

## A MULTIREOLUTION TENSOR SPLINE METHOD FOR FITTING FUNCTIONS ON THE SPHERE\*

TOM LYCHE<sup>†</sup> AND LARRY L. SCHUMAKER<sup>‡</sup>

**Abstract.** We present the details of a multiresolution method we proposed at the Taormina Wavelet Conference in 1993 (see “L-spline wavelets” in *Wavelets: Theory, Algorithms, and Applications*, C. Chui, L. Montefusco, and L. Puccio, eds., Academic Press, New York, pp. 197–212) which is suitable for fitting functions or data on the sphere. The method is based on tensor products of polynomial splines and trigonometric splines and can be adapted to produce surfaces that are either tangent plane continuous or almost tangent plane continuous. The result is a convenient compression algorithm for dealing with large amounts of data on the sphere. We give full details of a computer implementation that is highly efficient with respect to both storage and computational cost. We also demonstrate the performance of the method on several test examples.

**Key words.** multiresolution, spherical data compression, tensor splines

**AMS subject classifications.** 41A15, 42A10, 41A63, 65D07, 65D17, 65T60

**PII.** S1064827598344388

**1. Introduction.** In many applications (e.g., in geophysics, meteorology, medical modeling, etc.), one needs to construct smooth functions defined on the unit sphere  $S$  which approximate or interpolate data. As shown in [13], one way to do this is to work with tensor-product functions of the form

$$(1.1) \quad f(\theta, \phi) := \sum_{i=1}^m \sum_{j=1}^{\tilde{m}} c_{ij} \varphi_i(\theta) \tilde{\varphi}_j(\phi)$$

defined on the rectangle

$$H := \{(\theta, \phi) : -\pi/2 \leq \theta \leq \pi/2 \text{ and } 0 \leq \phi \leq 2\pi\},$$

where the  $\varphi_i$  are quadratic polynomial B-splines on  $[-\pi/2, \pi/2]$ , and the  $\tilde{\varphi}_j$  are periodic trigonometric splines of order 3 on  $[0, 2\pi]$ . With some care in the choice of the coefficients (see section 2), the associated *surface*

$$\mathcal{S}_f := \{f(\theta, \phi)\mathbf{v}(\theta, \phi) : (\theta, \phi) \in H\}$$

with  $\mathbf{v}(\theta, \phi) = [\cos(\theta) \cos(\phi), \cos(\theta) \sin(\phi), \sin(\theta)]^T$  will be tangent plane continuous.

In practice we often encounter very large data sets, and to get good fits using tensor product splines (1.1), a large number of knots is required, resulting in many basis functions and many coefficients. Since two spline spaces are nested if their

---

\*Received by the editors September 10, 1998; accepted for publication (in revised form) January 31, 2000; published electronically August 24, 2000.

<http://www.siam.org/journals/sisc/22-2/34438.html>

<sup>†</sup>Institutt for Informatikk, University of Oslo, P.O. Box 1080, Blindern 0316 Oslo, Norway (tom@ifi.uio.no). The research of this author was supported by NATO grant CRG951291. Part of the work was completed during a stay at Institut National des Sciences Appliquées and Laboratoire Approximation et Optimisation of Université Paul Sabatier, Toulouse, France.

<sup>‡</sup>Department of Mathematics, Vanderbilt University, Nashville, TN 37240 (s@mars.cas.vanderbilt.edu). The research of this author was supported by the National Science Foundation under grant DMS-9803340, by NATO grant CRG951291, and by the Army Research Office under grant DAAD 19-99-1-0160.

knot sequences are nested, one way to achieve a more efficient fit without sacrificing quality is to look for a multiresolution representation of (1.1), i.e., to recursively decompose it into splines on coarser meshes and corresponding correction (wavelet) terms. Then compression can be achieved in the standard way by thresholding out small coefficients; see [1], for example.

The paper is organized as follows. In section 2 we introduce notation and give details on the tensor product splines to be used here. In section 3 we describe the general decomposition and reconstruction algorithm in matrix form, while in section 4 we present a tensor version of the algorithms. The required matrices corresponding to the polynomial and trigonometric spline spaces, respectively, are derived in sections 5 and 6. Section 7 is devoted to details of implementing the algorithm, while in section 8 we discuss the case of almost tangent plane continuity. In section 9 we present test examples, and in section 10, several concluding remarks.

The method of this paper was presented at the Taormina Wavelet Conference in October 1993 but was not described in detail in the corresponding proceedings paper [8]. A very similar method based on exponential splines was developed independently in [2], [15], but with no implementation details. Alternative wavelet methods for the sphere can be found in [4], [5], [6], [10], [11]. The method in [4] uses discretizations of certain continuous wavelet transforms based on singular integral operators, while the method in [10] uses tensor functions based on polynomials and trigonometric polynomials. The methods in [6], [11] are based on triangulations. For a complete review of the various methods, see [5].

**2. Tangent plane continuous tensor splines.** Let  $\varphi_1, \dots, \varphi_m$  be the standard quadratic B-splines associated with the knot sequence

$$-\pi/2 = x_1 = x_2 = x_3 < x_4 < \dots < x_m < x_{m+1} = x_{m+2} = x_{m+3} = \pi/2.$$

Recall that  $\varphi_i$  is supported on the interval  $[x_i, x_{i+3}]$  and that the B-splines form a partition of unity on  $[-\pi/2, \pi/2]$ . Let  $T_1, \dots, T_{\tilde{m}}$  be the classical trigonometric B-splines of order 3 defined on the knot sequence  $\tilde{x}_1, \dots, \tilde{x}_{\tilde{m}+3}$ , where

$$0 = \tilde{x}_1 < \tilde{x}_2 < \dots < \tilde{x}_{\tilde{m}} < 2\pi$$

and  $\tilde{x}_{\tilde{m}+i} := \tilde{x}_i + 2\pi, i = 1, \dots, 3$ ; see section 6. Recall that  $T_j$  is supported on the interval  $[\tilde{x}_j, \tilde{x}_{j+3}]$ . Let

$$\tilde{\varphi}_j(x) = \begin{cases} T_j(x), & j = 1, \dots, \tilde{m} - 2, \\ T_j(x) + T_j(x - 2\pi), & j = \tilde{m} - 1, \tilde{m}, \end{cases}$$

be the associated  $2\pi$ -periodic trigonometric B-splines; see [12]. These splines can be normalized so that for  $\phi \in [0, 2\pi]$

$$(2.1) \quad 1 = \sum_{j=1}^{\tilde{m}} \cos\left(\frac{\tilde{x}_{j+2} - \tilde{x}_{j+1}}{2}\right) \tilde{\varphi}_j(\phi),$$

$$(2.2) \quad \cos(\phi) = \sum_{j=1}^{\tilde{m}} \cos\left(\frac{\tilde{x}_{j+1} + \tilde{x}_{j+2}}{2}\right) \tilde{\varphi}_j(\phi),$$

$$(2.3) \quad \sin(\phi) = \sum_{j=1}^{\tilde{m}} \sin\left(\frac{\tilde{x}_{j+1} + \tilde{x}_{j+2}}{2}\right) \tilde{\varphi}_j(\phi).$$

Since the boundaries of  $H$  corresponding to  $\theta = \pm\pi/2$  map to the north and south poles, respectively, a function  $f$  of the form (1.1) will be well defined on  $S$  if and only if

$$(2.4) \quad c_{1,j} = f_S \cos\left(\frac{\tilde{x}_{j+2} - \tilde{x}_{j+1}}{2}\right), \quad j = 1, \dots, \tilde{m},$$

and

$$(2.5) \quad c_{m,j} = f_N \cos\left(\frac{\tilde{x}_{j+2} - \tilde{x}_{j+1}}{2}\right), \quad j = 1, \dots, \tilde{m},$$

where  $f_S$  and  $f_N$  are the values at the poles. Now since  $f$  is  $2\pi$ -periodic in the  $\phi$  variable and is  $C^1$  continuous in both variables, we might expect that the corresponding surface  $\mathcal{S}_f$  has a continuous tangent plane at nonpolar points. However, since we are working in a parametric setting, more is needed. The following theorem shows that under mild conditions on  $f$  which are normally satisfied in practice, we do get tangent plane continuity except at the poles.

