Minimal energy surfaces using parametric splines

Gregory E. Fasshauer*a, Larry L. Schumakerb,1

a Department of Mathematics, Northwestern University, Evanston, IL 60208, USA
b Department of Mathematics, Vanderbilt University, Nashville, TN 37240, USA

Received March 1994; revised December 1994

Abstract

We explore the construction of parametric surfaces which interpolate prescribed 3D scattered data using spaces of parametric splines defined on a 2D triangulation. The method is based on minimizing certain natural energy expressions. Several examples involving filling holes and crowning surfaces are presented, and the role of the triangulation as a parameter is explored. The problem of creating closed surfaces is also addressed. This requires introducing spaces of splines on certain generalized triangulations.

Keywords: Minimal energy surfaces; Parametric splines; Generalized triangulations; Interpolation; Filling holes; Crowning

1. Introduction

This paper deals with the problem of constructing smooth parametric surfaces which interpolate to given scattered data in $\mathbb{R}^3$. This problem has been heavily studied in the literature. There are two main approaches:

1. construct a wireframe of curves that interpolate to the data, and then fill in the faces with a series of individual parametric patches by choosing their parameters so that they join together smoothly; see e.g. (Coons, 1964; Gordon, 1969; Peters 1991a; Sarraga, 1987, 1989, 1990; Shirman and Séquin, 1987, 1991) and references therein. This method is common in the CAGD world;

2. construct the surface as a patch complex defined parametrically in terms of coordinate functions on some convenient parameter domain, and choose their parameters to interpolate and to minimize some energy functional; see e.g. (Gmelig Meyling,
Here we consider parametric patch complexes whose components are drawn from linear spaces of polynomial splines defined on a triangulation of a parameter domain in $\mathbb{R}^2$. This leads to surfaces which exhibit $C^1$ smoothness as compared to the $G^1$ continuity achieved by most patch methods. The method involves minimizing a natural quadratic energy functional, and allows the inclusion of a penalty term to achieve approximate $C^2$ continuity of the interpolating surfaces. To illustrate the method, we present some numerical examples using cubic triangular patches for interpolating scattered data in $\mathbb{R}^3$, filling holes, and crowning surfaces. We also explore the possibility of adapting the method for closed surfaces.

The paper is organized as follows. In Section 2 we review the background material for surfaces defined by scalar-valued functions, and in Section 3 we describe the corresponding minimal energy approach. The definition of the linear space of smooth parametric splines, $\mathcal{SP}_d(\Delta)$, and a discussion of existing parametric surface fitting methods is presented in Section 4, while Section 5 deals with the energy of parametric surfaces. Interpolation of scattered points in $\mathbb{R}^3$ is the subject of Section 6, while in Section 7 we discuss filling n-sided holes and local editing of surfaces via crowning. In Section 8 we discuss the use of a $C^2$ penalty term. Finally, the problem of interpolating with closed surfaces is explored in Section 9. This requires the introduction of generalized triangulations and associated spline spaces.

2. Polynomial splines on triangulations

We begin by recalling the definition of polynomial splines on triangulations of the plane $\mathbb{R}^2$. Let $\mathcal{V} := \{(x_i, y_i)\}_{i=1}^V$ be a set of points in $\mathbb{R}^2$, and let $\mathcal{O}$ be its convex hull. A triangulation of $\mathcal{O}$ is defined as follows.

**Definition 1.** A collection $\Delta = \{T_i\}_{i=1}^N$ of triangles in the plane is called a triangulation (of $\mathcal{O}$) provided that

1. each triangle $T_i$ contains no points from $\mathcal{V}$ other than its vertices,
2. any two triangles either do not intersect, or share an edge,
3. the union $\mathcal{O}$ of the triangles $\{T_i\}_{i=1}^N$ is a connected set in $\mathbb{R}^2$.

For later use we introduce the following notation:

- $V_b :=$ number of boundary vertices,
- $V_i :=$ number of interior vertices,
- $V :=$ total number of vertices (data points),
- $E_b :=$ number of boundary edges,
- $E_i :=$ number of interior edges,
- $F :=$ total number of edges,
- $N :=$ number of triangles.
If we denote the \( \binom{d+2}{2} \) dimensional linear space of bivariate polynomials of total degree \( d \) by \( \mathcal{P}_d^{(2)} \), then given an arbitrary triangulation \( \triangle \), we can define an associated linear space of smooth piecewise polynomials.

**Definition 2.** Given two integers \( 0 \leq r < d \) and a triangulation \( \triangle \), the space of polynomial splines of degree \( d \) and smoothness \( r \) is defined by

\[
S_d^r(\triangle) := \{ g \in C^r(\Omega) : g|_{T_{\ell}} \in \mathcal{P}_d^{(2)}, \ \ell = 1, \ldots, N \}.
\]

Clearly \( S_d^r(\triangle) \) is a linear space. For the applications we have in mind, it usually suffices to work with \( C^1 \) continuity. It is known, see e.g. (Alfeld et al., 1987) and references therein, that for \( d > 3 \)

\[
\dim S_d^1(\triangle) = n_d := D + \sigma,
\]

where

\[
D := \binom{d+2}{2} + \binom{d}{2} E_1 - \frac{d^2 + 3d - 4}{2} V_1,
\]

and

\[
\sigma := \sum_{i=1}^{V_1} \sum_{j=1}^{d-1} (j + 2 - j e_i)_+.
\]

Here \( e_i \) is the number of edges with different slopes attached to the \( i \)th vertex. It is a widely accepted conjecture that this formula also holds for \( d = 3 \). Exact formulae for dimensions are known for many (but not all) other choices of \( r \) and \( d \), see e.g. (Alfeld et al., 1990; Schumaker, 1991).

For numerical purposes it is useful to think of \( S_d^1(\triangle) \) as the linear subspace of \( S_d^2(\triangle) \) defined by

\[
S_d^1(\triangle) := S_d^2(\triangle) \cap C^1(\Omega).
\]

Moreover, it is convenient to represent the elements in \( S_d^0(\triangle) \) in the well-known triangular Bernstein–Bézier form, cf. (Farin, 1986; de Boor, 1987). In particular, given \( g \in S_d^0(\triangle) \), we suppose that for each \( 1 \leq \ell \leq N \), the polynomial \( g^\ell(x, y) \) obtained by restricting \( g \) to triangle \( T^\ell \) is written in the form

\[
g^\ell(r, s, t) := \sum_{i+j+k=d} e_{i,j,k}^\ell B_{i,j,k}^d(r, s, t),
\]

where

\[
B_{i,j,k}^d(r, s, t) := \frac{d!}{i!j!k!} r^i s^j t^k, \quad i + j + k = d,
\]

are the classical bivariate Bernstein polynomials, and where \( (r, s, t) \) are the barycentric coordinates of the point \( (x, y) \) with respect to the vertices \( v_1^\ell, v_2^\ell, v_3^\ell \) of \( T^\ell \). As usual, we
associate the coefficients \( c_{i,j,k} \) with the equally spaced points \( \xi_{i,j,k} := (iv_1 + jv_2 + kv_3)/d \). It will be convenient to introduce the notation

\[
\xi_{i,j,k} := (\xi_{i,j,k}^1, \xi_{i,j,k}^2, \xi_{i,j,k}^3)
\]

for the corresponding control points lying in \( \mathbb{R}^3 \). The collection of all such control points is called the control net or Bézier net. We say that \( c_j \) is at the vertex \( v_j \), while \( c_{d-j-1,0} \) and \( c_{d-1,0} \) are in the first ring around \( v_1 \). Similarly, we say that \( c_{d-j,0} \), \( j = 0, \ldots, d \), are on the edge \( v_1v_2 \).

To force \( C^1 \) continuity between adjoining patches, we identify the coefficients on the common edges between neighboring triangles. If we renumber the remaining coefficients, we get a vector \( c := [c_1, \ldots, c_n]^T \) with

\[
n_c := V + (d - 1)E + \binom{d - 1}{2}N.
\]

Each spline in the space \( S^d_0(\Delta) \) is uniquely defined by its coefficient vector \( c \).

It is well known (cf. (Farin, 1986; de Boor, 1987)), that a spline in \( S^d_0(\Delta) \) will belong to \( C^1(\Omega) \) if and only if its coefficients satisfy a homogeneous linear system of equations of the form

\[
Hc = 0.
\]

Here \( c \) is the vector of coefficients defined above, and \( H \) is a sparse matrix of size \( n_{\text{cont}} \times n_c \), where \( n_{\text{cont}} \) is the total number of continuity conditions which are imposed.

Two patches sharing a common edge as in Fig. 1 will join together with \( C^1 \) continuity if and only if the following set of linear conditions are satisfied:

\[
c_i = rb_i + sa_i + tb_{i+1}, \quad i = 1, \ldots, d,
\]

where \( (r, s, t) \) are the barycentric coordinates of \( v_4 \) with respect to the triangle with vertices \( v_1, v_2, v_3 \), and the \( a_i, b_i, c_i \) are as shown in the figure. This implies that each row of the matrix \( H \) in (2.4) will contain at most four nonzero entries.

