Filling Polygonal Holes Using C^1 Cubic Triangular Spline Patches

Charles K. Chui[a] and Ming-Jun Lai[b]

Abstract

We use the method of energy minimization to fill polygonal holes by C^1 cubic triangular spline patches. We implement the method in MATLAB. Several numerical examples are shown.

§1. Introduction

In free-form surface modeling or surface design, we often encounter a polygonal hole when assembling several surface patches together. Usually these given surfaces patches are C^1 bicubic patches. We have to find a mending surface patch to fill the hole such that the modified surface is in C^1 or G^1 globally. In some applications, the mending surface patch to fill the hole may be required to satisfy certain interpolation condition. Several researchers have already tested some ideas, e.g., bicubic patches, subdivision algorithm, and etc. See, e.g., [Hahn'88], [Gregory and Zhou'94], [Qu'96], [Zheng and Ball'97]. In this paper, we propose to use C^1 cubic triangular spline patches to handle the filling problem. Our method can fill any number of sides of polygonal holes. However, we relax the boundary matching conditions a little bit. More precisely, our filling spline surface only matches the boundary values exactly (for bicubic patches which form a polygonal hole) while approximates the normal derivatives on the boundary of the hole. We use C^1 piecewise cubic triangular spline functions to construct the filling surfaces. Certainly there are many spline surfaces available to fill a polygonal hole. We shall minimize the potential energy of the filling surfaces to reduce the bumpiness of the surface. The relaxation of boundary matching, the application of the C^1 cubic triangular splines, and the energy minimization make our method more convenient and more flexible than those methods presented in the literature above.

The paper is organized as follows. We first describe our method to fill polygonal hole in §2. Then we discuss the existence and uniqueness of the filling spline surfaces in §3. We shall discuss the implementation of our method and present several examples to demonstrate the effectiveness of our method in §4. Finally, we give some remarks in §5.

---

a) Department of Mathematics, Texas A&M University, College Station, TX 77843. Research supported by the National Science Foundation GOALI program under Grant # DMI-9634833 and Boeing Company, St. Louis.

b) Department of Mathematics, University of Georgia, Athens, GA 30602. Research supported by the National Science Foundation under Grant # DMS-9870178.

§2. Bivariate Spline Method for Filling Polygonal Holes

Let us take time to describe our method for filling polygonal holes. For simplicity, we first assume that there exists a plane L such that the projection of the polygonal hole H on the plane L is a bounded domain with Lipschitz boundary. Here, a boundary $\partial\Omega$ of a domain in $\mathbf{R^2}$ is said to be Lipschitz if $\partial\Omega$ can be divided into a finite overlapping segments $p_1, \ldots, p_n$ such that $\partial\Omega = \bigcup_{T=1}^{n} p_i$ with $p_i \cap p_{i+1} \neq \emptyset, i = 1, \cdots, n-1$ and $p_n \cap p_1 \neq \emptyset$ and each segment p_i is a Lipschitz function. In general, we have to divide a hole H into several sub-holes by adding artificial boundaries such that for each sub-hole H_i, there exists a plane L_i so that the projection of H_i onto L_i is Lipschitz boundary of a bounded domain Ω_i. See Figure 7 for a hole which is divided into two sub-holes with one sub-hole is filled. We certainly divide H as less number of sub-holes as possible.

We then describe how to fill one of these sub-holes. For convenience, we assume that H can be projected to a plane L such that the projection is a Lipschitz boundary $\partial\Omega$ of a bounded domain Ω. We divide Ω into a collection $\Diamond$ of non-degenerate convex quadrilaterals. (See [Lai and Schumaker '97] for methods of partitioning a polygon into quadrilaterals.) For example, a H-domain is divided into quadrilaterals as shown in Figure 1.

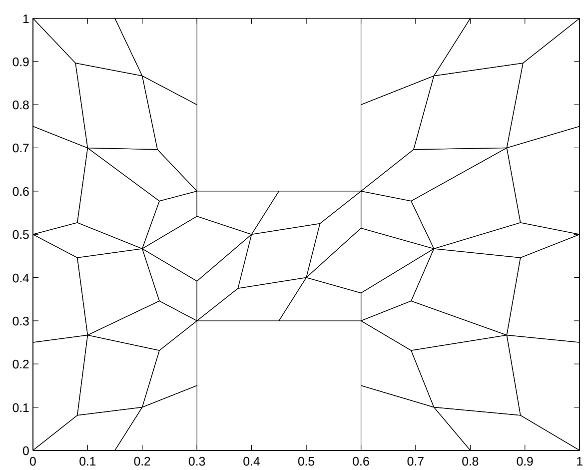

Figure 1. A quadrangulation of H-domain

Let $\diamondplus$ be the triangulation obtained from $\Diamond$ by adding two diagonals of each quadrilateral. We define

$$S_3^1(\diamondplus) = \{s \in C^1(\Omega) \quad s|_t \in \mathbf{P}_3, \forall t \in \diamondplus\}$$

to be the space of C^1 cubic spline functions over triangulation $\diamondplus$. We note that such a special triangulation is general enough to partition any polygon. Since Ω may not be a polygon, we use the curved edge quadrilaterals on the boundary $\partial\Omega$ if necessary. We let $V_b = \{V_1, \cdots, V_B\}$ be the collection of boundary vertices of $\diamondplus$ and $U_b = \{U_1, \cdots, U_B\}$ be the collection of all midpoint of all boundary edges of $\diamondplus$. Let