**THEOREM 2.1.** *Suppose  $f$  is a spline as in (1.1) that satisfies the conditions (2.4) and (2.5), and that in addition  $f(\theta, \phi) > 0$  for all  $(\theta, \phi) \in H$ . Then the corresponding surface  $\mathcal{S}_f$  is tangent plane continuous at all nonpolar points of  $S$ .*

*Proof.* Since  $f$  is a  $C^1$  spline, the partial derivatives  $f_\theta$  and  $f_\phi$  are continuous on  $H$ . Now  $t_1(\theta, \phi) := D_\theta[f(\theta, \phi)\mathbf{v}(\theta, \phi)]$  and  $t_2(\theta, \phi) := D_\phi[f(\theta, \phi)\mathbf{v}(\theta, \phi)]$  are two tangents to the surface  $\mathcal{S}_f$  at the point  $f(\theta, \phi)\mathbf{v}(\theta, \phi)$ . The normal vector to the surface at this point is given by the cross product  $\mathbf{n} := t_1 \times t_2$ . By the hypotheses,  $\mathbf{n}$  is continuous, and thus to ensure a continuous tangent plane, it suffices to show that  $\mathbf{n}$  has positive length (which insures that the surface does not have singular points or cusps). Using Mathematica, it is easy to see that

$$|\mathbf{n}(\theta, \phi)|^2 = f(\theta, \phi)^2 [\cos(\theta)^2 f(\theta, \phi)^2 + f_\phi(\theta, \phi)^2 + \cos(\theta)^2 f_\theta(\theta, \phi)^2],$$

which is clearly positive for all values of  $(\theta, \phi) \in H$  with  $\theta \neq \pm\pi/2$ . □

With some additional side conditions on the coefficients of  $f$ , we can make the surface  $\mathcal{S}_f$  also be tangent plane continuous at the poles. The required conditions (cf. [3], [13]) are that

$$c_{2,j} = c_{1,j} + \frac{(x_4 - x_3)}{2} \left[ A_S \cos\left(\frac{\tilde{x}_{j+1} + \tilde{x}_{j+2}}{2}\right) + B_S \sin\left(\frac{\tilde{x}_{j+1} + \tilde{x}_{j+2}}{2}\right) \right]$$

and

$$c_{m-1,j} = c_{m,j} - \frac{(x_{m+1} - x_m)}{2} \left[ A_N \cos\left(\frac{\tilde{x}_{j+1} + \tilde{x}_{j+2}}{2}\right) + B_N \sin\left(\frac{\tilde{x}_{j+1} + \tilde{x}_{j+2}}{2}\right) \right]$$

for  $j = 1, \dots, \tilde{m}$ , where  $A_S, B_S, A_N,$  and  $B_N$  are constants.

**3. Basic decomposition and reconstruction formulae.** Suppose  $\mathcal{V}_0, \mathcal{V}_1, \dots$  is a nested sequence of finite-dimensional linear subspaces of an inner-product space  $X$ , i.e.,

$$\mathcal{V}_0 \subset \mathcal{V}_1 \subset \dots \subset \mathcal{V}_k \subset \dots$$

Let

$$\mathcal{V}_k = \mathcal{V}_{k-1} \oplus \mathcal{W}_{k-1}$$

be the corresponding orthogonal decompositions.

For our application, it is convenient to express decomposition and reconstruction in matrix form, as is done, e.g., in [14]. Let  $\varphi_{k,1}, \dots, \varphi_{k,m_k}$  be a basis for  $\mathcal{V}_k$ , and let  $\psi_{k-1,1}, \dots, \psi_{k-1,n_{k-1}}$  be a basis for  $\mathcal{W}_{k-1}$ , where  $n_{k-1} = m_k - m_{k-1}$ . Then by the nestedness, there exists an  $m_k \times m_{k-1}$  matrix  $P_k$  such that

$$(3.1) \quad \varphi_{k-1}^T = \varphi_k^T P_k,$$

where

$$(3.2) \quad \varphi_k = (\varphi_{k,1}, \dots, \varphi_{k,m_k})^T.$$

The equation (3.1) is the usual *refinement relation*. Similarly, there exists an  $m_k \times n_{k-1}$  matrix  $Q_k$  such that

$$(3.3) \quad \psi_{k-1}^T = \varphi_k^T Q_k,$$

where

$$(3.4) \quad \psi_{k-1} = (\psi_{k-1,1}, \dots, \psi_{k-1,n_{k-1}})^T.$$

Let

$$G_k = (\langle \varphi_{k,i}, \varphi_{k,j} \rangle),$$

$$H_{k-1} = (\langle \psi_{k-1,i}, \psi_{k-1,j} \rangle)$$

be the Gram matrices of size  $m_k \times m_k$  and  $n_{k-1} \times n_{k-1}$ , respectively. It is easy to see that

$$(3.5) \quad H_{k-1} = Q_k^T G_k Q_k.$$

Clearly, the Gram matrices  $G_k$  and  $H_{k-1}$  are symmetric. The linear independence of the basis functions  $\phi_{k,i}$  and of  $\psi_{k,i}$  implies that both  $G_k$  and  $H_{k-1}$  are positive definite and thus nonsingular.

The following lemma shows how to decompose and reconstruct functions in  $\mathcal{V}_k$  in terms of functions in  $\mathcal{V}_{k-1}$  and  $\mathcal{W}_{k-1}$ .

LEMMA 3.1. *Let  $f_k = \varphi_k^T a_k$  be a function in  $\mathcal{V}_k$  associated with a coefficient vector  $a_k \in \mathbb{R}^{m_k}$ , and let*

$$(3.6) \quad f_k = f_{k-1} + g_{k-1}$$

be its orthogonal decomposition, where

$$f_{k-1} = \varphi_{k-1}^T a_{k-1} \in \mathcal{V}_{k-1}, \quad g_{k-1} = \psi_{k-1}^T b_{k-1} \in \mathcal{W}_{k-1}.$$

Then

$$a_{k-1} = G_{k-1}^{-1} P_k^T G_k a_k,$$

$$b_{k-1} = H_{k-1}^{-1} Q_k^T G_k a_k.$$

Moreover,

$$(3.7) \quad a_k = P_k a_{k-1} + Q_k b_{k-1}.$$

*Proof.* To find  $a_{k-1}$ , we take the inner-product of both sides of (3.6) with  $\varphi_{k-1,i}$  for  $i = 1, \dots, m_{k-1}$ . Using the refinement relation (3.1) and the orthogonality of the  $\varphi_{k-1,i}$  with  $\psi_{k-1,j}$ , we get

$$P_k^T G_k a_k = G_{k-1} a_{k-1},$$

which gives the formula for  $a_{k-1}$ . If we instead take the inner-products with  $\psi_{k-1}$ , we get the formula for  $b_{k-1}$ . In view of the linear independence of the functions  $\varphi_{k,1}, \dots, \varphi_{k,m_k}$ , the reconstruction formula (3.7) follows immediately from (3.6) and the refinement relations.  $\square$

Throughout the remainder of this paper we work with inner-products of the form  $\langle f, g \rangle = \int_a^b fg$ . In this case the function  $f_{k-1}$  in the representation (3.6) is nothing more than the best approximation of  $f_k$  from  $\mathcal{V}_{k-1}$  in the  $L_2$  norm on  $[a, b]$ .

**4. Tensor-product decomposition and reconstruction.** In this section we discuss decomposition and reconstruction of functions in tensor product spaces  $\mathcal{V}_k \times \tilde{\mathcal{V}}_\ell$ , where the  $\mathcal{V}_k$  are as in the previous section and where  $\tilde{\mathcal{V}}_\ell$  are similar subspaces of an inner-product space  $\tilde{X}$ . In particular, suppose

$$\tilde{\mathcal{V}}_0 \subset \tilde{\mathcal{V}}_1 \subset \dots \subset \tilde{\mathcal{V}}_\ell \subset \dots$$

and that

$$\tilde{\mathcal{V}}_\ell = \tilde{\mathcal{V}}_{\ell-1} \oplus \tilde{\mathcal{W}}_{\ell-1}.$$

Let  $P_k, Q_k, G_k$ , and  $H_k$  be as in the previous section, and let  $\tilde{P}_\ell, \tilde{Q}_\ell, \tilde{G}_\ell$ , and  $\tilde{H}_\ell$  be the analogous matrices associated with the spaces  $\tilde{\mathcal{V}}_\ell$ .

**THEOREM 4.1.** *Let  $f_{k,\ell} = \varphi_k^T A_{k,\ell} \tilde{\varphi}_\ell$  be a function in  $\mathcal{V}_k \times \tilde{\mathcal{V}}_\ell$  associated with a coefficient matrix  $A_{k,\ell}$ . Then  $f_{k,\ell}$  has the orthogonal decomposition*

$$(4.1) \quad f_{k,\ell} = f_{k-1,\ell-1} + g_{k-1,\ell-1}^{(1)} + g_{k-1,\ell-1}^{(2)} + g_{k-1,\ell-1}^{(3)},$$

with

$$\begin{aligned} f_{k-1,\ell-1} &= \varphi_{k-1}^T A_{k-1,\ell-1} \tilde{\varphi}_{\ell-1} \in \mathcal{V}_{k-1} \times \tilde{\mathcal{V}}_{\ell-1}, \\ g_{k-1,\ell-1}^{(1)} &= \varphi_{k-1}^T B_{k-1,\ell-1}^{(1)} \tilde{\psi}_{\ell-1} \in \mathcal{V}_{k-1} \times \tilde{\mathcal{W}}_{\ell-1}, \\ g_{k-1,\ell-1}^{(2)} &= \psi_{k-1}^T B_{k-1,\ell-1}^{(2)} \tilde{\varphi}_{\ell-1} \in \mathcal{W}_{k-1} \times \tilde{\mathcal{V}}_{\ell-1}, \\ g_{k-1,\ell-1}^{(3)} &= \psi_{k-1}^T B_{k-1,\ell-1}^{(3)} \tilde{\psi}_{\ell-1} \in \mathcal{W}_{k-1} \times \tilde{\mathcal{W}}_{\ell-1}, \end{aligned}$$

where the matrices  $A_{k-1,\ell-1}, B_{k-1,\ell-1}^{(1)}, B_{k-1,\ell-1}^{(2)}$ , and  $B_{k-1,\ell-1}^{(3)}$  are computed from the system of equations

$$(4.2) \quad \begin{aligned} G_{k-1} A_{k-1,\ell-1} \tilde{G}_{\ell-1} &= P_k^T D_{k,\ell} \tilde{P}_\ell, \\ G_{k-1} B_{k-1,\ell-1}^{(1)} \tilde{H}_{\ell-1} &= P_k^T D_{k,\ell} \tilde{Q}_\ell, \\ H_{k-1} B_{k-1,\ell-1}^{(2)} \tilde{G}_{\ell-1} &= Q_k^T D_{k,\ell} \tilde{P}_\ell, \\ H_{k-1} B_{k-1,\ell-1}^{(3)} \tilde{H}_{\ell-1} &= Q_k^T D_{k,\ell} \tilde{Q}_\ell \end{aligned}$$

with

$$D_{k,\ell} := G_k A_{k,\ell} \tilde{G}_\ell.$$

Moreover,

$$(4.3) \quad A_{k,\ell} = P_k A_{k-1,\ell-1} \tilde{P}_\ell^T + P_k B_{k-1,\ell-1}^{(1)} \tilde{Q}_\ell^T + Q_k B_{k-1,\ell-1}^{(2)} \tilde{P}_\ell^T + Q_k B_{k-1,\ell-1}^{(3)} \tilde{Q}_\ell^T.$$

