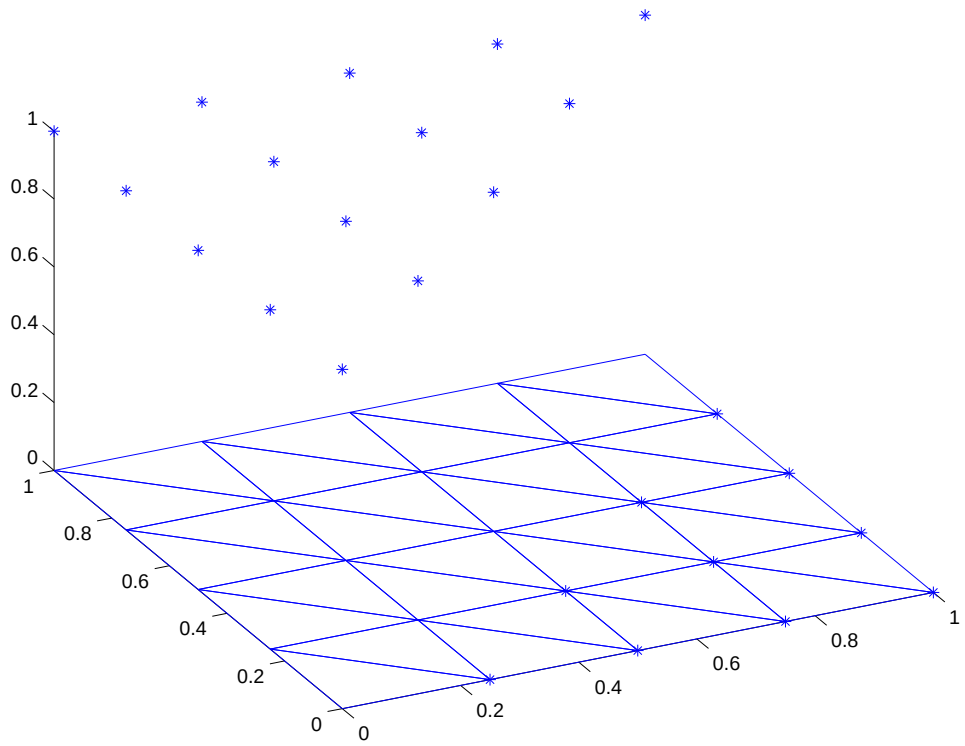

Fig. 4.10. A given data and triangulation

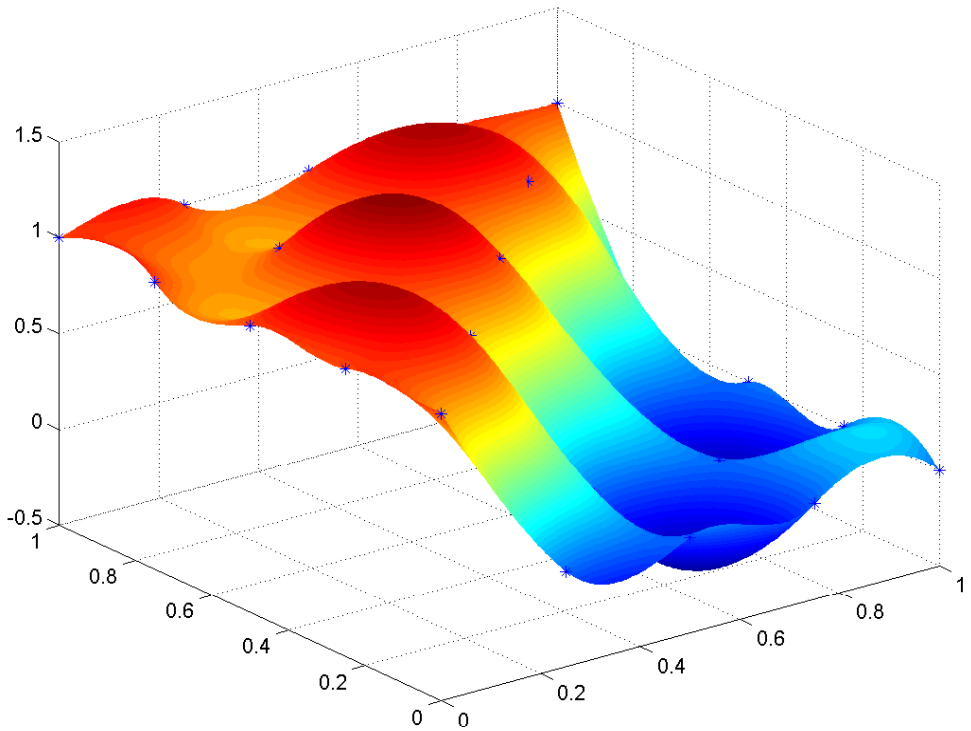

Fig. 4.11. C^1 cubic L_1 spline interpolation

of the given data set.

