

Scattered Data Fitting on Surfaces Using Projected Powell-Sabin Splines

Oleg Davydov^{1*} and Larry L. Schumaker²

¹ Department of Mathematics, University of Strathclyde, 26 Richmond Street,
Glasgow G1 1XH, Scotland

² Department of Mathematics, Vanderbilt University, Nashville, TN 37240

Abstract. We present C^1 methods for either interpolating data or for fitting scattered data associated with a smooth function on a two-dimensional smooth manifold Ω embedded into \mathbb{R}^3 . The methods are based on a local bivariate Powell-Sabin interpolation scheme, and make use of local projections on the tangent planes. The data fitting method is a two-stage method. We illustrate the performance of the algorithms with some numerical examples, which, in particular, confirm the $\mathcal{O}(h^3)$ order of convergence as the data becomes dense.

1 Introduction

Let Ω be a 2-dimensional smooth manifold. For simplicity we assume that Ω is compact and has no boundary. Suppose we are given the values of a (possibly unknown) smooth function f defined on Ω at a set of points X on Ω . Our aim is to construct a function s defined on Ω that approximates f . This problem arises frequently in practice, see Remark 1, but there do not seem to be many methods available for general manifolds. Several methods have been developed for the case when Ω is the sphere, see Remark 2.

Our approach to solving this problem is as follows. Suppose we have an atlas $\Phi = \{(U_\xi, \phi_\xi)\}_{\xi \in \Omega}$ for Ω , where for each $\xi \in \Omega$, U_ξ are open sets on Ω containing ξ , and ϕ_ξ are mappings of U_ξ into \mathbb{R}^2 . We assume that the ϕ_ξ depend smoothly on ξ in a certain sense, see [7]. Then for each $\xi \in \Omega$, we project the data locations into $\phi_\xi(U_\xi) \subset \mathbb{R}^2$, and use a local bivariate Powell-Sabin spline to compute the value $s(\xi)$ of the approximating function s . The resulting function s is C^1 on Ω , and it has the usual approximation properties of the Powell-Sabin spline. This approach is related to methods for interpolation and data fitting on manifolds introduced by Demjanovich [10, 11] and Pottmann [25], see Remarks 3 and 4.

In this paper we examine the case when Ω is a C^2 -surface, i.e. a compact 2-dimensional C^2 -manifold embedded in \mathbb{R}^3 . Our implementation is based on a natural atlas Φ defined by local projections onto the tangent planes at all points on the surface Ω , see Section 2.1. General smooth 2-manifolds are treated in [7], where we also prove the main theoretical results about smoothness of s and error bounds for interpolation and data fitting.

* Partially supported by the Edinburgh Mathematical Society Research Support Fund.

The paper is organized as follows. In Section 2 we introduce some basic concepts and notation. In Section 3 we first present a method for constructing an interpolant to the data on any smooth 2-dimensional manifold Ω embedded in \mathbb{R}^3 , assuming we are also given values for the gradients at each of the data points in X . Next, we describe a two-stage data fitting method that works only with the data $\{f_\xi\}_{\xi \in X}$, and which is more appropriate than interpolation for large data sets or noisy data. Theoretical results on C^1 smoothness of the interpolating or approximating functions on Ω , as well as error bounds for the approximation of smooth functions are given here without proof. Numerical examples are presented in Section 4 for both the sphere and for certain ring-type manifolds. We conclude the paper with remarks and references.

2 Preliminaries

2.1 Projection atlas

Let $\langle \cdot, \cdot \rangle$ be the usual inner product in \mathbb{R}^3 , and let $\|a\|_2$ be the Euclidean norm of any 3-vector a . Since Ω is embedded in \mathbb{R}^3 , it can be represented locally as a regular level surface of a C^2 function of three variables. More precisely, each point $\xi \in \Omega$ has a neighborhood \mathcal{G}_ξ in \mathbb{R}^3 such that $\mathcal{G}_\xi \cap \Omega = F_\xi^{-1}(0)$, where $F_\xi : \mathcal{G}_\xi \rightarrow \mathbb{R}$ is a C^2 function with nonzero gradient ∇F_ξ everywhere in $\mathcal{G}_\xi \cap \Omega$, see [17]. Then $n_\xi := \nabla F_\xi(\xi) / \|\nabla F_\xi(\xi)\|_2$ is a *normal vector* to Ω at ξ . Moreover, the *tangent plane* Γ_ξ is the unique plane in \mathbb{R}^3 that contains ξ and is orthogonal to n_ξ . Clearly, for all $\zeta \in \mathcal{G}_\xi \cap \Omega$, a normal vector to Ω at ζ can also be computed as $\nabla F_\xi(\zeta) / \|\nabla F_\xi(\zeta)\|_2$. It coincides with either n_ξ or $-n_\xi$. Clearly, $\langle n_\xi, \nabla F_\xi(\zeta) \rangle > 0$ for all $\zeta \in \mathcal{G}_\xi \cap \Omega$.

We are now ready to define an atlas associated with Ω . For each $\xi \in \Omega$, let U_ξ be the connected component of the open set $\{\zeta \in \Omega : \langle n_\xi, n_\zeta \rangle \neq 0\}$ that contains ξ . Then U_ξ is an open neighborhood of ξ . Clearly, the *orthogonal projection* $\pi_\xi : U_\xi \rightarrow \Gamma_\xi$ defined by

$$\pi_\xi(\zeta) = \zeta + \langle \xi - \zeta, n_\xi \rangle n_\xi, \quad \zeta \in U_\xi,$$

is invertible. Assuming that $\gamma_\xi^{[1]}, \gamma_\xi^{[2]}$ are orthogonal unit vectors in Γ_ξ such that $\gamma_\xi^{[1]} \times \gamma_\xi^{[2]} = n_\xi$, we can also write

$$\pi_\xi(\zeta) = \xi + \langle \zeta - \xi, \gamma_\xi^{[1]} \rangle \gamma_\xi^{[1]} + \langle \zeta - \xi, \gamma_\xi^{[2]} \rangle \gamma_\xi^{[2]}.$$

Define ϕ_ξ by the formula

$$\phi_\xi(\zeta) := [\langle \zeta - \xi, \gamma_\xi^{[1]} \rangle, \langle \zeta - \xi, \gamma_\xi^{[2]} \rangle]^T, \quad \zeta \in U_\xi.$$

We call $\Phi = \{(U_\xi, \phi_\xi)\}_{\xi \in \Omega}$ the *projection atlas* associated with Ω , and (U_ξ, ϕ_ξ) , $\xi \in \Omega$, are its *charts*.