*Proof.* To find the formula for  $A_{k-1,\ell-1}$ , we take the inner-product of both sides of (4.1) with  $\varphi_{k-1,i}$  for  $i = 1, \dots, m_{k-1}$  and with  $\tilde{\varphi}_{\ell-1,j}$  for  $j = 1, \dots, \tilde{m}_{\ell-1}$ . The formulae for the  $B_{k-1,\ell-1}^{(i)}$  are obtained in a similar way. The reconstruction formula (4.3) follows directly from (4.1) after inserting the refinement relations and using the linear independence of the components of the vectors  $\varphi_k$  and in  $\tilde{\varphi}_\ell$ .  $\square$

Note that computing the matrices  $A_{k-1,\ell-1}$  and  $B_{k-1,\ell-1}^{(i)}$  in a decomposition step can be done quite efficiently since several matrix products occur more than once, and we need only solve linear systems of equations involving the four matrices  $G_{k-1}$ ,  $H_{k-1}$ ,  $\tilde{G}_{\ell-1}$ , and  $\tilde{H}_{\ell-1}$ . As we shall see below, in our setting the first two of these are banded matrices, and the second two are periodic versions of banded matrices. All of them can be precomputed and stored in compact form.

**5. The decomposition matrices for the polynomial splines.** In this section we construct the matrices  $P_k$ ,  $Q_k$ , and  $G_k$  needed for the decomposition and reconstruction of quadratic polynomial splines on the closed interval  $[-\pi/2, \pi/2]$ . Consider the nested sequence of knots

$$-\pi/2 = x_1^k = x_2^k = x_3^k < x_4^k < \dots < x_{m_k}^k < x_{m_k+1}^k = x_{m_k+2}^k = x_{m_k+3}^k = \pi/2,$$

where

$$(5.1) \quad x_i^k = -\pi/2 + (i - 3)h_k, \quad i = 4, \dots, m_k,$$

with  $h_k = \pi/(m_k - 2)$  and  $m_k = 3 \cdot 2^k + 2$ . Let  $\{\varphi_{k,i}\}_{i=1}^{m_k}$  be the associated quadratic B-splines with supports on the intervals  $[x_i^k, x_{i+3}^k]$ ,  $i = 1, \dots, m_k$ . We assume they are normalized so that

$$(5.2) \quad \sum_{i=1}^{m_k} \varphi_{k,i} \equiv 1.$$

We choose this normalization since in performing compression we want to control the error in the uniform norm. For each  $k$ , the span  $\mathcal{V}_k$  of  $\varphi_{k,1}, \dots, \varphi_{k,m_k}$  is the  $m_k$  dimensional linear space of  $C^1$  quadratic splines with knots at the  $x_i^k$ . These spaces are clearly nested. In addition to the well-known *refinement relations*

$$(5.3) \quad \varphi_{k-1,i} = \frac{(\varphi_{k,2i-3} + 3\varphi_{k,2i-2} + 3\varphi_{k,2i-1} + \varphi_{k,2i})}{4}, \quad i = 3, \dots, m_{k-1} - 2,$$

a simple computation shows that

$$(5.4) \quad \begin{aligned} \varphi_{k-1,1} &= (4\varphi_{k,1} + 2\varphi_{k,2})/4, \\ \varphi_{k-1,2} &= (2\varphi_{k,2} + 3\varphi_{k,3} + \varphi_{k,4})/4, \\ \varphi_{k-1,m_{k-1}-1} &= (2\varphi_{k,m_k-1} + 3\varphi_{k,m_k-2} + \varphi_{k,m_k-3})/4, \\ \varphi_{k-1,m_{k-1}} &= (4\varphi_{k,m_k} + 2\varphi_{k,m_k-1})/4. \end{aligned}$$

Equations (5.3), (5.4) provide the entries for the matrix  $P_k$ . In particular, the first two and last two columns are determined by (5.4), while for any  $3 \leq i \leq m_{k-1} - 2$ , the  $i$ th column of  $P_k$  contains all zeros except for the four rows  $2i - 3, \dots, 2i$  which contain the numbers  $1/4, 3/4, 3/4$ , and  $1/4$ . For example,

$$P_1 = \frac{1}{4} \begin{pmatrix} 4 & 0 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 & 0 \\ 0 & 3 & 1 & 0 & 0 \\ 0 & 1 & 3 & 0 & 0 \\ 0 & 0 & 3 & 1 & 0 \\ 0 & 0 & 1 & 3 & 0 \\ 0 & 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 0 & 4 \end{pmatrix}.$$

In general,  $P_k$  has at most two nonzero entries in each row and at most four nonzero entries in each column.

In order to compute the entries of the matrices  $Q_k$ , we need explicit bases for the wavelet spaces  $\mathcal{W}_{k-1}$  on the interval  $[-\pi/2, \pi/2]$ . Although their construction is well known, for completeness we give details. Let  $n_{k-1} = m_k - m_{k-1} = 3 \cdot 2^{k-1}$ .

THEOREM 5.1. *Given  $k \geq 1$ , let*

$$\begin{aligned} \psi_{k-1,1} &:= \sum_{j=1}^6 q_{j,1} \varphi_{k,j}, \\ \psi_{k-1,n_{k-1}} &:= \sum_{j=1}^6 q_{j,1} \varphi_{k,m_k-j+1}, \end{aligned}$$

and for  $k \geq 2$ , let

$$\begin{aligned} \psi_{k-1,2} &:= \sum_{j=2}^8 q_{j,2} \varphi_{k,j}, \\ \psi_{k-1,n_{k-1}-1} &:= \sum_{j=2}^8 q_{j,2} \varphi_{k,m_k-j+1}, \end{aligned}$$

where

$$(5.5) \quad \begin{pmatrix} q_{1,1} \\ q_{2,1} \\ q_{3,1} \\ q_{4,1} \\ q_{5,1} \\ q_{6,1} \end{pmatrix} = \frac{1}{14} \begin{pmatrix} -6864 \\ 8346 \\ -4967 \\ 2083 \\ -406 \\ 14 \end{pmatrix}, \quad \begin{pmatrix} q_{2,2} \\ q_{3,2} \\ q_{4,2} \\ q_{5,2} \\ q_{6,2} \\ q_{7,2} \\ q_{8,2} \end{pmatrix} = \frac{1}{11} \begin{pmatrix} 780 \\ -1949 \\ 3481 \\ -3362 \\ 1618 \\ -319 \\ 11 \end{pmatrix}.$$

In addition, for  $k \geq 2$ , let

$$(5.6) \quad \begin{aligned} \psi_{k-1,i} &:= -\varphi_{k,2i-3} + 29\varphi_{k,2i-2} - 147\varphi_{k,2i-1} + 303\varphi_{k,2i} - 303\varphi_{k,2i+1} \\ &\quad + 147\varphi_{k,2i+2} - 29\varphi_{k,2i+3} + \varphi_{k,2i+4} \end{aligned}$$

for  $i = 3, \dots, n_{k-1} - 2$ , and set

$$\psi_{0,2} := -\varphi_{1,2} + \frac{5}{2}\varphi_{1,3} - \frac{9}{2}\varphi_{1,4} + \frac{9}{2}\varphi_{1,5} - \frac{5}{2}\varphi_{1,6} + \varphi_{1,7}.$$

Then  $\psi_{k-1,1}, \dots, \psi_{k-1,n_{k-1}}$  form a basis for  $\mathcal{W}_{k-1}$ .

