Fitting Scattered Data on Sphere-Like Surfaces using Spherical Splines

by

Peter Alfeld [1], Marian Neamtu [2], and Larry L. Schumaker [3]

Abstract. Spaces of polynomial splines defined on planar triangulations are very useful tools for fitting scattered data in the plane. Recently, [4, 5], using homogeneous polynomials, we have developed analogous spline spaces defined on triangulations on the sphere and on sphere-like surfaces. Using these spaces, it is possible to construct analogs of many of the classical interpolation and fitting methods. Here we examine some of the more interesting ones in detail. For interpolation, we discuss macro-element methods and minimal energy splines, and for fitting, we consider discrete least squares and penalized least squares.

1. Introduction

Let $\mathcal{S}$ be the unit sphere or a sphere-like surface (see Sect. 2 below) in $\mathbb{R}^3$. In addition, suppose that we are given a set of scattered points located on $\mathcal{S}$, along with real numbers associated with each of these points. The problem of interest in this paper is to find a function defined on $\mathcal{S}$ which either *interpolates* or *approximates* these data.

This problem arises in a variety of settings. For example, in geodesy, geophysics, and metereology, $\mathcal{S}$ is chosen to be some model of the earth. But it also

[1] Department of Mathematics, University of Utah, Salt Lake City, Utah 84112, alfeld@math.utah.edu. Supported by the National Science Foundation under grant DMS-9203859.

[2] Department of Mathematics, Vanderbilt University, Nashville, TN 37240, neamtu@orpheus.cas.vanderbilt.edu. Partially supported by Vanderbilt University Research Council.

[3] Department of Mathematics, Vanderbilt University, Nashville, TN 37240, s@mars.cas.vanderbilt.edu. Supported by the National Science Foundation under grant DMS-9500643.

comes up in very different situations — e.g., $\mathcal{S}$ might be part of the surface of an aircraft, see e.g. [14, 15, 55] and references therein.

It would take an extensive effort to compile a (even reasonably complete) list of the various methods which have been proposed for fitting scattered data on sphere-like surfaces. Some of these methods include

1) spherical harmonics, various types of singularity functions, and multipole expansions, see e.g. [22, 44]
2) local patches defined on a spherical triangulation of the data points [11, 47, 57, 68, 69]
3) spherical analogs of thin plate splines [32, 34, 79, 80, 81, 82]
4) tensor splines (after mapping the sphere to a rectangle) [19, 20, 37, 77, 78]
5) radial basis functions (spherical multiquadrics) [24, 27, 28, 45, 56, 61] and distance functions [15].

In this paper we discuss a new approach to this problem based on spaces of *spherical splines* which we introduced recently in [4, 5], see Sect. 2.4 for a precise definition. In [4, 5] we have shown that these splines have many properties in common with the classical polynomial splines on planar triangulations. These properties include a Bernstein-Bézier representation which is very useful for computations.

Because of the structure of spherical spline spaces, virtually any spline interpolation or approximation method for the planar scattered data problem has a spherical analog. The purpose of this paper is to develop several of the more useful and interesting of these methods.

The paper is organized as follows. First, in Sect. 2 we present some preliminaries, including sphere-like surfaces, the role of homogeneous functions, homogeneous Bernstein-Bézier polynomials, and spherical splines. In Sect. 3 we discuss various aspects of working with derivatives in the setting of homogeneous trivariate functions. The computation of integrals of spherical splines is addressed in Sect. 4. Sect. 5 is devoted to local interpolation methods, including quintic macro-elements, Clough-Tocher cubic, and Powell-Sabin quadratic elements. Minimal energy interpolation using cubic splines is the subject of Sect. 6, while discrete least squares and penalized least squares fitting are treated in Sect. 7. We report on numerical results in Sect. 8, and conclude the paper with a set of remarks pointing to open problems and new research directions.

2. Preliminaries

2.1. Sphere-like Surfaces

Suppose ρ is a continuous, positive, real-valued function defined on the unit sphere S in $\mathbb{R}^3$ centered at the origin. Then the surface $\mathcal{S}$ in $\mathbb{R}^3$ of the form

$$\mathcal{S} := \{\sigma(v) := \rho(v)v \ : \ v \in S\}$$

is called a *sphere-like surface*. For some of the methods to be discussed below, we need $\mathcal{S}$ to be smooth (for example for various methods involving derivatives). This can be achieved by requiring ρ to be sufficiently smooth. When $\rho \equiv 1$, $\mathcal{S}$ becomes the sphere S.

Since every point on $\mathcal{S}$ is uniquely associated with its radial projection on the sphere S, in solving our basic data fitting problem, in principle it suffices to consider only the sphere. However, when derivatives are involved, there are some subtle differences between working on the sphere and on a general sphere-like surface (see Remark 1). Thus, in the remainder of this paper we deal with general sphere-like surfaces $\mathcal{S}$.

As shown in our earlier papers [4,5], the key to working with functions on sphere-like surfaces is to consider them to be homogeneous trivariate functions. We explore this connection in the next section.

2.2. Homogeneous Functions

A trivariate function F is said to be *positively homogeneous* of degree $t \in \mathbb{R}$ provided that for every real number $a > 0$,

$$F(av) = a^t F(v), \qquad v \in \mathbb{R}^3 \setminus \{0\}\,.$$

In the sequel we shall drop the adjective "positively", and refer to such functions simply as *homogeneous*. There is a close relationship between functions defined on $\mathcal{S}$ and homogeneous functions.

Lemma 2.1. *Suppose f is a function defined on $\mathcal{S}$, and let $t \in \mathbb{R}$. Then*

$$F_t(v) := \frac{\|v\|^t f(\sigma(v/\|v\|))}{[\rho(v/\|v\|)]^t}$$

is the unique homogeneous extension of f of degree t to all of $\mathbb{R}^3 \setminus \{0\}$, i.e., $F_t|_{\mathcal{S}} = f$, and F_t is homogeneous of degree t.

Proof: The assertion is an immediate consequence of the fact that by definition, $\sigma(v/\|v\|) = v$ and $\rho(v/\|v\|) = \|v\|$ for all $v \in \mathcal{S}$. ■

This lemma shows that a given function f on $\mathcal{S}$ has infinitely many homogeneous extensions F_t, one for each real number t. If we only need values of f on $\mathcal{S}$, then the choice of t is obviously irrelevant. We will see later that this is true even if we require derivative values in directions that are tangent to $\mathcal{S}$. However, as we point out in Sect. 3.3, the choice of t is crucial when we need to work with values of derivatives of order 2 or larger.

2.3. Bernstein-Bézier Polynomials

In order to introduce the space of splines of interest, we first recall some definitions from [4]. Given a set of linearly independent vectors $v_1, v_2, v_3 \in \mathbb{R}^3$, any vector $v \in \mathbb{R}^3$ can be expressed uniquely as

$$v = b_1 v_1 + b_2 v_2 + b_3 v_3.$$

The numbers b_1, b_2, b_3, which are linear homogeneous functions of v, are called the *trihedral coordinates of v with respect to v_1, v_2, v_3*. The associated *trihedron* with vertices v_1,v_2,v_3 is the set

$$T := \{v \in \mathbb{R}^3 \ : \ b(v) \geq 0\}, \tag{2.1}$$

where $b(v) := (b_1, b_2, b_3)^T$. Given a nonnegative integer d, the functions

$$B^d_{ijk}(v) := \frac{d!}{i!j!k!} b_1^i b_2^j b_3^k, \qquad i+j+k = d,$$

are called the *homogeneous Bernstein basis polynomials of degree d* defined on T. Clearly, each B_{ijk} is a homogeneous function of degree d. It was shown in [4] that the B_{ijk} are linearly independent and span the $\binom{d+2}{2}$-dimensional space $\mathcal{H}_d$ of homogeneous polynomials of degree d.

A function of the form

$$p(v) = \sum_{i+j+k=d} c_{ijk} B^d_{ijk}(v), \tag{2.2}$$

with $c_{ijk} \in \mathbb{R}$, is called a *homogeneous Bernstein-Bézier (HBB)-polynomial of degree d*.

If T is a trihedron as in (2.1), then the set $T \cap \mathcal{S}$ is a surface triangle. We refer to it simply as a *triangle* on $\mathcal{S}$. Throughout the remainder of the paper we use the symbol T to stand for either a trihedron or a surface triangle depending on the context. If $\mathcal{S}$ is the sphere, then T is a classical spherical triangle. The restriction of a surface triangle T to the plane passing through the origin and through two of the vertices of T will be called an *edge* of T. If T is a spherical triangle, the edges are arcs of great circles.

When $\mathcal{S}$ is the sphere S, the trihedral coordinates b_1, b_2, b_3 of a point $v \in S$ relative to the vertices v_1, v_2, v_3 of a triangle T lying on S are just the spherical barycentric coordinates of v relative to T, see [4]. When T is a triangle on a general sphere-like surface $\mathcal{S}$, we will continue to refer to the b_i as *barycentric coordinates*.

As shown in [4], trihedral coordinates (and thus barycentric coordinates relative to triangles on general sphere-like surfaces) have almost all of the properties of the usual planar barycentric coordinates, except that they do not add up to 1.

In [4], we defined the restriction of an HBB-polynomial to the sphere S to be a *spherical Bernstein-Bézier (SBB-) polynomial*. Here we continue to use this terminology even when $\mathcal{S}$ is a general sphere-like surface. In the sequel we will write $\mathcal{P}_d$ for $\mathcal{H}_d$ restricted to $\mathcal{S}$. For a general $\mathcal{S}$, the spaces $\mathcal{P}_d$ are not nested, *i.e.*, $\mathcal{P}_d \not\subset \mathcal{P}_{d+1}$. In fact, $\mathcal{P}_d \cap \mathcal{P}_{d+1} = \{0\}$. Moreover, unless $\mathcal{S}$ is a special surface such as S, the space $\mathcal{P}_d$ does not contain constant functions for $d > 0$.

The restriction of an SBB-polynomial to an edge of a triangle T on a sphere-like surface $\mathcal{S}$ is a univariate function which we call a *circular Bernstein-Bézier (CBB-) polynomial*. CBB-polynomials were treated in detail in [3].

2.4. Spherical Splines

We say that a set of triangles $\Delta := \{T_i\}_1^N$ lying on a sphere-like surface $\mathcal{S}$ is a *triangulation* of $\mathcal{S}$ provided that $\mathcal{S} = \cup T_i$, and any two triangles intersect only at a common vertex or along an edge. As in the planar case, in general there are many different triangulations associated with a given set of vertices $\{v_i\}_{i=1}^V$, see e.g. [74].

A triangulation which covers all of $\mathcal{S}$ has been called a *total* triangulation [5]. It is well known that for a total triangulation, $N = 2V - 4$ and $E = 3V - 6$, where E is the number of edges of Δ.

Suppose r and d are nonnegative integers, and suppose $\mathcal{S}$ is sufficiently smooth. Then we call

$$\mathcal{S}_d^r(\Delta) := \{s \in C^r(\mathcal{S}) \ : \ s|_{T_i} \in \mathcal{P}_d, \ i = 1, \ldots, N\}$$

the *space of spherical splines of smoothness r and degree d*. It is the direct analog of the space of polynomial splines defined on a planar triangulation.

Spherical splines were introduced and studied in [4, 5], where almost all of the fundamental Bernstein-Bézier theory for dealing with piecewise polynomial functions on planar triangulations has been carried over to the spaces $\mathcal{S}_d^r(\Delta)$. This makes these spaces especially suitable for solving numerical problems associated with functions defined on $\mathcal{S}$, and in particular, for solving the basic interpolation and data fitting problems of this paper.

3. Derivatives

Although our basic interpolation problem involves matching function values to prescribed numbers, we will also discuss several methods which require the matching of derivative information. In this section we show how to compute directional derivatives of SBB- and HBB-polynomials, and of general functions on $\mathcal{S}$.

First we have to agree on what we mean by the derivative of a function f defined on a sphere-like surface $\mathcal{S}$. Suppose g is a given vector. Then we define the directional derivative $D_g f$ of f at a point $v \in \mathcal{S}$ by

$$D_g f(v) := D_g F(v) = g^T \nabla F(v), \tag{3.1}$$

where F is some homogeneous extension of f, and ∇F is the gradient of the trivariate function F. While a polynomial of degree d has a natural homogeneous extension to $\mathbb{R}^3$, as we saw in Lemma 2.1, a general function f on $\mathcal{S}$ has infinitely many different extensions. The value of its derivative may depend on which extension we take. We return to this point in Sect. 3.3, see also Remark 3.

3.1. Derivatives of HBB-polynomials

In this section we give explicit formulae for directional derivatives of HBB-polynomials defined on a trihedron T. We begin by giving formulae for the directional derivatives of the trihedral coordinate functions associated with T.

Lemma 3.1. *Let g be a given vector in $\mathbb{R}^3$. Then*

$$D_g b_i = b_i(g), \qquad i = 1, 2, 3.$$

Proof: We establish the result for $i = 1$. Let v_1, v_2, v_3 be the vertices of T, and let $v \in \mathbb{R}^3$. Then by Cramer's rule, $b_1 = D(v, v_2, v_3)/D(v_1, v_2, v_3)$, where D denotes the determinant of the associated 3×3 matrix. Now a simple calculation shows that $g^T \nabla b_1 = D(g, v_2, v_3)/D(v_1, v_2, v_3) = b_1(g)$, which is the desired result. ■

Proposition 3.2. *Suppose p is an HBB-polynomial. Then*

$$D_g p(v) = b^T(g) \nabla_b p,$$

where

$$\nabla_b := \left(\frac{\partial}{\partial b_1}, \frac{\partial}{\partial b_2}, \frac{\partial}{\partial b_3} \right)^T. \tag{3.2}$$

Proof: This follows immediately from the chain rule and Lemma 3.1. ■

We now turn to the problem of computing higher order derivatives of HBB-polynomials written in the form (2.2). Let $c^0_{ijk} := c_{ijk}$ be the Bézier coefficients of the polynomial p of degree d, and let $g_1, \ldots, g_m$, $1 \le m \le d$, be a set of direction vectors. For each $1 \le \ell \le m$, let $c^\ell_{ijk}, i + j + k = d - \ell$, be the intermediate values obtained in carrying out the de Casteljau algorithm [4] using $b(g_\ell)$. That is, c^ℓ_{ijk} is obtained from the recursion

$$c^\ell_{ijk} = b_1(g_\ell) c^{\ell-1}_{i+1,j,k} + b_2(g_\ell) c^{\ell-1}_{i,j+1,k} + b_3(g_\ell) c^{\ell-1}_{i,j,k+1}, \quad \ell = 1, \ldots, m.$$

It follows from this recursion that the c^ℓ_{ijk} depend on the vectors $g_1, \ldots, g_\ell$, but not on their ordering.