The \( C^1 \) continuity conditions (2.5) can also be expressed geometrically: they are satisfied if and only if

\[
a_i, b_i, b_{i+1}, c_i \text{ are coplanar}, \quad i = 1, \ldots, d.
\]
Since there are $d$ conditions for each edge, to enforce $C^1$ continuity, we have a total of
\[ n_{\text{cont}} = dE_1 = 3dV_l + dV_B - 3d \quad (2.7) \]
conditions. It is well known that not all of these conditions are independent; there are a number of redundancies. In using (2.4) as side conditions for minimization problems, it is important to remove as many of the redundant rows of $H$ as possible.

**Lemma 3.** The number of rows $n_r$ in the matrix $H$ corresponding to $C^1$ continuity conditions which are redundant is

\[
\begin{cases} 
    n_r = 2V_l + \sigma, & d \geq 4, \\
    n_r \geq 2V_l + \sigma, & d = 3.
\end{cases}
\]

**Proof.** Using (2.1), for $d \geq 4$ we know
\[
n_d = \dim S_j^l (\triangle) = \frac{(d + 1)(d + 2)}{2} + \frac{d(d - 1)}{2} E_1 - \frac{(d^2 + 3d - 4)}{2} V_l + \sigma
\]
\[= (d^2 - 3d + 2)V_l + \frac{d(d - 1)}{2} V_B - (d^2 - 3d - 1) + \sigma.
\]
The second expression follows immediately from the first and the well-known formulae
\[ E_B = V_B, \]
\[ E_1 = 3V_l + V_B - 3, \]
\[ N = 2V_l + V_B - 2. \]
Using these formulae on (2.3) also, we see that the number of coefficients in the vector $c$ is
\[ n_c = d^2V_l + \frac{d(d + 1)}{2} V_B - d^2 + 1. \]
It then follows that the number of linearly independent $C^1$ continuity conditions is
\[ n_{\text{ind}} := n_c - n_d = (3d - 2)V_l + dV_B - 3d - \sigma. \]
The number of redundant equations is then given by
\[ n_r = n_{\text{cont}} - n_{\text{ind}}, \]
where $n_{\text{cont}}$ is as in (2.7). For the case $d = 3$, formula (2.1) gives a lower bound for the dimension (see e.g. (Schumaker, 1979)), and the proof proceeds as above. □

As we mentioned above, it is not only important to know how many redundant rows the matrix $H$ in (2.4) has, but where precisely these redundancies are located. The following lemmas identify exactly where the redundancies are to be found.

**Lemma 4.** Let $v$ be an interior vertex with $m$ edges attached to it as in Fig. 2, and let $\lambda_i, i = 1, \ldots, m$, denote the associated $C^1$ continuity conditions in the first layer.
around \( v \). If the first \( m - 2 \) of these conditions are satisfied for some spline \( g \) in \( S_d^m(\triangle) \), then the two remaining conditions associated with edges \( m - 1 \) and \( m \) are automatically satisfied.

**Proof.** Clearly, the \( C^1 \) conditions defined by \( \lambda_1, \ldots, \lambda_{m-2} \) hold if and only if all of the points \( a_0, \ldots, a_m \) lie in one plane. But then the conditions associated with \( \lambda_{m-1} \) and \( \lambda_m \) automatically hold. \( \square \)

The following lemma shows that at a singular vertex, in addition to the two redundancies mentioned in Lemma 4, one of the continuity conditions in the second ring around \( v \) is always redundant (see also (Gmelig Meyling, 1987)).

**Lemma 5.** Let \( v \) be a singular interior vertex \( v \) as in Fig. 3. Then if any set of three of the continuity conditions in the second ring around \( v \) is satisfied for a spline \( g \) in \( S_d^m(\triangle) \), then the fourth second-ring condition is also automatically satisfied.
Proof. We show that one of the second-ring conditions depends on the other three second-ring conditions and two of the conditions in the first ring around \( v \). If we label the coefficients as in Fig. 3, we can write these six continuity conditions involved in matrix form as

\[
\begin{pmatrix}
    t_1 & r_1 & s_1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
    t_2 & -1 & r_2 & s_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
    0 & t_1 & 0 & 0 & 0 & s_1 & 0 & 0 & -1 & r_1 & 0 & 0 \\
    0 & 0 & t_2 & 0 & 0 & -1 & s_2 & 0 & 0 & r_2 & 0 & 0 \\
    0 & 0 & 0 & t_3 & 0 & 0 & -1 & s_3 & 0 & 0 & r_3 & 0 \\
    0 & 0 & 0 & 0 & t_4 & 0 & 0 & -1 & s_4 & 0 & 0 & r_4
\end{pmatrix}
\begin{pmatrix}
a_0 \\
a_1 \\
a_2 \\
a_3 \\
a_4 \\
b_1 \\
b_2 \\
b_3 \\
b_4
\end{pmatrix} = 0,
\]

where \((r_i, s_i, t_i)\) are the barycentric coordinates of \( v_{i-1} \) with respect to the triangle with vertices \( v_i, v_{i+1}, \) and \( v \) (and the counting is cyclically modulo 4). From the geometry of the singular vertex we know that all \( r_i \) are zero. Therefore, after dropping the last four columns and making some column and row interchanges, we obtain the following smaller matrix with the same rank as before:

\[
\begin{pmatrix}
    -1 & 0 & s_1 & 0 & 0 & 0 & 0 & 0 & t_1 \\
    0 & -1 & 0 & s_2 & 0 & 0 & 0 & 0 & t_2 \\
    0 & 0 & t_2 & 0 & 0 & -1 & s_2 & 0 & 0 \\
    0 & 0 & 0 & t_3 & 0 & 0 & -1 & s_3 & 0 \\
    0 & t_1 & 0 & 0 & -1 & s_1 & 0 & 0 & 0 \\
    0 & 0 & 0 & 0 & s_4 & 0 & 0 & -1 & 0
\end{pmatrix}
\]

We show this matrix has rank 5. Performing Gaussian elimination, we obtain

\[
\begin{pmatrix}
    -1 & 0 & s_1 & 0 & 0 & 0 & 0 & 0 & t_1 \\
    0 & -1 & 0 & s_2 & 0 & 0 & 0 & 0 & t_2 \\
    0 & 0 & t_2 & 0 & 0 & -1 & s_2 & 0 & 0 \\
    0 & 0 & 0 & t_3 & 0 & 0 & -1 & s_3 & 0 \\
    0 & 0 & 0 & -1 & s_1 & 0 & 0 & 0 & t_1t_2 \\
    0 & 0 & 0 & 0 & s_4 & 0 & 0 & -1 & t_1t_2
\end{pmatrix}
\]

Now multiplying the fifth row by \( s_4 \) and adding it to the last row changes all of the entries in the last row to zero. To see this, we use the identities of Lemma 6 below and the fact that \( s_i + t_i = 1 \) (since \( r_i = 0 \)) for \( i = 1, \ldots, 4 \).

Lemma 6. At a singular vertex \( v \), the barycentric coordinates satisfy the identities

\[s_1s_3 = 1, \quad s_2s_4 = 1.\]
Proof. Given the configuration of Fig. 3, we first write \( v_4 \) in terms of \( v_1, v_2 \) and \( v \) to obtain

\[
u_4 = s_1v_2 + t_1v. \tag{2.8}\]

Then we write \( v_2 \) in terms of \( v_3, v_4 \) and \( v \) to get

\[
u_2 = s_3v_4 + t_3v. \tag{2.9}\]

Now substituting (2.9) into (2.8), we have

\[
u_4 = s_1s_3v_4 + (s_1t_3 + t_1)v. \tag{2.10}\]

Since this equation must hold for arbitrary \( v_4 \) and \( v \), the first of the two identities follows. The second is obtained similarly. \( \square \)

3. Minimal energy interpolating splines

Let \( S^0_d(\Delta) \) be the space of \( C^0 \) continuous splines of degree \( d \) defined on a triangulation \( \Delta \) of a domain \( \Omega \). Each spline in this space is uniquely defined by the vector \( c \) of coefficients as described in Section 2.

We want to use this space of splines to solve various interpolation problems. We shall be interested in sets of \( n_{\text{int}} \) interpolation conditions which can be written in the form

\[
Bc = h, \tag{3.1}
\]

where \( B \) is a prescribed \( n_{\text{int}} \times n_c \) matrix, and \( h = (h_1, \ldots, h_{n_{\text{int}}}) \) is a prescribed vector. Each row of \( B \) corresponds to one interpolation condition. If a row of \( B \) corresponds to interpolation of function values and derivatives at specified points in \( \Omega \), then it contains at most \( \left( \frac{d+2}{2} \right) \) entries, see e.g. (Farin, 1986). (For example, the row of \( B \) corresponding to interpolation of a value at a vertex of \( \Delta \) has just one element in it, namely a 1 in the position corresponding to the coefficient associated with that vertex in the Bézier net.)

To get an interpolating spline in \( S^0_d(\Delta) \) which is \( C^1 \), we also have to enforce the smoothness conditions (2.4). In view of the discussion in Section 2, some of the equations in (2.4) are redundant. Removing the redundant equations, the \( C^1 \) continuity conditions reduce to the homogeneous system

\[
Ac = 0, \tag{3.2}
\]

where \( A \) is now an \( n_{\text{cont}} \times n_c \) matrix with \( n_{\text{cont}} = n_{\text{cont}} - n_t \). Here \( n_{\text{cont}} \) is the number of continuity conditions (2.7), and \( n_t \) is the number of those that are redundant (cf. Lemma 3).