*Proof.* The wavelets in (5.6) are just the well-known quadratic spline wavelets; see, e.g., [1]. The coefficients of the boundary wavelets  $\psi_{k-1,1}$ ,  $\psi_{k-1,n_{k-1}}$ ,  $\psi_{k-1,2}$ , and  $\psi_{k-1,n_{k-1}-1}$  can be computed by forcing orthogonality to  $\mathcal{V}_{k-1}$ . In view of (3.3), the wavelets  $\psi_{k-1,1}, \dots, \psi_{k-1,n_{k-1}}$  are linearly independent if and only if the matrix  $Q_k$  is of full rank. This follows since the submatrix of  $Q_k$  obtained by taking rows  $2, 4, \dots, 3 \cdot 2^{k-1}$  and  $3 \cdot 2^{k-1} + 3, 3 \cdot 2^{k-1} + 5, \dots, 3 \cdot 2^k + 1$  can easily be seen to be diagonally dominant. For an alternate proof of linear independence, see Lemma 11 of [7].  $\square$

Using elementary properties of B-splines, it is easy to see that

$$(5.7) \quad \begin{aligned} \psi_{k-1,i}(\pm\pi/2) &= 0, & i &= 2, \dots, n_{k-1} - 1, \\ D_\theta \psi_{k-1,i}(\pm\pi/2) &= 0, & i &= 3, \dots, n_{k-1} - 2. \end{aligned}$$

We now describe the matrices  $Q_k$ . By Theorem 5.1,

$$Q_1 = \begin{pmatrix} q_{1,1} & 0 & 0 \\ q_{2,1} & -1 & 0 \\ q_{3,1} & 5/2 & q_{6,1} \\ q_{4,1} & -9/2 & q_{5,1} \\ q_{5,1} & 9/2 & q_{4,1} \\ q_{6,1} & -5/2 & q_{3,1} \\ 0 & 1 & q_{2,1} \\ 0 & 0 & q_{1,1} \end{pmatrix}$$

and

$$Q_2 = \begin{pmatrix} q_{1,1} & 0 & 0 & 0 & 0 & 0 \\ q_{2,1} & q_{2,2} & 0 & 0 & 0 & 0 \\ q_{3,1} & q_{3,2} & -1 & 0 & 0 & 0 \\ q_{4,1} & q_{4,2} & 29 & 0 & 0 & 0 \\ q_{5,1} & q_{5,2} & -147 & -1 & 0 & 0 \\ q_{6,1} & q_{6,2} & 303 & 29 & 0 & 0 \\ 0 & q_{7,2} & -303 & -147 & q_{8,2} & 0 \\ 0 & q_{8,2} & 147 & 303 & q_{7,2} & 0 \\ 0 & 0 & -29 & -303 & q_{6,2} & q_{6,1} \\ 0 & 0 & 1 & 147 & q_{5,2} & q_{5,1} \\ 0 & 0 & 0 & -29 & q_{4,2} & q_{4,1} \\ 0 & 0 & 0 & 1 & q_{3,2} & q_{3,1} \\ 0 & 0 & 0 & 0 & q_{2,2} & q_{2,1} \\ 0 & 0 & 0 & 0 & 0 & q_{1,1} \end{pmatrix}.$$

For general  $k \geq 2$ , the nonzero elements in the third column of  $Q_k$  are repeated in columns  $4, \dots, n_{k-1} - 2$ , where in each successive column they are shifted down by two rows. The first two and last two columns of  $Q_k$  contain the same nonzero elements as  $Q_2$ . Clearly,  $Q_k$  has at most four nonzero entries in each row and at most eight nonzero entries in each column.

We now describe the Gram matrices  $G_k$ , which in general are symmetric and five-banded. To get  $G_k$ , we start with the matrix with  $66h_k/120$  on the diagonal,  $26h_k/120$  on the first subdiagonal, and  $h_k/120$  on the second subdiagonal. Then replace the entries in the  $3 \times 3$  submatrices in the upper left and lower right corners by

$$UL_k := \frac{h_k}{120} \begin{pmatrix} 24 & 14 & 2 \\ 14 & 40 & 25 \\ 2 & 25 & 66 \end{pmatrix}, \quad LR_k := \frac{h_k}{120} \begin{pmatrix} 66 & 25 & 2 \\ 25 & 40 & 14 \\ 2 & 14 & 24 \end{pmatrix}.$$

For example,

$$G_0 = \frac{h_0}{120} \begin{pmatrix} 24 & 14 & 2 & 0 & 0 \\ 14 & 40 & 25 & 1 & 0 \\ 2 & 25 & 66 & 25 & 2 \\ 0 & 1 & 25 & 40 & 14 \\ 0 & 0 & 2 & 14 & 24 \end{pmatrix}$$

and

$$G_1 = \frac{h_1}{120} \begin{pmatrix} 24 & 14 & 2 & 0 & 0 & 0 & 0 & 0 \\ 14 & 40 & 25 & 1 & 0 & 0 & 0 & 0 \\ 2 & 25 & 66 & 26 & 1 & 0 & 0 & 0 \\ 0 & 1 & 26 & 66 & 26 & 1 & 0 & 0 \\ 0 & 0 & 1 & 26 & 66 & 26 & 1 & 0 \\ 0 & 0 & 0 & 1 & 26 & 66 & 25 & 2 \\ 0 & 0 & 0 & 0 & 1 & 25 & 40 & 14 \\ 0 & 0 & 0 & 0 & 0 & 2 & 14 & 24 \end{pmatrix}.$$

**6. The decomposition matrices for the trigonometric splines.** In this section we present the matrices  $\tilde{P}_\ell$ ,  $\tilde{Q}_\ell$ , and  $\tilde{G}_\ell$  needed for the decomposition and reconstruction of periodic trigonometric splines of order 3 defined on the interval  $[0, 2\pi]$ . Suppose  $\ell \geq 1$  and that

$$(6.1) \quad \tilde{x}_i^\ell = (i - 1)\tilde{h}_\ell, \quad i = 1, \dots, \tilde{m}_\ell + 3,$$

is a nested sequence of knots, where  $\tilde{h}_\ell = 2^{(1-\ell)}\pi/3$  and  $\tilde{m}_\ell = 3 \cdot 2^\ell$ . Let

$$M_{\ell,i}(\phi) := T_{\tilde{h}_\ell}(\phi - \tilde{x}_i^\ell),$$

where

$$(6.2) \quad T_h(\phi) := \begin{cases} \frac{\sin(\phi/2)^2}{\sin(h/2)\sin(h)}, & 0 \leq \phi \leq h, \\ \frac{1}{\cos(h/2)} - \frac{\sin((\phi-h)/2)^2 + \sin((2h-\phi)/2)^2}{\sin(h/2)\sin(h)}, & h \leq \phi \leq 2h, \\ \frac{\sin((3h-\phi)/2)^2}{\sin(h/2)\sin(h)}, & 2h \leq \phi \leq 3h, \\ 0, & \text{otherwise} \end{cases}$$

is the usual trigonometric B-spline of order 3 associated with uniformly spaced knots  $(0, h, 2h, 3h)$ . Set

$$\tilde{\varphi}_{\ell,i}(\phi) = \begin{cases} M_{\ell,i}(\phi), & i = 1, \dots, \tilde{m}_\ell - 2, \\ M_{\ell,i}(\phi) + M_{\ell,i}(\phi - 2\pi), & i = \tilde{m}_\ell - 1, \tilde{m}_\ell. \end{cases}$$

For later use we define  $\tilde{\varphi}_{\ell, \tilde{m}_\ell+i} = \tilde{\varphi}_{\ell, i}$  for  $i = 1, \dots, 6$ . With this normalization, the  $\tilde{\varphi}_{\ell, i}$  satisfy (2.1)–(2.3), and it follows that

$$(6.3) \quad \sum_{i=1}^{\tilde{m}_\ell} \tilde{\varphi}_{\ell, i} \equiv \frac{1}{\cos(\tilde{h}_\ell)} \leq 2$$

for  $\ell \geq 1$ .

The span  $\tilde{\mathcal{V}}_\ell$  of  $\tilde{\varphi}_{\ell, 1}, \dots, \tilde{\varphi}_{\ell, \tilde{m}_\ell}$  is the space of periodic trigonometric splines of order 3. Clearly, these spaces are nested, and in fact we have the following *refinement relation*.

**THEOREM 6.1.** *For all  $\ell \geq 1$  and  $1 \leq i \leq \tilde{m}_{\ell-1}$ ,*

$$(6.4) \quad \tilde{\varphi}_{\ell-1, i} = u(\tilde{h}_\ell)\tilde{\varphi}_{\ell, 2i-1} + v(\tilde{h}_\ell)\tilde{\varphi}_{\ell, 2i} + v(\tilde{h}_\ell)\tilde{\varphi}_{\ell, 2i+1} + u(\tilde{h}_\ell)\tilde{\varphi}_{\ell, 2i+2},$$

where

$$u(h) := \frac{1}{4 \cos(h/2) \cos(h)}, \quad v(h) := \frac{\cos(h/2)}{\cos(h)} - u(h).$$

*Proof.* By nestedness and the nature of the support of  $T_h$ ,

$$T_{2h}(\phi) = u(h)T_h(\phi) + v(h)T_h(\phi - h) + w(h)T_h(\phi - 2h) + z(h)T_h(\phi - 3h)$$

for some numbers  $u, v, w, z$ . By symmetry, it is enough to compute  $u$  and  $v$ . To find  $u$ , we note that on  $[0, h]$ ,

$$T_{2h}(\phi) = u(h)T_h(\phi).$$

Then using (6.2) we can solve for  $u$ . To find  $v$  we note that

$$T_{2h}(2h) = u(h)T_h(2h) + v(h)T_h(h),$$

and then solve for  $v$  using (6.2). □

Theorem 6.1 can now be used to find the entries in the matrix  $\tilde{P}_\ell$  needed in section 2. In particular, each column has exactly the four nonzero elements  $u(\tilde{h}_\ell), v(\tilde{h}_\ell), v(\tilde{h}_\ell), u(\tilde{h}_\ell)$ , starting in the first row in column 1, and shifted down by two rows each time we move one column to the right (where in the last column the last two elements are moved to the top of the column). For example,

$$\tilde{P}_1 := \begin{pmatrix} u(\tilde{h}_1) & 0 & v(\tilde{h}_1) \\ v(\tilde{h}_1) & 0 & u(\tilde{h}_1) \\ v(\tilde{h}_1) & u(\tilde{h}_1) & 0 \\ u(\tilde{h}_1) & v(\tilde{h}_1) & 0 \\ 0 & v(\tilde{h}_1) & u(\tilde{h}_1) \\ 0 & u(\tilde{h}_1) & v(\tilde{h}_1) \end{pmatrix}.$$

Next we describe a basis for the *wavelet space*  $\tilde{\mathcal{W}}_{\ell-1}$  on  $[0, 2\pi]$  which has dimension  $\tilde{n}_{\ell-1} = 3 \cdot 2^{\ell-1}$  for  $\ell \geq 1$ .