Theorem 3.3. *For any* $0 \le m \le d$,

$$D_{g_1,\dots,g_m} p(v) := D_{g_1} \cdots D_{g_m} p(v) = \frac{d!}{(d-m)!} \sum_{i+j+k=d-m} c^m_{ijk} B^{d-m}_{ijk}(v). \tag{3.3}$$

Proof: By Lemma 3.1, for $i+j+k=d$,

$$\begin{aligned} D_{g_1} B^d_{ijk}(v) &= \frac{d!}{i!j!k!}[i b_1^{i-1} b_2^j b_3^k D_{g_1} b_1 + j b_1^i b_2^{j-1} b_3^k D_{g_1} b_2 + k b_1^i b_2^j b_3^{k-1} D_{g_1} b_3] \\ &= d[B^{d-1}_{i-1,j,k}(v) b_1(g_1) + B^{d-1}_{i,j-1,k}(v) b_2(g_1) + B^{d-1}_{i,j,k-1}(v) b_3(g_1)]. \end{aligned}$$

Substituting this in

$$D_{g_1} p(v) = \sum_{i+j+k=d} c_{ijk} D_{g_1} B^d_{ijk}(v)$$

and rearranging terms yields (3.3) for $m=1$. The general result follows by induction. ■

3.2. Derivatives of SBB-polynomials at Vertices and Along Edges

It is clear from the properties of trihedral coordinates that the values of an SBB-polynomial p at the vertices of its domain triangle are given by $p(v_1) = c_{d00}$, $p(v_2) = c_{0d0}$, and $p(v_3) = c_{00d}$. The derivatives of p at the vertices of T also have a simple form. For example, at $v = v_1$ we have

Corollary 3.4. *For all* $0 \le m \le d$,

$$D_{g_1,\dots,g_m} p(v_1) = \frac{d!}{(d-m)!} c^m_{d-m,0,0}. \tag{3.4}$$

If g is a vector in the plane spanned by the vertices v_1 and v_2, then $b_3(g) = 0$, and (3.4) only involves the coefficients $c_{d,0,0}, \dots, c_{d-m,m,0}$. For later use, we write out the formulae for the first and second derivatives at v_1:

$$D_g p(v_1) = d[b_1(g) c_{d,0,0} + b_2(g) c_{d-1,1,0}], \tag{3.5}$$

while

$$D^2_g p(v_1) := D_{g,g} p(v_1) = d(d-1)[b_1^2(g) c_{d,0,0} + 2 b_1(g) b_2(g) c_{d-1,1,0} + b_2^2(g) c_{d-2,2,0}]. \tag{3.6}$$

For the second order mixed derivatives, let g be as above and let h be a vector in the plane spanned by v_1 and v_3, so that $b_2(h) = 0$. In this case the formula (3.4) simplifies to

$$\begin{aligned} D_{g,h} p(v_1) = d(d-1)[&b_1(g) b_1(h) c_{d,0,0} + b_2(g) b_1(h) c_{d-1,1,0} \\ &+ b_1(g) b_3(h) c_{d-1,0,1} + b_2(g) b_3(h) c_{d-2,1,1}]. \end{aligned} \tag{3.7}$$

We also make a few remarks about *cross derivatives*. Consider the derivative D_h in the direction h, which does not lie in the plane e spanned by v_1 and v_2. Along e we have $b_3 \equiv 0$, and so by Theorem 3.3, for each $0 \le m \le d$, the m-fold cross-boundary derivative $D_h^m p$ reduces to an HBB-polynomial of degree $d-m$ on e. For example, if p is cubic ($d = 3$) and $m = 1$, then $D_h p(v)$ is the quadratic polynomial

$$D_h p(v) = 3[c^1_{200} b_1(v)^2 + 2c^1_{110} b_1(v) b_2(v) + c^1_{020} b_2(v)^2], \quad v \in e. \tag{3.8}$$

3.3. Derivatives and Degree of Homogeneity

It is clear from the definition (3.1) that in general the derivative of a function f defined on $\mathcal{S}$ depends on how f is homogeneously extended. The following result identifies an important case where it does not matter which extension we take. We will assume that $\mathcal{S}$ is smooth in the sense that it possesses a tangent plane at every point on $\mathcal{S}$.

Lemma 3.5. *Suppose f is a function on $\mathcal{S}$ and g is a tangent vector to $\mathcal{S}$ at a point v. Then the value of $D_g f(v)$ can be computed from (3.1) using any homogeneous extension of f.*

Proof: Let F be a homogeneous extension of f, and let $\mathcal{C}$ be a C^1 smooth curve on $\mathcal{S}$ passing through the point v, parameterized by a parameter θ such that $\mathcal{C}(\theta) = v$ and $\mathcal{C}'(\theta) = g$, for $\theta = 0$. By the chain rule we obtain

$$\frac{df(\mathcal{C}(\theta))}{d\theta}\Big|_{\theta=0} = \frac{dF(\mathcal{C}(\theta))}{d\theta}\Big|_{\theta=0} = g^T \nabla F(v) = D_g F(v). \tag{3.9}$$

This shows that $D_g F(v)$ does not depend on the degree of homogeneity of F since the left-hand side of (3.9) clearly depends only on $f = F|_{\mathcal{S}}$. ■

The following example shows that the situation is different for a derivative with respect to a vector g which is *not tangent* to $\mathcal{S}$.

Example 3.6. *Consider the two functions $F_0(v) = 1$ and $F_2(v) = \|v\|^2$ which are both homogeneous extensions of the same function $f = 1$ defined on $\mathcal{S} = S$, and let $g = (1,1,1)$. Then it is easy to check that $D_g F_0(1,0,0) = 0$ while $D_g F_2(1,0,0) = 2$.*

Lemma 3.5 also fails for higher derivatives, even if they are derivatives with respect to a single direction vector g which is tangent to $\mathcal{S}$.

Example 3.7. *Consider the two functions in Example 3.6, and let $g = (0,0,1)$. Then using (3.7), it is easy to check that $D_g^2 F_0(1,0,0) = 0$ while $D_g^2 F_2(1,0,0) = 2$.*

3.4. Estimating Derivatives from Scattered Data

The basic problem of interest in this paper is to find an unknown function f defined on $\mathcal{S}$ given only values of f at scattered data sites. However, several of the methods to be discussed below for constructing an interpolant to f require values for certain derivatives of f at the data sites. Sometimes these derivative values are given as part of the problem. In this case we can use them directly (but see the discussion in Sect. 3.3 above).

If the required derivative values are not prescribed, they have to be estimated from the given data. The problem of estimating derivatives numerically is non-trivial, and has been discussed extensively in the numerical analysis literature. In the usual bivariate case, one of the typical methods for estimating a derivative at a point v is to construct a low degree polynomial which fits a subset of the data which are associated with points lying near v, and then compute its derivative at v. A similar method can be used on a sphere-like surface, but the details of how to choose basis functions and how to choose the points to be included in the fit are not completely straightforward. For a more detailed discussion, see [51].

As we saw in Sect. 3.3, the values of higher order derivatives of a function f defined on $\mathcal{S}$ depend on how we extend f to $\mathbb{R}^3$. This means, for example, that if we take the derivative values from an HBB-polynomial p of degree m but we are using splines of degree $d \neq m$, we will not be using the same derivative values as we would if we converted p to a homogeneous function of degree d. Using the wrong derivative values can lead to a drastic loss of accuracy (see the example in Sect. 8.3 using the quintic macro-element method).

4. Integration of Spherical Polynomials

In many applications, *e.g.* in the finite element method or in minimal energy interpolation, it is necessary to compute integrals of piecewise polynomial functions. Evaluating integrals of spherical polynomials is considerably more difficult than in the planar case. Recall that for planar triangles, the integral of a Bernstein basis polynomial of degree d is equal to the area of the corresponding triangle divided by $d+1$. Thus, the value of the integral does not depend on the particular basis polynomial or on the precise shape of the triangle. Unfortunately, this attractive property does not carry over to an arbitrary sphere-like surface. In general, for two different surface triangles, the values of the integrals are different. This is true even if $\mathcal{S}$ is the sphere, unless the two triangles are similar. Moreover, the integrals of the Bernstein basis polynomials of degree d associated with a single triangle are also different in general.

There does not seem to be a simple explicit formula for integrals of SBB-polynomials. In fact, this difficulty arises already in the case of CBB-polynomials. As shown in [3], on the circle these polynomials are essentially trigonometric polynomials. Although recurrence relations exist for computing integrals of products of

trigonometric functions over an arbitrary interval (see *e.g.* [41], p. 130), a convenient closed-form formula does not seem to be available.

Throughout the remainder of this section we restrict our attention to the special case where $\mathcal{S} = S$ is the sphere. To compute integrals in this case, we propose a mapping of a surface triangle T to a planar triangle, namely the planar triangle whose vertices are the same as the vertices of T. This will enable us to use a standard technique of numerical integration for planar triangles.

Let A denote the matrix whose columns are the vertices v_1, v_2, v_3 of T, and let b be the vector of barycentric coordinates with respect to T. Moreover, let u be the vector of planar barycentric coordinates with respect to T such that $u \in U := \{u = (u_1, u_2, u_3) \ : \ u_1 + u_2 + u_3 = 1, u_1, u_2, u_3 \geq 0\}$, so that every point $v \in T$ can be expressed uniquely as $v = Ab = \sigma(Au/\|Au\|)$.

Now, the surface integral over a triangle T of a function f defined on T can be written as

$$\int_T f(v)ds = \int_U f\left(\sigma\left(Au/\|Au\|\right)\right)\|N(u)\|du := \int_0^1 \int_0^{1-u_1} f\left(\sigma\left(Au/\|Au\|\right)\right) du_2 du_1,$$

where ds is a surface area element on $\mathcal{S}$, and where $N(u)$ is the normal vector of $\mathcal{S}$ at the point $v = Ab = \sigma(Au/\|Au\|)$, in the parametrization of $\mathcal{S}$ induced by u, *i.e.*,

$$N(u) := \left(\frac{\partial}{\partial u_1} - \frac{\partial}{\partial u_3}\right)\sigma\left(\frac{Au}{\|Au\|}\right) \times \left(\frac{\partial}{\partial u_2} - \frac{\partial}{\partial u_3}\right)\sigma\left(\frac{Au}{\|Au\|}\right).$$

Since we have restricted our attention to the sphere, the length of $N(u)$ has a simple form, namely

$$\|N(u)\| = \frac{|\det\ A|}{\|Au\|^3},$$

which can be obtained by a straightforward calculation. Hence, for a spherical polynomial of degree d, viewed as a homogeneous polynomial of the same degree, we obtain

Proposition 4.1. *Let $p \in \mathcal{P}_d$. Then*

$$\int_T p ds = |det\ A| \int_U \frac{p(u)}{\|Au\|^{d+3}} du. \tag{4.1}$$

Having expressed surface integrals in terms of ordinary planar integrals, it is now possible to apply a numerical integration method designed for planar triangles in order to integrate arbitrary spherical polynomials. In our experiments we used a bivariate version of the trapezoidal rule. It is beyond the scope of this paper to present a more sophisticated numerical integration method. Instead, we refer the reader to [52] for a survey of various techniques.

5. Local Interpolation

A common approach to solving the scattered data interpolation problem in the planar case is the following:

1) construct a triangulation Δ with vertices at the given data points,
2) choose r and d,
3) for each triangle T in Δ, use the data at the vertices (along with additional derivative information at other points in T) to define a polynomial of degree d on T which interpolates the data in such a way that the resulting polynomial pieces join together to form a spline s in $\mathcal{S}_d^r(\Delta)$.

One advantage of this approach is that an interpolant s is constructed one triangle at a time, and the resulting method is completely *local* in the sense that the restriction of s to a triangle T depends only on the data in that triangle. Methods of this type are called *macro-element methods*. They have been successfully applied in many bivariate data fitting problems and in the conforming finite element method [46, 49, 72, 84].

This idea can be carried over to a sphere-like surface, and indeed, every macro-element which is known in the planar case has a spherical analog. We confine our discussion to just three examples (see Remark 13 for other possible choices):

1) quintic C^1 macro-elements
2) cubic C^1 elements on the Clough-Tocher split
3) quadratic C^1 elements on the 6-triangle Powell-Sabin split.

To define these macro-elements we need to use derivative information. If we are given the needed derivatives, we can use them. If not, we must estimate them (see Sect. 3.4).

5.1. A Quintic C^1 Element

In this section a C^1 smooth interpolating spline will be constructed which associates with each triangle T of the given triangulation a single quintic spherical polynomial. Let v_1, v_2, and v_3 be the vertices of T, and for convenience, let $v_4 = v_1$ and $v_5 = v_2$. For each $i = 1, 2, 3$, let $\hat{v}_i$ denote the center point of the edge from v_i to v_{i+1}. This point can be computed by projecting v_i and v_{i+1} radially back to the sphere, finding the center of the corresponding circular arc, and then projecting back up to $\mathcal{S}$.

To define some useful derivatives associated with T, let g_{ij} be a vector contained in the plane passing through v_i, v_j, and the origin, not parallel with v_i, $i, j = 1, 2, 3, i \neq j$. In addition, let h_i be a vector tangent to $\mathcal{S}$ at $\hat{v}_i$ which is not contained in the plane passing through v_i, v_{i+1} and the origin. We denote the derivative operators corresponding to g_{ij} and h_i by D_{ij} and D_i, respectively.

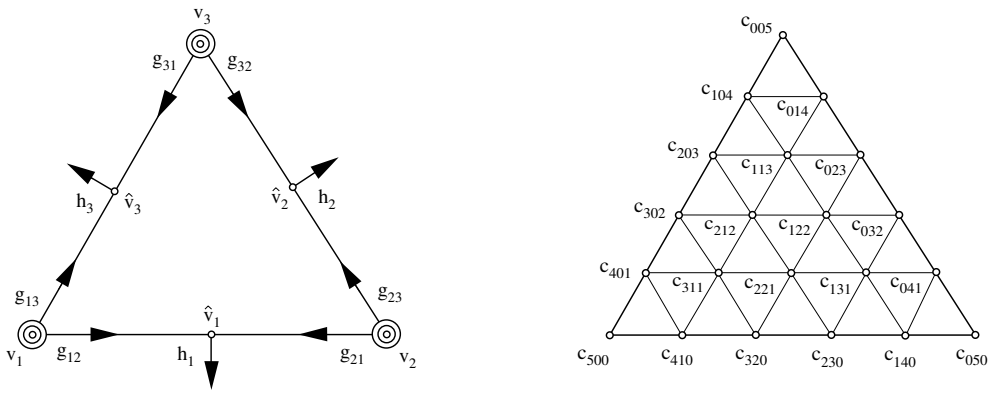

Fig. 1. The quintic macro-element.