Let us show that Φ is indeed an atlas in the sense of the standard definition of a C^1 -manifold, see e.g. [17]. This requires that $\phi_\xi : U_\xi \rightarrow \mathbb{R}^2$ is a homeomorphism

between U_ξ and an open subset of \mathbb{R}^2 . Moreover, for every $\xi, \zeta \in \Omega$, $\phi_{\zeta\xi} := \phi_\zeta \circ \phi_\xi^{-1} : \phi_\xi(U_\zeta \cap U_\xi) \rightarrow \phi_\zeta(U_\zeta \cap U_\xi)$ should be a C^1 mapping whenever $U_\xi \cap U_\zeta \neq \emptyset$.

Clearly, ϕ_ξ is invertible by the choice of U_ξ . Consider the coordinate system for \mathbb{R}^3 with coordinate vectors $\gamma_\xi^{[1]}, \gamma_\xi^{[2]}, n_\xi$ and origin ξ . For any $\mu \in U_\xi$, the equation $F_\mu = 0$ determines an implicit function $x^{[3]} = \delta_\mu(x^{[1]}, x^{[2]})$ in a neighborhood of $\phi_\xi(\mu)$, such that

$$\phi_\xi^{-1}(x^{[1]}, x^{[2]}) = \xi + x^{[1]}\gamma_\xi^{[1]} + x^{[2]}\gamma_\xi^{[2]} + \delta_\mu(x^{[1]}, x^{[2]})n_\xi.$$

Since $\langle n_\xi, \nabla F_\mu(\mu) \rangle = \langle n_\xi, n_\mu \rangle \|\nabla F_\mu(\mu)\|_2 \neq 0$, the implicit function theorem implies that $\delta_\mu(x^{[1]}, x^{[2]})$ is a C^2 function in a neighborhood of $\phi_\xi(\mu)$. Assuming $\mu \in U_\xi \cap U_\zeta$, we also have

$$(\phi_\zeta \circ \phi_\xi^{-1})(x^{[1]}, x^{[2]}) = \phi_{\zeta\xi}(x^{[1]}, x^{[2]}) = [\phi_{\zeta\xi}^{[1]}(x^{[1]}, x^{[2]}), \phi_{\zeta\xi}^{[2]}(x^{[1]}, x^{[2]})]^T,$$

where for $i = 1, 2$,

$$\phi_{\zeta\xi}^{[i]}(x^{[1]}, x^{[2]}) = \langle \xi - \zeta + x^{[1]}\gamma_\xi^{[1]} + x^{[2]}\gamma_\xi^{[2]} + \delta_\mu(x^{[1]}, x^{[2]})n_\xi, \gamma_\zeta^{[i]} \rangle \quad (1)$$

in a neighborhood of $\phi_\xi(\mu)$. Therefore $\phi_{\zeta\xi} : \phi_\xi(U_\xi \cap U_\zeta) \rightarrow \phi_\zeta(U_\xi \cap U_\zeta)$ is a C^2 mapping. Note that even if a C^1 mapping $\phi_{\zeta\xi}$ would suffice for our goals, we do need the assumption that Ω is a C^2 surface to ensure that the normal n_ξ changes smoothly with ξ , which guarantees the C^1 smoothness of our approximants, see [7].

For a C^1 function f defined in a neighborhood U of $\zeta \in \Omega$, we define $J_\zeta(f) : U \cap U_\zeta \rightarrow \mathbb{R}^{2 \times 2}$ by

$$J_\zeta(f)(\mu) := J(f \circ \phi_\zeta^{-1})(\phi_\zeta(\mu)), \quad \mu \in U \cap U_\zeta,$$

where for any smooth function $g : \mathbb{R}^2 \rightarrow \mathbb{R}^2$, $g = [g^{[1]}, g^{[2]}]^T$, $J(g)$ denotes its *Jacobian*

$$J(g) := \begin{bmatrix} \frac{\partial g^{[1]}}{\partial x^{[1]}} & \frac{\partial g^{[1]}}{\partial x^{[2]}} \\ \frac{\partial g^{[2]}}{\partial x^{[1]}} & \frac{\partial g^{[2]}}{\partial x^{[2]}} \end{bmatrix}.$$

We write

$$J_{\zeta\xi} := J_\xi(\phi_\zeta), \quad \text{on } U_\zeta \cap U_\xi,$$

so that

$$J_{\zeta\xi}(\mu) = J_\xi(\phi_\zeta)(\mu) = J(\phi_{\zeta\xi})(\phi_\xi(\mu)), \quad \mu \in U_\zeta \cap U_\xi,$$

is the Jacobian of $\phi_{\zeta\xi}$ evaluated at $\phi_\xi(\mu)$. Since $\phi_{\zeta\xi}^{-1} = \phi_{\xi\zeta}$, the well-known properties of the Jacobian imply

$$[J_{\zeta\xi}(\mu)]^{-1} = J_{\xi\zeta}(\mu). \quad (2)$$

For later use, we now obtain explicit formulas for the Jacobian $J_{\zeta\xi}(\xi) := J(\phi_{\zeta\xi})(\phi_\xi(\xi))$ and its determinant in the case of the projection atlas. By the

above construction, the implicit function $x^{[3]} = \delta_\xi(x^{[1]}, x^{[2]})$ is C^2 in a neighborhood of the origin $\phi_\xi(\xi)$. Moreover, it vanishes together with its gradient at the origin. Hence, by (1),

$$J_{\zeta\xi}(\xi) = [\langle \gamma_\zeta^{[i]}, \gamma_\xi^{[j]} \rangle]_{i,j=1,2}. \quad (3)$$

Clearly, the determinant of this matrix is the projection of $n_\zeta = \gamma_\zeta^{[1]} \times \gamma_\zeta^{[2]}$ on n_ξ , i.e.,

$$\det J_{\zeta\xi}(\xi) = \langle n_\zeta, n_\xi \rangle. \quad (4)$$

2.2 Projected gradients

Let $f \in C^1(\Omega)$, and let $f_\xi = f \circ \phi_\xi^{-1}$. Since Ω is embedded in \mathbb{R}^3 , the gradient $\nabla f_\xi = \left[\frac{\partial f_\xi}{\partial x^{[1]}}, \frac{\partial f_\xi}{\partial x^{[2]}} \right]$ of f_ξ can be identified with the 3-vector

$$\text{grad } f_\xi = \frac{\partial f_\xi}{\partial x^{[1]}} \gamma_\xi^{[1]} + \frac{\partial f_\xi}{\partial x^{[2]}} \gamma_\xi^{[2]}$$

lying in the tangent plane $\Gamma_\xi \subset \mathbb{R}^3$. We write

$$\text{grad}_\xi f(\mu) := (\text{grad } f_\xi)(\phi_\xi(\mu)), \quad \mu \in U_\xi,$$

for the gradient of f_ξ evaluated at $\phi_\xi(\mu)$. We call $\text{grad}_\xi f(\mu)$ the *projected gradient* of f at μ . It is easy to see that $\text{grad}_\xi f(\xi)$ coincides with the standard gradient of a function on a 2-surface in \mathbb{R}^3 , as defined for example in [28, p. 96]. We also need projected gradients when $\mu \neq \xi$.