$$V_{\partial\Omega} = V_b \cup U_b.$$

Next we define an energy functional

$$E(s) = \sum_{t \in \diamondplus} \int\int_t \left[\left(\frac{\partial^2}{\partial x^2} s \right)^2 + 2 \left(\frac{\partial^2}{\partial x \partial y} s \right)^2 + \left(\frac{\partial^2}{\partial y^2} s \right)^2 \right] dx dy$$

on $S_3^1(\diamondplus)$. Our method to fill a hole is to find $s^* \in S_3^1(\diamondplus)$ solving the following minimization problem:

$$\min\{E(s) : s|_{V_b} = h|_{V_b}, \frac{\partial s}{\partial n}|_{V_{\partial\Omega}} = \frac{\partial}{\partial n} h|_{V_{\partial\Omega}},\ s \in S_3^1(\diamondplus)\} \tag{2.1}$$

where h is a function defined in a neighborhood of $\partial\Omega$ which is the height of the surface patches from the plane L and $\frac{\partial}{\partial n}$ denotes the normal derivative operator along $\partial\Omega$.

When the hole H is divided more than one sub-holes, there is at least one part of each $\partial\Omega_i$ is not the projection of the original H. The above problem (2.1) may be modified as follows: letting $\partial\Omega'$ be a part of the boundary which is the projection of the original H, we replace $\partial\Omega$ in (2.1) with $\partial\Omega'$.

For certain applications, we are given several data inside the hole in addition to the given surface patches. Let us write the scattered data $\{x_i, y_i, z_i), i = 1, \ldots, P\}$ over the coordinate plane L. Then we add these vertices $V = \{(x_i, y_i), i = 1, \cdots, P\}$ in when partitioning the domain Ω into quadrilaterals. Now our method to fill the hole and meet the scattered data specified is to final $s^* \in S_3^1(\diamondplus)$ solving the following constrained minimization problem:

$$\begin{aligned} \min\{E(s) : & s|_{V_b} = h|_{V_b}, \frac{\partial}{\partial n} s|_{V_{\partial\Omega}} = \frac{\partial}{\partial n} h|_{V_{\partial\Omega}}, \\ & s(x_i, y_i) = z_i, i = 1 \cdots, P, \quad s \in S_3^1(\diamondplus), \} \end{aligned} \tag{2.2}$$

We remark here that using energy functional $E(s)$ to find a spline surface fitting given scattered data is a very popular method in computer aided geometric design (cf. e.g.., [Fasshauer and Schumaker'96] and the references there in). One of the advantages of minimizing energy functional is to reduce the bumpiness of the surface. Such features for filling holes is definitely needed for applications and have not been well studied in the literature so far. We shall call the problem (2.1) is the minimal energy filling problem. Similarly, the problem (2.2) is called the minimal energy filling and fitting.

§3. Main Results and Proof

Recall $\diamondplus$ is a triangulation of Ω. Let V and E be the numbers of all vertices and all edges of quadrangulation $\Diamond$ of Ω, respectively. It is known that the dimension of $S_3^1(\diamondplus)$ is $3V + E$. Let $\phi_i, i = 1, \cdots, N(= 3V + E)$ be the locally supported basic constructed in [Lai'96]. Let B be the number of boundary edges of $\Diamond$. Without loss of anything, we may assume that $\phi_i, i = 1, \cdots, M(= 4B)$ are those locally supported basis functions whose support contains at least one of boundary edges. We use them to approximate the boundary conditions in (2.1) and (2.2). Recall $V_b = \{V_1, \cdots, V_B\}$ is the collection of boundary vertices of $\Diamond$ and $U_b = \{U_1, \cdots, U_B\}$ is the collection of the midpoint of all boundary edges of $\diamondsuit$. By the properties of locally supported basis functions ϕ_i's (cf. [Lai '96]), we can find $C_i, i = 1, \cdots, M$ such that $S_b = \sum_{i=1}^{M} C_i\phi_i$ satisfies

$$S_b(V_i) = h(V_j) \tag{3.1}$$

$$\frac{\partial}{\partial x}S_b(V_j) = \frac{\partial}{\partial x}h(V_j), \frac{\partial}{\partial y}S_b(V_j) = \frac{\partial}{\partial y}h(V_j) \tag{3.2}$$

$$\frac{\partial}{\partial n}S_b(U_j) = \frac{\partial}{\partial n}h(U_j) \tag{3.3}$$

for $j = 1, \cdots, B$. Indeed, letting $\frac{\partial}{\partial t}$ denote the tangential derivative along the boundary $\partial\Omega$, $\frac{\partial}{\partial n}h|_{V\partial\Omega}$ and $\frac{\partial}{\partial t}h|_{V\partial\Omega}$ imply $\frac{\partial}{\partial x}h|_{V\partial\Omega}$ and $\frac{\partial}{\partial y}h|_{V\partial\Omega}$. Thus, we can have (3.2). Such an S_b is uniquely determined by the interpolation conditions (3.1) –(3.3). Thus, for any $S \in S_3^1(\diamondplus)$ satisfying the boundary condition in (3.1)–(3.3), letting

$$S = S_b + \sum_{i=M+1}^{N} C_i\phi_i,$$

$E(S)$ is thus a function of the $N - M$ coefficients $\{C_i\}_{i=M+1}^{n}$.