Depending on the interpolation conditions to be enforced, there may not exist any such spline. Thus for the rest of this section we assume that the following hypothesis is satisfied:

\[
\mathcal{U} := \{ g \in S^0_d(\Delta) : Ac = 0 \text{ and } Bc = h \} \neq \emptyset. \tag{3.3}
\]
While $\mathcal{U}$ itself is not a linear subspace, it is the translate of one, say $\mathcal{U}^0$. Assuming (3.3) holds, let $n_f$ be the dimension of $\mathcal{U}^0$. Then $n_f$ is the number of free parameters for elements in $\mathcal{U}$. In order to get visually pleasing smooth surfaces, it has been suggested by several authors (e.g. (Schmidt, 1982; Alfeld, 1984b; Gmelig Meyling, 1987)) to use these free parameters to minimize some kind of energy functional for the surface.

As can be found in classical books on mechanics (e.g. (Timoshenko, 1959)), the energy of a thin plate of arbitrary shape under small deflections is given by

$$\int \int \left\{ (f_{xx} + f_{yy})^2 - 2(1 - \nu)(f_{xx}f_{xy} - f_{xy}^2) \right\} \, dx \, dy,$$

where the parameter $\nu$ is a constant (Poisson's ratio) depending on the material at hand. We choose $\nu = 0$ to simplify the integrand as much as possible ($\nu = 0.3$ might be a more realistic setting, since it corresponds to aluminum or steel). This means that on each triangle $T_i$, we should measure the energy by

$$J_i := \int \int_{T_i} \left\{ (f_{xx})^2 + 2(f_{xy})^2 + (f_{yy})^2 \right\} \, dx \, dy,$$

so that the total energy is given by

$$J := \sum_{i=1}^{N} J_i. \quad (3.4)$$

We can now define what we mean by a minimal energy spline interpolant.

**Definition 7.** A spline $g \in \mathcal{U}$ is called a minimal energy $C^1$ interpolating spline provided

$$J(g) \leq J(u) \quad \text{for all } u \in \mathcal{U}.$$

It turns out that the coefficients (Bézier net) of a minimal energy interpolating spline can be found by solving a sparse system of linear equations. To state this formally, we need some notation. As shown in (Quak and Schumaker, 1990), the energy expression (3.4) can be written as a quadratic form

$$J = c^T Q c, \quad (3.5)$$

where $Q$ is an $n_c \times n_c$ symmetric positive semi-definite matrix. It was also shown in (Quak and Schumaker, 1990) that for cubic splines, the matrix $Q$ has 13 essential entries, while in the quartic case, there are 26. The formulae for these entries are explicitly listed there in terms of inner products of gradient vectors of the barycentric coordinates $(r, s, t)$ of a point $(x, y)$. Using these formulae, the energy matrix $Q$ can be assembled without performing any integrations. This is a major advantage over any method which uses a more complicated energy functional (such as the methods mentioned in Section 5), or which uses numerical integration techniques to evaluate the energy integral.
Suppose \( A \) is the \( n_{\text{ind}} \times n_c \) matrix of (3.2) describing the continuity conditions for a spline \( g \in \mathcal{S}\) to lie in \( \mathcal{S}\). Suppose \( B \) is an \( n_{\text{int}} \times n_c \) matrix describing a set of \( n_{\text{int}} \) interpolation conditions as in (3.1). Then the following is well known:

**Theorem 8.** Assume (3.3) holds. Then there is a unique solution to the following constrained quadratic minimization problem:

\[
\min c^T Q c, \quad \text{subject to } Ac = 0, \quad Bc = h.
\]

It can be found by solving the linear system

\[
\begin{pmatrix}
  Q & C^T \\
  C & 0
\end{pmatrix}
\begin{pmatrix}
  c \\
  \lambda
\end{pmatrix}
= \begin{pmatrix}
  0 \\
  b
\end{pmatrix},
\]

where

\[
C := \begin{pmatrix}
  A \\
  B
\end{pmatrix}, \quad b := \begin{pmatrix}
  0 \\
  h
\end{pmatrix},
\]

and \( \lambda \) is an \( n_{\text{ind}} + n_{\text{int}} \) vector (whose components are called Lagrange multipliers). Moreover, the spline \( g \) with Bézier net \( c \) is the minimal energy interpolating spline.

In the next sections we will show how this approach can be carried over to the construction of parametric surfaces.

4. Parametric surfaces

First we briefly review the difficulties inherent in the parametric setting. In recent years there has been considerable interest in \( G^1 \) (oriented tangent plane or geometric) continuity. One reason for this is that, in general, it is not possible to construct closed surfaces defined over a standard parameter domain in \( \mathbb{R}^2 \) which maintain parametric \( C^1 \) continuity (Herron, 1985).

Geometric continuity is weaker than parametric \( C^1 \) continuity, and allows more flexibility in constructing smooth surfaces. However, it has one major drawback: the space of \( G^1 \) splines is a nonlinear space. This is due to the fact that the necessary and sufficient \( G^1 \) continuity conditions are nonlinear (see e.g. (Farin, 1986)). One way to formulate the conditions for tangent plane continuity is

\[
\det(D_c, D_R, D_L) = 0,
\]

where \( D_c \) stands for the derivative along the common boundary curve of two adjoining surface patches, and \( D_R \) and \( D_L \) denote the cross-boundary derivatives into the right and the left patch respectively.

Many attempts have been made to derive sufficient (linear) conditions for \( G^1 \) continuity (see e.g. (Farin, 1986; Piper, 1987; Sarraga, 1987, 1989, 1990; Shirman and Séquin, 1987, 1991; Hahn, 1989a,b; Peters 1990b, 1991a)). There has also been a discussion of more theoretical concepts, such as dimensions, for these "linearized" \( G^1 \) spline spaces.
However, the surface interpolation schemes in the literature not only share the common problem of how to linearize, but the need for a sensible way of determining the remaining free coefficients. The various methods use different heuristics to give the user so-called shape parameters to obtain nice looking surfaces. Most of these schemes have default values which produce surfaces with a poor distribution of curvature. This fact was pointed out in the extensive study of many of the aforementioned schemes in (Lounsbery et al., 1992) and (Mann et al., 1992).

We examine a different approach to smooth parametrically defined surfaces. Instead of working with reparametrizations of one standard parameter domain for all pieces, such as the unit square or unit triangle, we use a different parameter domain for each individual polynomial piece, i.e. in our case the reparametrizations are built into one overall parameter domain. In particular, we use a triangulation as the parameter domain for our surface and define each coordinate as a scalar-valued spline as described in Section 2. We can therefore define a linear space of smooth parametric splines.

**Definition 9.** Given two integers $0 \leq r < d$ and a triangulation $\triangle$ of the parameter domain in the $(u, v)$ plane, the space of parametric polynomial splines of degree $d$ and smoothness $r$ is defined by

$$\mathcal{S}_d^r(\triangle) := \{(g_1, g_2, g_3)^T : g_j \in \mathcal{S}_d^r(\triangle), j = 1, 2, 3\}. \quad (4.1)$$

Using a triangulation as the parameter domain for the surface corresponds to affine reparametrizations in the general theory of geometric continuity for parametric surfaces. This means we have a $(1, 1, 1)$ match in the sense of (Peters, 1990b, 1991a). As Peters has shown, this does not allow for the full flexibility of (nonlinear) reparametrizations, but in almost all $G^1$ schemes in the literature the authors do not take full advantage of this possibility either.

However, there are great advantages to working with a fixed triangulation of the parameter domain and treating each coordinate of our parametric patch as a spline in $\mathcal{S}_d^r(\triangle)$. For one thing we do not have to worry at all about the vertex enclosure problem which gives so much trouble in the existing literature on smooth parametric surfaces (and which has up to date only been solved for the symmetric case, see e.g. (Van Wijk, 1986; Sarraga, 1987, 1989, 1990; Du, 1988; Hahn, 1989b; Peters, 1992)). Our "reparametrizations" are as simple as can be: we reparametrize with the identity and therefore match Hahn's constraint (Hahn, 1989a, Theorem 7.1) trivially.

From a computational point of view it is also very advantageous to treat each coordinate separately, as we will show in the next section.

5. Minimal energy interpolating parametric surfaces

Some early remarks on minimization of an energy expression in order to obtain smooth parametric surfaces can be found in (Hosaka, 1969). However, no details are given there. Most work on this subject has been done in several recent papers (see e.g.
Hagen and Schulze use a variational approach for the nonlinear energy functional
\[
\int_S (\kappa_1^2 + \kappa_2^2) \, dS,
\]
where \(\kappa_1\) and \(\kappa_2\) are the principal curvatures of the surface \(S\). They use biquintic patches to obtain \(G^2\) smooth surfaces.

Moreton and Sequin choose to minimize the variation in the derivatives of the curvature, i.e.
\[
\int_S \left[ \left( \frac{d\kappa_n}{d\hat{e}_1} \right)^2 + \left( \frac{d\kappa_n}{d\hat{e}_2} \right)^2 \right] \, dS,
\]
where \(\kappa_n\) is the normal curvature, and \(\hat{e}_1\) and \(\hat{e}_2\) are the principal directions. They evaluate this energy integral numerically using Lobatto quadrature. Moreton and Sequin also use biquintic patches but do not even enforce full \(G^1\) continuity. Instead they only add the necessary and sufficient \(G^1\) conditions of DeRose (1990) as a penalty term (see also Section 7). In this way they achieve approximate \(G^1\) continuity similarly to (DeRose and Mann, 1992). However, they do have \(G^2\) continuity at the vertices (a sufficient condition proposed by Peters (1991a) to avoid the vertex enclosure problem).