**THEOREM 6.2.** *Given  $\ell \geq 1$ , let*

$$(6.5) \quad \tilde{\psi}_{\ell-1, i} = \sum_{j=0}^7 \tilde{q}_j(\tilde{h}_\ell)\tilde{\varphi}_{\ell, 2i+j-1}, \quad i = 1, \dots, \tilde{n}_{\ell-1},$$

where

$$\begin{aligned} \tilde{q}_0(h) &= 1, \\ \tilde{q}_1(h) &= \frac{-h + 5h \cos(h) - h \cos(2h) - 3 \sin(h)}{D(h)}, \\ \tilde{q}_2(h) &= \frac{3h - 7h \cos(h) - 5h \cos(2h) + 3 \sin(3h)}{D(h)}, \\ \tilde{q}_3(h) &= \frac{-2h - 7h \cos(h) + 4h \cos(2h) - 4h \cos(3h) + 3 \sin(3h)}{D(h)}, \end{aligned}$$

and

$$\tilde{q}_{7-j}(h) = -\tilde{q}_j(h), \quad j = 0, \dots, 3,$$

with

$$D(h) := 2h + h \cos(h) - 3 \sin(h).$$

Then  $\tilde{\psi}_{\ell-1,1}, \dots, \tilde{\psi}_{\ell-1,\tilde{n}_{\ell-1}}$  is a basis for the space  $\tilde{\mathcal{W}}_{\ell-1}$ .

*Proof.* To construct wavelets in  $\tilde{\mathcal{W}}_{\ell-1}$ , we apply [8, Theorem 5.1], which gives explicit formulae for the  $\tilde{q}_i$  in terms of inner-products of  $\tilde{\varphi}_{\ell,i}$  with  $\tilde{\varphi}_{\ell-1,j}$ . To show that  $\tilde{\psi}_{\ell-1,1}, \dots, \tilde{\psi}_{\ell-1,\tilde{n}_{\ell-1}}$  are linearly independent, it suffices to show that  $\tilde{Q}_\ell$  is of full rank. To see this, we construct an  $\tilde{n}_{\ell-1} \times \tilde{n}_{\ell-1}$  matrix  $B_\ell$  by moving the last column of  $\tilde{Q}_\ell$  in front of the first column and then selecting rows  $2, 4, \dots, \tilde{m}_\ell$ . We now show that this matrix is strictly diagonally dominant, and thus of full rank.

First, we note that in each row of  $B_\ell$  the element on the diagonal is  $\tilde{q}_3(\tilde{h}_\ell)$  while the sum of the absolute values of the off-diagonal elements is  $|\tilde{q}_1(\tilde{h}_\ell)| + |\tilde{q}_5(\tilde{h}_\ell)| + |\tilde{q}_7(\tilde{h}_\ell)|$ . A simple computation shows that each of the functions  $D(h)$  and  $r_i(h) := \tilde{q}_i(h)D(h)$  has a Taylor expansion which is an alternating series. In particular, using the first two terms of each series, we get

$$\begin{aligned} D(h) &> h^5 [1/60 - h^2/1260] > 0, \\ r_1(h) &< h^5 [-29/60 + 26h^2/315] < 0, \\ r_2(h) &> h^5 [49/20 - 89h^2/105] > 0, \\ r_3(h) &< h^5 [-101/20 + 1009h^2/420] < 0 \end{aligned}$$

for  $0 \leq h \leq \pi/3 = \tilde{h}_1$ . Now it is easy to see that

$$a(h) := [|\tilde{q}_3(h)| - |\tilde{q}_1(h)| - |\tilde{q}_5(h)| - |\tilde{q}_7(h)|]D(h) = -r_3(h) - r_2(h) + r_1(h) - D(h)$$

also has an alternating series expansion, and we get

$$a(h) > h^5 [71/10 - 103h^2/70] > 0$$

for the same range of  $h$ . This shows that  $B_\ell$  is strictly diagonally dominant, and the proof is complete.  $\square$

The formulae for the  $\tilde{q}_i$  in Theorem 6.2 are not appropriate for small values of  $\tilde{h}_{\ell-1}$ . In this case we can use the following Taylor expansions:

$$\begin{aligned} \tilde{q}_0(h) &= 1, \\ \tilde{q}_1(h) &= -29 + \frac{25}{7}h^2 - \frac{103}{588}h^4 + \frac{1255}{271656}h^6 + \dots, \\ \tilde{q}_2(h) &= 147 - \frac{307}{7}h^2 + \frac{3301}{588}h^4 - \frac{545273}{1358280}h^6 + \dots, \\ \tilde{q}_3(h) &= -303 + \frac{908}{7}h^2 - \frac{3131}{147}h^4 + \frac{642583}{339570}h^6 + \dots. \end{aligned}$$

Rather than computing them each time we need them, we can precompute and store the necessary values of  $\tilde{q}_1(\tilde{h}_\ell)$ ,  $\tilde{q}_2(\tilde{h}_\ell)$ , and  $\tilde{q}_3(\tilde{h}_\ell)$  for various levels  $\ell$ ; see Table 1. We can now describe the matrix  $\tilde{Q}_\ell$  needed in section 2 for decomposing and reconstructing with trigonometric splines. For  $\ell = 1$  we have

$$\tilde{Q}_1 = \begin{pmatrix} \tilde{q}_0 + \tilde{q}_6 & \tilde{q}_4 & \tilde{q}_2 \\ \tilde{q}_1 + \tilde{q}_7 & \tilde{q}_5 & \tilde{q}_3 \\ \tilde{q}_2 & \tilde{q}_0 + \tilde{q}_6 & \tilde{q}_4 \\ \tilde{q}_3 & \tilde{q}_1 + \tilde{q}_7 & \tilde{q}_5 \\ \tilde{q}_4 & \tilde{q}_2 & \tilde{q}_0 + \tilde{q}_6 \\ \tilde{q}_5 & \tilde{q}_3 & \tilde{q}_1 + \tilde{q}_7 \end{pmatrix},$$

where all  $\tilde{q}_i$  are evaluated at  $\tilde{h}_1$ . For  $\ell \geq 2$ , each column of  $\tilde{Q}_\ell$  contains the eight entries  $\tilde{q}_0, \tilde{q}_1, \tilde{q}_2, \tilde{q}_3, \tilde{q}_4, \tilde{q}_5, \tilde{q}_6, \tilde{q}_7$ , evaluated at  $\tilde{h}_\ell$ . In particular, these entries start in row 1 in column 1 and are shifted down by two each time we move one column to the right (where in the last three columns, entries falling below the last row are moved to the top). Clearly,  $\tilde{Q}_\ell$  has exactly four nonzero entries in each row. For example,

$$\tilde{Q}_2 = \begin{pmatrix} \tilde{q}_0 & 0 & 0 & \tilde{q}_6 & \tilde{q}_4 & \tilde{q}_2 \\ \tilde{q}_1 & 0 & 0 & \tilde{q}_7 & \tilde{q}_5 & \tilde{q}_3 \\ \tilde{q}_2 & \tilde{q}_0 & 0 & 0 & \tilde{q}_6 & \tilde{q}_4 \\ \tilde{q}_3 & \tilde{q}_1 & 0 & 0 & \tilde{q}_7 & \tilde{q}_5 \\ \tilde{q}_4 & \tilde{q}_2 & \tilde{q}_0 & 0 & 0 & \tilde{q}_6 \\ \tilde{q}_5 & \tilde{q}_3 & \tilde{q}_1 & 0 & 0 & \tilde{q}_7 \\ \tilde{q}_6 & \tilde{q}_4 & \tilde{q}_2 & \tilde{q}_0 & 0 & 0 \\ \tilde{q}_7 & \tilde{q}_5 & \tilde{q}_3 & \tilde{q}_1 & 0 & 0 \\ 0 & \tilde{q}_6 & \tilde{q}_4 & \tilde{q}_2 & \tilde{q}_0 & 0 \\ 0 & \tilde{q}_7 & \tilde{q}_5 & \tilde{q}_3 & \tilde{q}_1 & 0 \\ 0 & 0 & \tilde{q}_6 & \tilde{q}_4 & \tilde{q}_2 & \tilde{q}_0 \\ 0 & 0 & \tilde{q}_7 & \tilde{q}_5 & \tilde{q}_3 & \tilde{q}_1 \end{pmatrix},$$

where all  $\tilde{q}_i$  are evaluated at  $\tilde{h}_2$ . Table 1 gives the values of  $\tilde{q}_1(\tilde{h}_k)$ ,  $\tilde{q}_2(\tilde{h}_\ell)$ , and  $\tilde{q}_3(\tilde{h}_\ell)$  for  $\ell = 1, \dots, 12$  needed for the  $\tilde{Q}_\ell$ .