We are now ready to prove the main results in this paper.

Theorem 3.1. *There exists a unique solution $S^* \in S_3^1(\diamondplus)$ solving problem* (2.1) *with approximate boundary conditions (3.1)–(3.3).*

Proof: Since $E(s)$ is a function of $\{C_{M+1}, \ldots, C_N\}$, in order to minimize $E(s)$ we let $\frac{\partial}{\partial C_j}E(s) = 0$ for each $C_j \in \{C_{M+1}, \ldots, C_N\}$. For notation convenience, we let

$$\oplus\phi_i = (\frac{\partial^2}{\partial x^2}\phi_i, \sqrt{2}\frac{\partial^2}{\partial x\partial y}\phi_i, \frac{\partial^2}{\partial y^2}\phi_i)^T$$

be a vector of three components and denote the inner product of three vectors of three components by

$$\langle\oplus\phi_i, \oplus\phi_j\rangle = \int_\Omega \left(\frac{\partial^2}{\partial x^2}\phi_i\frac{\partial^2}{\partial x^2}\phi_j + \frac{\partial^2}{\partial y^2}\phi_i\frac{\partial^2}{\partial y^2}\phi_j + 2\frac{\partial^2}{\partial x\partial y}\phi_i\frac{\partial^2}{\partial x\partial y}\phi_j\right) dxdy.$$

Thus taking the partial of $E(s)$ with respect to each of the $N - M$ unknown coefficients, we have

$$\begin{aligned}\frac{\partial}{\partial C_j}E(s) =& 2\sum_{t\in\Delta}\int_t \left[\left(\sum_{i=1}^{N} C_i\frac{\partial^2}{\partial x^2}\phi_i\right)\frac{\partial^2}{\partial x^2}\phi_j + \left(\sum_{i=1}^{N} C_i\frac{\partial^2}{\partial y^2}\phi_i\right)\frac{\partial^2}{\partial y^2}\phi_j + \right.\\ &\left. 2\left(\sum_{i=1}^{N} C_i\frac{\partial^2}{\partial x\partial y}\phi_i\right)\frac{\partial^2}{\partial x\partial y}\phi_j\right] dxdy\end{aligned}$$

$$
\begin{aligned}
=&2\sum_{i=1}^{N} C_i \sum_{t\in\Delta}\int_t \Bigg(\frac{\partial^2}{\partial x^2}\phi_i\frac{\partial^2}{\partial x^2}\phi_j+\frac{\partial^2}{\partial y^2}\phi_i\frac{\partial^2}{\partial y^2}\phi_j+\\
&2\frac{\partial^2}{\partial x\partial y}\phi_i\frac{\partial^2}{\partial x\partial y}\phi_j\Bigg)\,dxdy\\
=&2\sum_{i=M+1}^{N} C_i\langle\oplus\phi_i,\oplus\phi_j\rangle+2\sum_{i=1}^{M}C_i\langle\oplus\phi_i,\oplus\phi_j\rangle.
\end{aligned}
$$

To minimize $E(s)$, we want $\frac{\partial}{\partial C_j}E(s) = 0$ for all $j \in \{M+1,\ldots,N\}$. From the calculation of $\frac{\partial}{\partial C_j}E(s)$ above, it follows that

$$
\sum_{i=M+1}^{N} C_i\langle\oplus\phi_i,\oplus\phi_j\rangle = -\sum_{i=1}^{M}C_i\langle\oplus\phi_i,\oplus\phi_j\rangle
$$

for each $j \in \{M+1,\ldots,N\}$. Thus we have a system of $N-M$ equations in $N-M$ unknown coefficients $\{C_{M+1},\ldots,C_N\}$. Denote by A the coefficient matrix associated with the linear system which is an $N-M$ square matrix composed of the inner-products $\langle\oplus\phi_i,\oplus\phi_j\rangle$'s.

Let $\mathbf{c}$ be the vector containing the unknown coefficients and $\mathbf{b}$ be the right-hand side vector. Then the system $A\mathbf{c}=\mathbf{b}$ becomes:

$$
\begin{bmatrix}
\langle\oplus\phi_{M+1},\oplus\phi_{M+1}\rangle & \langle\oplus\phi_{M+2},\oplus\phi_{M+1}\rangle & \ldots & \langle\oplus\phi_N,\oplus\phi_{M+1}\rangle\\
\vdots & & & \vdots\\
\langle\oplus\phi_{M+1},\oplus\phi_N\rangle & \langle\oplus\phi_{M+2},\oplus\phi_N\rangle & \ldots & \langle\oplus\phi_N,\oplus\phi_N\rangle
\end{bmatrix}
\begin{bmatrix} C_{M+1}\\ \vdots\\ C_N\end{bmatrix}
= -\begin{bmatrix}\sum_{i=1}^{M} C_i\langle\oplus\phi_i,\oplus\phi_{M+1}\rangle\\ \vdots\\ \sum_{i=1}^{M} C_i\langle\oplus\phi_i,\oplus\phi_N\rangle\end{bmatrix}.
$$

Clearly, A is symmetric. We now further show A is positive definite.