The method proposed by Celniker and Gossard uses an energy functional very similar to ours, but they also evaluate the energy integrals by Gaussian quadrature. Their method does not allow for arbitrary closed surfaces and uses "12 dof" triangles, corresponding to the well-known Clough-Tocher split (see also Section 6).

In (Kallay, 1992) it is not clear whether the author calculates the entries of his matrix \(A\) and right hand side \(b\) in advance, or whether this is done numerically. Also Kallay's analysis is for tensor-product \(B\)-spline surfaces only, which are not as flexible as triangular elements.

Most of the above methods are computationally quite expensive since they either work with nonlinear energy expressions, or use numerical integration techniques to evaluate the quadratic energy integrals.

We now propose a method which avoids all of the difficulties mentioned above, and cuts the amount of work required to solve the interpolation problem to essentially one third by treating each coordinate separately. In particular, given a parametric spline \(g \in \mathcal{S}^d_{\Delta}\), we define its energy by
\[
\tilde{J} := \sum_{\ell=1}^N \tilde{J}_\ell, \quad (5.1)
\]
where now
\[
\tilde{J}_\ell\cdot(g) := \int_{\tau^\ell} (g_{uu} \cdot g_{uu} + 2g_{uv} \cdot g_{uv} + g_{vv} \cdot g_{vv}) \, du \, dv. \quad (5.2)
\]
As before, $g = [g_1(u,v), g_2(u,v), g_3(u,v)]^T$, and "·" denotes the dot product. Multiplying out the dot products and regrouping we obtain

$$\mathcal{J}_{V}(g) = \int_{T} \left( (g_{1uu})^2 + 2(g_{1uv})^2 + (g_{1vv})^2 \right) \, du \, dv$$

$$+ \int_{T} \left( (g_{2uu})^2 + 2(g_{2uv})^2 + (g_{2vv})^2 \right) \, du \, dv$$

$$+ \int_{T} \left( (g_{3uu})^2 + 2(g_{3uv})^2 + (g_{3vv})^2 \right) \, du \, dv,$$

which shows that the energy decouples, and that the total energy of the surface patch is equal to the sum of the coordinate energies.

Suppose now that the Bézier nets associated with the splines $g_1$, $g_2$, $g_3$ are $c_x$, $c_y$, and $c_z$, respectively. Then the parametric spline $g$ is $C^1$ on $\triangle$ provided that

$$\tilde{A} \begin{pmatrix} c_x \\ c_y \\ c_z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix},$$

where

$$\tilde{A} := \begin{pmatrix} A & 0 & 0 \\ 0 & A & 0 \\ 0 & 0 & A \end{pmatrix},$$

and $A$ is the matrix in (3.2). To force $g$ to interpolate values in $\mathbb{R}^3$ with respect to the linear functionals defined by the matrix $B$ in (3.1), we can simply require that each of its components interpolate given values, i.e., if $h_x$, $h_y$, and $h_z$ are given vectors of $n_{int}$ real numbers, we require

$$\tilde{B} \begin{pmatrix} c_x \\ c_y \\ c_z \end{pmatrix} = \begin{pmatrix} h_x \\ h_y \\ h_z \end{pmatrix}, \quad (5.3)$$

where

$$\tilde{B} := \begin{pmatrix} B & 0 & 0 \\ 0 & B & 0 \\ 0 & 0 & B \end{pmatrix}.$$

Now a minimal energy interpolating parametric spline is defined exactly as in Definition 7, but using the energy expression (5.1) and the set of splines

$$\tilde{\mathcal{U}} := \{g \in \mathcal{S} \mathcal{P}_d^0(\triangle): \tilde{A}c = 0 \text{ and } \tilde{B}c = h\}, \quad (5.4)$$

where $c = (c_x, c_y, c_z)^T$ and $h = (h_x, h_y, h_z)^T$. 
Definition 10. A spline \( g \in \tilde{U} \) is called a minimal energy \( C^1 \) interpolating parametric spline provided

\[
\tilde{J}(g) \leq \tilde{J}(u) \quad \text{for all} \quad u \in \tilde{U}.
\]

As in Section 2, the coefficients (Bézier net) of a minimal energy interpolating spline can be found by solving a sparse system of linear equations. Let \( \tilde{Q} \) be the \( 3n_c \times 3n_c \) matrix

\[
\tilde{Q} := \begin{pmatrix} Q & 0 & 0 \\ 0 & Q & 0 \\ 0 & 0 & Q \end{pmatrix}.
\]

Theorem 11. Assume \( \tilde{U} \neq \emptyset \). Then there is a unique solution to the following constrained quadratic minimization problem:

\[
\min (c_x^T c_x, c_y^T c_y, c_z^T c_z) \tilde{Q} \begin{pmatrix} c_x \\ c_y \\ c_z \end{pmatrix},
\]

subject to

\[
\tilde{A} \begin{pmatrix} c_x \\ c_y \\ c_z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad \tilde{B} \begin{pmatrix} c_x \\ c_y \\ c_z \end{pmatrix} = \begin{pmatrix} h_x \\ h_y \\ h_z \end{pmatrix}.
\]

It can be found by solving the linear system with multiple right-hand sides

\[
\begin{pmatrix} Q & C^T \\ C & 0 \end{pmatrix} \begin{pmatrix} \lambda_x \\ \lambda_y \\ \lambda_z \end{pmatrix} = \begin{pmatrix} b_x \\ b_y \\ b_z \end{pmatrix},
\]

where \( C \) is the matrix in (3.6) and

\[
b_x := \begin{pmatrix} 0 \\ h_x \end{pmatrix}, \quad b_y := \begin{pmatrix} 0 \\ h_y \end{pmatrix}, \quad b_z := \begin{pmatrix} 0 \\ h_z \end{pmatrix}.
\]

Here \( \lambda_x, \lambda_y, \lambda_z \) are \( n_{\text{ind}} + n_{\text{nt}} \) vectors (whose components are the Lagrange multipliers). Moreover, the parametric spline \( g \) with Bézier net \( c \) is the minimal energy interpolating spline.

Proof. Because of the nature of the matrices \( \tilde{A}, \tilde{B}, \) and \( \tilde{Q} \), this quadratic minimization problem decouples into three problems of the same type as in Theorem 8. □

The linear systems that arise from this global surface construction grow quite rapidly with the size of the triangulation. Therefore it is essential to have fast and accurate methods for solving large (partially sparse) linear systems. We have experimented with several methods for solving the problem: direct sparse matrix solvers, cf. (Duff, 1977; Sherman, 1978; Zlatev et al, 1981; Eisenstat et al., 1982), iterative sparse matrix solvers (conjugate gradient, generalized minimum residual, iterative refinement)
three-stage relaxation methods on the energy part (Dyn and Ferguson, 1983), quadratic minimization (Powell, 1989). Certainly, those methods which consist of a factorization as well as a solve phase have a considerable advantage in treating problems with multiple right-hand sides. We have found that we can reliably solve problems with system matrices of size up to 3000 × 3000 in only a few seconds on a Silicon Graphics Iris4D workstation using direct methods. Most other methods require setting a number of parameters whose default choice is not obvious for our type of problems, since the structure of the system matrix depends very heavily on the geometry of the domain triangulation.

Due to the global nature of our method, we find that it is particularly well suited for constructing smooth interpolating surfaces with moderately sized constellations, or localized problems such as filling n-sided holes, filleting or local modifications of existing surfaces via crowning. We discuss some of these applications in Sections 6 and 7.

One possibility for dealing with interpolation problems involving a larger number of points would be to create several surfaces which interpolate locally to smaller subsets of the data, and then blend the constructed parts together to form an overall smooth surface. This would also permit working on smaller pieces of the problem where the surface is relatively “flat”.

6. Interpolation of points in \( \mathbb{R}^3 \)

In this section we consider the following problem.

**Problem 12.** Suppose \( \Gamma := \{ (y_i^1, y_i^2, y_i^3) \}_{i=1}^n \) are given points in \( \mathbb{R}^3 \), and that \( \mathcal{F} := \{ F_i \}_{i=1}^n \) is a set of nonintersecting triangles in \( \mathbb{R}^3 \) with vertices at the points \( \Gamma \). Let \( S \) be the surface in \( \mathbb{R}^3 \) defined by the union of the triangles in \( \mathcal{F} \). Find an interpolating parametric surface which is \( C^1 \), and which has a shape similar to \( S \).

The surface \( S \) can be thought of as a kind of “control surface”. It is continuous and passes through the points in the set \( \Gamma \). For the remainder of this section, we assume that it is a relatively “flat” open surface (we come back to the general case including closed surfaces in Section 9).