Lemma 1. *For any $\xi \in \Omega$ and $\zeta \in U_\xi$, the projected gradient $\text{grad}_\zeta f(\zeta)$ is the orthogonal projection of $\text{grad}_\xi f(\zeta)$ onto Γ_ζ . In particular,*

$$\text{grad}_\zeta f(\zeta) = \text{grad}_\xi f(\zeta) - \langle \text{grad}_\xi f(\zeta), n_\zeta \rangle n_\zeta, \quad (5)$$

and

$$\text{grad}_\xi f(\zeta) = \text{grad}_\zeta f(\zeta) - \frac{\langle \text{grad}_\zeta f(\zeta), n_\xi \rangle}{\langle n_\zeta, n_\xi \rangle} n_\zeta, \quad \text{if } \langle n_\zeta, n_\xi \rangle \neq 0, \quad (6)$$

where n_ζ and n_ξ are the unit normal vectors to Γ_ζ and Γ_ξ , respectively.

Proof. We have

$$\text{grad}_\xi f(\zeta) = \frac{\partial f_\xi}{\partial x^{[1]}}(\phi_\xi(\zeta)) \gamma_\xi^{[1]} + \frac{\partial f_\xi}{\partial x^{[2]}}(\phi_\xi(\zeta)) \gamma_\xi^{[2]}.$$

Its projection onto Γ_ζ is therefore

$$\begin{aligned} & \left(\frac{\partial f_\xi}{\partial x^{[1]}}(\phi_\xi(\zeta)) \langle \gamma_\xi^{[1]}, \gamma_\zeta^{[1]} \rangle + \frac{\partial f_\xi}{\partial x^{[2]}}(\phi_\xi(\zeta)) \langle \gamma_\xi^{[2]}, \gamma_\zeta^{[1]} \rangle \right) \gamma_\zeta^{[1]} \\ & + \left(\frac{\partial f_\xi}{\partial x^{[1]}}(\phi_\xi(\zeta)) \langle \gamma_\xi^{[1]}, \gamma_\zeta^{[2]} \rangle + \frac{\partial f_\xi}{\partial x^{[2]}}(\phi_\xi(\zeta)) \langle \gamma_\xi^{[2]}, \gamma_\zeta^{[2]} \rangle \right) \gamma_\zeta^{[2]}. \end{aligned}$$

This last expression coincides with $\text{grad}_\zeta f(\zeta)$, since

$$\nabla f_\zeta(\phi_\zeta(\zeta)) = \nabla f_\xi(\phi_\xi(\zeta))J_{\xi\zeta}(\zeta) = \nabla_\xi f(\zeta) [(\gamma_\xi^{[i]}, \gamma_\zeta^{[j]})]_{i,j=1,2}$$

by the chain rule and (3).

The formulas (5) and (6) for the projection and inverse projection, respectively, follow immediately. ■

2.3 Consistent triangulations

Given a finite set \mathcal{V} of points in Ω , let \mathcal{T} be a set of triples $\tau = \{v, u, w\}$ of points $v, u, w \in \mathcal{V}$ such that

- any two triples have at most two common points,
- any pair of points in \mathcal{V} belong to at most two different triples in \mathcal{T} , and
- for any $v \in \mathcal{V}$, the set of all triples containing v forms a *cell*, i.e. $\{\tau \in \mathcal{T} : v \in \tau\} = \{\tau_i : i = 1, \dots, n\}$ for some $n \geq 3$, where $\tau_i = \{v, v_i, v_{i+1}\}$, with v_1, \dots, v_n all different, and $v_{n+1} = v_1$.

If these conditions are satisfied, we say that \mathcal{T} is a *triangulation* of Ω with *vertices* \mathcal{V} . We say that two vertices v_1, v_2 are *connected* in \mathcal{T} if there is a triple $\tau \in \mathcal{T}$ containing both v_1 and v_2 . This definition of a triangulation of a manifold Ω is described by connectivity of vertices only, and does not involve “edges” or “triangles” on Ω . Indeed, \mathcal{T} is essentially an *abstract simplicial complex* [21] with vertices in Ω .

Given $\xi \in \Omega$, assuming that all vertices of $\tau = \{v, u, w\} \in \mathcal{T}$ are in U_ξ , we denote by $\phi_\xi(\tau)$ the (open) planar triangle in B_ξ with vertices $\phi_\xi(u), \phi_\xi(v), \phi_\xi(w)$. Note that the triangle $\phi_\xi(\tau)$ may be degenerate. We set $\mathcal{T}_\xi = \{\tau \in \mathcal{T} : \tau \subset U_\xi\}$ and $\Delta_\xi = \{\phi_\xi(\tau) : \tau \in \mathcal{T}_\xi\}$.

Definition 1. A triangulation \mathcal{T} of Ω is said to be *consistent* with the projection atlas $\Phi = \{(U_\xi, \phi_\xi)\}_{\xi \in \Omega}$ if for any $\xi \in \Omega$,

- every triangle $T \in \Delta_\xi$ is non-degenerate,
- Δ_ξ is a planar triangulation of $P_\xi := \cup_{T \in \Delta_\xi} \overline{T}$,
- $\phi_\xi(\xi)$ lies in the interior of P_ξ .

For any $\xi \in \Omega$, let \mathcal{V}_ξ be the set consisting of vertices of all $\tau \in \mathcal{T}_\xi$ such that $\phi_\xi(\xi)$ lies in the closure of $\phi_\xi(\tau)$, i.e.

$$\mathcal{V}_\xi := \{v \in \mathcal{V} \cap U_\xi : \phi_\xi(\xi) \in \overline{\phi_\xi(\tau)} \text{ for some } \tau \in \mathcal{T}_\xi \text{ with a vertex at } v\}.$$

For a consistent triangulation \mathcal{T} , it is not difficult to check that $\mathcal{V}_\zeta \subseteq \mathcal{V}_\xi$ for all $\zeta \in \Omega$ sufficiently close to ξ , see [7]. Moreover, if ξ is a vertex in \mathcal{V} , then \mathcal{V}_ξ consists of ξ and all vertices connected to it. For any point $\xi \in \Omega \setminus \mathcal{V}$, the set \mathcal{V}_ξ contains either three or four points, depending on whether $\phi_\xi(\xi)$ belongs to the interior of a triangle in Δ_ξ or it lies on a common edge of two such triangles.