Finally, we describe the Gram matrices.

**THEOREM 6.3.** *For  $\ell \geq 1$ , the  $3 \cdot 2^\ell \times 3 \cdot 2^\ell$  Gram matrix  $\tilde{G}_\ell$  associated with the*

TABLE 1  
 Trigonometric spline wavelet coefficients for various  $\ell$ .

$\ell$	$\tilde{q}_1(\tilde{h}_\ell)$	$\tilde{q}_2(\tilde{h}_\ell)$	$\tilde{q}_3(\tilde{h}_\ell)$
1	-25.288158402784911895	105.15263361113964758	-184.01710881949438326
2	-28.033943811096385992	135.39009725820806026	-269.00057271914225083
3	-28.756039535012008061	144.02032194736046124	-294.20897139729685258
4	-28.938855942719881876	146.25016593522229565	-300.78362470238002386
5	-28.984704348047217637	146.81223291013457079	-302.44473588115810747
6	-28.996175484404513950	146.95303891951439472	-302.86111072242944246
7	-28.999043833434183593	146.98825852278080426	-302.96527310098241998
8	-28.999760956004299506	146.99706455524617238	-302.99131798899351388
9	-28.999940238853933503	146.99926613409589417	-302.99782947935722381
10	-28.999985059704287025	146.99981653322924416	-302.99945736872110255
11	-28.999996264925496984	146.99995413328889043	-302.99986434211038783
12	-28.999999066231338323	146.99998853332107132	-302.99996608552322897

$\tilde{\varphi}_{\ell,i}$  is given by

$$\tilde{G}_\ell := \begin{pmatrix} I_{00} & I_{01} & I_{02} & 0 & \cdots & 0 & I_{02} & I_{01} \\ I_{01} & I_{00} & I_{01} & I_{02} & 0 & \cdots & 0 & I_{02} \\ I_{02} & I_{01} & I_{00} & I_{01} & I_{02} & \cdots & 0 & 0 \\ & \ddots & & \ddots & & \ddots & & \\ & & \ddots & & \ddots & & \ddots & \\ 0 & 0 & \cdots & I_{02} & I_{01} & I_{00} & I_{01} & I_{02} \\ I_{02} & 0 & \cdots & 0 & I_{02} & I_{01} & I_{00} & I_{01} \\ I_{01} & I_{02} & 0 & \cdots & 0 & I_{02} & I_{01} & I_{00} \end{pmatrix},$$

where

$$I_{02} := \int_{\tilde{x}_i^\ell}^{\tilde{x}_{i+1}^\ell} \tilde{\varphi}_{\ell,i} \tilde{\varphi}_{\ell,i-2} = \gamma_\ell [4\tilde{h}_\ell + 2\tilde{h}_\ell \cos(2\tilde{h}_\ell) - 3 \sin(2\tilde{h}_\ell)],$$

$$I_{01} := \int_{\tilde{x}_i^\ell}^{\tilde{x}_{i+2}^\ell} \tilde{\varphi}_{\ell,i} \tilde{\varphi}_{\ell,i-1} = \gamma_\ell [-4\tilde{h}_\ell - 20\tilde{h}_\ell \cos(2\tilde{h}_\ell) + 6 \sin(2\tilde{h}_\ell) + 3 \sin(4\tilde{h}_\ell)],$$

$$I_{00} := \int_{\tilde{x}_i^\ell}^{\tilde{x}_{i+3}^\ell} \tilde{\varphi}_{\ell,i}^2 = \gamma_\ell [4\tilde{h}_\ell \cos(2\tilde{h}_\ell) + 8\tilde{h}_\ell \cos(4\tilde{h}_\ell) + 24\tilde{h}_\ell - 6 \sin(2\tilde{h}_\ell) - 6 \sin(4\tilde{h}_\ell)],$$

with

$$\gamma_\ell := \frac{1}{64 \sin(\tilde{h}_\ell)^4 \cos(\tilde{h}_\ell)^2}.$$

Moreover,

$$\tilde{G}_0 := \begin{pmatrix} I_{00} & I_{01} + I_{02} & I_{01} + I_{02} \\ I_{01} + I_{02} & I_{00} & I_{01} + I_{02} \\ I_{01} + I_{02} & I_{01} + I_{02} & I_{00} \end{pmatrix}.$$

TABLE 2  
*Inner products of trigonometric B-splines for various  $\ell$ .*

$\ell$	$I_{00}/\tilde{h}_\ell$	$I_{01}/\tilde{h}_\ell$	$I_{02}/\tilde{h}_\ell$
0	2.000000000000000000	0.9423311143775626914	0.05766888562243730858
1	0.7173865882718287392	0.29529339212946177894	0.012679980401290518133
2	0.5863256235682111689	0.23350674787359713392	0.009228825204542694601
3	0.5587848830466661676,	0.22072647211850900468	0.008547276418657547047
4	0.5521783423263619826	0.21767257645539869275	0.008386225140326574126
5	0.5505434780490859489	0.21691758424366982979	0.008346519562452568337
6	0.5501358004443718685	0.21672936114999970712	0.008336627601639628844
7	0.5500339457967328606	0.21668233810682990252	0.008334156757445556228
8	0.5500084861795729324	0.21667058439043531220	0.008333539180427623907
9	0.5500021215280431720	0.21666764608909212581	0.008333384794548569195
10	0.5500005303809576733	0.21666691152174074237	0.008333346198602246618
11	0.5500001325951735985	0.21666672788040191760	0.008333336549648380680
12	0.5500000331487892859	0.21666668197009840015	0.008333334137411958859

*Proof.* Using (6.2), the necessary integrals can be computed directly.  $\square$

The formulae in Theorem 6.3 are clearly not appropriate for small values of  $\tilde{h}_\ell$ , in which case the following formulae can be used:

$$\begin{aligned}
 I_{02} &= \frac{\tilde{h}_\ell}{120} \left[ 1 + \frac{31}{21}\tilde{h}_\ell^2 + \frac{134}{105}\tilde{h}_\ell^4 + \frac{2971}{3465}\tilde{h}_\ell^6 + \dots \right], \\
 I_{01} &= \frac{13\tilde{h}_\ell}{60} \left[ 1 + \frac{295}{273}\tilde{h}_\ell^2 + \frac{146}{195}\tilde{h}_\ell^4 + \frac{299}{693}\tilde{h}_\ell^6 + \dots \right], \\
 I_{00} &= \frac{11\tilde{h}_\ell}{20} \left[ 1 + \frac{71}{77}\tilde{h}_\ell^2 + \frac{674}{1155}\tilde{h}_\ell^4 + \frac{12233}{38115}\tilde{h}_\ell^6 + \dots \right].
 \end{aligned}$$

Table 2 contains the values of  $I_{00}/\tilde{h}_\ell$ ,  $I_{01}/\tilde{h}_\ell$ , and  $I_{02}/\tilde{h}_\ell$  for  $\ell = 1, \dots, 12$ .

**7. Implementation.**

**7.1. Decomposition.** The decomposition procedure begins with a tensor spline of the form (1.1) based on polynomial splines  $\varphi_{k,i}(\theta)$  at a given level  $k \geq 1$  and periodic trigonometric splines  $\tilde{\varphi}_{\ell,j}(\phi)$  at a given level  $\ell \geq 1$  with coefficient matrix  $C := A_{k,\ell}$  of size  $m_k \times \tilde{m}_\ell$ . To carry out one step of the decomposition, we solve the systems (4.2) for  $A_{k-1,\ell-1}$ ,  $B_{k-1,\ell-1}^{(1)}$ ,  $B_{k-1,\ell-1}^{(2)}$ ,  $B_{k-1,\ell-1}^{(3)}$  and set

$$(7.1) \quad C = \begin{pmatrix} A_{k-1,\ell-1} & B_{k-1,\ell-1}^{(1)} \\ B_{k-1,\ell-1}^{(2)} & B_{k-1,\ell-1}^{(3)} \end{pmatrix}.$$

To continue the decomposition, we now carry out the same procedure on the matrix  $A_{k-1,\ell-1}$ . This process can be repeated at most  $\min(k, \ell) - 1$  times, where at each step the new spline coefficients and wavelet coefficients are stored in  $C$ . Thus, the entire decomposition process requires no additional storage beyond the original coefficient matrix.

The systems of equations (4.2) which have to be solved to carry out each step of the decomposition process can be efficiently solved due to their tensor nature. All of the matrices that have to be inverted are symmetric positive definite and are either banded or periodic banded. In particular,  $G_k$  is five-banded and  $H_k$  is seven-banded, while  $\tilde{G}_\ell$  and  $\tilde{H}_\ell$  are periodic five-banded and seven-banded, respectively. In our implementation we created special code to compute the band-Cholesky decompositions.

The matrices  $H_k, G_k, \tilde{G}_\ell,$  and  $\tilde{H}_\ell$  are fixed and do not depend on the particular spline being processed. Thus, they can be precomputed and stored in the program.