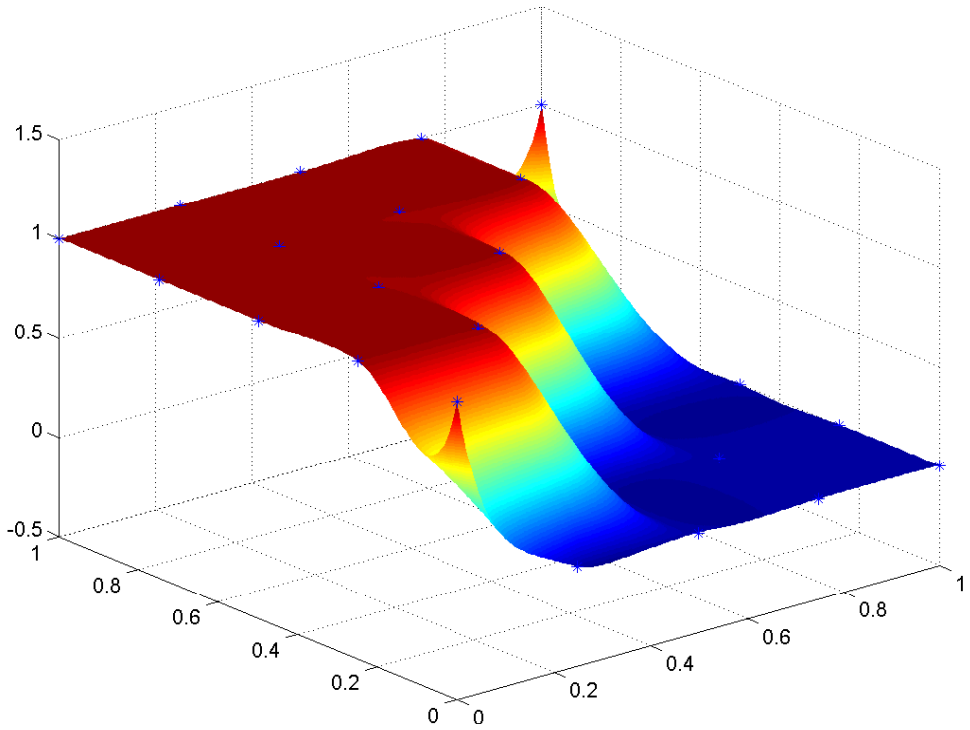

Fig. 4.12. C^1 quintic L_1 spline interpolation

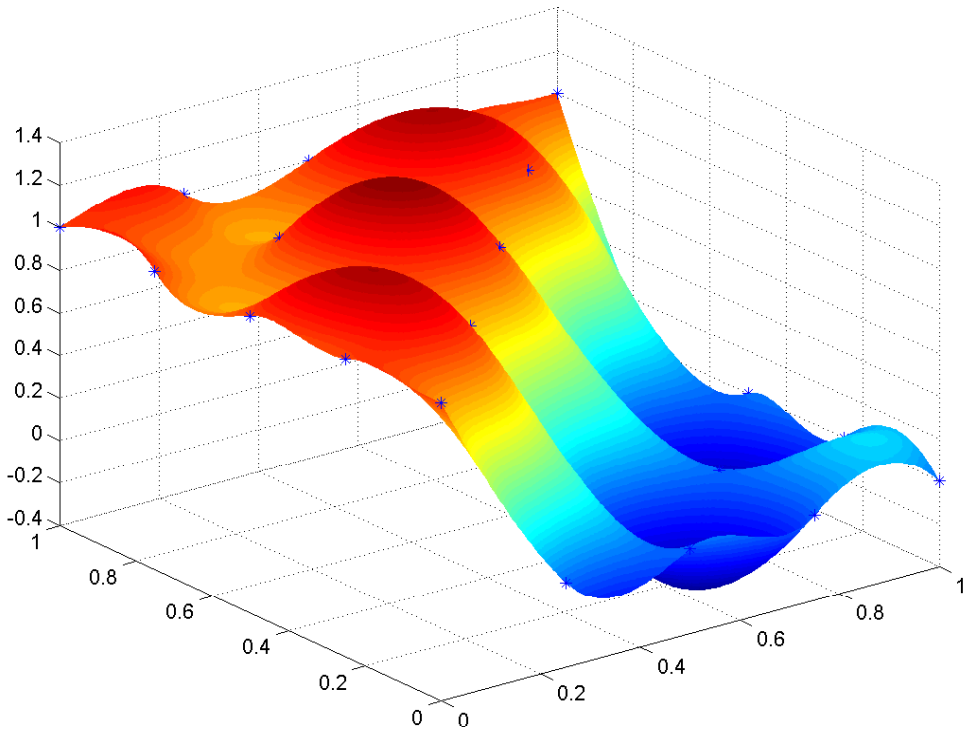

Fig. 4.13. C^1 cubic spline interpolation using the minimal energy method

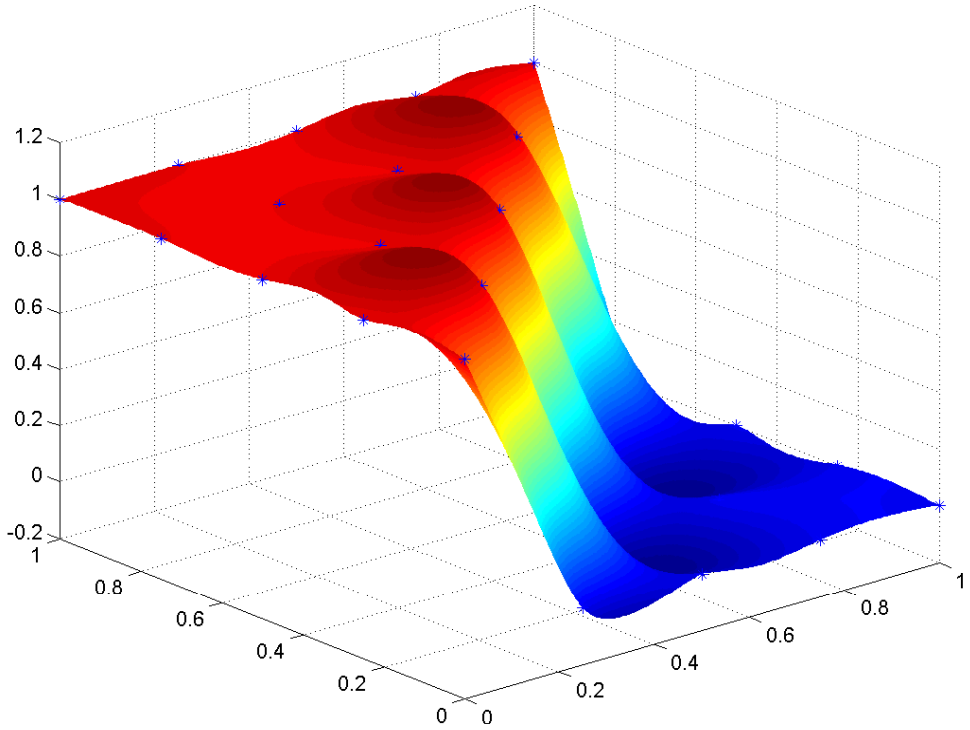

Fig. 4.14. C^1 quintic spline interpolation using the minimal energy method

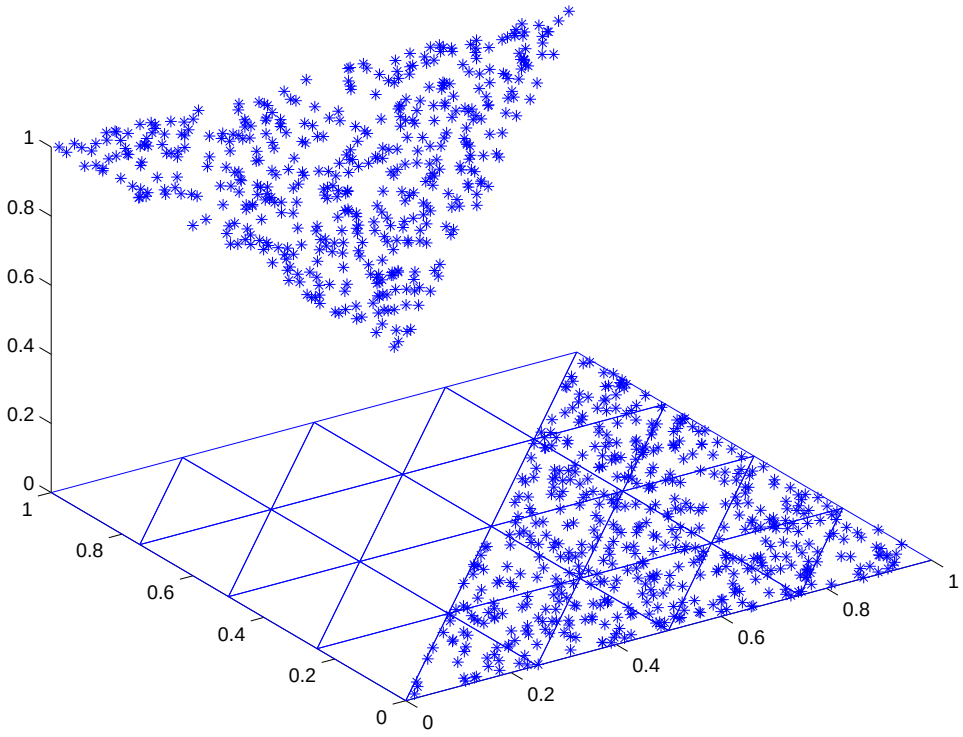

Fig. 4.15. A given set of 1000 random data and triangulation

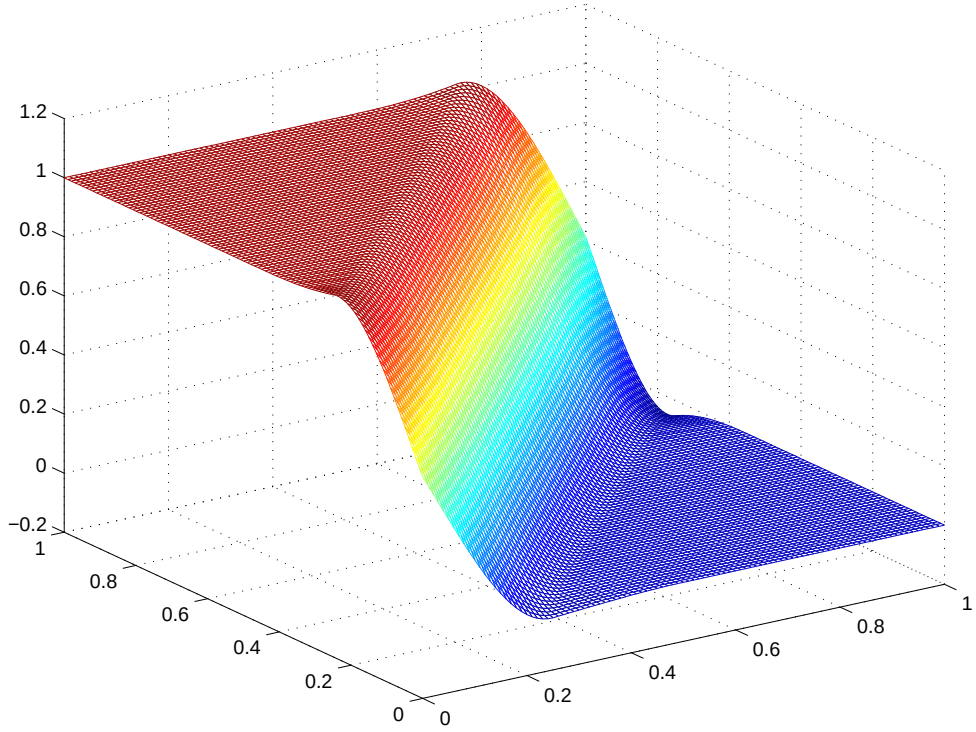

Fig. 4.16. A C^1 cubic spline fitting by L_1 spline smoothing method

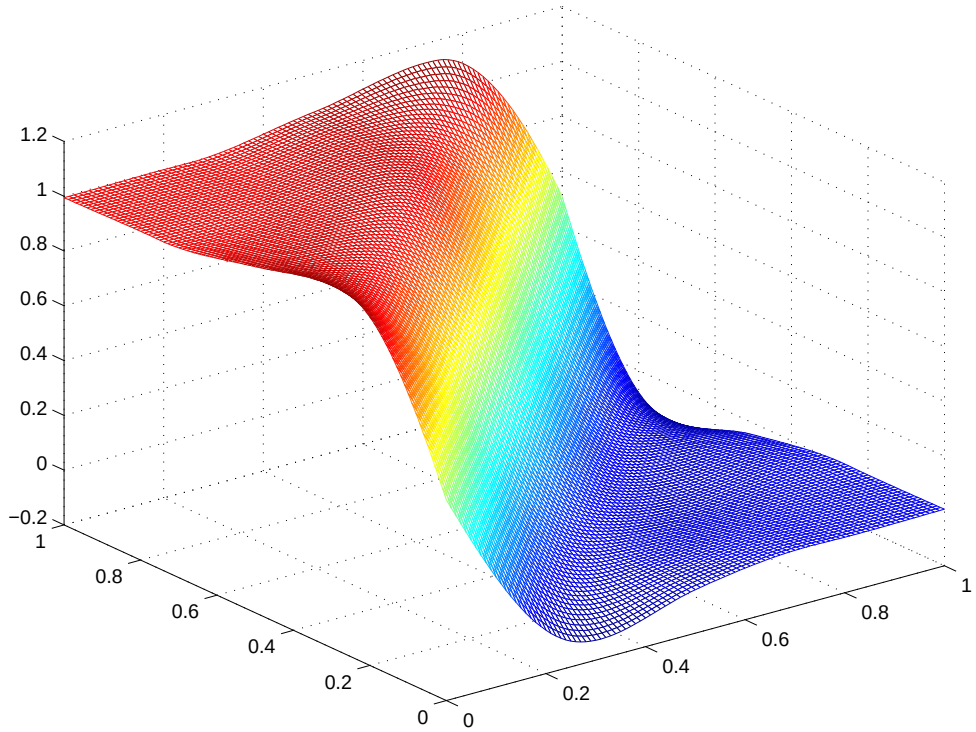

Fig. 4.17. A C^1 cubic spline fitting by the penalized least squares method

§5. Some Extensions