**Algorithm 13.**

1. Choose a plane \( \pi \) in \( \mathbb{R}^3 \) which is as “parallel” to the surface \( S \) as possible, and introduce a \( (u, v) \) Cartesian coordinate system on \( \pi \). Choose a projection direction.
2. For each \( i = 1, \ldots, n \), let \( (u_i, v_i) \) be the point on \( \pi \) obtained by parallel projecting the point \( (y_i^1, y_i^2, y_i^3) \) onto \( \pi \).
3. Let \( \Delta \) be the triangulation in \( \pi \) obtained by projecting the edges of the triangles \( F_i \) onto \( \pi \).
4. Find a minimal energy parametric spline \( g = (g_1, g_2, g_3)^T \) in the space \( SP_d(\Delta) \) which interpolates in the sense that

\[
g_j(u_i, v_i) = y_i^j, \quad i = 1, \ldots, n, \quad j = 1, 2, 3.
\]
The projection in step 2 can be in any arbitrary direction, as long as no triangles in \( \mathbb{R}^3 \) are mapped into triangles in \( \pi \) with zero area. To the extent possible, it is desirable to preserve the ratios of the areas of the original triangles. Usually we would project perpendicularly onto the plane \( \pi \). In step 4 we have required that the spline \( g \) interpolate the points \( \gamma_i \) at the vertices of the triangulation \( \Delta \). The reason for this is that for splines defined on triangulations, little is known about interpolation at points other than the vertices. For practical problems, we may choose \( d = 3, 4 \), i.e., cubic and quartic splines. For the remainder of this section we work with \( C^1 \) cubic splines.

To illustrate this algorithm, we consider the following example. For the sake of reproducibility, we give explicit coordinates for the points in \( \Gamma \), and lists of the triangles in \( \mathcal{F} \) forming the control surface \( S \).

**Example 14.** Let \( \gamma_1 = (0,0,1) \), \( \gamma_2 = (1,0,1) \), \( \gamma_3 = (1,0,0) \), \( \gamma_4 = (1,1,0) \), \( \gamma_5 = (0,1,0) \), \( \gamma_6 = (0,1,1) \), and \( \gamma_7 = (.8, .8, .8) \). Suppose the indices of the vertices of the triangles \( \mathcal{F} \) are given by \((1,2,6), (2,3,4), (4,5,6), (2,7,6), (2,4,7), \) and \((4,6,7)\).

**Discussion.** These 7 points lie on a rounded corner of a cube, i.e., on a suitcase corner. Fig. 4b shows the projection of \( \Gamma \) onto the plane \( x + y + z = 1 \), and the associated triangulation \( \Delta \). For each component of the spline, we have

\[
\begin{align*}
n_c &= \text{total number of parameters} = 37, \\
n_d &= \text{conjectured dimension} = 21, \\
n_{\text{int}} &= \text{number of interpolation conditions} = 7, \\
n_f &= \text{number of free parameters for the minimization} = 14.
\end{align*}
\]

The total number of free parameters for the parametric spline is \( 3 \times 14 = 42 \). The matrix in the linear system of Theorem 11 is of size \( 60 \times 60 \). Fig. 4a shows the surface corresponding to the cubic \( C^1 \) parametric spline interpolant. It is a reasonable approximation to the suitcase corner. □

Clearly, for a given set of points \( \Gamma \), if we choose to connect them in different ways to get a different control surface, we get a different parametric spline interpolant.

**Example 15.** Let \( \Gamma \) be as in Example 14, but suppose \( \mathcal{F} \) now consists of the following set of triangles: \((1,2,7), (2,3,7), (3,4,7), (4,5,7), (5,6,7), \) and \((6,1,7)\).

**Discussion.** Fig. 4d shows the projection of \( \Gamma \) onto the plane \( \pi \) of Example 14, and the associated triangulation \( \Delta \). Here the control surface \( S \) is geometrically identical with the one in Example 14, but consists of different triangles. Fig. 4c shows the surface corresponding to the cubic \( C^1 \) parametric spline interpolant. The change in the parameter domain did not cause a major change in the shape of the surface. □

For a given set of points \( \Gamma \) and control surface \( S \), the triangulation \( \Delta \) underlying the parametric spline space depends quite heavily on the choice of the projection plane \( \pi \) in step 1 of our algorithm. The following example illustrates this effect.
Fig. 4. A suitcase corner with three different parameterizations.
Example 16. Let \( \Gamma \) and \( \mathcal{F} \) be as in Example 14, but suppose the projection plane \( \Pi \) is now defined by \( x + 5y + z = 1 \) and the projection direction is given by \( (5,2,3)^T \).

Discussion. Fig. 4f shows the projection of \( \Gamma \) onto the plane \( \Pi \) and the associated triangulation \( \Delta \). It is, of course, just a distortion of the triangulation in Fig. 4b. Fig. 4e shows the associated cubic \( C^1 \) parametric spline interpolant. In this case the suitcase corner has been significantly distorted. □

To get a better approximation to the suitcase corner discussed in the above examples, we can interpolate at more points.

Example 17. Let \( \Gamma \) consist of the 7 points of Example 14 plus the following additional 9 points: \( (1,0,.5), (1,.5,0), (1,.5,.5), (.5,0,1), (.5,.5,1), (.5,1,0), (.5,1,.5), (0,.5,1), \) and \( (0,1,.5) \). The plane \( \Pi \) and projection direction are the same as in Example 14.

Discussion. We do not list the triangles in \( \mathcal{F} \) since it is clear what they are from Fig. 5b. For each component of the spline, we now have

\[
\begin{align*}
n_c &= \text{total number of parameters} = 100, \\
n_d &= \text{conjectured dimension} = 45, \\
n_{\text{int}} &= \text{number of interpolation conditions} = 16, \\
n_f &= \text{number of free parameters for the minimization} = 29.
\end{align*}
\]

The total number of free parameters for the parametric spline is \( 3 \times 29 = 87 \). The matrix in the linear system of Theorem 11 is of size \( 171 \times 171 \). Fig. 5a shows the associated cubic \( C^1 \) parametric spline interpolant, which is now an excellent approximation to the suitcase corner. □
To conclude this section, we look at an even more extreme parametrization.

**Example 18.** Let \( T \) be as in Example 14, and suppose we choose the triangulation shown in Fig. 6b.

**Discussion.** This triangulation has only four boundary edges, but since it has 7 vertices, we can still construct a cubic \( C^1 \) parametric spline surface which interpolates the given data. Fig. 6a shows the resulting surface. The reason for its extreme shape is due to the fact that here we have not specified a reasonable control surface, and so the triangular patches connect the given points on our "suitcase corner" in an unnatural way.

7. **Filling \( n \)-sided holes and crowning**

The problem of filling an \( n \)-sided hole in a surface is one of the central problems of CAGD. For our purposes, we assume that the hole is cut out of an already existing smooth piecewise polynomial surface, and that the boundary of the hole consists of \( n \) curves. The problem is to create a surface patch which fills the hole, and which joins smoothly to the outside.

The problem of crowning a surface is closely related. We cut an \( n \)-sided hole in a given surface, and then fill it with a patch which is close to the original surface, but which may be raised or lowered somewhat. This can be accomplished by making the patch interpolate some specified point(s) in \( \mathbb{R}^3 \) in addition to matching the boundary curves and the cross derivatives along the boundary curves of the hole.

There have been many suggestions for smoothly filling in an \( n \)-sided hole in an already existing smooth surface. The available methods include blending (convex combination) schemes (Gregory, 1986; Gregory and Hahn, 1989), splitting schemes with triangular (Shirman and Séquin, 1987; Peters, 1990a) or rectangular pieces (Hahn, 1989a, b; Sarraga, 1990), overlap patches (Varady, 1990), subdivision (Doo and Sabin, 1978;
Storry and Ball, 1989; Peters, 1993), and methods that use special nonplanar domains (Sabin, 1983; Hosaka and Kimura, 1984; Loop and DeRose, 1989, 1990). For some good surveys on this subject see (Gregory et al., 1990; Peters, 1990b; Varady, 1990).

In order to create a surface to fill a hole with a patch maintaining $C^1$ parametric continuity, we need to make a compatibility assumption on the initial surface surrounding the hole. We assume that it is given in piecewise polynomial form, and that the derivatives across the boundaries of the hole are such that they maintain $C^{1,1}$ continuity at the corners of the hole (for similar assumptions see (Gregory et al., 1990; Loop and DeRose, 1990)).

In this section we examine the use of our parametric spline method to solve this problem. To apply the method, we have to create a planar triangulation $\Delta$ to serve as the domain of the spline. In this setting, it seems natural to choose $\Delta$ so that it has $n$ vertices on the boundary.

To make our surface patch join smoothly with the rest of the surface, we require it to interpolate the surface values at each of the boundary vertices of $\Delta$, and in addition

1. choose the degree of the patch to match the degree of the outside surface patches;
2. construct the interpolant so that the boundary curves match the boundary curves of the hole, and so that the cross boundary derivatives of the patch match the cross-boundary derivatives defined by the outside surface.

One way to create the triangulation $\Delta$ is to choose some point in $\mathbb{R}^3$ which we want our crowned surface to go through, connect it to the $n$ points on the boundary of the hole, and then follow the first three steps in Algorithm 13. This typically gives a triangulation of an $n$-gon with one interior vertex. Unfortunately, using low degree $C^1$ spline spaces we do not get enough free parameters to match all of the required boundary information (in the cubic case), or there are very few free parameters (e.g., in the quartic case). Thus, this triangulation usually has to be further subdivided.