Lemma 5.1. *The following 21 data uniquely determine an SBB-polynomial p of degree 5 on T:*

1) $p(v_i)$,
2) $D_{i,i+1}p(v_i)$, $D_{i,i+2}p(v_i)$,
3) $D^2_{i,i+1}p(v_i)$, $D_{i,i+1}D_{i,i+2}p(v_i)$, $D^2_{i,i+2}p(v_i)$,
4) $D_i p(\hat{v}_i)$,

for $i = 1, 2, 3$.

Proof: Suppose p is written in SBB-form, and that its Bézier coefficients are numbered as in Fig. 1. (For simplicity, in our illustration of Bernstein-Bézier nets for spherical macro-elements in Fig. 1 and also in the figures below, we have flattened out the spherical triangles and depicted the domain points as if they were equally spaced (they are not).) We now show that the 6 coefficients closest to vertex v_1 are completely determined by the data in items 1)–3) for $i = 1$. Indeed, $c_{500} = p(v_1)$, and by (3.5),

$$c_{410} = [D_{12}p(v_1)/5 - \alpha_{12}c_{500}]/\beta_{12},$$

where $(\alpha_{12}, \beta_{12}, 0) = b(g_{12})$ are the barycentric coordinates of g_{12} relative to T. Note that β_{12} is nonzero by the assumption that g_{12} is not parallel to v_1. Similarly,

$$c_{401} = [D_{13}p(v_1)/5 - \alpha_{13}c_{500}]/\gamma_{13},$$

where $(\alpha_{13}, 0, \gamma_{13}) = b(g_{13})$, and where $\gamma_{13} \neq 0$. For the second derivatives at v_1 we have, using (3.6) and (3.7),

$$\begin{aligned} D^2_{12}p(v_1)/20 &= \alpha^2_{12}c_{500} + 2\alpha_{12}\beta_{12}c_{410} + \beta^2_{12}c_{320}, \\ D_{12}D_{13}p(v_1)/20 &= \alpha_{12}\alpha_{13}c_{500} + \beta_{12}\alpha_{13}c_{410} + \alpha_{12}\gamma_{13}c_{401} + \beta_{12}\gamma_{13}c_{311}, \\ D^2_{13}p(v_1)/20 &= \alpha^2_{13}c_{500} + 2\alpha_{13}\gamma_{13}c_{401} + \gamma^2_{13}c_{302}. \end{aligned}$$

Then c_{320}, c_{311}, and c_{302} can be immediately computed. The situation at v_2 and v_3 is analogous.

Once the above coefficients have been computed, the coefficient c_{221} is uniquely determined by $D_1 p(\hat{v}_1)$. Namely, by (3.3),

$$D_1 p(\hat{v}_1) = 5 \sum_{i+j+k=4} c^1_{ijk} B^4_{ijk}(\hat{v}_1), \tag{5.1}$$

where

$$c^1_{ijk} = \alpha_1 c_{i+1,j,k} + \beta_1 c_{i,j+1,k} + \gamma_1 c_{i,j,k+1}, \quad i+j+k = 4,$$

and $(\alpha_1, \beta_1, \gamma_1) = b(h_1)$. At this point the only unknown quantity on the right-hand side of (5.1) is the coefficient c_{221}. Since it is multiplied by $\gamma_1 B^4_{220}(\hat{v}_1)$, which is nonzero by our assumption on h_1, we can solve (5.1) for it.

The coefficients c_{122} and c_{212} are determined in the same way from $D_1 p(\hat{v}_2)$ and $D_2 p(\hat{v}_3)$, respectively, and the proof is complete. ■

Lemma 5.1 shows how to construct a quintic polynomial on a surface triangle using only values and derivatives at the vertices and at the centers of the edges. Using this macro-element we can now construct an interpolating quintic spline.

Theorem 5.2. *Let Δ be a triangulation corresponding to a set of vertices $\{v_i\}_{i=1}^V$. Suppose we are given function, first, and second derivative information as in 1) – 3) of Lemma 5.1 at each of the vertices. In addition, suppose we are given a value for a cross-boundary derivative at the center of each edge of Δ. Then there exists a unique spline $s \in \mathcal{S}^1_5(\Delta)$ which interpolates these data.*

Proof: By Lemma 5.1, the given data uniquely define a quintic SBB-polynomial on each triangle of Δ. It remains to show that these polynomials join together with C^1 continuity across the edges of Δ to form a spline in $\mathcal{S}^1_5(\Delta)$. The argument is virtually the same as in the planar case. Suppose p and $\tilde{p}$ are two such polynomials defined on triangles T and $\tilde{T}$ which share an edge e joining the vertices v_1 and v_2. Then along e they reduce to quintic CBB-polynomials satisfying

$$D^k_{12} p(v_1) = D^k_{12} \tilde{p}(v_1), \quad D^k_{21} p(v_2) = D^k_{21} \tilde{p}(v_2), \qquad k = 0, 1, 2.$$

These six conditions imply that $p = \tilde{p}$ on e. Now, in view of the discussion in Sect. 3.2, the cross-boundary derivatives $q := D_1 p$ and $\tilde{q} := D_1 \tilde{p}$ reduce to quartic CBB-polynomials on e satisfying

$$D^k_{12} q(v_1) = D^k_{12} \tilde{q}(v_1), \quad D^k_{21} q(v_2) = D^k_{21} \tilde{q}(v_2), \qquad k = 0, 1.$$

Since we also have $q(\hat{v}) = \tilde{q}(\hat{v})$ at the center point $\hat{v}$ of e, we conclude that $q = \tilde{q}$ on e. This establishes the C^1 continuity between p and $\tilde{p}$ across e. The same argument works for every edge, and the proof is complete. ■

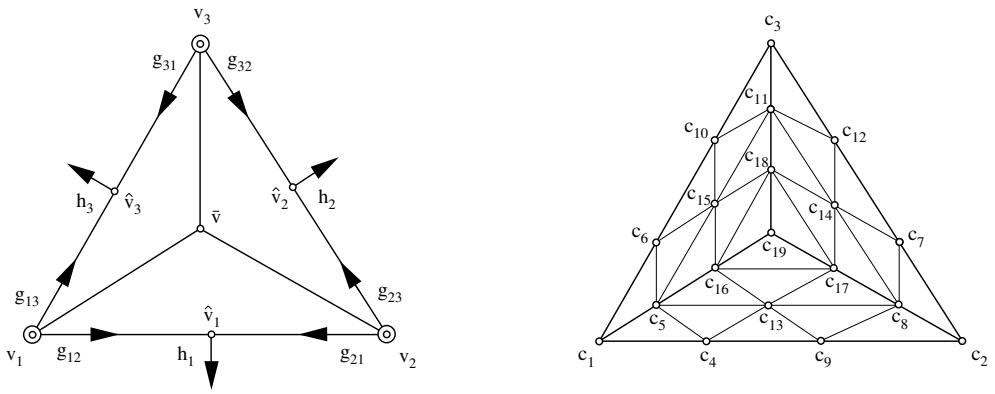

Fig. 2. The Clough-Tocher macro-element.

By construction, this quintic macro-element actually exhibits C^2 continuity at each of the vertices. In this sense it is a *superspline*, see [5].

5.2. A Clough-Tocher Element

The macro-element in the previous subsection is piecewise quintic. In order to work with lower degree polynomials while maintaining C^1 continuity, we have to subdivide each triangle in Δ into subtriangles. It is well known in the planar case that cubics can be used if each triangle T is split into three subtriangles. In this section we discuss this method in the setting of a sphere-like surface.

Given a triangle T with vertices v_1, v_2, v_3, let

$$\bar{v} := \sigma \left(\frac{v_1 + v_2 + v_3}{\|v_1 + v_2 + v_3\|} \right)$$

be its *center*. If we connect $\bar{v}$ to each of the vertices of T, we get three subtriangles. This is called the *Clough-Tocher split* of the triangle, see Fig. 2.

We now show how to construct a cubic C^1 spline on the split triangle T using function and derivative values at the vertices along with cross derivatives at the centers of the edges of T. Let D_{ij} and D_i be the directional derivatives introduced at the beginning of Sect. 5.1.

Lemma 5.3. *The following 12 pieces of data uniquely determine a C^1 cubic spline s on the Clough-Tocher split of T:*

1) $s(v_i)$,

2) $D_{i,i+1}s(v_i)$, $D_{i,i+2}s(v_i)$,

3) $D_i s(\hat{v}_i)$,

for $i = 1, 2, 3$.

Proof: Suppose we number the 19 Bézier coefficients of s as in Fig. 2. Then $c_1 = s(v_1)$. Moreover,

$$c_4 = [D_{12}s(v_1)/3 - \alpha_{12}s(v_1)]/\beta_{12},$$

where $(\alpha_{12}, \beta_{12}, 0)$ are the barycentric coordinates of g_{12} relative to the triangle $T = \langle v_1, v_2, v_3 \rangle$. Similarly, c_6 is determined by $D_{13}s(v_1)$. The coefficient c_5 can now be determined from the C^1 continuity condition across the edge $\langle v_1, \bar{v} \rangle$. Similarly, the coefficients c_2, c_7, c_8, c_9 are determined by the data at v_2, while c_3, c_{10}, c_{11}, c_{12} are determined by the data at v_3.

We now show that c_{13} is determined by the value of the cross-boundary derivative $D_1 p(\hat{v}_1)$. Suppose $(\alpha_1, \beta_1, \gamma_1)$ are the barycentric coordinates relative to the triangle $\langle \bar{v}, v_1, v_2 \rangle$ of the vector h_1, and let $(0, a_2, a_3)$ be the barycentric coordinates of the center point $\bar{v}_1$ relative to the same triangle. Then by (3.8) the cross-boundary derivative at $\hat{v}_1$ is given by

$$\begin{aligned} D_1 p(\hat{v}_1)/3 = {} & a_2^2(\alpha_1 c_5 + \beta_1 c_1 + \gamma_1 c_4) \\ & + 2a_2 a_3(\alpha_1 c_{13} + \beta_1 c_4 + \gamma_1 c_9) + a_3^2(\alpha_1 c_8 + \beta_1 c_9 + \gamma_1 c_2). \end{aligned}$$

Since $\hat{v}_1$ is at the center of the edge $\langle v_1, v_2 \rangle$, a_2 and a_3 are not zero. Moreover, γ_1 cannot be zero for this direction, and we can now solve for c_{13} in terms of the other (known) quantities. The coefficients c_{14} and c_{15} are similarly computed from the cross-boundary derivatives at $\hat{v}_2$ and $\hat{v}_3$.

We now show that the remaining coefficients of s are uniquely determined by C^1 continuity. Let $(\bar{\alpha}, \bar{\beta}, \bar{\gamma})$ be the barycentric coordinates of the center point $\bar{v}$ relative to T. Then by the C^1 conditions across the edges inside of T, we have

$$\begin{aligned} c_{16} &= (\bar{\alpha}c_5 + \bar{\beta}c_{13} + \bar{\gamma}c_{15})/3, \\ c_{17} &= (\bar{\alpha}c_{13} + \bar{\beta}c_8 + \bar{\gamma}c_{14})/3, \\ c_{18} &= (\bar{\alpha}c_{15} + \bar{\beta}c_{14} + \bar{\gamma}c_{11})/3, \\ c_{19} &= (\bar{\alpha}c_{16} + \bar{\beta}c_{17} + \bar{\gamma}c_{18})/3. \end{aligned}$$

This completes the proof that s is uniquely determined by the given data. ■

As in the planar case, it is easy to show that the macro-element constructed in Lemma 5.3 is actually C^2 at the center of the triangle T. We now show how to use the macro-element described in Lemma 5.3 to solve the interpolation problem using C^1 cubic splines.

Theorem 5.4. *Let Δ be a triangulation corresponding to a set of vertices $\{v_i\}_{i=1}^V$, and let Δ_{CT} be the triangulation obtained by splitting each triangle in Δ about its center to create three subtriangles. Suppose we are given function and first derivative values at each of the points $\{v_i\}_{i=1}^V$, along with a value for a cross-boundary derivative at the center of each edge of Δ. Then there exists a unique spherical spline $s \in \mathcal{S}_3^1(\Delta_{CT})$ which interpolates these data.*

Proof: The prescribed data uniquely determine a C^1 cubic spline on each of the triangles of Δ. We now show that these splines join together smoothly across the edges of Δ to form a spline $s \in \mathcal{S}_3^1(\Delta_{CT})$. Let $e = \langle v_1, v_2 \rangle$ be one of the edges of Δ, and let p and $\tilde{p}$ be the cubic CBB-polynomials obtained by restricting the cubic polynomial pieces of s on either side of e to e. By construction,

$$D_{12}^k p(v_1) = D_{12}^k \tilde{p}(v_1), \quad D_{21}^k p(v_2) = D_{21}^k \tilde{p}(v_2), \qquad k = 0, 1.$$

These four pieces of data uniquely determine a cubic CBB-polynomial, and it follows that $p = \tilde{p}$.

Now consider the cross-boundary derivatives $D_1 p$ and $D_1 \tilde{p}$ of the two adjoining cubics. By (3.8), these two functions are quadratic CBB-polynomials which have common values at the three points v_1, $\hat{v}_1$, and v_2. We conclude that $D_1 p = D_1 \tilde{p}$, and the proof is complete. ■

5.3. A Powell-Sabin Element

The macro-element in the previous subsection is piecewise cubic. If we want to work with even lower degree polynomials while maintaining C^1 continuity, we have to subdivide each domain triangle into more subtriangles. It is well known in the planar case that quadratics can be used if each triangle T is split into six subtriangles as indicated in Fig. 3.

If v_1, v_2, v_3 are the vertices of T, we define its *incenter* as the point on $\mathcal{S}$ which is obtained by radially projecting the incenter of the planar triangle with vertices $v_i/\|v_i\|, i = 1, 2, 3$, onto $\mathcal{S}$. Given a triangulation Δ, we denote the incenter of the j-th triangle by $\bar{v}_j$ for $j = 1, \ldots, N$. As in the planar case, if the incenters of two neighboring triangles sharing an edge e are connected with an arc (*i.e.*, the curve segment connecting these two incenters obtained as the intersection of $\mathcal{S}$ with a plane passing through the two points and the origin), then that arc intersects e at some point $\hat{v}$ in the interior of e.