Next we note that L_1 spline methods can also be generalized to deal with other surface design problems. One problem is the smooth connection of multiple surface patches. Let us consider two given smooth surface patches S_1 and S_2 which are to be connected in a smooth fashion. Assume that they can be projected onto a plane P, that is, S_1 and S_2 are two functions defined on P. Let Ω_1 and Ω_2 be the domains of S_1 and S_2, respectively. Let Ω be a domain which connects Ω_1 and Ω_2 in a desirable fashion. Suppose that Γ_1 and Γ_2 are the intersection of Ω with Ω_1 and Ω_2, respectively. Then we consider the following problem: find $s_f \in S_d^r(\triangle)$ such that

$$\begin{aligned} E(s_f) = \min\{E(s) : & s \in S_d^r(\triangle), s|_{\Gamma_1} = S_1|_{\Gamma_1}, s|_{\Gamma_2} = S_2|_{\Gamma_2} \\ & \frac{\partial}{\partial \mathbf{n}}(s - S_1)|_{\Gamma_1} = 0, \frac{\partial}{\partial \mathbf{n}}(s - S_2)|_{\Gamma_2} = 0\}. \end{aligned} \tag{5.1}$$

Similar to the arguments in §2 and §3, we can show the existence of such spline functions and compute the spline function which connects the two given surface patches. We omit the details here. See Figs. 5.1–5.3 at the end of this section.

Another possible extension of the L_1 spline methods is to compute polygonal hole filling. Let us be more precise. Assume that a given smooth surface S has a hole which can be projected onto a plane P and that the projection of the hole is a polygonal domain Ω. We let g denote the function value of the given surface S