Suppose $\mathbf{c}$ is an $(N-M)\times 1$ non-zero vector. Now

$$
\begin{aligned}
\mathbf{c}^T A\mathbf{c} =& \sum_{j=M+1}^{N}\sum_{i=M+1}^{N} c_ic_j\langle\oplus\phi_i,\oplus\phi_j\rangle\\
=&\sum_{t\in\Delta}\int_t\Bigg[\sum_{i=M+1}^{N}\sum_{j=M+1}^{N} c_ic_j\frac{\partial^2}{\partial x^2}\phi_i\frac{\partial^2}{\partial x^2}\phi_j+\sum_{i=M+1}^{N}\sum_{j=M+1}^{N} c_ic_j\frac{\partial^2}{\partial y^2}\phi_i\frac{\partial^2}{\partial y^2}\phi_j+\\
&\sum_{i=M+1}^{N}\sum_{j=M+1}^{N} c_ic_j\frac{\partial^2}{\partial x\partial y}\phi_i\frac{\partial^2}{\partial x\partial y}\phi_j\Bigg]\,dxdy\\
=&\sum_{t\in\Delta}\int_t\Bigg[(\sum_{i=M+1}^{N} c_i\frac{\partial^2}{\partial x^2}\phi_i)^2+(\sum_{i=M+1}^{M} c_i\frac{\partial^2}{\partial y^2}\phi_i)^2+2(\sum_{i=M+1}^{N} c_i\frac{\partial^2}{\partial x\partial y}\phi_i)^2\Bigg]\,dxdy.
\end{aligned}
$$

The equation above implies that $\mathbf{c}^T A\mathbf{c} \geq 0$. Suppose $\mathbf{c}^T A\mathbf{c} = 0$. Then each of the three terms in the equation above must be 0 over each triangle $t \in \Delta$. This implies

$$\sum_{i=M+1}^{N} c_i \frac{\partial^2}{\partial x^2}\phi_i = 0, \qquad \sum_{i=M+1}^{N} c_i \frac{\partial^2}{\partial y^2}\phi_i = 0, \qquad \sum_{i=M+1}^{N} c_i \frac{\partial^2}{\partial x \partial y}\phi_i = 0.$$

Letting $f(x,y) = \sum_{i=M+1}^{N} c_i\phi_i(x,y)$, then the above three equations become

$$\frac{\partial^2}{\partial x^2} f(x,y) = 0, \qquad \frac{\partial^2}{\partial y^2} f(x,y) = 0, \qquad \frac{\partial^2}{\partial x \partial y} f(x,y) = 0,$$

which implies $f(x,y)$ is a linear polynomial on each triangle $t \in \Delta$. Since $f \in C^1(\Omega)$, f is the same linear polynomial over Ω. Since $\phi_i(V_j) = 0$ for all $i \in \{M+1, \ldots, N\}$ for all boundary vertices $V_j \in V_b$, $f(x,y) = 0$ on the boundary of Ω and hence, $f(x,y) = 0$ for all $(x,y) \in \Omega$. Since $\{\phi_i\}_{i=M+1}^{N}$ are basis functions and therefore are linearly independent, we know that $C_i = 0$ for all $i \in \{M+1, \ldots, N\}$. Thus $\mathbf{c} = 0$. Therefore, $\mathbf{c}^T A\mathbf{c} > 0$ for all non-zero vectors $\mathbf{c}$ which implies A is positive definite. ■

Similarly we can prove the following theorem for problem (2.2).

Theorem 3.2. *There exists a unique solution $S^* \in S_3^1(\diamondplus)$ solving problem (2.2) with approximate boundary conditions (3.1)–(3.3).*

Certainly, s^* satisfying approximating boundary conditions (3.1) –(3.3) may not fit the given surface patches well. We may refine the quadrangulation $\diamondsuit$ by connecting the intersection of the two diagonals of each quadrilateral to the midpoint of each of its four sides. Let $\diamondsuit_1$ be such a refinement of $\diamondsuit$. $\diamondsuit_1$ is called the uniform refinement of $\diamondsuit$. In general let $\diamondsuit_n$ be the uniform refinement of $\diamondsuit_{n-1}, n = 2, \cdots$. Let $\diamondplus_n$ be the triangulation obtained from $\diamondsuit_n$. When n is large, $s^* \in S_3^1(\diamondplus_n)$ will well approximate the boundary of the surface patches along the hole H. We shall demonstrate the effectiveness of our method in the next section.

§4. Numerical Experiments

We have implemented the bivariate spline space $S_3^1(\diamondplus)$ in MATLAB and used the spline functions to fill polygonal holes. In the following four examples we demonstrate the effectiveness of our method for filling holes.

Example 1. *We consider a surface patch given on the usual coordinate plane with a polygonal hole in Fig. 2. We use the C^1 cubic splines to fill the hole in Fig. 3.*

Example 2. *We consider a surface patch which has a non-convex polygonal hole. We use a C^1 cubic spline to fill the hole in Fig. 4.*

Example 3. *We consider a surface patch given on the usual coordinate plane with a polygonal hole. We are also given several scattered data to be interpolated. We find a C^1 cubic spline to fill the hole and interpolate the given data in Fig. 5.*