Starting with the triangulation Δ, we now construct a refined triangulation Δ_{PS} where each of the original triangles of Δ has been split into six subtriangles. For each triangle we simply connect its incenter to its three vertices and the

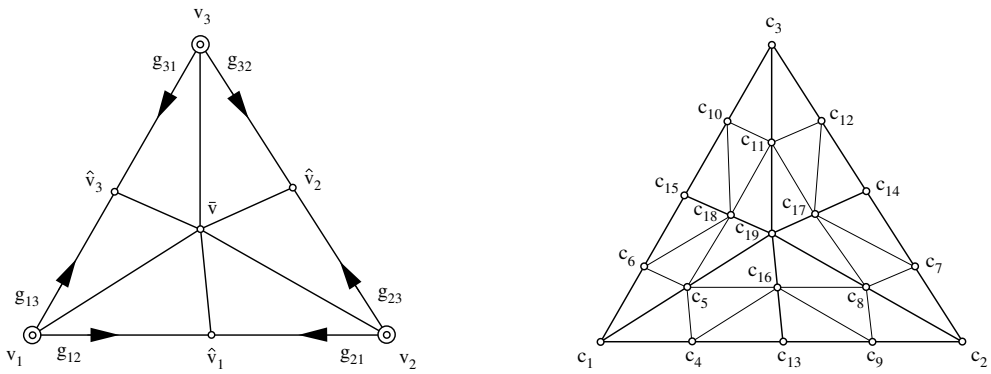

Fig. 3. The Powell-Sabin macro-element.

three points on its edges singled out above. The result is called the *Powell-Sabin refinement of* Δ. Fig. 3 shows the split of one triangle.

We now restrict our attention to one of the original triangles $T \in \Delta$. Suppose its vertices are v_1, v_2, v_3, its incenter is $\bar{v}$, and the intersection points on the edges are $\hat{v}_1$, $\hat{v}_2$, and $\hat{v}_3$. On T a typical element $s \in \mathcal{S}_2^0(\Delta_{PS})$ consists of 6 SBB-polynomials, and the corresponding Bézier net has 19 coefficients which we number as in Fig. 3. Let D_{ij} and D_i be the directional derivatives introduced at the beginning of Sect. 5.1.

Lemma 5.5. *The following 9 pieces of data uniquely determine a C^1 quadratic spline s on the Powell-Sabin split of T:*

1) $s(v_i)$,

2) $D_{i,i+1}s(v_i)$, $D_{i,i+2}s(v_i)$,

for $i = 1, 2, 3$.

Proof: Suppose we number the Bézier coefficients of s as in Fig. 3. Then $c_1 = s(v_1)$. Moreover,

$$c_4 = [D_{12}s(v_1)/2 - \alpha_{12}s(v_1)]/\beta_{12}, \tag{5.2}$$

where $(\alpha_{12}, \beta_{12}, 0)$ are the barycentric coordinates of the vector g_{12} relative to T Similarly, c_6 is determined by $D_{13}s(v_1)$ and $s(v_1)$. The coefficient c_5 can now be computed from the C^1 continuity condition across the edge $\langle v_1, \bar{v}\rangle$. Similarly, the coefficients c_2, c_7, c_8, c_9 are determined by the data at v_2, while c_3, c_{10}, c_{11}, c_{12} are determined by the data at v_3.

We now show that the remaining coefficients are uniquely determined by C^1 continuity. Let $(\alpha_i, \beta_i, \gamma_i)$ be the barycentric coordinates of $\hat{v}_i$ relative to the triangle T, and let $\bar{\alpha}$, $\bar{\beta}$, $\bar{\gamma}$ be the barycentric coordinates of $\bar{v}$ relative to the same

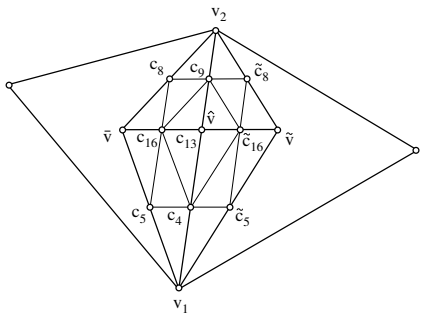

Fig. 4. Adjoining Powell-Sabin macro-elements.

triangle. Then the C^1 continuity conditions across the interior edges of the split triangle hold if and only if

$$
\begin{aligned}
c_{13} &= \alpha_1 c_4 + \beta_1 c_9, \\
c_{16} &= \alpha_1 c_5 + \beta_1 c_8, \\
c_{14} &= \beta_2 c_7 + \gamma_2 c_{12}, \\
c_{17} &= \beta_2 c_8 + \gamma_2 c_{11}, \\
c_{15} &= \alpha_3 c_6 + \gamma_3 c_{10}, \\
c_{18} &= \alpha_3 c_5 + \gamma_3 c_{11}, \\
c_{19} &= \bar{\alpha} c_5 + \bar{\beta} c_8 + \bar{\gamma} c_{11}.
\end{aligned}
$$

This completes the proof of the lemma. ■

We now show how to use the macro-element constructed in Lemma 5.5 to solve the scattered data interpolation problem using C^1 quadratic splines.

Theorem 5.6. *Let Δ be a triangulation corresponding to a set of vertices $\{v_i\}_{i=1}^V$, and let Δ_{PS} be the Powell-Sabin triangulation obtained by splitting each triangle into six subtriangles using the incenters as above. Suppose we are given function and first derivative values at each of the points $\{v_i\}_{i=1}^V$. Then there exists a unique spline $s \in \mathcal{S}_2^1(\Delta_{PS})$ which interpolates these data.*

Proof: It was shown in Lemma 5.5 that the data uniquely determine C^1 quadratic splines on each of the triangles of Δ. It remains to check that these splines join together smoothly to form a spline $s \in \mathcal{S}_2^1(\Delta_{PS})$. To show this, it suffices to consider two such splines s and $\tilde{s}$ which share a common edge e of Δ. Let v_1 and v_2 be the endpoints of e, and let $\hat{v}$ be the intersection point of the edge connecting the incenters $\bar{v}$ and $\tilde{v}$ of the corresponding adjoining triangles (cf. Fig. 4). We have to show that s and $\tilde{s}$ join with C^1 continuity across $e = \langle v_1, v_2 \rangle$.

Let q and $\tilde{q}$ be the quadratic CBB-splines obtained by restricting s and $\tilde{s}$ to e. Both q and $\tilde{q}$ are C^1 quadratic CBB-splines along e with one knot at the point $\hat{v}$. By construction,

$$D_1^k q(v_i) = D_1^k \tilde{q}(v_i), \qquad k = 0, 1, \quad i = 1, 2,$$

where D_1 is a cross-boundary derivative for the edge e. It is easy to see that these four pieces of data uniquely determine such a quadratic CBB-spline, and we conclude that $q = \tilde{q}$ on e, establishing that the pieces join continuously along e. Now consider the restrictions $q_{_D}$ and $\tilde{q}_{_D}$ to e of the derivatives $D_1 s$ and $D_1 \tilde{s}$. These are C^0 linear CBB-splines with one knot at the point $\hat{v}$. By the construction $q_{_D}(v_i) = \tilde{q}_{_D}(v_i)$, $i = 1, 2$. We now show that $q_{_D}(\hat{v}) = \tilde{q}_{_D}(\hat{v})$.

Suppose $\hat{v} = \alpha_1 v_1 + \beta_1 v_2$, and let $(\tilde{\alpha}, 0, \tilde{\gamma})$ be the barycentric coordinates of $\tilde{v}$ relative to the triangle $\langle \bar{v}, v_1, \hat{v} \rangle$. Note that $(\tilde{\alpha}, 0, \tilde{\gamma})$ are also the barycentric coordinates of $\tilde{v}$ relative to the triangle $\langle \bar{v}, v_2, \hat{v} \rangle$. Then by construction,

$$\begin{aligned}
c_{16} &= \alpha_1 c_5 + \beta_1 c_8, \\
\tilde{c}_{16} &= \alpha_1 \tilde{c}_5 + \beta_1 \tilde{c}_8, \\
c_{13} &= \alpha_1 c_4 + \beta_1 c_9, \\
\tilde{c}_5 &= \tilde{\alpha} c_5 + \tilde{\gamma} c_4, \\
\tilde{c}_8 &= \tilde{\alpha} c_8 + \tilde{\gamma} c_9.
\end{aligned}$$

Combining these equations, we see that

$$\begin{aligned}
\tilde{c}_{16} &= \alpha_1(\tilde{\alpha} c_5 + \tilde{\gamma} c_4) + \beta_1(\tilde{\alpha} c_8 + \tilde{\gamma} c_9) \\
&= \tilde{\alpha}(\alpha_1 c_5 + \beta_1 c_8) + \tilde{\gamma}(\alpha_1 c_4 + \beta_1 c_9) \\
&= \tilde{\alpha} c_{16} + \tilde{\gamma} c_{13}.
\end{aligned}$$

This implies that the cross derivatives $q_{_D}$ and $\tilde{q}_{_D}$ also match at $\hat{v}$, and we conclude that $q_{_D} = \tilde{q}_{_D}$ on all of e, which completes the proof. ■

5.4. Dependence on Extensions

In Sect. 3.3 we showed that first order directional derivatives taken with respect to vectors which are tangent to $\mathcal{S}$ do not depend on the degree of homogeneity of the function, while higher order derivatives do. As a consequence we immediately conclude

1) The Powell-Sabin and Clough-Tocher methods do not depend on how we compute derivatives. Thus, if we are interpolating a known function f, its degree

of homogeneity is irrelevant, and if we are estimating the needed first derivatives by SBB-polynomials, it does not matter how we extend these (see the discussion in Sect. 3.3).

2) The quintic macro-element method does depend on how we compute derivatives. Thus, if we compute these from a known function f, the result will depend on how we extend it to $\mathbb{R}^3$ (however, see Remark 7). Moreover, if we estimate derivatives by a low degree SBB-polynomial, the result will depend on how we view it as a trivariate function (*i.e.*, what degree of homogeneity we assign to it in computing the derivative estimates). We illustrate this effect for quintics in Sect. 8.

6. A Global Interpolation Method

In this section we discuss a method for constructing a spherical spline which satisfies

$$s(v_i) = f_i, \qquad i = 1, \ldots, V, \tag{6.1}$$

for prescribed data $\{(v_i, f_i)\}_{i=1}^V$ associated with a sphere-like surface $\mathcal{S}$. The method involves minimizing an appropriate energy functional, and is *global* in the sense that the coefficients of the spline depend on all of the data, and are determined simultaneously (by solving a linear system of equations). We restrict our discussion to C^1 splines, although the approach also works for smoother splines.

6.1. $\mathcal{S}_d^1(\Delta)$ as a Subspace of $\mathcal{S}_d^0(\Delta)$

As pointed out in [5], every spline $s \in \mathcal{S}_d^0(\Delta)$ is uniquely associated with a vector $c = (c_1, \ldots, c_M)$, where

$$M := \dim\, \mathcal{S}_d^0(\Delta) = V + (d-1)E + \binom{d-1}{2} N. \tag{6.2}$$

We can think of c as consisting of an ordered list of the Bernstein-Bézier coefficients of the SBB-polynomial pieces of s, using the convention that when two such pieces join along an edge, then the corresponding Bernstein-Bézier coefficients associated with that edge are identified with each other and included in the list just once.

It was shown in [5] that a spline $s \in \mathcal{S}_d^0(\Delta)$ is C^r continuous ($r \geq 1$) across an edge of the triangulation, if and only if a corresponding set of

$$n_r := d + (d-1) + \cdots + (d-r+1) = r(2d-r+1)/2$$

linear homogeneous conditions on c are satisfied. As in the planar case, these conditions involve coefficients closest to the edge under consideration. Thus a spline $s \in \mathcal{S}_d^0(\Delta)$ belongs to $\mathcal{S}_d^r(\Delta)$ if and only if

$$Gc = 0, \tag{6.3}$$

where G is the an $n_r E \times M$ matrix expressing the smoothness conditions (see [5] for the explicit formulae).

In general, the full set (6.3) of smoothness conditions contain some redundant equations. The problem of determining which – or even just how many – of these equations are redundant is by no means simple, and in fact, is tantamount to computing the dimension of $\mathcal{S}_d^r(\Delta)$. This problem remains unsolved for $d < 3r + 2$, see [5]. For more on redundancy, see Sect. 6.3.

6.2. Minimal Energy Interpolation

Suppose $\{v_i\}_{i=1}^V$ are given data sites on a sphere-like surface $\mathcal{S}$ and let Δ be a triangulation on $\mathcal{S}$ with vertices at the data sites. In this section we show how to compute the coefficients of a spline $s \in \mathcal{S}_d^1(\Delta)$ so that (6.1) is satisfied.

We begin by showing how to calculate those Bernstein-Bézier coefficients of s associated with domain points falling at the vertices of Δ. Suppose we order the vector c so that the i-th coefficient is associated with the vertex v_i, for $i = 1, \ldots, V$. Since the value of an SBB-polynomial at each of the vertices of its domain triangle is equal to the value of the Bernstein-Bézier coefficient associated with that vertex, to make a spline $s \in \mathcal{S}_d^0(\Delta)$ satisfy (6.1), we simply have to set

$$c_i = f_i, \qquad i = 1, \ldots, V. \tag{6.4}$$

The remaining $M - V = (d-1)E + \binom{d-1}{2}N$ coefficients of s are free, and can be used to make s have some additional desirable property.

A typical way to use these extra degrees of freedom is to minimize a functional $\mathcal{E}(c)$ measuring the smoothness of s. The problem becomes particularly simple if we choose $\mathcal{E}(c)$ to be a quadratic functional, i.e.,

$$\mathcal{E}(c) = c^T Q c,$$

with some $M \times M$ symmetric positive definite matrix Q. Thus our problem becomes

$$\text{mininimize } \mathcal{E}(c), \qquad \text{subject to (6.4)}.$$

By introducing a *Lagrange multiplier* vector λ of length V, it is easy to see (cf. [26, p. 236]) that c solves this problem if and only if

$$\begin{pmatrix} Q & I \\ I & 0 \end{pmatrix} \begin{pmatrix} c \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ g \end{pmatrix},$$

where I is the $V \times V$ identity matrix and $g := (f_1, \ldots, f_V)^T$.

Since we have not enforced any smoothness conditions, the minimal energy spline constructed above is only continuous. In many applications we would like

our interpolant to be at least C^1. As observed in Sect. 6.1, a spline in $\mathcal{S}_d^0(\Delta)$ will belong to $\mathcal{S}_d^r(\Delta)$ if and only if the linear homogeneous conditions (6.3) are satisfied. Thus, to construct an interpolating spline in $\mathcal{S}_d^r(\Delta)$, we seek c solving the problem

$$\text{minimize } \mathcal{E}(c), \qquad \text{subject to (6.4) and (6.3).}$$

To find the solution of this problem, we solve the linear system

$$\begin{pmatrix} Q & I^T & G^T \\ I & 0 & 0 \\ G & 0 & 0 \end{pmatrix} \begin{pmatrix} c \\ \lambda \\ \gamma \end{pmatrix} = \begin{pmatrix} 0 \\ g \\ 0 \end{pmatrix}, \tag{6.5}$$

where γ is an $n_r E$ vector of additional Lagrange multipliers.