In addition to consistency, we will need the following assumption, specifically related to the Powell-Sabin spline:

$$\text{for every } \xi \in \Omega, \text{ if } \phi_\xi(\xi) \in \overline{T} \text{ for a } T \in \Delta_\xi, \text{ then } \Delta_\xi \text{ also includes} \quad (7)$$

three triangles sharing edges with T .

We extend \mathcal{V}_ξ to $\tilde{\mathcal{V}}_\xi$ by adding to \mathcal{V}_ξ the vertices of the triangles described in (7).

3 Interpolation and data fitting

3.1 An interpolation method

Let \mathcal{T} be a consistent triangulation of Ω . We assume that \mathcal{T} is fine enough for (7) to hold. Let $D := \{a_v, c_v\}_{v \in \mathcal{V}}$, where a_v are real numbers and c_v are 3-vectors in Γ_v . We now show how to construct a C^1 function $s_{\mathcal{T}}$ defined on Ω that satisfies the interpolation conditions

$$s_{\mathcal{T}}(v) = a_v, \quad \text{grad}_v s_{\mathcal{T}}(v) = c_v, \quad \text{all } v \in \mathcal{V}. \quad (8)$$

Algorithm 1. *Given $\xi \in \Omega$, compute $s_{\mathcal{T}}(\xi)$:*

1. Let $T := \langle w_1, w_2, w_3 \rangle$ be a triangle in Δ_ξ such that $\phi_\xi(\xi) \in \overline{T}$, and let $T_1 := \langle w_4, w_3, w_2 \rangle$, $T_2 := \langle w_5, w_1, w_3 \rangle$, and $T_3 := \langle w_6, w_2, w_1 \rangle$ be the three triangles in Δ_ξ sharing edges with T , see Figure 1(left).
2. Let T_{PS} be the Powell-Sabin split of T into six triangles obtained by connecting the incenter w of T to the incenters of T_1, T_2, T_3 , and to the vertices w_1, w_2, w_3 , see Figure 1(right).
3. Let $g_i := c_{v_i} - \frac{\langle c_{v_i}, n_\xi \rangle}{\langle n_{v_i}, n_\xi \rangle} n_{v_i}$, where $v_i = \phi_\xi^{-1}(w_i)$, for $i = 1, 2, 3$.
4. Let $s_{\mathcal{T}}(\xi)$ be the value at $\phi_\xi(\xi)$ of the Powell-Sabin C^1 quadratic spline s_ξ defined on T_{PS} that interpolates the values $\{a_{v_i}\}_{i=1}^3$ and the gradients corresponding to $\{g_i\}_{i=1}^3$ at the vertices $\{w_i\}_{i=1}^3$, see [26].

Since the Powell-Sabin interpolant in step 4 is uniquely defined by the values $\{(a_{v_i}, g_i)\}_{i=1}^3$ at the vertices $\{w_i\}_{i=1}^3$, it follows that $s_{\mathcal{T}}$ is uniquely defined by the data D . By construction, $s_{\mathcal{T}}$ satisfies (8). It would be tempting to consider $s_{\mathcal{T}}$ to be a spline on a partition of the manifold Ω obtained by drawing curves on Ω between connected vertices of \mathcal{T} . Indeed, $s_{\mathcal{T}}$ possesses a kind of piecewise structure, see [7]. However, this does not seem to be of a practical significance. What is important is that $s_{\mathcal{T}}$ is a smooth function.

Theorem 1 ([7]). *The interpolant $s_{\mathcal{T}}$ defined by Algorithm 1 is a C^1 function on the manifold Ω .*

Suppose that $s_{\mathcal{T}}(f)$ is the interpolant $s_{\mathcal{T}}$ corresponding to the data

$$a_v := f(v), \quad c_v := \text{grad}_v f(v), \quad \text{all } v \in \mathcal{V},$$

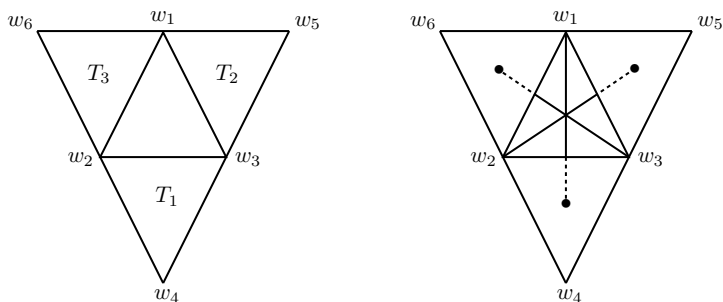


Fig. 1. The triangles in Algorithm 1.

where f is a smooth function defined on Ω . Denote by h the *mesh size* of \mathcal{T} , by which we mean in the current setting the maximum distance in \mathbb{R}^3 between any pair of vertices $v \in \mathcal{V}$ connected in \mathcal{T} . By actually connecting these pairs of vertices by straight line segments, we obtain a 2-dimensional triangulation in \mathbb{R}^3 . Let α be the *smallest angle* appearing in its triangles.

Theorem 2 ([7]). *If f is a sufficiently smooth real function on Ω , then*

$$\|f - s_{\mathcal{T}}(f)\|_{C(\Omega)} \leq K h^3, \quad (9)$$

where K is a constant depending only on f and α .

Note that this error bound crucially depends on using the above formulas for the transformation of the gradients in step 3 of Algorithm 1 and in step 1(d) of Algorithm 2. These formulas are related to (6) and (5), respectively.

3.2 A two-stage data fitting method

In practice we are frequently given only values of an unknown function f at a set X of scattered data points on the manifold Ω . In this case we can use a two-stage method to construct an approximation. First we select a consistent triangulation \mathcal{T} of Ω satisfying (7). Let \mathcal{V} be the set of vertices of \mathcal{T} . Note that we do not require that the vertices be located at the data points of X , and the number of vertices may be much smaller than the number of data points.

In the *first stage* of the algorithm we compute approximations to the values $\{f(v), \text{grad}_v f(v)\}_{v \in \mathcal{V}}$ based on the data $\{f(\xi)\}_{\xi \in X}$. We perform these calculations in the sets $B_v := \phi_v(U_v) \subset \mathbb{R}^2$ using techniques available for local fitting of bivariate data. To carry this out, we suppose that

$$X \text{ is sufficiently dense so that } X \cap U_v \neq \emptyset \text{ for each } v \in \mathcal{V}. \quad (10)$$

Experience with the bivariate case [8] suggests that for each $v \in \mathcal{V}$, we compute both $a_v \approx f(v)$ and $c_v \approx \text{grad}_v f(v)$ by averaging several estimates

of the same quantities based on different sets of nearby data. It follows from the consistency of \mathcal{T} that for each vertex $v \in \mathcal{V}$, all vertices of \mathcal{T} connected to v belong to the set U_v .