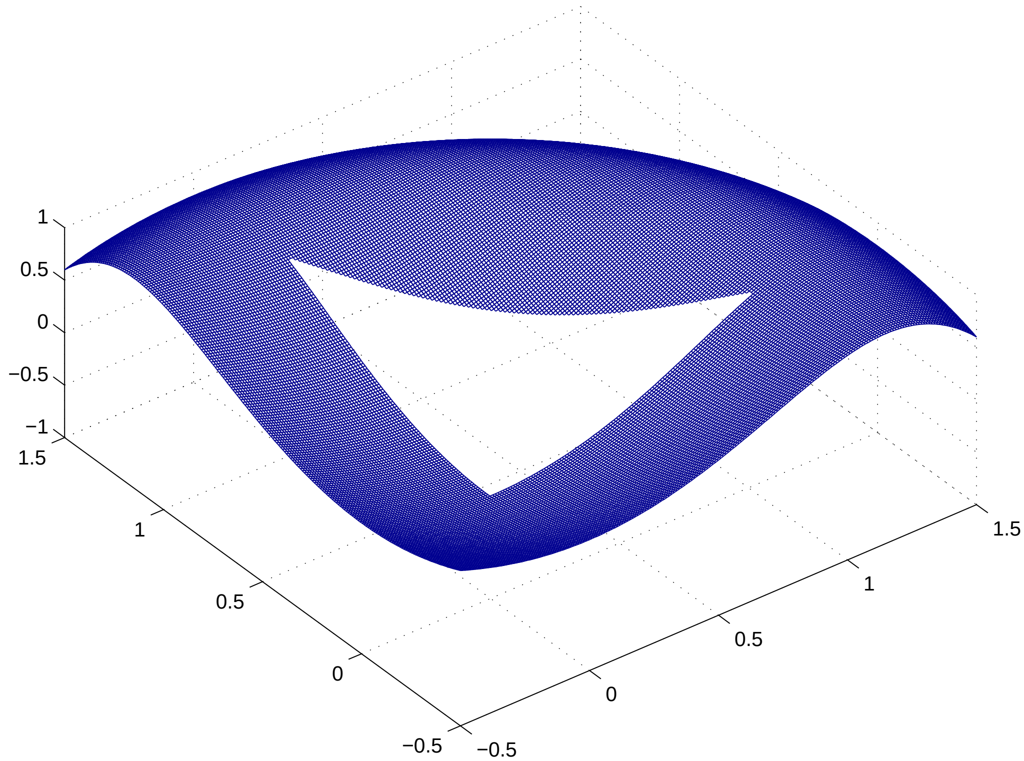

Figure 2. A surface with polygonal hole.

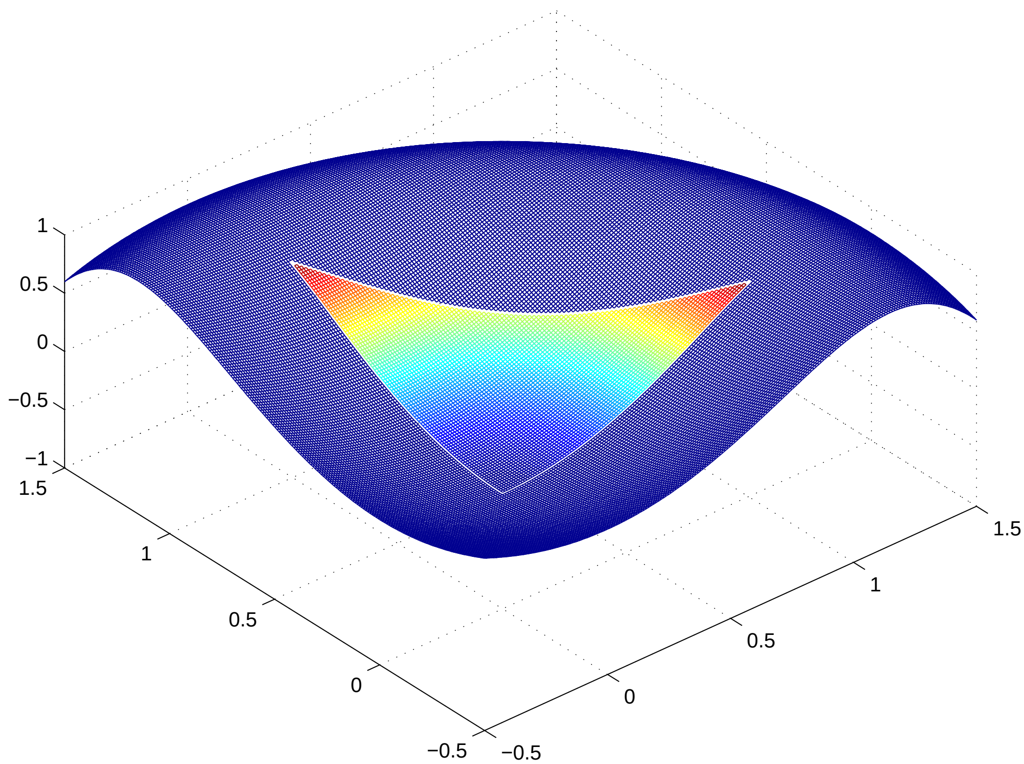

Figure 3. The surface with hole filling by a C^1 cubic spline

Example 4. *We are given two surface patches. One has a polygonal hole while the other patch is located inside the hole. We need to fill the hole in-between. See Figures 6 and 7.*