The alternative is to construct the triangulation $\Delta$ directly in a $(u,v)$-plane. Given an $n$-gon which is to be the domain $\Omega$ of our spline space, we propose two natural ways to triangulate $\Omega$. The first corresponds to triangulating the $n$ gon with a single interior vertex, and then splitting each of the resulting triangles into three subtriangles as is done in the well-known Clough–Tocher method of finite element theory.

**Triangulation 1.** (See Fig. 7a.)

1. Find the centroid of the vertices surrounding the hole and connect all the vertices to the centroid.
2. Find the centroid of each of the subtriangles constructed in step 1 and connect the centroids to the vertices of the subtriangles.

Our second triangulation involves the same number of interior vertices, but positioned in a somewhat more symmetric way.

**Triangulation 2.** (See Fig. 7b.)

1. Find the centroid of the vertices surrounding the hole.
2. Find the midpoints of the boundary edges (marked with $*$) and then the midpoints (marked with $\bullet$) of the line segments connecting the boundary midpoints and the
3. Connect all of the vertices marked with $•$ and also connect each of them with the centroid of the hole.

4. Connect each of the boundary vertices with the two closest vertices marked with $•$.

Neither triangulation requires $Ω$ to be convex. However, it should be such that its centroid lies inside the boundary polygon.

The following theorem shows that if we use cubic $C^1$ splines matching values and derivatives on the boundary of Triangulation 1, the basic hypothesis (5.4) for the existence of an interpolating spline is satisfied.

**Theorem 19.** Suppose we are given an $n$-sided hole in a piecewise polynomial surface with (at most) cubic boundary curves and (at most) quadratic cross-boundary derivatives, and satisfying the basic compatibility condition discussed above. Let $Δ_1$ be a triangulation as shown in Fig. 7a. Then there exists a unique $C^1$ cubic parametric spline such that the corresponding surface fills the hole, matches the given surface smoothly ($C^1$) along the boundary, and has a prescribed value and tangent plane at the parameter point corresponding to the center vertex of the triangulation.

**Proof.** The fact that there is a unique scalar-valued $C^1$ cubic spline which takes on given values and gradients at the $n$ boundary vertices and the center vertex of $Δ_1$ is well known from finite element theory (see e.g. (Alfeld, 1984a)). We can think of the resulting surface as a kind of macro element consisting of $3n$ cubic triangular patches which fill the hole and join with the outside surface with $C^1$ smoothness across all patch boundaries. $\Box$

It is not known if the analogous result holds for Triangulation 2 using cubic splines, but our computational experience suggests that it does. We discuss the use of quartic splines with Triangulation 2 at the end of this section.

**Example 20.** Consider the function $f(x,y) = x^2 + y^2$ defined on a regular hexagon surrounding the origin with one vertex at $(2,0)$. Suppose we create a symmetric six-
sided hole surrounding the origin as shown in Fig. 8. Fill this hole with a patch which interpolates the value 1.5 at (0,0) with a horizontal tangent plane.

**Discussion.** We use Triangulation 2 and cubic $C^1$ (scalar) splines. Fig. 8 shows the resulting surface with the fill. The views on the left correspond to a slightly higher viewing angle. The top two images show lines corresponding to a few steps of subdivision of the patches. These clearly reveal the hole which has been filled. The middle two images show a curvature plot of the fill (unfortunately, this is hard to depict in black and white). The bottom images show the surface and the patch using Gouraud shading.

As observed above, if we use $C^1$ cubic splines to fill a hole which has been triangulated as in Triangulation 2, there is no guarantee that the resulting system of equations is nonsingular. However, we can establish the nonsingularity for $C^1$ quartic splines.

**Theorem 21.** Suppose we are given an n-sided hole surrounded by a piecewise polynomial surface with (at most) quartic boundary curves and (at most) cubic cross-boundary derivatives, and satisfying the basic compatibility condition discussed above. Let $\Delta_2$ be a triangulation as shown in Fig. 7b. Then there exists a unique $C^1$ quartic parametric spline such that the corresponding surface fills the hole and matches the given surface smoothly ($C^1$) along the boundary. Moreover, at the middle splitting point, we can specify the position of the surface and the tangent plane to it, provided $n$ is odd. If $n$ is even, then we can specify the same information at the middle point provided $\prod_{i=1}^{n} r_i \neq \prod_{i=1}^{n} t_i$ (see the proof for details).

**Proof.** Throughout the proof we will refer to Fig. 9 and the notation used there. We will show that, given positional and cross-boundary information from the outside surface, we have enough free coefficients inside the hole to satisfy a $C^1$ join across all patch boundaries while choosing the center point and its tangent plane arbitrarily.

The key to the proof is the analysis of a cyclic nonhomogeneous linear system involving the coefficients numbered 1 through $n$ in the figure.

Let us denote all coefficients determined by the outside world by $\ast$. If we choose those marked with $\odot_1$ arbitrarily, then the coefficients represented by $\bullet_1$ are determined by $C^1$ continuity conditions across the respective edges. In the next step we choose the $\odot_2$ coefficients freely. Together with the $\ast$, $\odot_1$ and $\bullet_1$ coefficients they uniquely determine those marked with $\bullet_2$. In order to fix the tangent planes at the vertices of the inner ring, we set the coefficients with $\odot_3$ and then also have those with $\bullet_3$. The last set of coefficients we get to choose freely is those denoted by $\odot_4$ in the second layer around the center vertex; they then determine the $\bullet_4$ coefficients (in case the $\bullet_4$ coefficient cannot be determined because the quadrilateral has a straight edge through the center vertex, we get to choose this coefficient also). Now we are left with the coefficients numbered 1 through $n$ and $n+1$ through $3n$. They are involved in one continuity condition across each of the edges emanating from the center vertex, and in two $C^1$ conditions across the boundaries between the two rings of triangles (the continuity conditions are crosshatched in the figure). If we label the latter edges counterclockwise from $e_1$ to $e_n$, then we can
Fig. 8. Filling a symmetric 6-sided hole.
write the first set of continuity conditions as follows:

\[
\begin{align*}
c_{2i+n-1} &= t_i c_i + \alpha_i, \\
c_{2i+n} &= r_i c_i + \beta_i,
\end{align*}
\]  

(7.1)

where \(i = 1, \ldots, n\), the \(c\)'s are the unknown coefficients numbered as in the figure, and \((r_i, s_i, t_i)\) are the barycentric coordinates associated with the quadrilateral to both sides of edge \(e_i\), where we always express the center vertex as \(r_i\) times the “right” vertex of edge \(e_i\) plus \(s_i\) times the associated “outside” vertex plus \(t_i\) times the “left” vertex of edge \(e_i\). The quantities \(\alpha_i\) and \(\beta_i\) on the right-hand side contain the already determined coefficients also present in the continuity conditions.

The second set of continuity conditions across the innermost edges, labelled counterclockwise with \(e_{n+1}\) through \(e_{2n}\), turns out to be

\[
\begin{align*}
c_{2i-n-1} &= s_i c_{2i-n-2} + \gamma_i, \quad i = n+1, \ldots, 2n,
\end{align*}
\]  

(7.2)

where now the barycentric coordinates \((r_i, s_i, t_i)\) reflect the shapes of two neighboring triangles when going around the center vertex counterclockwise. Again the \(\gamma_i\) contain already set coefficients. This set of equations is of a cyclic nature, so we therefore have to identify the coefficient \(c_n\) as \(c_{3n}\).

We can now combine (7.1) and (7.2) and obtain a cyclic linear system in the coefficients \(c_1\) through \(c_n\) which has the following structure
\[
\begin{pmatrix}
-s_{n+1}r_1 & t_2 & 0 & 0 & \ldots & 0 \\
0 & -s_{n+2}r_2 & t_3 & 0 & \ldots & 0 \\
\vdots & \ddots & \ddots & \ddots & \vdots \\
0 & \ddots & \ddots & \ddots & 0 \\
0 & 0 & \ddots & \ddots & 0 \\
t_1 & 0 & 0 & \ldots & 0 & -s_{2n}r_n
\end{pmatrix}
\begin{pmatrix}
c_1 \\
c_2 \\
\vdots \\
c_n
\end{pmatrix} =
\begin{pmatrix}
\rho_1 \\
\rho_2 \\
\vdots \\
\rho_n
\end{pmatrix},
\]

where \((r_i, s_i, t_i)\) are as above and the \(\rho_i (i = 1, \ldots, n)\) are the known right-hand sides. We can easily calculate the determinant of the coefficient matrix as

\[
\text{Det} = (-1)^n \prod_{i=n+1}^{2n} s_i \prod_{i=1}^{n} r_i + (-1)^{n-1} \prod_{i=1}^{n} t_i.
\]

Since we have \(\prod_{i=n+1}^{2n} s_i = (-1)^n\) due to the fact that the cycle around the center vertex closes on itself and barycentric coordinates can be viewed as ratios of areas, the expression for the determinant simplifies to

\[
\text{Det} = \prod_{i=1}^{n} r_i + (-1)^{n-1} \prod_{i=1}^{n} t_i.
\]

We now have to distinguish two cases:

Case 1 \((n \text{ is odd})\). In this case we have

\[
\text{Det} = \prod_{i=1}^{n} r_i + (-1)^{n-1} \prod_{i=1}^{n} t_i \neq 0,
\]

since in our particular split all relevant quadrilaterals are convex, which means that all of the \(r_i\) and \(t_i\) \((i = 1, \ldots, n)\) are positive. So the linear system has a unique solution and we can fill in a hole with an odd number of vertices.