There are two problems with this approach. First it may happen that there is no spline in $\mathcal{S}_d^0(\Delta)$ satisfying both the interpolation conditions (6.4) and the smoothness conditions (6.3). This is not a problem if d is large, since as shown in [5], if $d \geq 3r+2$, then each coefficient associated with a vertex can be independently set, and so an interpolant in $\mathcal{S}_d^r(\Delta)$ always exists. Thus, we can always interpolate with $\mathcal{S}_d^1(\Delta)$ for $d \geq 5$. Moreover, following the arguments in [6], it can be shown that interpolation is also possible in $\mathcal{S}_4^1(\Delta)$. As in the planar case, we conjecture that interpolation is also possible with cubic C^1 splines. On the other hand, it can easily be seen that $\mathcal{S}_2^1(\Delta)$ is not large enough to solve the interpolation problem in general.

The other problem arises when there are redundancies in the side conditions (6.3). In this case the system (6.5) becomes singular. Of course, what we really have in this case is an *underdetermined* but consistent system, which can still be solved by standard techniques. Ideally redundancies should be removed if possible, however, as doing so also reduces the size of the system.

Although the minimal energy approach discussed in this section leads to rather large systems of equations, we should point out that for the kind of energy functional we intend to use (see Sect. 6.4) the matrix Q is rather sparse. Moreover, for C^1 continuity, each of the smoothness conditions involves only 4 coefficients, and so the rows of G corresponding to smoothness conditions have only 4 entries in them. Finally, the rows corresponding to interpolation conditions have only one entry in them.

6.3. Redundancies for $\mathcal{S}_3^1(\Delta)$

Since we intend to present some numerical examples of minimal energy interpolation based on C^1 cubic splines, in this section we discuss what is known about redundancies in this case.

There are two types of known redundancies for $\mathcal{S}_3^1(\Delta)$. They occur in the C^1 continuity conditions in the first ring around each vertex, and also in the second

ring around each vertex. Here we are using standard Bernstein-Bézier terminology, see e.g. [4]: the coefficients in the ℓ-th ring around the vertex v_1 have the form $c_{d-\ell,j,k}$ with $j+k=\ell$.

The first type of redundancy is easy to deal with. Indeed, as in the planar case, it is easy to see that if $e_1,\ldots,e_m$ are the edges attached to a vertex v, then the two C^1 conditions associated with the two edges e_{m-1} and e_m are redundant, and can simply be dropped.

The second type of redundancy occurs only in connection with singular vertices. A vertex v is *singular* provided it is connected to four vertices $v_1,\ldots,v_4$ where the pair v_1, v_3 lies in one plane passing through v and the origin, and the pair v_2, v_4 lies in another. In this case one of the four ring-2 C^1 continuity conditions is redundant, and can be dropped.

In the planar case, singular vertices are isolated, and it is no problem to remove the redundant second ring conditions. However, on the sphere it is possible that two adjoining vertices are both singular, see Sect. 7 of [5]. Thus, there can be additional redundancies due to the interaction between singular vertices. It is possible to construct an algorithm to eliminate such redundancies by examining all closed paths connecting triangles in Δ.

6.4. Choice of the Quadratic Functional

It remains to discuss appropriate choices of the quadratic functional $\mathcal{E}(c)$. For a general sphere-like surface, it is not clear what a good choice of an energy functional may be. Therefore, in this subsection we restrict our discussion to the special case $\mathcal{S}=S$. In this case, there are two types of natural choices of energy functionals, which are both analogs of the well-known thin-plate functional for functions on the plane. They are of the form

$$\mathcal{E}(f) := \int_S (Of)^2 ds, \tag{6.6}$$

where O is a differential operator. Wahba [79] defined the family of operators

$$O := (\Delta^*)^{m/2}, \tag{6.7}$$

where m is an even integer and where Δ^* is the Laplace-Beltrami operator on S. This operator is a natural analog of the familiar Laplace operator. In fact it is a restriction of the Laplace operator to S. The definition for the case where m is odd is more complicated and can be found in [79].

A different set of operators has been suggested by Freeden [34], who considered operators which are defined by

$$O := (\Delta^*)_m^q := (\Delta^*+\lambda_0)^{q_0}\cdots(\Delta^*+\lambda_m)^{q_m}, \tag{6.8}$$

where $q := (q_0, \ldots, q_m)$ is a vector of positive integers, m is an arbitrary nonnegative integer, and $\lambda_i := i(i+1), i = 0, \ldots, m$. In some sense the second type of operators are more natural since they annihilate all spherical polynomials up to degree m, while (6.7) only annihilates constants. We note that in both cases (6.6) is invariant under rotations of the coordinate system. If we set $m = 2$ in (6.7) and $m = 0, q_0 = 1$ in (6.8), the corresponding functionals are identical and are equal to

$$\mathcal{E}(f) = \int_S (\Delta^* f)^2 ds, \tag{6.9}$$

which is a functional annihilating constants.

In order to compute the entries of the energy matrix Q for a spline s, we need to calculate the energy contributions $\mathcal{E}_{T_i}(s)$ from each single triangle T_i, $i = 1, \ldots, N$, of the given triangulation Δ. If we denote by c_i the vector of the Bézier coefficients of the polynomial $p_i = s|_{T_i}$, then

$$\mathcal{E}_{T_i}(s) = \mathcal{E}_{T_i}(p_i) = c_i^T Q_i c_i,$$

where Q_i is a symmetric $D \times D$ Gram matrix corresponding to the collection of Bernstein basis polynomials defined on T_i, where $D := \dim \mathcal{P}_d = \binom{d+2}{2}$. In particular, if B_j and B_k are the Bernstein basis polynomials associated with the coefficients corresponding to the j-th and k-th entry of vector c_i, the jk-th entry of Q_i is given by

$$(Q_i)_{jk} = \int_{T_i} OB_j OB_k ds, \quad j, k = 1, \ldots, D.$$

The above integrals can be computed based on the considerations in Sect. 4.

In order to calculate the functions OB_j for operators O of types (6.7) and (6.8), one needs to evaluate expressions involving the Laplace-Beltrami operator. We recall a useful formula for Δ^* applied to a homogeneous function of degree d restricted to S and hence, in particular, to an SBB-polynomial p of degree d [53]:

$$\Delta^* p = \Delta p - d(d+1)p. \tag{6.10}$$

In (6.10) we abused the notation slightly: on the right-hand side, Δp should be viewed as the Laplace operator applied to a trivariate function, homogeneous of degree d, which is then restricted to S. The formula reduces the problem of finding $\Delta^* p$ to the problem of computing Δp. We will now give an expression for Δp which involves partial derivatives with respect to barycentric coordinates rather than Cartesian coordinates. Let x_i, $i = 1, 2, 3$, denote the Cartesian coordinates in $\mathbb{R}^3$ and let e_i, $i = 1, 2, 3$, be the unit coordinate vectors. Then, if we view p as a

trivariate homogeneous function expressed in terms of the barycentric coordinates with respect to T, the chain rule gives

$$\frac{\partial p}{\partial x_i} = \sum_{j=1}^{3} \frac{\partial p}{\partial b_j} \frac{\partial b_j}{\partial x_i} = b^T(e_i) \nabla_b p, \quad i = 1, 2, 3,$$

where ∇_b is the gradient taken relative to the variables b_1, b_2, and b_3 as defined in (3.2). As a consequence, we have

$$\frac{\partial^2 p}{\partial x_i^2} = b^T(e_i) H(p) b(e_i), \quad i = 1, 2, 3,$$

and hence

$$\Delta p = \sum_{i=1}^{3} b^T(e_i) H_b(p) b(e_i).$$

Here $H_b(p)$ is the 3×3 Hessian matrix

$$H_b(p) := \left(\frac{\partial^2 p}{\partial b_j \partial b_k} \right)_{j,k=1}^{3},$$

computed relative to the variables b_1, b_2, b_3.

7. Approximation Methods

As in the planar case, there are two situations where it may not be appropriate to interpolate a set of scattered data:

1) if the data are noisy
2) if there is an extremely large number of data.

Both situations occur frequently in practice. Indeed, measured data are almost always subject to some noise, and it simply does not make sense to interpolate the measured values exactly. What we should be doing is smoothing out the noise by some approximation technique. As for the second case, it is not at all uncommon to have hundreds of thousands of data, or even many more, in which case it is generally not appropriate to try to construct an interpolating spline with hundreds of thousands of coefficients.

In this section we discuss analogs of two common methods for dealing with noisy data – discrete least squares, and penalized least squares. Both of these methods are global in nature, and involve solving large sparse systems of linear equations.

7.1. Discrete Least Squares

Suppose that we are given noisy measurements $f_i = f(v_i) + \varepsilon_i$ at n points on a sphere-like surface for an unknown function f. Here ε_i represent unknown errors. The method proceeds as follows:

1) choose a triangulation Δ with V vertices ($V \leq n$),
2) choose a spline space $\mathcal{S}_d^r(\Delta)$,
3) find the spline $s \in \mathcal{S}_d^r(\Delta)$ which minimizes

$$L(s) := \sum_{i=1}^{n} [s(v_i) - f_i]^2.$$

We discuss each of the steps separately.

The choice of the vertices to be triangulated has to be done by the user – there is no simple automatic way to do it. Certainly we will not use all of the data points as vertices (since then we can interpolate with a zero error), and in fact, it is not necessary to choose any of the data points as vertices. Ideally, the user will have some idea of the complexity of the function being fit, in which case more vertices would be inserted where the data change more rapidly.

The choice of spline space is also up to the user. Generally it is advisable to work with low degree spaces ($d \leq 5$), depending on the amount of smoothness required. Often C^1 is adequate.

To carry out step 3), we would like to express $L(s)$ as a quadratic form involving some coefficient vector, and thus we need a set of basis functions for our spline space. Appropriate locally supported basis functions have been described in [5] for $d \geq 3r + 2$, although they can be quite complicated to describe and compute. In practice it may be more desirable to work with spline spaces $\mathcal{S}_d^r(\Delta)$ with $d < 3r+2$. However, for these values of r and d, we do not even know the dimension of the spline space, let alone a basis for it. We now describe a way around this difficulty.

Recall that the spline space $\mathcal{S}_d^r(\Delta)$ is just the subspace of $\mathcal{S}_d^0(\Delta)$ satisfying the set of homogeneous smoothness conditions (6.3). Our approach is to look for a minimum of $L(s)$ in $\mathcal{S}_d^0(\Delta)$ subject to (6.3). As a basis for $\mathcal{S}_d^0(\Delta)$ we can use the splines $\{B_i\}_{i=1}^M$, where M is the dimension of $\mathcal{S}_d^0(\Delta)$ as given in (6.2), and where B_i is the unique spline whose coefficient vector (see Sect. 6.1) is $(0, \ldots, 0, 1, 0, \ldots, 0)$, where the 1 appears in the i-th position. These basis functions are locally supported (with support at most the star of a vertex).

Given data sites $\{v_j\}_{j=1}^n$ on the sphere, let

$$B := (B_{ij})_{i=1,j=1}^{n \quad M}$$

be the $n \times M$ *observation matrix* where $B_{ij} = B_i(v_j)$. Then it is easy to see that

$$L(s) = L(c) = c^T B^T B c - 2c^T B^T g + g^T g, \tag{7.1}$$

where $g = (f_1, \ldots, f_n)^T$ is the column n-vector containing the values to be fit, and c is the vector of coefficients describing s in terms of the basis $\{B_i\}$.

It is now clear that the problem of minimizing $L(s)$ over $s \in \mathcal{S}_d^r(\Delta)$ is equivalent to minimizing $L(c)$ over $\mathcal{S}_d^0(\Delta)$ subject to the linear side conditions (6.3). This problem can be solved numerically by introducing Lagrange multipliers γ. We are led to the linear system

$$\begin{pmatrix} B^T B & G^T \\ G & 0 \end{pmatrix} \begin{pmatrix} c \\ \gamma \end{pmatrix} = \begin{pmatrix} B^T g \\ 0 \end{pmatrix}, \tag{7.2}$$

where G is the matrix in (6.3).

As was the case for minimal energy splines, it is important to remove redundancies in the side conditions describing smoothness. Since the B_i have local support, both $B^T B$ and the block matrix in (7.2) are sparse.

7.2. Penalized Least Squares

In some fitting problems, particularly when the data are especially noisy, it may be useful to replace the standard discrete least squares problem by a *penalized least squares* problem. The idea is to minimize a combination

$$K(c) := L(c) + \lambda \mathcal{E}(c),$$

where $\mathcal{E}(c)$ is a measure of energy as used in Sect. 6.4, and $L(c)$ is the sum of squares of the errors as in (7.1). The parameter λ controls the trade-off between these two quantities, and is typically chosen in the interval $[0, \infty)$, see [39].

The penalized least squares problem can be solved in exactly the same way as the discrete least squares problem. Indeed, assuming that the energy is given by the quadratic form $\mathcal{E}(c) = c^T Q c$, the only change in (7.2) is that the matrix $B^T B$ in the upper left-hand corner has to be replaced by $B^T B + \lambda Q$.

In performing penalized least squares in practice, we still have to choose the parameter λ. If λ is 0, we get the least squares spline. If λ is very large, we are essentially solving a minimal energy problem over the space

$$\{s \in \mathcal{S}_d^r(\Delta) \ : \ Os = 0\},$$

where O is the operator describing the energy term $\mathcal{E}(c)$. In general, good results can be expected with very small values of λ. For more on how to select the smoothing parameter λ, see [39].

8. Examples

In this section we report on our computational experience with the methods presented above. We discuss the following issues:

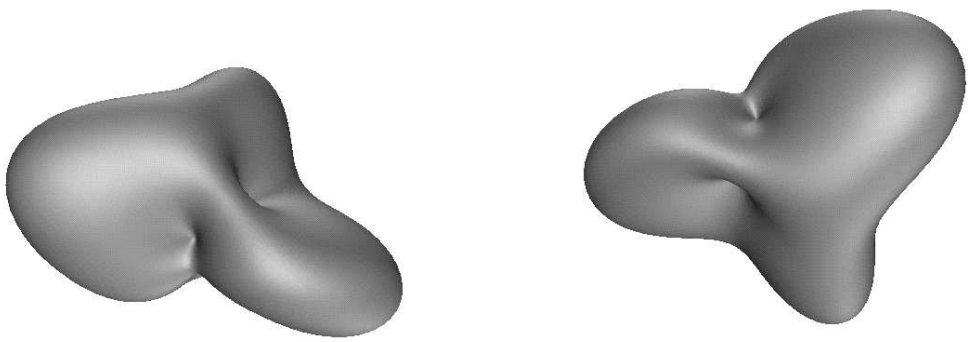

Fig. 5. Graph of f^*.

1) implementation, storage requirements, and speed,
2) convergence behavior, approximation orders,
3) the effect of long and thin triangles in the triangulation,
4) the effect of singular or near-singular vertices,
5) conditioning of linear systems for global methods,
6) accuracy and numerical stability of algorithms.