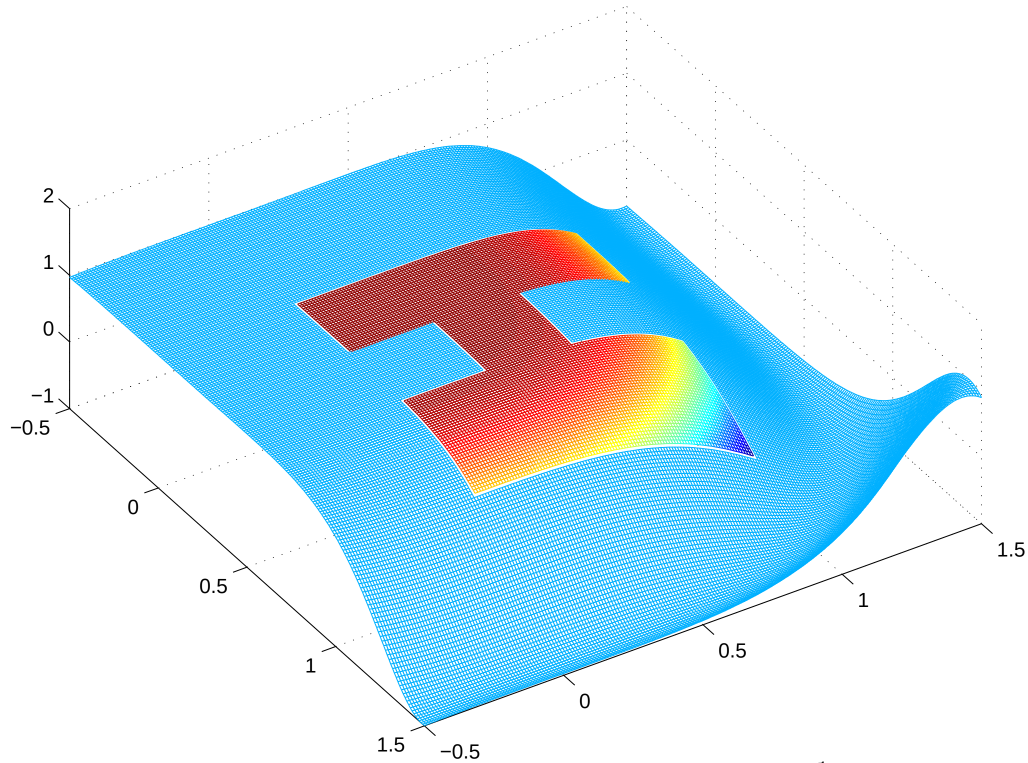

Figure 4. A surface with a non-convex hole is filled by a C^1 cubic spline

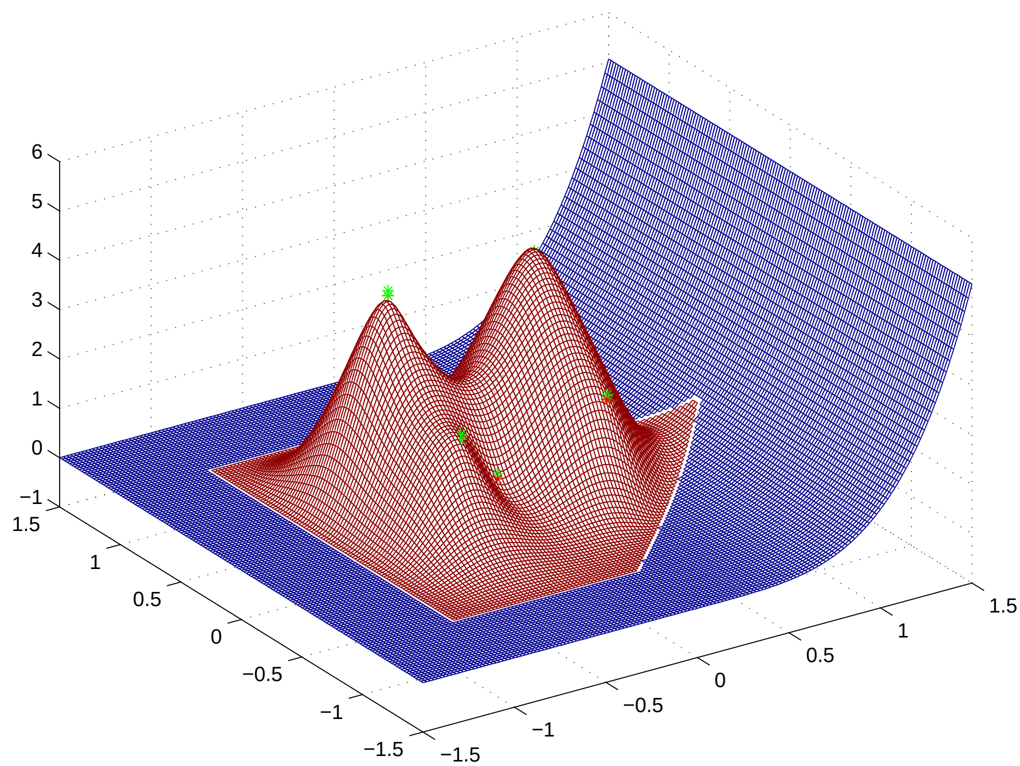

Figure 5. A C^1 cubic spline filling the hole and fitting the scattered data

Example 5. *We consider a surface patch with a hole. We divide the hole into two parts and we only fill one of them. Note that the one we fill has a free boundary.*