Case 2 \((n \text{ is even})\). Now the determinant turns out to be

\[
\text{Det} = \prod_{i=1}^{n} r_i - \prod_{i=1}^{n} t_i,
\]

and we cannot positively say whether this expression is nonzero. Therefore a unique solution for the even case is guaranteed to exist for data that satisfy this condition.

It is easy to construct examples to show that the condition in Theorem 21 for the case \(n \text{ even}\) is sufficient, but not necessary (any symmetric setting will do). It is obvious from the proof of this theorem that instead of specifying just the center point of the split with its tangent plane, we could also set all other points marked with a \(\circ\) in Fig. 9.

Instead of using \(C^1\) cubic or quartic splines, we could also fill holes using a \(C^1\) quadratic spline defined on the triangulation obtained by splitting each initial triangle into \(6n\) subtriangles according to the well-known Powell–Sabin split (cf. (Dierckx, 1993)).
This would assure that values and tangent planes can be specified at all vertices of the original triangulation, and requires that fewer coefficients be determined (and stored). (Here there are $14n + 1$ coefficients as compared with $15n + 1$ for the cubic scheme on the Clough-Tocher split). However, working with quadratics only also restricts the maximal degree of the surface representation on the outside to quadratics, which might be too low for many applications.

The macro elements discussed here can be viewed as locally supported elements similar to B-splines. However, it is not clear if there is a way of using them to construct a basis for $S_d^1(\Delta)$ or $S_P^d(\Delta)$.

### 8. Adding a $C^2$ penalty term

Most surface interpolation methods (cf. those mentioned in Sections 4 and 5) achieve only $C^1$ or $G^1$ continuity, and some of them have only approximate $G^1$ continuity. Using our approach, however, it is not much additional work to incorporate conditions that yield approximate $C^2$ continuity, and in fact we can do this with total degree cubic patches as opposed to e.g. the (bi-)quintic patches used by Hagen and Schulze (1987), or Morcón and Séquin (1992) who achieve only approximate $G^1$ continuity with $G^2$ continuity at the vertices!

The idea is to add the $C^2$ continuity conditions to the energy functional we are minimizing in the form of a least squares penalty term. We first discuss this idea in the case of surfaces corresponding to scalar-valued splines. It will also suffice to look only at the case of cubic splines; the general case is completely analogous.

As is well known, cf. (Farin, 1986), a $C^1$ cubic spline is $C^2$ continuous across an edge of the triangulation (cf. Fig. 10), if and only if the sum

$$
\sum_{i=1}^{2} \left[ e_i r - (s^2 e_i m + 2sx e_i m + 2ty e_i m + 2st e_i m + t^2 e_i m) \right]^2
$$

is zero. Here $(r,s,t)$ are the barycentric coordinates associated with the edge, and the coefficients are as in Fig. 10.
Thus, to make a spline $g \in S_1^1(\Delta)$ be approximately $C^2$, we should minimize the sum of all such terms across all interior edges of $\Delta$. In view of the quadratic form of the expressions in this sum, it can be written in the form

$$c^T P c.$$ 

Thus, we look for a spline which minimizes a combination of energy and this smoothness term:

$$J_\omega(g) = c^T (Q + \omega P) c.$$ 

Here $\omega$ is a parameter controlling the relative weight of energy versus smoothness. For more on penalized least squares in approximation theory and the use of generalized cross validation for the choice of the smoothing parameter, see (von Golitschek and Schumaker, 1990).

Interpolating splines which minimize $J_\omega$ can be found by solving a simple linear system of equations. Indeed, Theorem 8 remains valid if we simply replace $Q$ by $Q + \omega P$. Thus to compute these splines, the only extra work that has to be performed is the creation of the entries of the matrix $P$. This can be done in the same step in which the $C^1$ continuity conditions are being constructed, and therefore does not require much overhead.

The extension of this idea to surfaces defined by parametric splines is obvious. In particular, Theorem 11 also holds if we replace $Q$ by $Q + \omega P$. One could slightly modify the $C^2$ penalty term defined above by assigning individual weights $\omega_i$, $i = 1, \ldots, N$, to each triangular patch. This would give an even better control over the smoothing effect.

To see the effects of the $C^2$ penalty term, we solve a simple scalar interpolation problem both with and without the penalty term.

**Example 22.** Consider the function

$$f(x, y) = (1 + 2e^{-3|9(x^2+y^2)^{1/2}-6.7|})^{-1/2}$$

defined on a regular hexagon with one vertex at $(2,0)$. Let $\Delta$ be the Delaunay triangulation corresponding to two concentric hexagonal rings and the origin. Find a cubic $C^1$ spline on $\Delta$ which interpolates function values at the 13 vertices of $\Delta$.

**Discussion.** The surface on the left in Fig. 11 is constructed with $\omega = 0$, while for the surface on the right $\omega = 100$. Areas of high positive or negative Gaussian curvature are highlighted (unfortunately, in this gray-scale rendering one cannot distinguish between the two extremes). Nevertheless, the $C^2$ smoothing effect is clearly demonstrated. This high degree of smoothness is achieved using only cubic splines.

9. Closed surfaces

In this section we explore the possibility of using the above ideas to create closed surfaces which interpolate given sets of points in $\mathbb{R}^3$. We restrict our attention to closed surfaces which are topologically equivalent to the sphere.
Problem 23. Suppose \( \Gamma := \{ (\gamma_1^i, \gamma_2^i, \gamma_3^i) \}_{i=1}^n \) are given points in \( \mathbb{R}^3 \), and that \( \mathcal{F} := \{ F_i \}_{i=1}^n \) is a set of nonintersecting triangles in \( \mathbb{R}^3 \) with vertices at the points \( \Gamma \). Suppose the surface \( S \) in \( \mathbb{R}^3 \) defined by the union of the triangles in \( \mathcal{F} \) is closed. Find an interpolating parametric surface which is smooth, and which has a shape similar to \( S \).

As in Section 6, the surface \( S \) can be thought of as a kind of "control surface". To model \( S \) with a parametric spline, we shall use an analog of Algorithm 13. Before describing an algorithm, we need to introduce two new ideas: (1) generalized triangulations, and (2) splines on generalized triangulations. The triangulations introduced in the following definition are generalizations of ordinary planar triangulations in the sense that triangles are allowed to lie on top of each other.

Definition 24. Let \( \Delta = \{ T_i \}_{i=1}^N \) be a collection of triangles, and let \( \mathcal{V} = \{ V_i \}_{i=1}^V \) be the set of all vertices involved. We say that \( \Delta \) is a generalized triangulation of

\[
\Omega := \bigcup_{i=1}^N T_i
\]

provided that
1. each triangle \( T_i \) contains no points from \( \mathcal{V} \) other than its vertices,
2. any two triangles either do not intersect at all, or share one, two, or three edges,
3. no more than two triangles share any edge,
4. \( \Omega \) is connected.

This definition extends the notion of generalized triangulations introduced in (Alfeld et al., 1993). There generalized triangulations were defined by projecting a set of 3D triangles in a special way.

Here is an example showing how such a generalized triangulation can be created by projecting a control surface as described in Problem 23 onto a plane.
Example 25. Find a closed surface which interpolates the six points \( v_1 = (1, 0, 0), v_2 = (0, 1, 0), v_3 = (-1, 0, 0), v_4 = (0, -1, 0), v_5 = (0, 0, 1), \) and \( v_6 = (0, 0, -1) \).

Discussion. These six points form the vertices of an octahedron in \( \mathbb{R}^3 \), each of whose faces is a triangle. We can consider the union of these eight triangles to be a closed control surface \( S \). Now if we project \( S \) along the \( z \)-axis onto the \( xy \)-plane, the eight triangular faces are mapped into eight planar triangles, four of which lie on top, and four of which lie on the bottom, see Fig. 12. Note that the images of \( v_5 \) and \( v_6 \) both appear to be at the origin, but \( v_5 \) is on top, and \( v_6 \) is on the bottom.

For later use, we need some additional terminology.

Definition 26. Given a generalized triangulation \( \Delta \), an edge which is shared by two triangles is called an interior edge. All other edges are called boundary edges. Vertices which lie on boundary edges are called boundary vertices, and all others are called interior vertices. An interior edge is said to be folded provided that one of the two triangles sharing that edge is on top of the other. A vertex is said to be folded provided it lies on a folded edge. A folded vertex is said to be exceptional provided that its two adjacent folded edges are not collinear.

Clearly, if a generalized triangulation arises by collapsing a closed control surface as above, then all edges and vertices are interior. In the generalized triangulation shown on the left in Fig. 12, the edges drawn with solid lines are folded edges, while those drawn with dashed lines are not. The vertices \( v_5 \) and \( v_6 \) are ordinary vertices (not folded), while \( v_i, i = 1, \ldots, 4 \), are folded exceptional vertices.