**7.2. Thresholding.** Typically, at each step of the decomposition, many of the entries in the matrices  $B_{k-j, \ell-j}^{(i)}$  of wavelet coefficients will be quite small. Thus, to achieve compression, these can be removed by a thresholding process. In view of (5.7), tangent plane continuity will be maintained at the poles if we retain all coefficients in the first two and last two rows of these matrices. As a guide to the thresholding process, we now estimate the effect of removing a set of wavelet coefficients.

**THEOREM 7.1.** *Suppose*

$$s = (\varphi_{k-1}^T \quad \psi_{k-1}^T) C \begin{pmatrix} \tilde{\varphi}_{\ell-1} \\ \tilde{\psi}_{\ell-1} \end{pmatrix},$$

where  $C$  is the matrix in (7.1) and  $\varphi_{k-1}^T, \psi_{k-1}^T, \tilde{\varphi}_{\ell-1}, \tilde{\psi}_{\ell-1}$  are vectors of basis functions as in (3.2) and (3.4). Given  $\varepsilon > 0$ , let  $\hat{s}$  be the associated spline where the coefficients  $c_{ij}$  in  $C$  are set to zero whenever

$$|c_{ij}| < \begin{cases} \varepsilon & \text{if } c_{ij} \in B_{k-1, \ell-1}^{(1)} \cup B_{k-1, \ell-1}^{(2)}, \\ \varepsilon/300 & \text{if } c_{ij} \in B_{k-1, \ell-1}^{(3)}. \end{cases}$$

Then

$$(7.2) \quad \|s - \hat{s}\|_\infty \leq 4000 \varepsilon.$$

*Proof.* Let  $Z_\nu$  be the sets of indices  $i, j$  of coefficients in  $B_{k-1, \ell-1}^{(\nu)}$  which are set to zero. Then

(7.3)

$$s - \hat{s} = \sum_{(i,j) \in Z_1} c_{ij} \varphi_{k-1, i} \tilde{\psi}_{\ell-1, j} + \sum_{(i,j) \in Z_2} c_{ij} \psi_{k-1, i} \tilde{\varphi}_{\ell-1, j} + \sum_{(i,j) \in Z_3} c_{ij} \psi_{k-1, i} \tilde{\psi}_{\ell-1, j}.$$

We now estimate the size of the first sum. Let  $\tilde{q}_{rj}^{(\ell)}$  be the entries of the matrix  $\tilde{Q}_\ell$ , so that  $\tilde{\psi}_{\ell-1, j} = \sum_r \tilde{q}_{rj}^{(\ell)} \tilde{\varphi}_{\ell, r}$ . Then using  $|c_{ij}| < \varepsilon$  along with (6.3) and (5.2), we have

$$\left| \sum_{(i,j) \in Z_1} c_{ij} \varphi_{k-1, i} \sum_r \tilde{q}_{rj}^{(\ell)} \tilde{\varphi}_{\ell, r} \right| \leq \varepsilon \sum_r \left( \sum_j |\tilde{q}_{rj}^{(\ell)}| \right) \tilde{\varphi}_{\ell, r} \leq 2\varepsilon \|\tilde{Q}_\ell\|_\infty,$$

where  $\|\tilde{Q}_\ell\|_\infty$  is the matrix norm subordinate to the vector infinity norm.

Similar estimates hold for the other two sums in (7.3), and thus

$$(7.4) \quad |(s - \hat{s})(\theta, \phi)| \leq \varepsilon \left( 2\|\tilde{Q}_\ell\|_\infty + \|Q_k\|_\infty + \frac{2\|\tilde{Q}_\ell\|_\infty \|Q_k\|_\infty}{300} \right).$$

It is clear that

$$\|Q_k\|_\infty = \frac{51363}{77} < 668,$$

and using Mathematica, it can be seen that

$$\|\tilde{Q}_\ell\|_\infty = \frac{128\tilde{h}_\ell \cos(\frac{\tilde{h}_\ell}{2})^2 \sin(\frac{\tilde{h}_\ell}{2})^4}{2\tilde{h}_\ell + \tilde{h}_\ell \cos(\tilde{h}_\ell) - 3 \sin(\tilde{h}_\ell)} \leq 480.$$

The last inequality follows by expanding the fraction into an alternating series in  $\tilde{h}_\ell$  around 0. Inserting these estimates in (7.4) gives (7.2).  $\square$

The large size of the constant in (7.2) is due to the way in which the wavelets have been scaled, and in fact, is quite pessimistic. In practice we have observed numerically that  $\|s - \hat{s}\|_\infty$  is usually about 100 times as large as  $\varepsilon$ .

Theorem 7.1 leads immediately to several reasonable thresholding algorithms. For example, we can simply remove coefficients as described in the theorem. Alternatively (and this is what we do in the experiments below), at the  $j$ th level we can use a threshold of  $\varepsilon/2^{j-1}$  for the coefficients in  $B_{k-j,\ell-j}^{(1)} \cup B_{k-j,\ell-j}^{(2)}$  and a threshold of  $\varepsilon/(300 \cdot 2^{j-1})$  for the coefficients in  $B_{k-j,\ell-j}^{(3)}$ . This level-dependent thresholding scheme retains more coefficients at the coarser levels.

**7.3. Reconstruction.** In view of (4.3), to carry out one reconstruction step simply involves matrix multiplication involving the matrices  $P_k, Q_k, \tilde{P}_\ell,$  and  $\tilde{Q}_\ell$ . These matrices are fixed and do not depend on the particular spline being operated on. Since they are also very sparse, we can simply store their entries in the program.

**7.4. Storage.** Due to their special structure, all of the fixed matrices needed to describe both decomposition and reconstruction can be stored in band form. This will require  $\mathcal{O}(\max(N, \tilde{N}))$  storage, assuming that the largest starting coefficient matrix is of size  $N \times \tilde{N}$ . It is clear that in carrying out both decomposition and reconstruction, all of the coefficients arising in the intermediate steps can be stored in the original coefficient matrix  $C$ . Thus, it follows that the *total amount of storage* needed is  $N \times \tilde{N} + \mathcal{O}(\max(N, \tilde{N}))$ .

**7.5. Complexity.** To carry out one decomposition step on an initial coefficient matrix  $C$  of size  $N \times \tilde{N}$ , we need to solve  $\tilde{N}$  systems of size  $N \times N$  and then  $N$  systems of size  $\tilde{N} \times \tilde{N}$ . Since these systems are at most seven-banded (or periodic seven-banded) and positive definite, the total operation count is  $\mathcal{O}(N \times \tilde{N})$ . Since at each level of decomposition the number of coefficients to be dealt with is reduced by a factor of 4, it follows that the *total operation count for decomposition* is of order  $\mathcal{O}(N \times \tilde{N})$ .

To compute the total operation count for reconstruction, we first discuss the step that produces the coefficients at the finest level. This step requires multiplication by matrices with at most eight nonzero elements in each row or column. Thus the complexity for this step is  $\mathcal{O}(N \times \tilde{N})$ . Adding up the operations required for the coarser steps, it follows that the *total operation count for reconstruction* is also  $\mathcal{O}(N \times \tilde{N})$ .

**7.6. Conditioning.** Using Mathematica, we computed the sup-norm condition numbers of the matrices  $G_k, H_k, \tilde{G}_\ell,$  and  $\tilde{H}_\ell$  for  $1 \leq k \leq 9$ . The results are shown in Table 3. Since none of the condition numbers exceeds 13, we conclude that the algorithm is highly robust.

TABLE 3  
Condition numbers of the Gram matrices.

$k$	$G_k$	$\tilde{G}_k$	$H_k$	$\tilde{H}_k$
1	12.9273	8.76272	4.09545	4.85975
2	12.8056	7.77962	4.94326	4.36433
3	12.8048	7.56795	5.01714	4.25737
4	12.8048	7.51687	5.01762	4.23155
5	12.8048	7.50421	5.01762	4.22515
6	12.8048	7.50105	5.01762	4.22355
7	12.8048	7.50026	5.01762	4.22316
8	12.8048	7.50007	5.01762	4.22306
9	12.8048	7.50002	5.01762	4.22303

TABLE 4  
Computational times.

$k$	$\ell$	Coefficients	Seconds
6	6	37,248	.87
7	7	148,224	3.51
8	8	591,360	14.19
8	9	1,182,720	28.76
9	9	2,362,368	57.41

**7.7. Timing.** Due to the efficient coding, both decomposition and reconstruction are extremely fast. Table 4 lists the computational times for full decomposition and reconstruction for various choices of  $k$  and  $\ell$ . The table clearly shows that the computational time depends linearly on the number of coefficients in the initial spline. The computation was done on a Sun Ultra 1 with a 167 MHz Sun UltraSPARC CPU with 512 KB cache.

**8. Almost tangent plane continuity.** In the method described above, we do not allow the thresholding process to remove coefficients in the first two and last two rows of the wavelet matrices  $B_{k-j,\ell-j}^{(i)}$ . This is required to guarantee tangent plane continuity at the poles and is a disadvantage since retaining these coefficients reduces compression rates.