8.1. The Testing Environment

We use the following abbreviations for our methods: QT (quintic), CT (Clough-Tocher cubic), PS (Powell-Sabin quadratic), ME (minimal energy cubic), LS (least squares cubic), PL (penalized least squares cubic).

All six of these schemes have been implemented in FORTRAN 77 and tested on an SGI Indigo 2. Except for the construction of the triangulations (which were done with a package of Renka [69, 70]) all calculations were carried out in double precision.

All of our tests were conducted on the sphere. The local interpolation methods were tested with data sets of up to 16,386 vertices and up to 32,768 triangles. The global methods were tested on data sets of up to 258 vertices and up to 512 triangles. We ran LS and PL with up to 100,000 data points. We discuss space and time limitations in Sect. 8.2.

We have done testing on a number of functions, but most of the results reported here are based on the test function

$$f^*(x,y,z) = 1 + x^8 + e^{2y^3} + e^{2z^2} + 10xyz. \tag{8.1}$$

For those methods requiring gradient or Hessian information, the needed derivatives were hand coded (with the assistance of *Mathematica*).

To assist in understanding the behavior of the methods, we visually examined the surfaces

$$\{f(v)v \ : \ v \in S\} \tag{8.2}$$

corresponding to our test function and its various interpolants and approximants. In addition to permitting a comparison of shapes, this provided a way to visualize errors, smoothness of derivatives, and curvature by color coding these quantities on the surfaces. These kinds of maps are best appreciated as color images, which unfortunately, we were not able to include in this paper. To give an idea of what our basic test function looks like, we present two views of it in Fig. 5. These figures were obtained with Geomview 1.5 which is available from the Software Development Group, Geometry Center, 1300 South 2nd St, Suite 500, Minneapolis, MN 55454, U.S.A, register@geom.umn.edu. We also used Explorer 2.2.2 available from IRIS Explorer Center, PO Box 50, Oxford OX2 8JU, UK, infodesk@nag.co.uk.

While (8.2) is the most natural way to visualize a surface defined on the sphere, it is certainly not the only way. For some other approaches, see e.g. [12, 29, 62, 63], and references therein.

8.2. A Basic Comparison of the Methods

Tables 1 and 2 summarize some of the basic properties of the first five methods (PL is virtually the same as LS). The entries labeled *Storage* give the amount of space required to store the set of coefficients of the spline associated with a triangulation with V vertices. The table shows that the ME and LS methods require the least storage (which is to be expected, since they are based on cubics, and do not involve split triangles). For the range of problems considered, this is not an important factor.

The entries labeled *Exactness* describe the sets of SBB-polynomials for which the methods give exact results. Thus, for example, if we use the QT method to interpolate data coming from an SBB-polynomial in the space $\mathcal{P}_5$, we get an exact fit. The LS method reproduces cubics (since it is based on cubics). However, even though ME and PL are also based on cubics, they do not reproduce them since $N_{\Delta^*} \cap \mathcal{S}_3^0(\Delta) = \{0\}$, where N_{Δ^*} is the null space of the Laplace-Beltrami operator.

The entries labeled *Time* give the time in seconds required to compute the corresponding interpolant (or approximant) based on a triangulation with 258 vertices. The times for LS were computed for a problem involving 100,000 data sites. As the table shows, the local methods are all very fast, and exhibit comparable running times. The global methods required considerably more time, but most of their running time was devoted to solving the associated linear systems. For

	QT	CT	PS
Storage	$25V - 48$	$27V - 52$	$24V - 46$
Exactness	$\mathcal{P}_5$	$\mathcal{P}_3$	$\mathcal{P}_2$
Order	h^6	h^4	h^3
Time	1 sec.	1 sec.	1 sec.

Table 1. Comparison of local methods.

	ME	LS
Storage	$9V - 16$	$9V - 16$
Exactness	none	$\mathcal{P}_3$
System Size	$17V$	$16V$
Time	200 sec.	125 sec.

Table 2. Comparison of global methods.

this purpose we have tested several public domain sparse matrix solvers (the times reported here are based on a package called Y12M).

In Table 1 the entries labeled *Order* give the order of approximation to be expected in using the method as a power of a parameter h measuring the diameter of the largest triangle in the triangulation. Theoretical results on approximation order of spherical splines will be presented elsewhere. In Sect. 8.3 we give numerical results which support the expected orders of convergence.

In Table 2 the entries labeled *System Size* give the sizes of the linear systems of equations that have to be solved in terms of the number V of vertices of the underlying spherical triangulation. For the global methods, the sizes of the linear systems are the main determinators of the amount of space and time needed to run the program. Because of the space requirements of our sparse system solver, we were not able to run substantially larger problems.

8.3. Numerical Results

To illustrate the performance of our methods, we created a sequence $\Delta_1, \Delta_2, \ldots, \Delta_7$ of regular triangulations, where the number of vertices of Δ_ℓ is $2^{2\ell} + 2$ and the number of triangles is $2^{2\ell+1}$. This sequence was created as follows. Δ_1 is the Delaunay triangulation associated with the 6 vertices of a regular octahedron, *i.e.*, with the points $\pm e_i$, $i = 1, 2, 3$, where e_i are the Cartesian coordinate vectors. This triangulation consists of the 8 quadrantal spherical triangles. Then for each $\ell = 2, \ldots, 7$, we computed the vertices of Δ_ℓ from those of $\Delta_{\ell-1}$ by adding the midpoints of each edge of $\Delta_{\ell-1}$ to the vertices of $\Delta_{\ell-1}$. This amounts to splitting each triangle of $\Delta_{\ell-1}$ into four subtriangles in a standard way, and in fact each of these triangulations is a Delaunay triangulation of its vertex set. The triangulation

Δ_7 involves $16,386$ vertices and $32,768$ triangles. For later use, we note that the mesh sizes h_ℓ of the triangulations Δ_ℓ behave essentially like $2^{-\ell}$ since in refining $\Delta_{\ell-1}$ to get Δ_ℓ, we are approximately halving the size of each triangle. (It is not exactly $1/2$ since we are measuring distances on the surface of S along great circles).

Tables 3 and 5 list the errors for all five methods applied to the test function f^*. The corresponding estimated rates of convergence are shown in Tables 4 and 6, respectively. All reported errors are relative errors, defined by

$$\frac{\max_{v\in V_\ell} |s(v) - f^*(v)|}{\max_{v\in V_\ell} |f^*(v)|},$$

where f^* is the test function and s is its interpolant or approximant. The sets V_ℓ vary with the triangulation Δ_ℓ, and are made up of points of the form

$$\frac{iv_1 + jv_2 + kv_3}{\|iv_1 + jv_2 + kv_3\|}, \qquad i + j + k = 2^{10-\ell},$$

where v_1, v_2, and v_3 are vertices of a typical triangle in Δ_ℓ. These sets are all of the same size, approximately 1,050,000 points, but do not always contain the same points.

Table 4 clearly confirms our expectations for the convergence orders for the local macro-element methods (recall that each refinement of the triangulation reduces the diameter of the largest triangle by a factor of approximately $1/2$, so we expect the ratios of the errors to behave like 2^k, where k is the order). In particular, QT gave order 5, CT order 3, and PS order 2.

The columns labeled QT* in Tables 3 and 4 are included to show what happens when the second derivative information required for QT is not computed properly. In the form given in (8.1), the function f^* is *not* homogeneous of order 5, and so by the discussion in Sect. 5.4, we cannot expect order 5 convergence if we compute second derivatives directly from (8.1). Indeed as the tables show, if we do compute second derivatives in this way, we seem to be getting order 2 convergence. The order 5 convergence shown in the columns labeled QT corresponds to computing the second derivatives from the order 5 homogeneous extension of f^* (cf. Lemma 2.1).

The situation is less clear-cut for the global methods. The approximation order of $\mathcal{S}_3^1(\Delta)$ is unknown in general, even for the planar case. It has been shown [17] that in that case the approximation order of $\mathcal{S}_3^1(\Delta)$ on a uniform type-1 triangulation is three. This has also been confirmed experimentally in the tests presented in [36]. Since our sequence of spherical triangulations Δ_ℓ is an analog of uniform type-1 planar triangulations, we expect that for such triangulations, the approximation order of the homogeneous spline space $\mathcal{S}_3^1(\Delta)$ is also 3.

In Sect. 7.1 we have chosen to work with a basis for the entire space $\mathcal{S}_d^0(\Delta)$. As in the planar case [1] and [36], it is also possible to work with bases for certain

	QT	QT*	CT	PS
Δ_1	8.1554 (-2)	2.9288 (-1)	1.0647 (-1)	5.5912 (-1)
Δ_2	7.3667 (-3)	1.1328 (-1)	4.4236 (-2)	7.8296 (-2)
Δ_3	6.0130 (-4)	3.0328 (-2)	1.0490 (-2)	2.1020 (-2)
Δ_4	2.0279 (-5)	7.8996 (-3)	1.0721 (-3)	2.0461 (-3)
Δ_5	3.8139 (-7)	1.9925 (-3)	7.6170 (-5)	2.2841 (-4)
Δ_6	6.1964 (-9)	5.0004 (-4)	4.8942 (-6)	2.8834 (-5)
Δ_7	9.5359 (-11)	1.2607 (-4)	3.0294 (-7)	3.5994 (-6)

Table 3. Approximation errors for local methods.

	QT	QT*	CT	PS
Δ_1/Δ_2	11.07	2.59	2.41	7.14
Δ_2/Δ_3	12.25	3.74	4.22	3.72
Δ_3/Δ_4	29.65	3.84	9.78	10.27
Δ_4/Δ_5	53.17	3.96	14.08	8.96
Δ_5/Δ_6	61.55	3.99	15.56	7.92
Δ_6/Δ_7	64.98	3.97	16.16	8.01

Table 4. Approximation rates for local methods.

	ME	LS	LS-2
Δ_1	4.0333 (-1)	8.0373 (-2)	6.6153 (-5)
Δ_2	2.8149 (-1)	6.2699 (-2)	4.9083 (-5)
Δ_3	2.8027 (-2)	1.1354 (-2)	4.8470 (-6)
Δ_4	1.6184 (-3)	1.0286 (-3)	3.8391 (-7)

Table 5. Approximation errors for global methods.

	ME	LS	LS-2
Δ_1/Δ_2	1.43	1.28	1.35
Δ_2/Δ_3	10.04	5.52	10.13
Δ_3/Δ_4	17.32	11.04	12.63

Table 6. Approximation rates for global methods.

smaller subspaces of $\mathcal{S}_d^0(\Delta)$. We should also note that although we have used Lagrange multipliers to convert the problem of minimizing a quadratic function subject to linear side constraints into an equivalent linear system of equations,

these types of constrained minimization problems can also be solved directly using any one of a variety of available algorithms, see [26, 35].

8.4. Effect of Thin Triangles

It is common practice in finite element computations to avoid the use of thin triangles. In this section we explore the performance of each of our methods in the presence of thin triangles. The tests were performed on a sequence of triangulations τ_ε obtained by adding one additional vertex to the regular triangulation Δ_1 with 6 vertices discussed in Sect. 8.3 above. For given ε, the new vertex was placed at the point with spherical coordinates $(\varepsilon, 0)$ in degrees. We then created the corresponding Delaunay triangulation τ_ε. Since one of the vertices of Δ_1 is at the point $(1, 0, 0)$ with spherical coordinates $(0, 0)$, when ε is small the triangulation τ_ε has two thin triangles, one attached to each pole. We examined the sequence $\varepsilon_i = 10^{-i}$ for $i = 1, \ldots, 9$.

Our first tests were conducted on the local macro-element methods QT and CT. In order to be able to observe the effect of the thin triangles more clearly, we generated the data from the function $f \equiv x + y + z$ rather than f^*, since (with no roundoff error) both methods give exact results for this function. We do not bother to give a table of results, since for both QT and CT, we got virtually the same accuracy (approximately 10^{-15}) for all ε_i. This suggests that these methods are not sensitive to the presence of thin triangles in the triangulation.

Next we tested PS with the same sequence of triangulations. For this method we generated the data from the function $f \equiv 1$, since (with no roundoff error) the PS method produces an exact fit for this function. The results are presented in Table 7, which shows that thin triangles have a clear effect on the accuracy of this method. We lose essentially one digit of decimal accuracy each time ε_i is reduced by a factor of 10.

There may be several causes for the dropoff in accuracy in the presence of increasingly small triangles. We believe that the main reason is that for a very long and thin triangle, the incenter will be very close to two vertices at one end. This leads to intersection points on the long edges which are very nearly at the ends of the edges. In that case the computation of Bézier coefficients in the first ring around those vertices loses accuracy because of the small size of the barycentric coordinate appearing in the denominator of (5.2). This does not happen for the QT method since the original triangles are not split. Neither does it happen for the CT method since there the split point is chosen to be the center of the triangle, which is not close to any of the vertices.

The effect of thin triangles is much more pronounced for the minimal energy method ME and for the least squares LS and penalized least squares PL methods. In fact, the condition numbers became so bad that our sparse system solver failed

ε	error
1 (-1)	5.11 (-15)
1 (-2)	7.77 (-15)
1 (-3)	1.10 (-13)
1 (-4)	1.63 (-12)
1 (-5)	1.12 (-11)
1 (-6)	1.08 (-10)
1 (-7)	1.09 (-9)
1 (-8)	1.01 (-8)
1 (-9)	1.36 (-7)

Table 7. Consequences of thin triangles for PS.

for ME already for $\varepsilon = .01$, while for PL it failed for $\varepsilon = .001$.

8.5. Condition Numbers

The global methods LS and ME require solving linear systems of equations which are generally large and sparse. In practice we recommend solving them with sparse matrix methods.

For both methods, the matrices of interest have a block structure (cf. (6.5) and (7.2)) where the sizes of the entries in the matrix G describing smoothness conditions is of order 1. However, the sizes of the entries in the matrices Q and B^TB appearing in these systems can often be much larger or much smaller than those in G. In particular, the entries of Q are obtained by computing integrals of second derivatives of the SBB-polynomial basis functions B_{ijk}, and these become large for triangles with one or more short edges. The entries of B^TB in the LS method are sums of products of values of these basis functions, and can be either very large or very small (in absolute value), depending on how many data points fall in a given triangle.