Algorithm 2. Given $\{f(\xi)\}_{\xi \in X}$, compute $\{a_v, c_v\}_{v \in \mathcal{V}}$:

1. For each $v \in \mathcal{V}$,
 - (a) Let $v_0 := v$, and let $v_1, v_2, \dots, v_n \in \mathcal{V}$ be the set of all vertices of \mathcal{T} connected to v . Let $\tilde{v}_i = \phi_v(v_i)$, $i = 1, \dots, n$.
 - (b) Choose a set $\tilde{X}_v \subset \phi_v(X \cap U_v)$ of points in B_v near $\phi_v(v)$.
 - (c) Compute a bivariate approximation p_v defined on B_v based on the data $\{f_v(\xi)\}_{\xi \in \tilde{X}_v}$, where $f_v := f \circ \phi_v^{-1}$.
 - (d) Store the numbers $a_{v,v_i} := p_v(\tilde{v}_i)$ and vectors $c_{v,v_i} := \text{grad} p_v(\tilde{v}_i) - \langle \text{grad} p_v(\tilde{v}_i), n_{v_i} \rangle n_{v_i}$ for $i = 0, \dots, n$.
2. For each $v \in \mathcal{V}$, set

$$a_v := \frac{1}{n+1} \sum_{i=0}^n a_{v_i,v}, \quad c_v := \frac{1}{n+1} \sum_{i=0}^n c_{v_i,v}.$$

In the *second stage* of the algorithm we construct our approximant $s_{\mathcal{T}}$ as the interpolant (8) to the data $\{a_v, c_v\}_{v \in \mathcal{V}}$ obtained from Algorithm 2.

We have not specified how \mathcal{T} is selected and how the steps 1(b) and 1(c) of Algorithm 2 are to be performed. However, the overall performance of the two-stage method will depend significantly on the particular techniques used in these steps. We discuss two numerical examples in Section 4, using recently developed adaptive techniques based on local least squares fitting by bivariate polynomials and radial basis functions [5, 6, 8].

We now give an error bound for this method in terms of the mesh size h and the approximation power of the local approximations p_v .

Let $\kappa(\xi)$ be the minimum of $\langle n_\xi, n_v \rangle$ over all $v \in \mathcal{V} \cap U_\xi$ such that $\xi = \phi_\xi(\xi)$ belongs to the closure of a triangle of Δ_ξ attached to $\phi_\xi(v)$, and let

$$\kappa = \min_{\xi \in \Omega} \kappa(\xi).$$

By the definition of U_ξ , it follows that $\kappa > 0$. Moreover, $\kappa \rightarrow 1$ as the mesh size h of \mathcal{T} goes to 0.

For each $v \in \mathcal{V}$, let N_v be the union of all triangles of Δ_v attached to v , and let p_v be the bivariate approximation to $f_v = f \circ \phi_v^{-1}$, as in Algorithm 2.

Theorem 3 ([7]). Let $s_{\mathcal{T}}$ be the approximant to a sufficiently smooth function f on Ω constructed by the above two-stage scattered data fitting method. Then

$$\|f - s_{\mathcal{T}}\|_{C(\Omega)} \leq K \left[C_f h^3 + \max_{v \in \mathcal{V}} \left\{ \|f_v - p_v\|_{C(N_v)} + h \|\text{grad} f_v - \text{grad} p_v\|_{C(N_v)} \right\} \right],$$

where the constant K depends only on κ and the smallest angle α , and C_f depends only on f .

Clearly, this theorem shows that if the local approximations p_v are powerful enough to guarantee an $\mathcal{O}(h^3)$ error, then the overall error of the two-stage method is also $\mathcal{O}(h^3)$. This asymptotic behavior of the error is confirmed by the numerical examples in the next section.

4 Numerical examples

4.1 Scattered data fitting on the unit sphere

For our first example, we choose the manifold Ω to be the unit sphere in \mathbb{R}^3 defined by $x^2 + y^2 + z^2 = 1$. As a test function, we take the function

$$f_1(x, y, z) := 1 + x^8 + e^{2y^3} + e^{2z^2} + 10xyz$$

used in the examples in [1]. As in [1], we visualize this function as a kind of offset surface to the manifold, i.e., we plot

$$\{\xi + f_1(\xi)n_\xi : \xi \in \Omega\},$$

where n_ξ denotes the unit outer normal to the manifold Ω at ξ , see Figure 2.

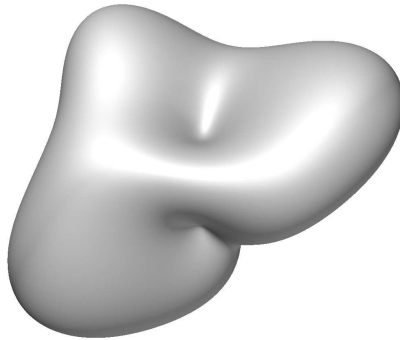


Fig. 2. Test function f_1 on the sphere.

In order to study the behavior of the error as a function of mesh size, we use the nested sequence of triangulations introduced in [1]. Let \mathcal{T}_0 be the regular octahedron with vertices at $\pm e_i$, $i = 1, 2, 3$, where e_i are the Cartesian coordinate vectors. This triangulation has 6 vertices and 8 triangles. We now define \mathcal{T}_n by repeated refinement, where \mathcal{T}_n is obtained from \mathcal{T}_{n-1} by adding vertices at the midpoints of the great circle arcs connecting neighboring vertices of \mathcal{T}_{n-1} , and then splitting each triangle in \mathcal{T}_{n-1} into four subtriangles using these new vertices. The number of vertices of \mathcal{T}_n is $V_n = 2^{2n+2} + 2$, and the number of triangles is 2^{2n+3} . The triangulations \mathcal{T}_1 and \mathcal{T}_2 are shown in Figure 3. Note

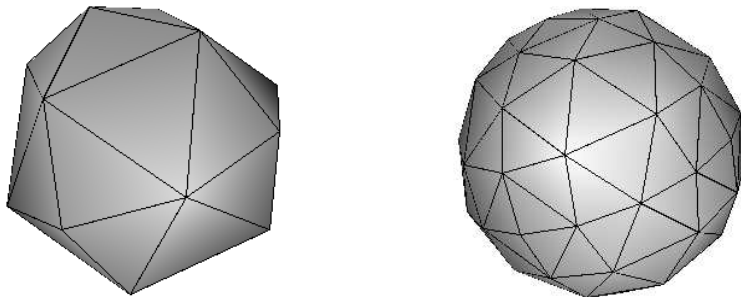


Fig. 3. Triangulations \mathcal{T}_1 and \mathcal{T}_2 .

that \mathcal{T}_0 is not sufficiently fine for our Powell-Sabin interpolant to be defined, see Section 3.1.