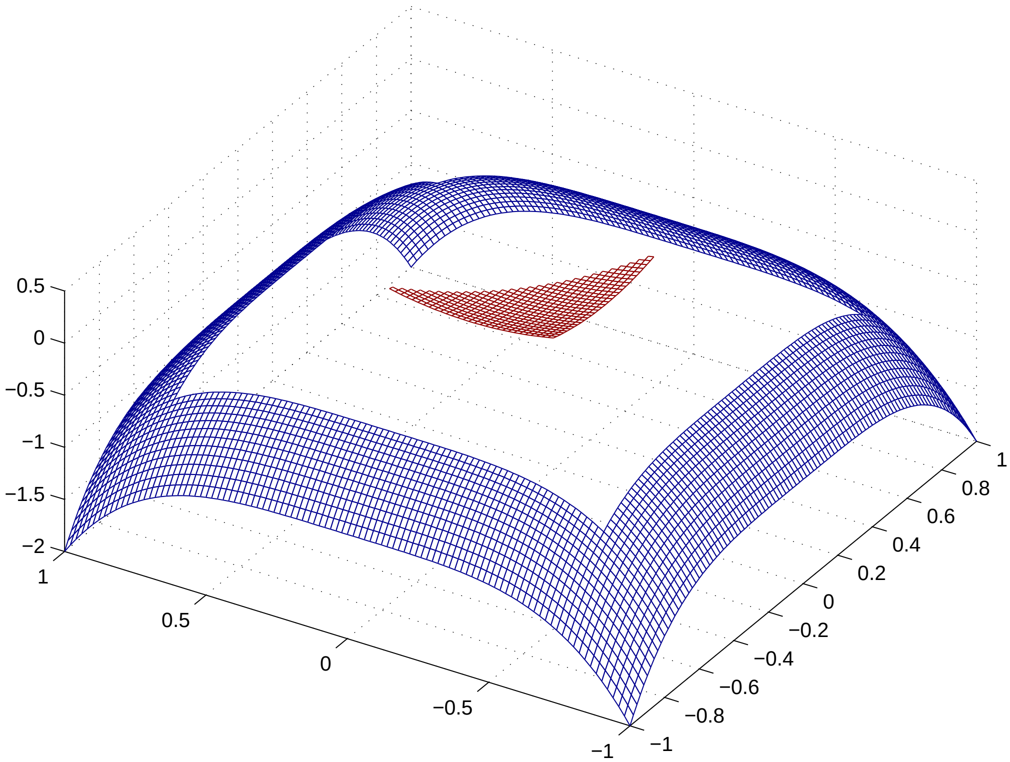

Figure 6. Two given surface patches

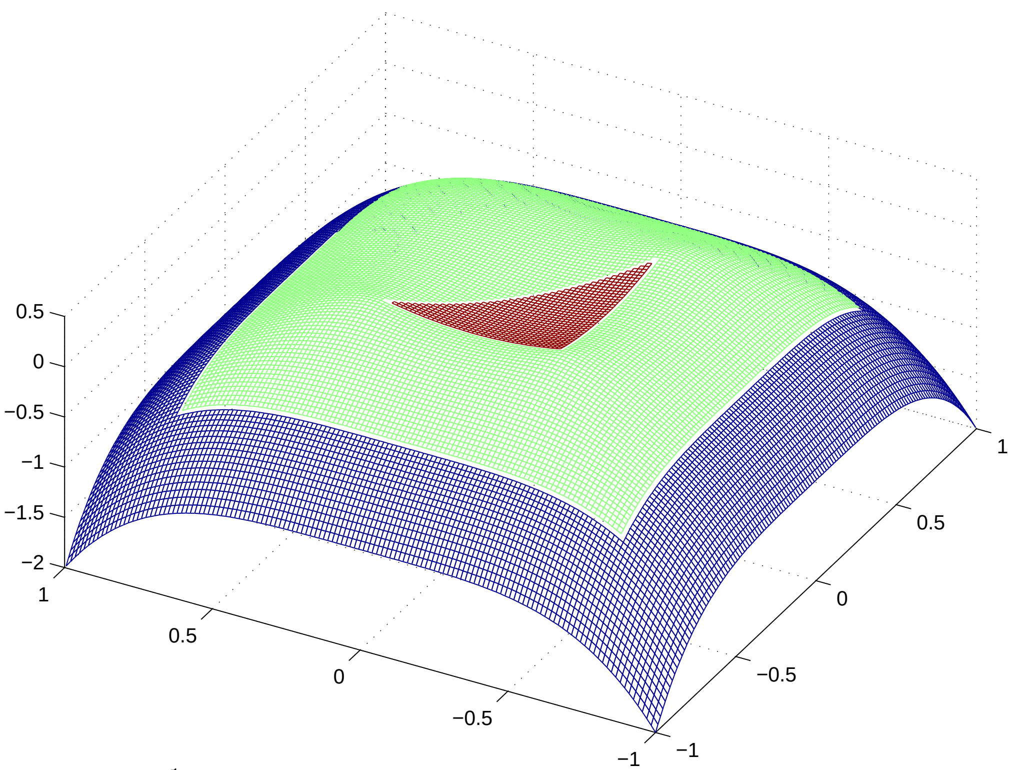

Figure 7. A C^1 cubic spline filling the hole in-between two surface patches

§5. Remarks

We have the following remarks in order.