Our numerical experience suggests that while the condition numbers of the matrices Q and B^TB are quite good, the condition numbers of the block matrices in (6.5) and (7.2) are often very large, indicating the need for some kind of scaling. While there are a variety of general-purpose scaling strategies which can be applied to a linear system, we have experimented with a simple scaling strategy which appears to be considerably more effective than standard techniques like row equilibration. To scale the system (6.5) which arises in the ME method, we multiply the entries in Q by a positive constant α. By the block structure of the system, the vector $(c, \lambda, \gamma)^T$ is a solution of the original system if and only if $(c, \alpha\lambda, \alpha\gamma)^T$ is a solution of the new system. Similarly, to scale the system (7.2) which arises in the LS method, we multiply the entries in B^TB and the right-hand side vector

	C_1	C_α	α	min	max	size
Δ_1	2.1 (3)	10	.0139	15.5	6.1 (2)	238
Δ_2	1.1 (5)	26	.0088	17.8	1.2 (4)	510
Δ_3	8.8 (6)	61	.0015	37.6	2.8 (5)	1054

Table 8. Optimal scaling parameters and condition numbers for ME.

	C_1	C_α	α	min	max	n_{min}
Δ_1	1.6 (6)	18	.0017	1.9 (0)	2.0 (3)	1,247
Δ_2	1.5 (4)	29	.0392	1.6 (-1)	2.5 (3)	308
Δ_3	1.6 (3)	54	.1316	2.0 (-2)	7.1 (2)	40

Table 9. Optimal scaling parameters and condition numbers for LS.

$B^T g$ by a positive constant α. Again the solution of the new system gives the same coefficient vector c, but a changed Lagrange multiplier vector γ.

To see how effective this scaling strategy is, we conducted some tests using LINPACK to compute condition numbers of the systems. We used the same triangulations Δ_1, Δ_2, and Δ_3, as in Section 8.3. For each triangulation, we determined a (near) optimal scaling by examining a series of α values, and performing a golden section search.

Tables 8 and 9 show the results for ME and LS (applied to a set of 10,000 random points), respectively. In both tables the column labeled C_1 lists the condition numbers of the associated systems without scaling. The second column labeled C_α gives the improved condition numbers obtained by using the associated value of α in column 3. In Table 8 the fourth and fifth columns list the smallest (nonzero) and largest entries in the matrix Q, while the the last column gives the number of equations in the system. In Table 9 the fourth and fifth columns list the smallest (nonzero) and largest entries in the matrix $B^T B$, while the last column gives the minimum number n_{min} of data points per triangle.

Clearly, scaling has a highly significant impact on condition number for both methods. Indeed, although the original condition numbers range from order 10^3 to 10^6, in all cases we are able to reduce them to very satisfactory values. Note that as the number of triangles in the triangulation increases, for the ME method the values of C_1 increase while the optimal values of α decrease. The reverse happens for the LS method.

There is, of course, a vast literature on the subject of preconditioning linear systems, and we make no claim of having developed an optimal way to scale the systems arising in LS and ME methods. However, the above tests indicate that the simple method suggested above is quite effective. Its obvious drawback is that it

requires a parameter α that depends on the data and at present can be determined only empirically. Moreover, our computations suggest that the size of the condition numbers obtained are quite sensitive to the choice of scaling parameters.

8.6. The Effect of Near-singular Vertices

The global methods ME, LS, and PL involve minimizing a quadratic functional over the space $\mathcal{S}_3^0(\Delta)$ of continuous spherical cubic splines, subject to side conditions enforcing interpolation and C^1 continuity. As discussed in Sect. 6.3, certain of the smoothness conditions are redundant when a vertex is singular. This means that the dimension of the spline space changes as a vertex approaches singularity (usually, the dimension jumps by one when a vertex becomes singular). The change in dimension corresponds to a change in the rank of the linear system defining the solution. It might be expected that such changes in the dimension would have a significant impact on the results when comparing a triangulation with a singular vertex to one which is obtained by perturbing the vertex slightly.

To test the effect of near-singularity, we start with the regular triangulation Δ_2 of Sect. 8.3 which has 18 vertices, six of which are singular. For each $i = 0, \ldots, 12$ we perturb the spherical coordinates (θ, ϕ) of each of the singular vertices by random multiples of $2^{-i}\varepsilon$, where ε is 4 degrees. Thus in stepping down one line in Tables 10 and 11, the perturbations are exactly halved, except that the last line corresponds to Δ_1 itself. In this case the known redundant equations are removed, which accounts for the much smaller condition number. In both tables we present the corresponding values of the condition numbers corresponding to the optimal scaling for Δ_2 given in Tables 8 and 9. The corresponding condition numbers without scaling would be significantly larger.

For both ME and LS we choose the function $f(x, y, z) = x + y + z$. Without roundoff error it is fit exactly by the LS method, and so the errors listed in the third column of Table 11 are due to the effects of the numerical computation. The LS test was done with a set of 10,000 random data points on the sphere. Unfortunately, we do not have a function that is reproduced exactly by the ME method, and so we cannot exhibit the subtle effects of near singularity on the error.

As we move down the rows of the tables, we observe a significant increase in the condition number of the associated linear system (to the point that the estimated condition number equals the reciprocal of the round-off unit). In particular, the condition numbers increase by approximately a factor of 10 each time the angular perturbation of the singular vertices is reduced by a factor of 2.

8.7. Guidelines

We have not done an extensive numerical comparison of spherical spline methods against various alternative methods, and thus are not in a position to make

ε	Condition Number
2^2	6.3 (5)
2^1	9.2 (6)
2^0	1.4 (8)
2^{-1}	2.2 (9)
2^{-2}	3.6 (10)
2^{-3}	5.7 (11)
2^{-4}	9.1 (13)
2^{-5}	1.5 (14)
2^{-6}	2.3 (15)
2^{-7}	3.4 (16)
2^{-8}	8.3 (17)
2^{-9}	4.4 (17)
2^{-10}	1.9 (17)
0	10.3

Table 10. Consequences of near-singular vertices for ME.

ε	Condition Number	Error
2^2	9.1 (5)	6.5 (-15)
2^1	1.2 (7)	9.4 (-15)
2^0	1.8 (8)	8.4 (-14)
2^{-1}	2.8 (9)	4.5 (-14)
2^{-2}	4.4 (10)	3.4 (-14)
2^{-3}	7.1 (11)	1.6 (-13)
2^{-4}	1.1 (13)	6.0 (-14)
2^{-5}	1.8 (14)	2.5 (-13)
2^{-6}	2.9 (15)	1.0 (-12)
2^{-7}	4.6 (16)	3.2 (-12)
2^{-8}	7.3 (18)	3.1 (-12)
2^{-9}	1.2 (20)	6.6 (-12)
2^{-10}	1.9 (20)	2.4 (-11)
0	17.8	5.0 (-15)

Table 11. Consequences of near-singular vertices for LS.

any recommendations here. However, we can offer the following general guidelines concerning the choice among the various spline methods treated here:

1) If the number of data is very large, or if the values to be fit are noisy, the user should strongly consider using least squares or penalized least squares.

2) There are many advantages to using a local method rather than a global one. In particular, our PS, CT, and QT methods are extremely fast, can be used on large data sets, and are both stable and accurate. However, all three methods depend on having good values for derivatives. In particular, if second derivatives are available, it is hard to beat QT in terms of speed and accuracy. However, these methods may not perform as well in cases where the derivatives have to be estimated, particularly if the number of data points available to do the derivative estimation is small.

3) The advantage of the ME interpolation method is that it does not require any derivative information. The main disadvantage is that it requires solving a large system of equations which seems to become less well-conditioned with size. In addition, in general we can expect an ME interpolant to be somewhat smoother than the macro-element methods PS, CT, and QT discussed above (although all methods are C^1). The reason is that ME minimizes energy, which in a certain sense corresponds to minimizing curvature of the surface.

4) For highly noisy data, we have found that the penalized least squares method performs significantly better than simple least squares. The main difficulty in using it is to choose an appropriate smoothing parameter.

9. Remarks

Remark 1. In principle any interpolation or approximation problem on a sphere-like surface can be converted into a similar problem on the sphere by the simple expedient of replacing the data sites v_i by their projections $v_i/\|v_i\|$ onto the sphere. However, this is not always desirable for a number of reasons. The data may contain directional information (for example velocities) which cannot be transferred to the unit sphere without a redefinition and recomputation involving σ. The projection may contain a geometric distortion which changes a simple function defined on points in $\mathbb{R}^3$ to a more complicated function on the sphere. Finally, and perhaps most importantly, in the special case that $\mathcal{S}$ is the surface of earth or an equipotential surface of gravity, σ is very complicated, only partially known, and even controversial in some places on earth, see [22]. Throughout this paper we therefore employed the more general setting provided by a sphere-like surface.

Remark 2. If a triangulation covers only a part of a a sphere-like surface $\mathcal{S}$, we call it a *partial* triangulation. In this paper we have discussed only total triangulations. However, all of the methods discussed here can easily be extended to partial triangulations.

Remark 3. Suppose g is a tangent vector to a sphere-like surface $\mathcal{S}$ at a point $v \in \mathcal{S}$. The first derivative of f in the direction g can also be defined equivalently

as

$$D_g f(v) := \frac{df(\mathcal{C}(\theta))}{d\theta}|_{\theta=0},$$

where $\mathcal{C}(\theta)$ is defined as in the proof of Lemma 3.5. Higher order *pure* derivatives can also be defined in this parametric way. However, this approach does not work for mixed derivatives of the form D_{g_1,g_2}, since in general a tangent vector g to $\mathcal{S}$ at v is no longer tangent to $\mathcal{S}$ if we move away from the point v.

Remark 4. An identity similar to (4.1) also holds for integrals of CBB-polynomials along edges of spherical triangles except that in this case the exponent $d+3$ in (4.1) should be replaced by $d+2$.

Remark 5. In view of (6.10) it may be useful to express a spherical polynomial in terms of spherical harmonics as

$$p = s_d + s_{d-2} + \cdots + s_{(1-(-1)^d)/2},$$

where s_k is a spherical harmonic of degree k. This simplifies the calculation of Δ^* since for all k, Δs_k vanishes and hence

$$\Delta^* p = -(d(d+1)s_d + (d-2)(d-1)s_{d-2} + \ldots + (1-(-1)^d)s_{(1-(-1)^d)/2}.$$

However, it is not clear to us how to do the above factorization efficiently (for a discussion of this problem, see [33]). Also, if f is a spherical harmonic and if $b(v)$ are the barycentric coordinates of v with respect to a triangle T, then $f(b(v))$ may not be a spherical harmonic. Hence if the factorization is expressed as a function f of b, this function will generally depend on T. For example, if T is the quadrantal triangle, $2b_1^2 - b_2^2 - b_3^2$ is a spherical harmonic of degree 2, but this is not true for a general triangle T. This is in contrast to spherical polynomials, since we know that if $f(b)$ is a spherical polynomial, then it is so independently of the given triangle.

Remark 6. It is well known in the planar case that for $d \geq 3r+2$, the spline spaces $\mathcal{S}_d^r(\Delta)$ have approximation power h^{d+1}, where h is the mesh size of the triangulation, *i.e.*, the diameter of the largest triangle in Δ. We will report on analogous results for the sphere elsewhere.

Remark 7. As discussed in Sect. 5, the quintic macro-element method depends on how the required derivatives are computed (*i.e.*, on the degree of the homogeneous extension used). However, in a weaker sense, the quintic method is invariant under homogeneous extensions. Suppose s is a quintic spherical spline obtained by the method of Sect. 5.1, interpolating the prescribed values and derivatives of a given function f, which is also viewed as a homogeneous function of degree five. Next, let $t \in \mathbb{R}$ and let s_t be the quintic spherical spline obtained similarly as s with the exception that now all needed derivative values are computed based on viewing

both functions s_t and f as homogeneous functions of degree t rather than five. It can be shown that then s_t is identical with s, for all t. This fact may be useful in situations where the underlying function f is unknown and where only the degree of homogeneity of f used in calculating second order derivatives is given. In this case, in order to maintain good approximation properties of the quintic method, one would have to view s as a function of exactly the same degree of homogeneity.

Remark 8. Spherical Bernstein-Bézier polynomials are a special case of certain spherical spline functions which are analogs of simplex splines in the plane. These *spherical simplex splines* are locally supported smooth functions whose pieces are spherical polynomials, see [54, 59] for a discussion.

Remark 9. If the data to be fitted are reasonably uniformly distributed, it would make sense to use one of the "regular" triangulations introduced in Sect. 8.3, or similar triangulations which can be obtained by starting with the set of 12 vertices corresponding to a regular icosahedron (which have the advantage that they do not contain singular vertices). The problem of computing sets of points on the sphere which are well spread-out is of considerable importance in several other fields, and has an extensive literature, see *e.g.*, [67].

Remark 10. As shown in [23, 64, 66, 71] for the planar case, in constructing a spline interpolant or approximant, in some cases it is very useful to make the triangulation depend on the data. This can be accomplished by choosing some appropriate criterion (such as smoothness of the resulting surface), and then adjusting the triangulation by swapping edges. This approach carries over immediately to the spherical setting. For some algorithms for finding best triangulations using edge swapping, see [74, 75]. In case there are many vertices in the triangulation, one may also want to consider removing vertices whose presence has little or no effect on the quality of the approximant [50].

Remark 11. In the planar case, it is possible to use C^1 cubic splines without splitting the triangles by working with rational functions rather than polynomials – see [30, 40]. This method has been extended to the sphere in [51].

Remark 12. It is straightforward to construct a parametric surface defined on a sphere-like surface $\mathcal{S}$ simply by defining three coordinate functions $s_i(v)$ defined for $v \in \mathcal{S}$. As in the planar case [25], one advantage of this approach to modeling complicated functions is that the resulting surface has true C^r smoothness, and not just geometric continuity. We present parametric results elsewhere.

Remark 13. We do not attempt to list all possible planar scattered data methods — for more comprehensive lists, see the survey papers [16, 31, 76]. For more on the planar analogs of our PS, CT, and QT methods, see [20, 21, 10]. Global interpolation methods using $\mathcal{S}_3^1(\Delta)$ were investigated by [36, 37, 42]. For alternative

macro-element methods, some with higher-order smoothness, see [2, 18, 49, 72, 83], and references therein. There are are a large number of interpolation methods based on blending; here we mention just one [43].

Remark 14. Our approach to constructing splines on sphere-like surfaces is to use restrictions of homogeneous functions. For some other methods based on restrictions of trivariate functions, see [7, 8, 9, 13].

Remark 15. In Sect. 8 we have reported on the performance of a minimal energy method based on C^1 cubic splines and using the energy measure (6.9) based on the Laplace-Beltrami operator. This functional annihilates constants, but constants are not in the space $\mathcal{S}_3^1(\Delta)$, so as indicated in Table 2, the method is exact only for the zero function. This suggests that this measure of energy and this space are probably not well-matched. The main impetus for discussing this method is that it is the most natural analog of planar methods studied by several authors.