To get test data, we use a simple spherical random number generator, see Remark 6. In particular, for each $n = 1, 2, \dots, 6$, we generate $3V_n = 3(2^{2n+2} + 2)$ random points. We choose this number since as shown in Section 3.1, the Powell-Sabin interpolant is uniquely defined by $3V_n$ degrees of freedom. We then evaluate the test function f_1 at these points, and create approximants s_n by the two-stage method described in Section 3.2.

In the first stage we use least squares multiquadric fitting as described in [6] with the following parameter values: minimum number of points $M_{min} = 25$, maximum number of points $M_{max} = 100$, separation $S = 35$, and scaling $\delta = 0.2$ if $n = 1$, $\delta = 1.0$ if $n = 2$, $\delta = 2.0$ if $n = 3$, $\delta = 3.0$ if $n \in \{4, 5\}$, $\delta = 4.0$ if $n = 6$. See [6] for the exact meaning of M_{min} , M_{max} , S and δ . The choice of increasing values for the δ 's as the number of data points increases is motivated by numerical results in [6]. In our implementation we make use of the corresponding subroutines from the software library TSFIT [9].

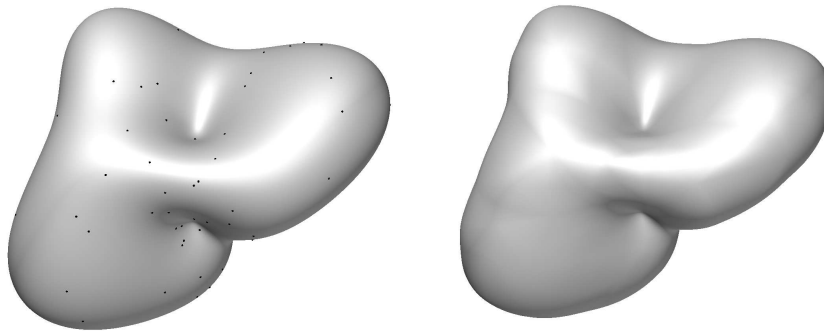
The results of our experiments are presented in Table 1. In the column labelled *Two Stage*, we list the relative maximum errors $\|f_1 - s_n\|_\infty / \|f_1\|_\infty$ for $n = 1, \dots, 6$. For comparison purposes, in the column labelled *Exact*, we list the relative errors when the exact function values and projected gradients are used instead of the approximate values obtained from first stage fitting. In the column labelled *Spherical PS* we also list the relative errors corresponding to using the spherical Powell Sabin interpolant of [1] based on the exact function values and gradients at the vertices of \mathcal{T}_n .

The table shows that the errors for the three methods are comparable, and indeed for $n \geq 2$ are almost identical. To test the rate of convergence, in the column labelled *Ratio* we list the ratios of the errors of our two-stage method, i.e., $\|f_1 - s_{n-1}\| / \|f_1 - s_n\|$ for $n = 2, \dots, 6$. Since the mesh size decreases by approximately $1/2$ at each refinement step, and since the error should be of size $\mathcal{O}(h^3)$, these ratios tend to 8 as they should.

Table 1. Tests with random data on the sphere.

	# data	Two Stage	Ratio	Exact	Spherical PS
\mathcal{T}_1	54	2.74×10^{-1}		8.08×10^{-2}	7.83×10^{-2}
\mathcal{T}_2	198	2.15×10^{-2}	12.7	2.17×10^{-2}	2.10×10^{-2}
\mathcal{T}_3	774	2.10×10^{-3}	10.2	2.10×10^{-3}	2.05×10^{-3}
\mathcal{T}_4	3078	2.41×10^{-4}	8.7	2.41×10^{-4}	2.28×10^{-4}
\mathcal{T}_5	12294	2.92×10^{-5}	8.3	2.92×10^{-5}	2.88×10^{-5}
\mathcal{T}_6	49158	3.65×10^{-6}	8.0	3.65×10^{-6}	3.60×10^{-6}

To illustrate the performance of the two-stage method with even fewer data points, we recomputed the spline fit corresponding to the triangulation \mathcal{T}_2 , but with only 100 random points on the sphere. The parameters for the first step were taken to be $M_{min} = 25$, $M_{max} = 100$, $S = 25$ and $\delta = 0.8$. In this case the maximum relative error was 2.18×10^{-2} . Figure 4 shows the test function f_1 along with the data sites and the approximant computed using the two-stage method.

**Fig. 4.** Test function with 100 data sites (left) and its approximation (right).

4.2 Ring type surfaces

As a second example, we consider the ring-type surfaces used in the examples presented in [25]. Given a real number $0 \leq a < 1$ and an integer $m \geq 0$, we define a smooth 2-manifold in \mathbb{R}^3 parametrically via

$$x = [2 + (1 + a \cos mu) \cos v] \cos u, \quad (11)$$

$$y = [2 + (1 + a \cos mu) \cos v] \sin u, \quad (12)$$

$$z = [1 + a \cos mu] \sin v, \quad (13)$$

where (u, v) runs over the parameter domain $[0, 2\pi) \times [0, 2\pi)$. When $m = 0$, this corresponds to a torus with outer radius $3 + a$ and inner radius $1 - a$. In general it is a surface of genus 1.

For our experiments, we choose $a = 0.3$ and $m = 5$, giving rise to the manifold depicted in Figure 5 (left). As a test function, we now take

$$f_2(x, y, z) := (1 + x^8 + y^3 + z^2)/4000.$$

The corresponding surface is shown in Figure 5 (right).

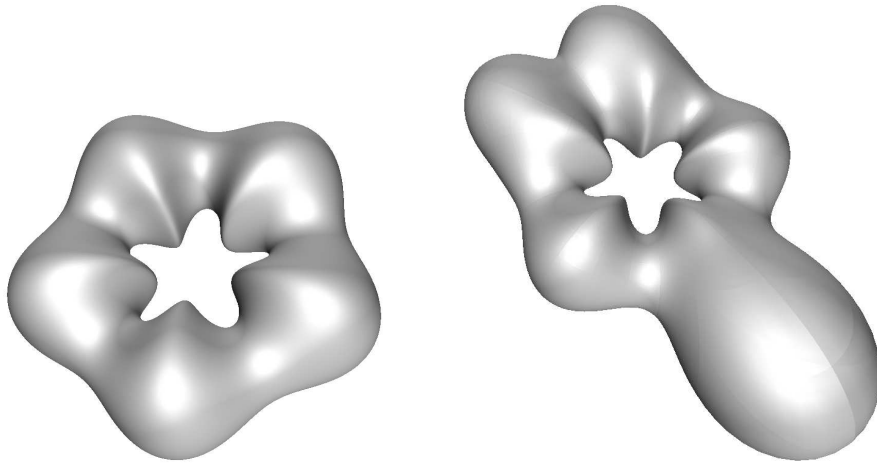


Fig. 5. Ring-type surface with $a = 0.3$ and $m = 5$ (left), and the test function f_2 visualized as an offset surface (right).