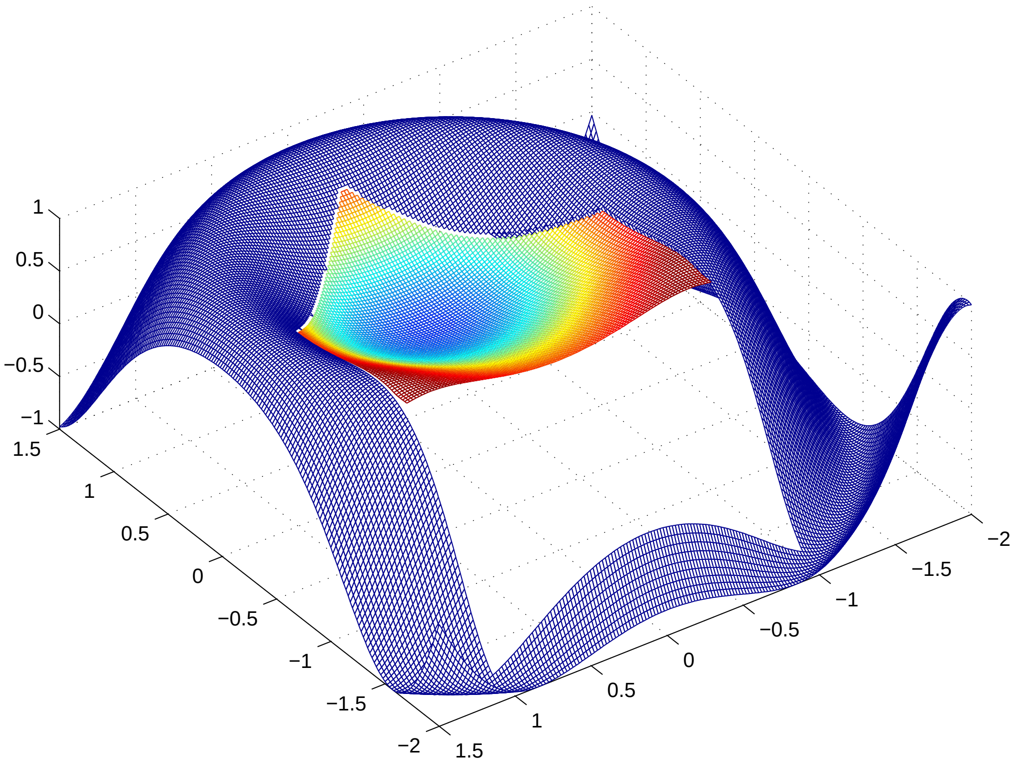

Figure 8. A C^1 cubic spline filling one of sub-holes

Remark 1. *In §2, we introduced a function h which is the height of the given surface patch to a projection plane L. It is easy to compute the function h. The computation of $\frac{\partial}{\partial n}h$ is a little bit difficult. One may estimate the quantity of $\frac{\partial}{\partial n}h$ as follows. Let $\hat{p}$ be the point of interest on the projected boundary on the plane. Offset $\hat{p}$ outwards to p and then beam both $\hat{p}$ and p up to intersect the boundary patches to obtain $h_{\hat{p}}$ and h_p. We use*

$$\frac{h_p - h(\hat{p})}{dist(p, \hat{p})}$$

to approximate $\frac{\partial}{\partial n}h$ at $\hat{p}$.

Remark 2. *Although one may use C^1 cubic splines based on Clough-Tocher's refinement Δ_{CT} of a triangulation Δ to fill the polygonal holes, we note that the dimension of $S_3^1(\Delta_{CT})$ is much larger than $S_3^1(\diamondplus)$ assuming the vertices of Δ are the vertices of $\Diamond$. For the efficiency of solving the linear systems associated with the minimal energy filling, we prefer $S_3^1(\diamondplus)$ to $S_3^1(\Delta_{CT})$.*

Remark 3. *When the given surface patches are bicubic and the projection of the hole is a polygon, the C^1 cubic minimal energy filling spline will meet the boundary of the hole exactly since the spline piece S_b on the boundary $\partial\Omega$ reproduces C^1 cubic polynomial pieces on the boundary of the hole.*

References

[1] G. Fasshauer and L. L. Schumaker, *Minimal energy surfaces using parametric splines*, Computer Aided Geometric Design, **13** (1996), pp. 45–79.

[2] J. A. Gregory and J. Zhou, *Filling polygonal holes with bicubic patches*,Comput. Aided. Geom. Design, 11(1994), pp. 391-410.

[3] J. Hahn,*Filling polygonal holes with rectangular patches, Theory and Practice of Geometric Modeling*, Spring 1989, pp. 81-91.

[4] M. J. Lai, Scattered data interpolation and approximation by C^1 piecewise cubic polynomials, Comput. Aided Geom. Design, 13(1996), pp. 81–88.

[5] M. J. Lai and L. L. Schumaker, On the approximation power of splines on triangulated quadrangulations, SIAM J. Num. Anal., vol. 36 (1999), pp. 143–159.

[6] R. Qu, *Filling polygonal holes by a subdivision algorithm*, Ann. Numer. Math. 3 (1996) pp. 333-343.

[7] D. Storry and A. A. Ball, *Design of an n-sided surface patch from Hermite data*, Comput. Aided Geom. Design, 6(1989), pp. 111–120.

[8] J. J. Zheng and A. A. Ball, *Control point surfaces over non-four-sided ares*, Comput. Aided Geom. Design, 14(1997), pp. 807–821.