References

1. Alfeld, P., Triangular extrapolation, Mathematics Research Center Rpt. 2707, 1984.
2. Alfeld, P., A bivariate C^2 Clough-Tocher scheme, Comput. Aided Geom. Design **1** (1984), 257–267.
3. Alfeld, P., M. Neamtu, and L. L. Schumaker, Circular Bernstein-Bézier polynomials, in *Mathematical Methods for Curves and Surfaces* (Morten Dæhlen, Tom Lyche, Larry L. Schumaker, eds), Vanderbilt University Press (Nashville & London), 1995, 11–20.
4. Alfeld, P., M. Neamtu, and L. L. Schumaker, Bernstein-Bézier polynomials on spheres and sphere-like surfaces, Comput. Aided Geom. Design , to appear.
5. Alfeld, P., M. Neamtu, and L. L. Schumaker, Dimension and local bases of homogeneous spline spaces, SIAM J. Math. Anal., to appear.
6. Alfeld, P., B. Piper, and L. L. Schumaker, An explicit basis for C^1 quartic bivariate splines, SIAM J. Numer. Anal. **24** (1987), 891–911.
7. Bajaj, C., F. Bernardini, and G. Xu, Automatic reconstruction of surfaces and scalar fields from 3D scans, Computer Graphics **29, 2** (1995), to appear.
8. Bajaj, C., F. Bernardini, and G. Xu, Adaptive reconstruction of surfaces and scalar fields from dense scattered trivariate data, Computer Science Technical Report, TR 95-028, Purdue University, 1995.
9. Bajaj, C., and G. Xu, Modeling and visualization of scattered function data on curved surfaces, in *Fundamentals of Computer Graphics*, J. Chen, N. Thalmann, Z. Tang, and D. Thalmann (eds.), World Scientific Publishing Co., 1994, 19–29.

10. Barnhill, R. E., and G. Farin, C^1 quintic interpolation over triangles: two explicit representations, Int. J. Numer. Meth. Engr. **17** (1981), 1763–1778.

11. Barnhill, R. E., and T. A. Foley, Methods for constructing surfaces on surfaces, in *Geometric Modeling, Methods and Applications* (H. Hagen and D. Roller, eds), Springer Verlag (Berlin), 1991, 1–15.

12. Barnhill, R. E., G. T. Makatura, and S. E. Stead, A new look at higher dimensional surfaces through computer graphics, in *Geometric Modeling: Algorithms and New Trends* (G. E. Farin, ed), SIAM Publications (Philadelphia), 1987, 123–130.

13. Barnhill, R. E., K. Opitz, and H. Pottmann, Fat surfaces: a trivariate approach to triangle-based interpolation on surfaces, Comput. Aided Geom. Design **9** (1992), 365–378.

14. Barnhill, R. E., B. R. Piper, and K. L. Rescorla, Interpolation to arbitrary data on a surface, in *Geometric Modeling: Algorithms and New Trends* (G. E. Farin, ed), SIAM Publications (Philadelphia), 1987, 281–290.

15. Barnhill, R. E., and H. S. Ou, Surfaces defined on surfaces, Comput. Aided Geom. Design **7** (1990), 323–336.

16. Boor, C. de, Multivariate piecewise polynomials, Acta Numerica (1993), 65–109.

17. Boor, C. de, and K. Höllig, Approximation power of smooth bivariate pp functions, Math. Z. **197** (1988), 343–363.

18. Dahmen, W., R. H. J. Gmelig Meyling, and J. H. M. Ursem, Scattered data interpolation by bivariate C^1 piecewise quadratic functions, Approx. Theory Appl. **6** (1990), 6–29.

19. Dierckx, P., Algorithms for smoothing data on the sphere with tensor product splines, Computing **32** (1984), 319–342.

20. Dierckx, P., *Curve and Surface Fitting with Splines*, Clarendon Press, Oxford, 1993.

21. Dierckx, P., S. van Leemput, and T. Vermeire, Algorithms for surface fitting using Powell-Sabin splines, IMA J. Numer. Anal. **12(2)** (1992), 271–299.

22. Dragomir, V. C. et al., *Theory of the Earths Shape*, Elsevier Sc. Publ., New York, 1982.

23. Dyn, N., D. Levin, and S. Rippa, Algorithms for the construction of data dependent triangulations, in *Algorithms for Approximation II* (J. C. Mason and M. G. Cox, eds), Chapman & Hall (London), 1990, 185–192.

24. Fasshauer, G., Adaptive least squares fitting with radial basis functions on the sphere, in *Mathematical Methods for Curves and Surfaces* (Morten Dæhlen,

Tom Lyche, Larry L. Schumaker, eds), Vanderbilt University Press (Nashville & London), 1995, 141–150.

25. Fasshauer, G., and L. L. Schumaker, Minimal energy surfaces using parametric splines, Comput. Aided Geom. Design , to appear.

26. Fletcher, R., *Practical Methods of Optimization*, Second Edition, John Wiley & Sons, Chichester, 1987.

27. Foley, T. A., Interpolation to scattered data on a spherical domain, in *Algorithms for Approximation II* (J. C. Mason and M. G. Cox, eds), Chapman & Hall (London), 1990, 303–310.

28. Foley, T. A., D. A. Lane, G. M. Nielson, R. Franke, and H. Hagen, Interpolation of scattered data on closed surfaces, Comput. Aided Geom. Design **7** (1989), 303–312.

29. Foley, T. A., D. A. Lane, G. M. Nielson, and R. Ramaraj, Visualizing functions over a sphere, IEEE Comp. Graphics & Appl. **10** (1990), 32–40.

30. Foley, T. A., and K. Opitz, Hybrid cubic Bézier triangle patches, in *Mathematical Methods in Computer Aided Geometric Design II* (T. Lyche and L. Schumaker, eds), Academic Press (New York), 1992, 275–286.

31. Franke, R., and G. M. Nielson, Scattered data interpolation and applications: A tutorial and survey, in *Geometric Modeling* (H. Hagen and D. Roller, eds), Springer Verlag (Berlin), 1991, 131–160.

32. Freeden, W., On spherical spline interpolation and approximation, Math. Meth. Appl. Sci. **3** (1981), 551–575.

33. Freeden, W., and R. Reuter, Exact computation of spherical harmonics, Computing **32** (1984), 365–378.

34. Freeden, W., Spherical spline interpolation-basic theory and computational aspects, J. Comp. Appl. Math. **11** (1985), 367–375.

35. George, A., M. T. Heath, and E. Ng, A comparison of some methods for solving sparse linear least squares problems, SIAM J. Sci. Stat. Comp. **4** (1983), 177–187.

36. Gmelig Meyling, R. H. J., Approximation by cubic C^1 splines on arbitrary triangulations, Numer. Math. **51** (1987), 65–85.

37. Gmelig Meyling, R. H. J., and P. Pfluger, B-spline approximation of a closed surface, IMA J. Numer. Anal. **7** (1987), 73–96.

38. Gmelig Meyling, R. H. J., and P. Pfluger, Smooth interpolation to scattered data by bivariate piecewise polynomials of odd degree, Comput. Aided Geom. Design **7** (1990), 439–458.

39. Golitschek, M. von, and L. L. Schumaker, Data fitting by penalized least squares, in *Algorithms for Approximation II* (J. C. Mason and M. G. Cox, eds), Chapman & Hall (London), 1990, 210–227.

40. Goodman, T. N. T., and H. B. Said, A C^1 triangular interpolant suitable for scattered data interpolation, Comm. Appl. Numer. Meth. **7** (1991), 479–485.

41. Gradshteyn, I. S., and I. M. Ryzhik, *Table of Integrals, Series, and Products*, Academic Press, 1980.

42. Grandine, T. A., An iterative method for computing multivariate C^1 piecewise polynomial interpolants, Comput. Aided Geom. Design **4** (1987), 307–319.

43. Gregory, J. A., A C^1 Triangular Interpolation Patch for Computer Aided Geometric Design, Comp. Graphics and Image Proc. **13** (1980), 80–87.

44. Hardy, R. L., and W. M. Göpfert, Least squares prediction of gravity anomalies, geoidal undulations, and deflection of the vertical with multiquadric harmonic functions, Geophy. Res. Letters **2** (1981), 423–426.

45. Hardy, R. L., and S. A. Nelson, A multiquadric-biharmonic representation and approximation of disturbing potential, Geophys. Res. Letters **13** (1986), 18–21.

46. Laghchim-Lahlou, M., and P. Sablonnière, C^r finite elements of HCT, PS and FVS types, in *Proc. Fifth Intern. Symposium on Numerical Methods in Engineering, Vol. 2* (R. Gruber, J. Periaux and R. P. Shaw, eds), Springer-Verlag (Berlin), 1989, 163–168.

47. Lawson, C. L., C^1 surface interpolation for scattered data on a sphere, Rocky Mountain J. Math. **14** (1984), 177–202.

48. Le Méhauté, A., Approximation of derivatives in $\mathbb{R}^n$ application: construction of surfaces in $\mathbb{R}^n$, in *Approximation Theory and Spline Functions* (S. P. Singh, J. H. W. Burry, and B. Watson, eds), Reidel (Dordrecht), 1984, 361–378.

49. Le Méhauté, A., A finite element approach to surface reconstruction, in *Computation of Curves and Surfaces* (W. Dahmen, M. Gasca and C. Micchelli, eds), Kluwer (Dordrecht, Netherlands), 1990, 237–274.

50. Le Méhauté, A., and Y. Lafranche, A knot removal strategy for scattered data in $\mathbb{R}^2$, in *Mathematical Methods in Computer Aided Geometric Design* (T. Lyche and L. Schumaker, eds), Academic Press (New York), 1989, 419–426.

51. Liu, Xiaoyu, and L. L. Schumaker, Hybrid cubic Bézier patches on spheres and sphere-like surfaces, manuscript, 1995.

52. Lyness, J. N., and R. Cools, A survey of numerical cubature over triangles, Proc. of Symposia in Applied Mathematics **48** (1994), 127–150.

53. Müller, C., *Spherical Harmonics*, Springer Lecture Notes in Mathematics, Vol. 17, 1966.

54. Neamtu, M., Homogeneous simplex splines, preprint, 1995.

55. Nielson, G. M., Modeling and visualizing volumetric and surface-on-surface data, in *Focus on Scientific Visualization* (H. Hagen, H. Mueller, and G. Nielson, eds), Springer-Verlag (Berlin), 1992, 219–274.

56. Nielson, G. M., Scattered data modeling, IEEE Comp. Graph. Appl. **13** (1993), 60–70.

57. Nielson, G. M., and R. Ramaraj, Interpolation over a sphere based upon a minimum norm network, Comput. Aided Geom. Design **4** (1987), 41–58.

58. Opitz, K., and H. Pottmann, Computing shortest paths on polyhedra: applications in geometric modeling and scientific visualization, Int. J. Comp. Geom. Applic. **4** (1994), 165–178.

59. Pfeifle, R., and H.-P. Seidel, Spherical triangular B-splines with application to data fitting, preprint, 1994.

60. Pottmann, H., Interpolation on surfaces using minimum norm networks, Comput. Aided Geom. Design **9** (1992), 51–67.

61. Pottmann, H., and M. Eck, Modified multiquadric methods for scattered data interpolation over a sphere, Comput. Aided Geom. Design **7** (1990), 313–321.

62. Pottmann, H., H. Hagen, and A. Divivier, Visualizing functions on a surface, Journal of Visualization and Computer Animation **2** (1991), 52–58.

63. Pottmann, H., and K. Opitz, Curvature analysis and visualization for functions defined on Euclidean spaces or surfaces, Comput. Aided Geom. Design **11** (1994), 655–674.

64. Quak, E., and L. L. Schumaker, Cubic spline fitting using data dependent triangulations, Comput. Aided Geom. Design **7** (1990), 293–301.

65. Quak, E., and L. L. Schumaker, Calculation of the energy of a piecewise polynomial surface, in *Algorithms for Approximation II* (J. C. Mason and M. G. Cox, eds), Chapman & Hall (London), 1990, 134–143.

66. Quak, E., and L. L. Schumaker, Least squares fitting by linear splines on data dependent triangulations, in *Curves and Surfaces* (P.-J. Laurent, A. Le Méhauté, and L. L. Schumaker, eds), Academic Press (New York), 1991, 387–390.

67. Rakhmanov, E., E. B. Saff, and Y. M. Zhou, Minimal discrete energy on the sphere, Math. Res. Let. **1** (1994), 647–662.

68. Ramaraj, R., Interpolation and display of scattered data over a sphere, M.S. Thesis, Arizona State University, 1986.

69. Renka, R. J., Interpolation of data on the surface of a sphere, ACM Trans. Math. Software **10** (1984), 417–436.

70. Renka, R. J., Algorithm 623: Interpolation on the surface of a sphere, ACM Trans. Math. Software **10** (1984), 437–439.

71. Rippa, S., Long and thin triangles can be good for linear interpolation, SIAM J. Numer. Anal. **29** (1992), 257–270.

72. Sablonnière, P., Composite finite elements of class C^2, in *Topics in Multivariate Approximation* (C. K. Chui, L. L. Schumaker, and F. Utreras, eds), Academic Press (New York), 1987, 207–217.

73. Schmidt, R. M., Eine Methode zur Konstruktion von C^1-Flächen zur Interpolation unregelmässiger Daten, in *Multivariate Approximation Theory II* (W. Schempp and K. Zeller, eds), Birkhäuser (Basel), 1982, 343–361.

74. Schumaker, L. L., Triangulation methods in CAGD, IEEE Computer Graphics and Applications **13** (1993), 47–52.

75. Schumaker, L. L., Computing optimal triangulations using simulated annealing, Comput. Aided Geom. Design **10** (1993), 329–345.

76. Schumaker, L. L., Applications of multivariate splines, Proceedings of Symposia in Applied Mathematics, Vol. 48, AMS, 1994, 177–203.

77. Schumaker, L. L, and C. Traas, Fitting scattered data on spherelike surfaces using tensor products of trigonometric and polynomial splines, Numer. Math. **60** (1991), 133–144.

78. Traas, C. R., Smooth approximation of data on the sphere with splines, Computing **38** (1987), 177–184.

79. Wahba, G, Spline interpolation and smoothing on the sphere, SIAM J. Sci. Statist. Comput. **2** (1981), 5–16.

80. Wahba, G., Errata: Spline interpolation and smoothing on the sphere, SIAM J. Sci. Statist. Comput. **3** (1982), 385–386.

81. Wahba, G., Vector splines on the sphere, with application to the estimation of vorticity and divergence from discrete, noisy data, in *Multivariate Approximation Theory II* (W. Schempp and K. Zeller, eds), Birkhäuser (Basel), 1982, 407–429.

82. Wahba, G., Surface fitting with scattered noisy data on Euclidean $d-$space and on the sphere, Rocky Mountain J. Math. **14** (1984), 281–299.

83. Whelan, T., A representation of a C^2 interpolant over triangles, Comput. Aided Geom. Design **3** (1986), 53–66.

84. Zlamal, M, On the finite element method, Numer. Math. **12** (1968), 394–409.