on $\partial\Omega$ and h the normal derivative value of S on $\partial\Omega$. Let $\triangle$ be a triangulation of Ω. Then we can formulate the following problem: find $s_f \in S_d^r(\triangle)$ such that

$$E(s_f) = \min\{E(s) : s \in S_d^r(\triangle), s|_{\partial\Omega} = g \text{ and } \frac{\partial}{\partial \mathbf{n}} s|_{\partial\Omega} = h\}. \tag{5.2}$$

Similar to the arguments in §2 and §3, we can show the existence such spline functions and compute the spline function which fills the polygonal hole.

Example 5.1. *We are given two smooth surface patches S_1 and S_2 as shown in Fig. 5.1. We first use the L_1 spline method (5.1) to construct a connecting surface which connects the two surface patches in C^1 fashion in Fig. 5.2. Then we use the minimal energy method to construct a connecting surface joining the two patches in C^1 fashion using C^1 cubic bivariate splines. See Fig. 5.3.*

References

[1] Bloomfield, P. and W. L. Steiger, 1983, *Least Absolute Deviations: Theory, Applications, and Algorithms*, Birkhäuser, Boston, 1983.

[2] Farin, G. 1986, Triangular Bernstein-Bézier patches, Comput. Aided Geom. Design, 3(1986), 83–127.

[3] Fasshauer, G., and L. L. Schumaker, 1996, Multi-patch parametric surfaces with minimal energy, Comp. Aided Geom. Design, 13(1996), pp. 45–79.

[4] Lavery, J. E. 2000, Shape-preserving, multiscale fitting of univariate data by cubic L_1 smoothing splines, Comp. Aided Geom. Design, 17(2000), pp. 715–727.