For comparison purposes, we generate a sequence of triangulations \mathcal{T}_n , $n = 1, \dots, 5$, by starting with nested three directional meshes in the parameter domain $[0, 2\pi) \times [0, 2\pi)$ with vertices (u_i, v_j) given by

$$u_i = \frac{2\pi i}{24 \cdot 2^{n-1}}, \quad v_j = \frac{2\pi j}{15 \cdot 2^{n-1}},$$

and mapping these vertices onto the ring-type surface. The triangulations \mathcal{T}_1 and \mathcal{T}_2 are shown in Figure 6. The number of vertices of \mathcal{T}_n is $V_n = 90 \cdot 4^n$, and the number of triangles is $180 \cdot 4^n$. To generate data for our experiments, we evaluate the test function f_2 at $3V_n$ random points on the surface. As in Section 4.1, we use local multiquadric fitting [6] with the following parameter values: $M_{min} = 25$, $M_{max} = 100$, $S = 15$, and $\delta = 2.0$ if $n = 1$, $\delta = 3.0$ if $n \geq 2$. Table 2 shows the relative maximum error for the two-stage method and for the exact interpolant. The somewhat irregular convergence rate seen in the

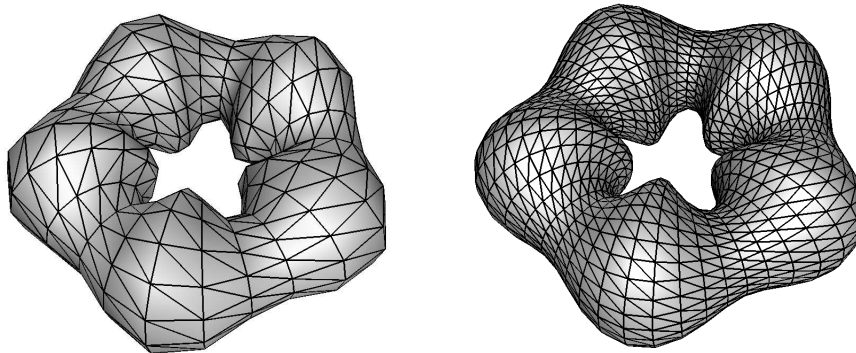


Fig. 6. Triangulations \mathcal{T}_1 and \mathcal{T}_2 .

Table 2. Tests with random data on the ring type surface.

	# data	Two Stage	Ratio	Exact
\mathcal{T}_1	1080	5.57×10^{-2}		5.50×10^{-2}
\mathcal{T}_2	4320	2.84×10^{-2}	1.96	2.84×10^{-2}
\mathcal{T}_3	17280	4.01×10^{-3}	7.08	4.01×10^{-3}
\mathcal{T}_4	69120	9.18×10^{-5}	43.7	9.17×10^{-5}
\mathcal{T}_5	276480	1.14×10^{-5}	8.05	1.14×10^{-5}

'ratio' column is probably due to the fact that we are using easy to generate triangulations that are not well adapted to this particular manifold.

Figure 7 shows the \mathcal{T}_2 -approximation computed using 1000 random data, with parameters $M_{min} = 25$, $M_{max} = 100$, $S = 15$, and $\delta = 2.0$. The relative error of this approximation is 2.89×10^{-2} .

5 Remarks

Remark 1. The problem of fitting functions defined on surfaces arises in many applications, see [1–4, 10–14, 19, 20, 23, 25, 27, 29], and references therein. Used parametrically, such functions can be applied to the problem of modelling surfaces of arbitrary topological type from point clouds, see [15, 16, 30].

Remark 2. Many of the papers mentioned in the above remark deal with the sphere in \mathbb{R}^3 . For a survey of interpolation and scattered data fitting methods on the sphere, see [12]. For some specific methods, see [14, 19, 20, 22, 23, 27, 29].

Remark 3. The method of this paper is closely related to work of Demjanovich [10, 11]. He also computes an interpolant s at a point ξ on the manifold by projecting into the tangent plane and using a bivariate finite element interpolation method. A key difference is that for each evaluation point ξ , his method involves interpolation of the original function at the points in the tangent plane

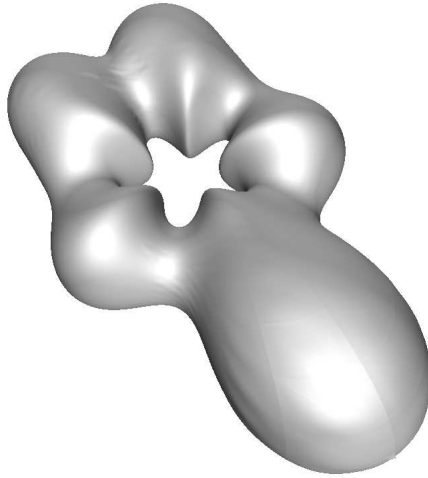


Fig. 7. Powell-Sabin approximation computed using 1000 data points.

Γ_ξ determined by the finite element scheme, whereas in our methods we only interpolate projected gradients corresponding to the vertices of the underlying triangulation \mathcal{T} , compare steps 3 and 4 of Algorithm 1. Therefore, our interpolation operator only requires function values and gradients at the vertices of \mathcal{T} , which makes it possible to design a two-stage scattered data fitting method. In [10, 11] only interpolation with Courant hat functions has similar properties for general manifolds, but it does not produce a C^1 interpolant.

Remark 4. The method of this paper is also closely related to work of Pottmann [25], which also makes use of projected gradients. (It is not difficult to see that our equation (6) describes the π -transform of [25].) However, instead of using local approximation methods to estimate gradients, he constructs a kind of minimum norm network.

Remark 5. Here we have made use of the standard bivariate C^1 quadratic Powell-Sabin macro-element to solve the interpolation problem in the tangent plane. Its key feature is that it is constructed from only nine pieces of data, the values and gradients at the three vertices of the macro-triangle. Using the same data, we can also construct an interpolant based on the classical C^1 *reduced Clough-Tocher macro-element*. It is based on a split of the macro-triangle into three subtriangles (typically using the barycenter), and is a cubic polynomial on each piece. Along each edge its cross derivative is restricted to be a linear polynomial. Yet another possibility is a modified quadratic *Powell-Sabin macro-element on a 12-split* [26], where the cross derivatives are assumed linear rather than piecewise linear on the edges of the macro-triangles. Note that with either a Clough-Tocher or 12-split Powell-Sabin macro-element the assumption (7) will not be needed.