We now introduce a space of splines defined on a generalized triangulation. It should consist of scalar-valued functions defined on the triangles of \( \Delta \). Thus, given an integer \( d > 0 \), we define the associated space of piecewise polynomials to be

\[
P_d^\Delta := \{ g : g|_T \in P_d^{(2)}, \text{ for all } T \in \Delta \},
\]

where as before, \( P_d^{(2)} \) is the space of bivariate polynomials of total degree \( d \).

Each piece of an element \( g \in P_d^\Delta \) is a polynomial of total degree \( d \) on a planar triangle, and thus can be expressed in Bernstein–Bézier form as described in Section 1.
This means we can define an associated coefficient vector $c$ in the same way as in Section 2 for splines on ordinary triangulations (where the coefficients of patches which are associated with adjoining triangles and which lie on the common edge between the two triangles are identified). We denote the length of $c$ by $n_c$ as before. We can now describe a kind of spline space where adjoining patches are required to "match" along their common edge.

**Definition 27.** Let $\Delta$ be a generalized triangulation, and let $0 \leq r < d$ be given integers. We define the space of splines by

$$S_r^d(\Delta) := \{ g \in \mathcal{P}_d(\Delta) : H^r e = 0 \text{ for all interior edges } e \},$$

where $H^r e$ is the $m_r \times n_c$ matrix with $m_r := d + (d - 1) + \cdots + (d - r + 1)$ which describes the $C^r$ continuity across the edge $e$, except that if $e$ is a folded edge, then in computing the barycentric coordinates, we first fold out the edge. Thus, for example in Fig. 12, since the edge between $v_1$ and $v_2$ is a folded edge and $v_6$ is underneath, we first fold out the triangle on the bottom before computing the barycentric coordinates $(r, s, t)$ of $v_6$ in terms of the triangle with vertices $v_2$, $v_5$, and $v_1$. The conditions across nonfolded edges are exactly as in the case of ordinary splines.

The space of splines defined on generalized triangulations introduced in (Alfeld et al., 1993) is not the same as ours, since there the space was defined using the usual continuity conditions even for folded edges (without folding them out).

It is clear that spaces of parametric splines on a generalized triangulation can be defined in exactly the same way as was done for ordinary triangulations, see Definition 9. Any such parametric spline defines a surface in $\mathbb{R}^3$ defined on the parameter domain $\Omega$.

Because of the unusual nature of the space $S_r^d(\Delta)$, we should say something about the smoothness of the resulting parametric surface at a folded vertex $v$ when $r = 1$. First, if $v$ is not exceptional, then the surface is visually smooth at $v$ in the usual sense, i.e., it has a continuous tangent plane in a neighborhood of $v$. The reason for this is that in this case folding out the bottom triangles can be regarded as a reparametrization based on an ordinary triangulation.

If $v$ is a folded exceptional vertex, then as the following lemma shows, the Bézier coefficients in the first ring around $v$ must all be equal to the coefficient at the vertex, and thus (as is well known), the surface may have a cusp at $v$. This corresponds to a singular parameterization. Such cusps can be removed by appropriately setting Bézier coefficients in the second ring (cf. (Neamtu, 1991; Peters, 1991b; Reif, 1993)). We illustrate this in Fig. 14.

**Theorem 28.** Let $v$ be an exceptional folded vertex with $m+k$ edges attached. Suppose the coefficients of $g \in S_r^d(\Delta)$ satisfy the conditions $H^1 e = 0$ for all edges $e$ attached to the vertex $v$. Then all coefficients in the first ring around $v$ must have the same value as the coefficient at $v$.

**Proof.** For the proof it will suffice to consider only those triangles of the generalized triangulation $\Delta$ which are attached to the vertex $v$. Assume the triangles have vertices
$v_i, v_{i+1}, v$, and that those on top correspond to $i = 1, \ldots, m$, while those on the bottom correspond to $i = m, \ldots, m+k$, where $v_{m+k+1} = v_1$. Now cut along edge $v_{m+k}$ and fold out all of the bottom triangles. Let $\tilde{v}_i$ be the new location of the vertex $v_i$ for $i = m+1, \ldots, m+k-1$. By the assumption that $v$ is exceptional, the vertex $v_{m+k}$ moves to two different positions, $\tilde{v}_{m+k}$ and $\hat{v}_{m+k}$ associated with the folded out triangles $v_{m+k-1} \tilde{v}_{m+k}$ and $v_{m+k} \hat{v}_{m+k}$.

Let the Bézier coefficient associated with $v$ be $c_0$. By the $C^1$ continuity conditions, all of the points $(v_i, c_i)$ for $i = 1, \ldots, m$ as well as the points $(\tilde{v}_i, c_i)$ for $i = m+1, \ldots, m+k$ and $(\hat{v}_{m+k}, c_{m+k})$ lie on a plane passing through $(v, c_0)$. In addition, the points $(v_i, c_i)$ for $i = m, \ldots, m+k+1$ must also lie on a plane passing through $(v, c_0)$. Since the points $v_1, v, v_{m+k}$ are noncollinear, the two planes coincide with a common plane. This plane must be the horizontal plane through $(v, c_0)$ since it contains the points $(v_{m+k}, c_{m+k})$, $(\tilde{v}_{m+k}, c_{m+k})$, $(\hat{v}_{m+k}, c_{m+k})$. We conclude that $c_i = c_0$ for all $i = 1, \ldots, m+k$.

Theorem 28 implies that if $g = (g_1, g_2, g_3)^T$ is a parametric spline, then the corresponding control points (which are then 3-vectors) all collapse to the control point at $v$.

To solve Problem 23 using the space of splines $S^1_1(\Delta)$ defined on a generalized triangulation, we now introduce the total energy of such a spline surface as the sum of the individual energies on triangles, i.e., $J(g)$ can be defined exactly as in (5.1). Now we can define a minimal energy interpolating parametric spline just as in Definition 10: it minimizes the quadratic form $J(g)$ subject to smoothness and interpolation conditions. As in Theorem 11, this reduces to solving a linear system of equations similar to (5.5).

We can now present an algorithm for interpolating data in $\mathbb{R}^3$ with a closed parametric spline surface.

**Algorithm 29.**

1. Choose a plane $\pi$ in $\mathbb{R}^3$ and introduce a $(u, v)$ Cartesian coordinate system on it. Choose a projection direction.
2. For each $i = 1, \ldots, n$, let $(u_i, v_i)$ be the point on $\pi$ obtained by projecting the point $(y^1_i, y^2_i, y^3_i)$ onto $\pi$.
3. Let $\Delta$ be the generalized triangulation in $\pi$ obtained by parallel projecting the edges of the triangles $F_i$ onto $\pi$.
4. Find a minimal energy parametric spline $g = (g_1, g_2, g_3)^T$ in the space $S^1_1(\Delta)$ which interpolates in the sense that $g_j(u_i, v_i) = y^j_i, \quad i = 1, \ldots, n, \quad j = 1, 2, 3$. (9.1)

The projection in step 2 can be in any arbitrary direction, as long as no triangles in $\mathbb{R}^3$ are mapped into triangles in $\pi$ with zero area.

Fig. 13a shows the result of using splines in $S^1_1(\Delta)$ with the generalized triangulation shown in Fig. 13b to interpolate the data in Example 25. In this example the singularities in the parametrization occur at the corners of the square, which are folded exceptional vertices in this case. The resulting surface is smooth across the interiors of all edges, but as the figure shows, because of the singularities, there are cusps at the corners.
To smooth out the cusps at the corners, we use the idea mentioned above of setting the Bézier coefficients in the second ring appropriately. To do this, we have to refine the generalized triangulation of Fig. 12 in such a way as to separate the folded exceptional vertices, and to provide enough coefficients to adjust. The top half of our refined generalized triangulation is shown in Fig. 14b.

Fig. 14a shows the minimal energy parametric spline in $SP^1_3(\Delta)$ which interpolates the data of Example 25 based on this triangulation. Here we have set two control points in the second layer around each folded exceptional vertex to form a vertical tangent plane there. It should be noted that although the surface looks smooth, it has infinite curvature at folded corners.

A disadvantage when using generalized triangulations and singular parametrizations is that, as soon as we define a common tangent plane at a vertex (e.g. by specifying its normal vector, instead of explicitly fixing certain coefficients as we did for Fig. 14), the decoupling of the three spline coordinates is destroyed and the size of the system matrix grows by a factor of 9. However, it does have the block structure mentioned in Section 5. We have not attempted to exploit this fact.
Another problem associated with the use of generalized triangulations and singular parametrizations is that the redundancies in the $C^1$ continuity conditions behave differently than in the regular case. In the regular case one simply ignores the last two conditions in the ring around an interior vertex as well as an additional one at a singular vertex (see also Section 2). In the case of a singular parametrization, this is not necessarily true, and now we have to use some method such as singular value decomposition to numerically find the dependent rows of the matrix $A$ describing the continuity conditions.

References

Coons, S. (1964), Surfaces for computer aided design, MAC-TR-4, MIT, Cambridge, MA.


Kallay, M. (1992), Constrained optimization in surface design, EDS, Bellevue, WA.


Peters, J. (1993), Smooth free-form surfaces over irregular meshes generalizing quadratic splines, CSD-TR-92-063, Purdue University, West Lafayette, IN.


Powell, M.J.D. (1989), TOLMIN: A Fortran package for linearly constrained optimization calculations, Dept. of Applied Mathematics and Theoretical Physics, University of Cambridge.