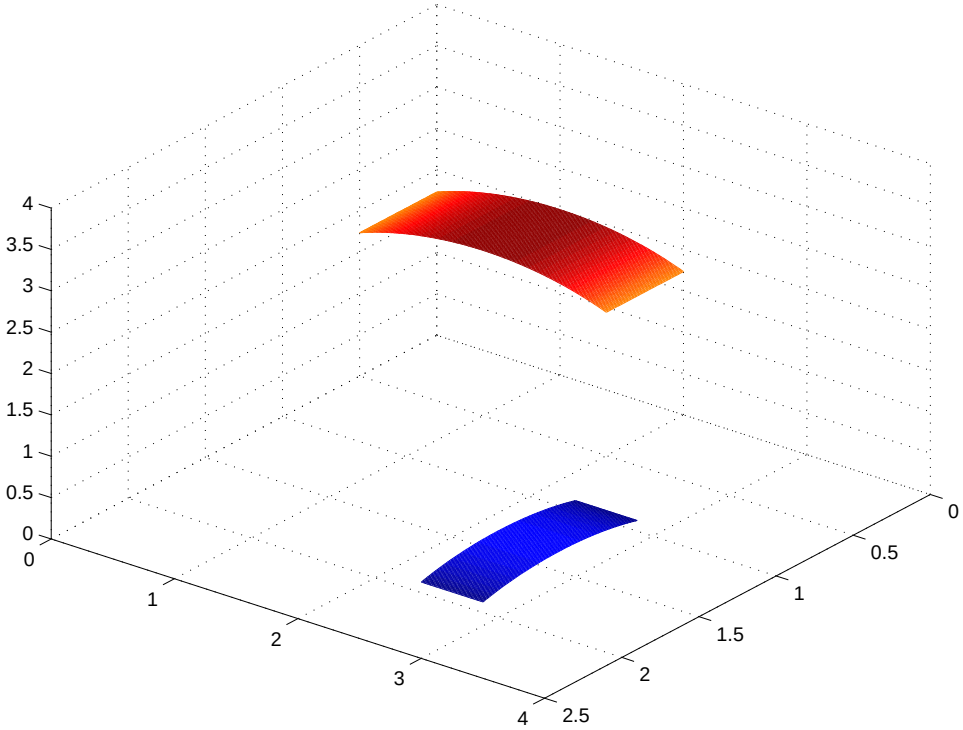

Fig. 5.1. Two given surface patches

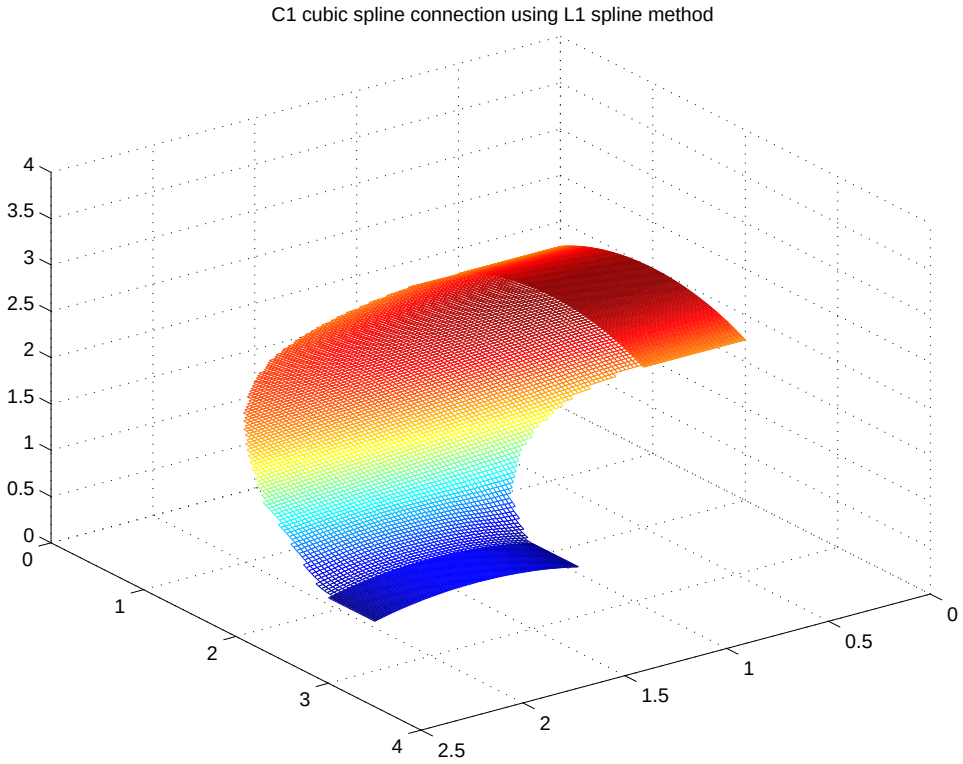

Fig. 5.2. The L_1 spline connection of the two given surface patches

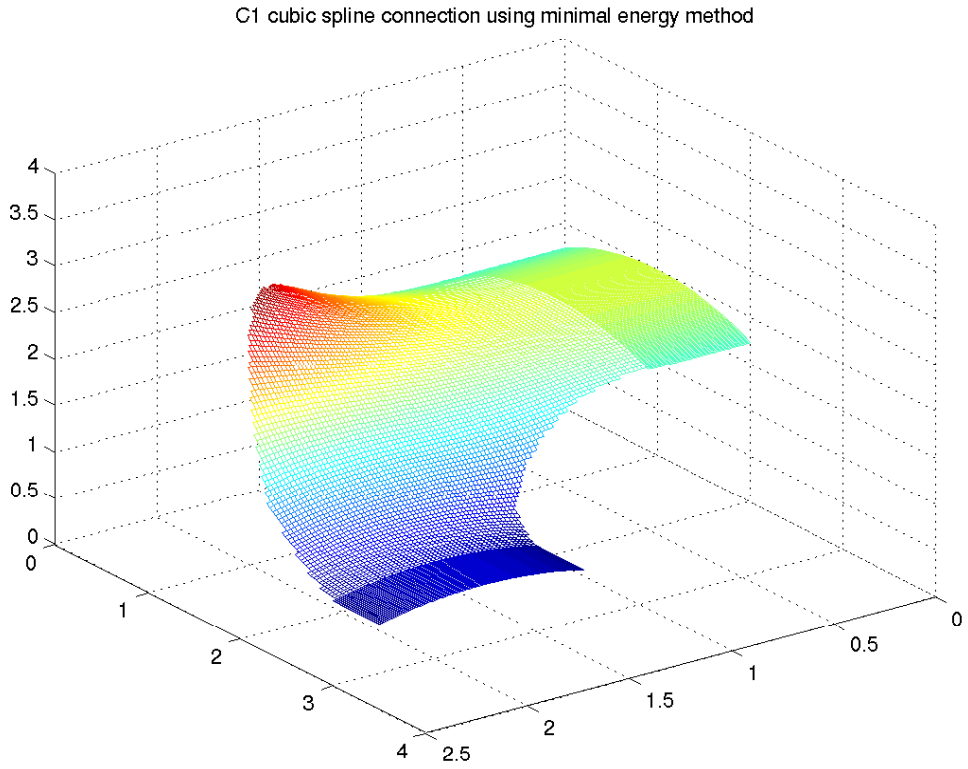

Fig. 5.3. The minimal energy method for connecting the two surface patches

[5] Lavery, J. E., 2001, Shape-preserving, multiscale interpolation by bi- and multivariate cubic L_1 splines, Comp. Aided Geom. Design, 18(2001), pp. 321–343.
[6] Meketon, M., 1985, Least absolute value regression, unpublished manuscript.
[7] Vanderbei, R. J. Meketon, M. J. and B. A. Freedman, 1986, A modification of Karmarkar's linear programming algorithm, Algorithmica, 1(1986), pp. 395–407.