Remark 6. To simulate scattered data for our numerical experiments in Section 4, we generated pseudo-random points uniformly distributed on the test surfaces. For the *surface of the sphere* we use the following method described e.g. in [24]: To obtain a point (x, y, z) on the unit sphere, generate two pseudo-random real numbers z and t uniformly distributed in $[-1, 1]$ and $[0, 2\pi)$, respectively, and then compute $x = \rho \cos t$ and $y = \rho \sin t$, where $\rho = \sqrt{1 - z^2}$. For the *ring-type surface* defined parametrically by (11)–(13), the points $r(u, v) = (x(u, v), y(u, v), z(u, v))$ will be uniformly distributed on the surface if $(u, v) \in [0, 2\pi) \times [0, 2\pi)$ are chosen according to the probability distribution $p(u, v) = \alpha \|r_u \times r_v\|_2$, with $\alpha \in \mathbb{R}$ such that $\int_{[0, 2\pi) \times [0, 2\pi)} p(u, v) du dv = 1$. It is not difficult to see that $\|r_u \times r_v\|_2^2 = \psi^2[(\psi')^2 + (2 + \psi \cos v)^2]$, where $\psi(u) = 1 + a \cos mu$. Using this explicit formula, we employ the well-known von Neumann rejection method to generate the points.

References

1. P. Alfeld, M. Neamtu and L. L. Schumaker, Fitting scattered data on sphere-like surfaces using spherical splines, *J. Comp. Appl. Math.* **73** (1996), 5–43.
2. R. E. Barnhill and T. A. Foley, Methods for constructing surfaces on surfaces, in “Geometric Modeling: Methods and Their Applications”, Springer, Berlin, 1991, pp. 1–15.
3. R. E. Barnhill, B. R. Piper and S. E. Stead, A multidimensional surface problem: pressure on a wing, *Comput. Aided Geom. Design* **2** (1985), 185–187.
4. R. E. Barnhill and H. S. Ou, Surfaces defined on surfaces, *Comput. Aided Geom. Design* **7** (1990), 323–336.
5. O. Davydov, R. Morandi and A. Sestini, Local hybrid approximation for scattered data fitting with bivariate splines, *Comput. Aided Geom. Design* **23** (2006), 703–721.
6. O. Davydov, A. Sestini and R. Morandi, Local RBF approximation for scattered data fitting with bivariate splines, in “Trends and Applications in Constructive Approximation,” (M. G. de Bruin, D. H. Mache, and J. Szabados, Eds.), ISNM Vol. 151, Birkhäuser, 2005, pp. 91–102.
7. O. Davydov and L. L. Schumaker, Interpolation and scattered data fitting on manifolds using projected Powell-Sabin splines, manuscript, 2007.
8. O. Davydov and F. Zeilfelder, Scattered data fitting by direct extension of local polynomials to bivariate splines, *Advances in Comp. Math.* **21** (2004), 223–271.
9. O. Davydov and F. Zeilfelder, TSFIT: A Software Package for Two-Stage Scattered Data Fitting, 2005, available under GPL from <http://www.maths.strath.ac.uk/~aas04108/tsfit/>
10. Yu. K. Demjanovich, Construction of spaces of local functions on manifolds, *Metody Vychisl.* (1985) no. 14, 100–109.
11. Yu. K. Demjanovich, Local approximations of functions given on manifolds, *Amer. Math. Soc. Transl. (2)* **159** (1994), 53–76.
12. G. Fasshauer and L. L. Schumaker, Scattered data fitting on the sphere, in “Mathematical Methods for Curves and Surfaces II”, M. Dæhlen, T. Lyche, and L. L. Schumaker (eds.), Vanderbilt University Press, Nashville, 1998, 117–166.
13. T. A. Foley, D. A. Lane, G. M. Nielson, R. Franke, and H. Hagen, Interpolation of scattered data on closed surfaces, *Comput. Aided Geom. Design* **7** (1990), 303–312.

14. W. Freeden, Spherical spline interpolation-basic theory and computational aspects, *J. Comp. Appl. Math.* **11** (1985), 367–375.
15. C. M. Grimm and J. F. Hughes, Modeling surfaces of arbitrary topology using manifolds, in “Proceedings of SIGGRAPH 95,” 1995, pp. 359–368
16. Xianfeng Gu, Ying He, and Hong Qin, Manifold splines, in “Proceedings of ACM Symposium on Solid and Physical Modeling,” 2005, pp. 27–38.
17. M. W. Hirsch, *Differential Topology*, Springer-Verlag, 1976.
18. M.-J. Lai and L. L. Schumaker, *Spline Functions on Triangulations*, Cambridge University Press, Cambridge, 2007.
19. C. L. Lawson, C^1 surface interpolation for scattered data on a sphere, *Rocky Mountain J. Math.* **14** (1984), 177–202.
20. T. Lyche and L. L. Schumaker, A multiresolution tensor spline method for fitting functions on the sphere, *SIAM J. Sci. Computing* **22** (2000), 724–746.
21. C. R. F. Maunder, *Algebraic Topology*, Van Nostrand Reinhold Company, London, 1970.
22. M. Neamtu and L. L. Schumaker, On the approximation order of splines on spherical triangulations, *Adv. in Comp. Math.* **21** (2004), 3–20.
23. G. M. Nielson and R. Ramaraj, Interpolation over a sphere based upon a minimum norm network, *Comput. Aided Geom. Design* **4** (1987), 41–58.
24. J. O’Rourke, *Computational Geometry in C* (2nd Ed.), Cambridge University Press, 1998.
25. H. Pottmann, Interpolation on surfaces using minimum norm networks, *Computer Aided Geometric Design* **9** (1992), 51–67.
26. M. J. D. Powell and M. A. Sabin, Piecewise quadratic approximations on triangles, *ACM Trans. Math. Software* **3** (1977), 316–325.
27. L. L. Schumaker and C. Traas, Fitting scattered data on spherelike surfaces using tensor products of trigonometric and polynomial splines, *Numer. Math.* **60** (1991), 133–144.
28. J. A. Thorpe, *Elementary Topics in Differential Geometry*, Springer, 1979.
29. G. Wahba, Spline interpolation and smoothing on the sphere, *SIAM J. Sci. Stat. Computing* **2** (1981), 5–16.
30. L. Ying and D. Zorin, A simple manifold-based construction of surfaces of arbitrary smoothness, *ACM Transactions on Graphics* **23** (2004), 271–275.