It is easy to see that the number of retained coefficients at the poles at level  $k, \ell$  is at most  $6\tilde{m}_\ell$ , and thus after performing  $\min(k, \ell) - 1$  decomposition steps, at most  $12\tilde{m}_\ell$  coefficients are retained at the poles. Since the number of initial coefficients is  $m_k \times \tilde{m}_\ell$ , it follows that the compression rate is bounded above by

$$\frac{m_k \times \tilde{m}_\ell}{12\tilde{m}_\ell} = \frac{m_k}{12} \approx 2^{k-2}.$$

The limit on compression rates with tangent plane continuity is thus 64 and 128 for  $k = 8$  and  $k = 9$ , respectively.

It is easy to modify the thresholding process described in section 7.2 to get much higher compression rates at the expense of losing exact smoothness at the poles. There are two possibilities:

( $T^0$ ) We allow coefficients in the second and next-to-last rows of the matrices  $B_{k-j,\ell-j}^{(i)}$  to be removed by thresholding,

( $T^{-1}$ ) We allow all coefficients in matrices  $B_{k-j,\ell-j}^{(i)}$  to be removed by thresholding.

In both cases we get a reconstructed surface that is still tangent plane continuous at all nonpolar points. With strategy  $T^0$ , we lose exact tangent-plane continuity at the poles (but still have it approximately). With strategy  $T^{-1}$ , we may even lose continuity of the surface at the poles. If we are willing to accept minor artifacts at the poles, then much higher compression rates can be achieved. The corresponding algorithms based on thresholding strategies  $T^0$  and  $T^{-1}$  both require exactly the same storage as in the restricted case, and also have the same efficiency and stability properties. We compare the performance of the various algorithms in the following section.

**9. Examples.** To test the general performance of the algorithms, we begin with the following simple example.

*Example 1.* Let  $k = 8$  and  $\ell = 9$ , and let  $s$  be the tensor spline with coefficients

$$c_{ij} = \cos((\tilde{x}_{j+2}^9 - \tilde{x}_{j+1}^9)/2), \quad i = 1, \dots, m_8, \quad j = 1, \dots, \tilde{m}_9.$$

*Discussion.* Since the normalized quadratic B-splines form a partition of unity, it follows from (2.1) that with these coefficients,  $s \equiv 1$  for all  $(\theta, \phi) \in H$ , i.e, the corresponding surface is exactly the unit sphere. In this case the coefficient matrix is of size  $770 \times 1536$  and involves 1,182,720 coefficients. To test the algorithms, we performed decomposition with various values of  $\varepsilon$ , including zero. In all cases, after reconstruction we got coefficients that were correct to machine accuracy (working in double precision).  $\square$

To illustrate the ability of our multiresolution approach to achieve high levels of compression while retaining important features of a surface, we created a tensor spline fit to a smooth surface with a number of bumps.

*Example 2.* Let  $\mathcal{B}$  be the surface shown in the upper left-hand corner of Figure 1.

*Discussion.* The surface  $\mathcal{B}$  was created by fitting a spline  $f_{8,8}$  to data created by choosing 10 random sized subrectangles at random positions in  $H$  and adding tensor product quadratic B-splines of maximum height  $3/4$  with support on each such rectangle to the constant values corresponding to the unit sphere. For  $k = \ell = 8$ , the coefficient matrix is of size  $770 \times 768$  and involves 591,360 coefficients. To test the algorithms, we performed decomposition with the thresholding values  $\varepsilon_r = 10^{-r}$  for  $r = 1, \dots, 9$ . Table 5 shows the number of coefficients  $nco$  remaining at each level for the case where  $\varepsilon = .001$ . Almost  $3/4$  of the coefficients are removed in the first step of decomposition, and after seven steps we end up with only 9734 coefficients (which amounts to a 60:1 compression ratio). Table 6 shows the differences between the original coefficients and the coefficients obtained after reconstruction. The table lists both the *maximum norm*

$$e_\infty := \max_{ij} |c_{ij} - \tilde{c}_{ij}|$$

and the *average  $\ell_1$  norm*

$$e_1 := \frac{\sum_{ij} |c_{ij} - \tilde{c}_{ij}|}{m\tilde{m}},$$

where  $c_{ij}$  are the original coefficients and  $\tilde{c}_{ij}$  are the reconstructed ones.

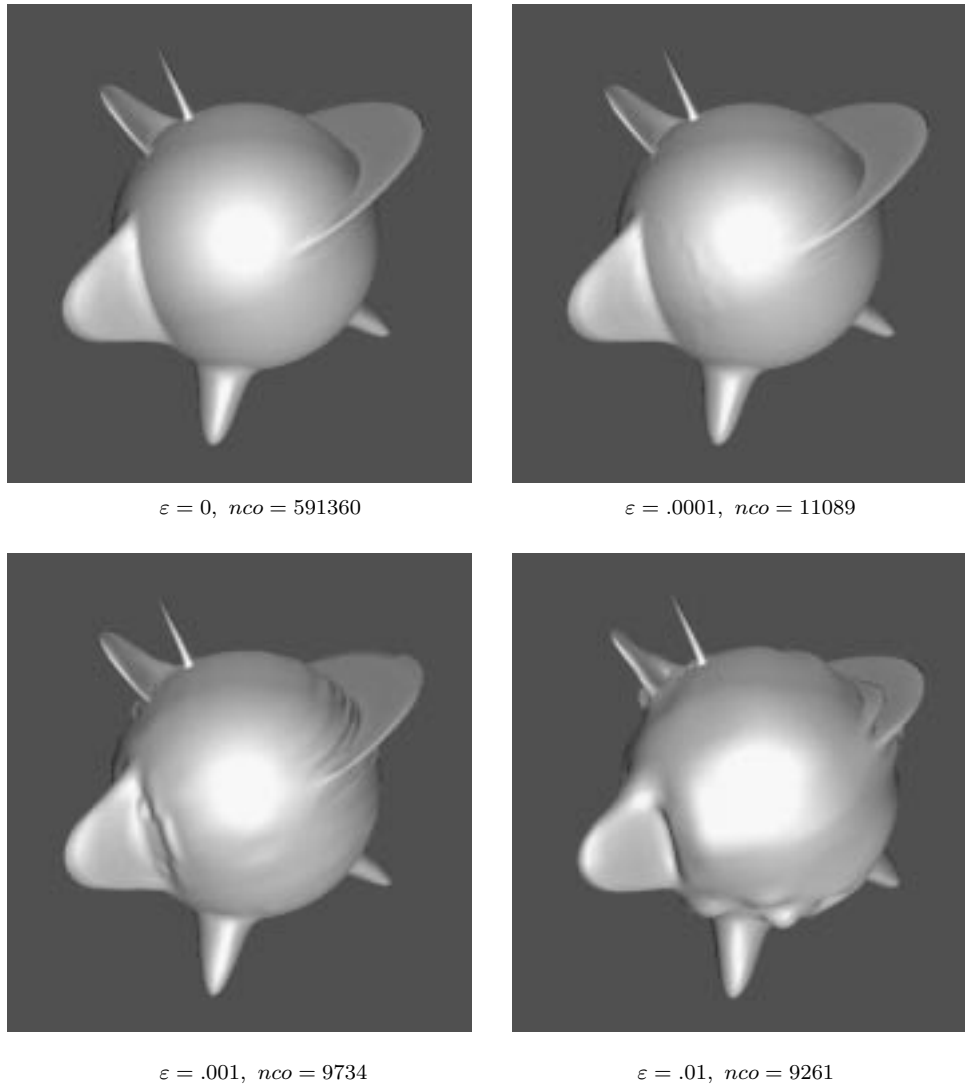
FIG. 1. *Compressed surfaces for Example 2.*

TABLE 5  
*Number, nco, of retained coefficients for Example 2 with  $\varepsilon = .001$ .*

Step	<i>nco</i>
0	591360
1	152832
2	44160
3	17486
4	11209
5	9863
6	9737
7	9734

TABLE 6  
Coefficient errors in Example 2 for selected  $\varepsilon$ .

$\varepsilon$	$nco$	$e_\infty$	$e_1$
0	591360	0	0
10 (-9)	110466	5.54 (-7)	2.75 (-8)
10 (-8)	74332	6.93 (-6)	2.34 (-7)
10 (-7)	44967	5.16 (-5)	1.80 (-6)
10 (-6)	25409	3.69 (-4)	1.38 (-5)
10 (-5)	14992	3.07 (-3)	7.93 (-5)
10 (-4)	11089	1.16 (-2)	4.56 (-4)
10 (-3)	9734	3.92 (-2)	2.62 (-3)
10 (-2)	9261	1.80 (-1)	1.49 (-2)
10 (-1)	9192	6.22 (-1)	3.93 (-2)

TABLE 7  
Example 2 with  $\varepsilon = .0001$  using  $T^0$  and  $T^{-1}$  thresholding.

Step	$nco$	Step	$nco$
0	591360	0	591360
1	150528	1	148224
2	40846	2	37390
3	13945	3	9929
4	7812	4	3521
5	6728	5	2316
6	6704	6	2278
7	6703	7	2277

The surfaces corresponding to the values  $\varepsilon = 0, 10^{-4}, 10^{-3}, 10^{-2}$  are shown in Figure 1. At  $\varepsilon = .0001$  we get near perfect looking reconstruction, while at  $\varepsilon = .001$  the major features are reproduced with only small wiggles in the surface. At  $\varepsilon = .01$  we have somewhat larger errors in the surface.  $\square$

We now present analogous results for Example 2 using the modified thresholding procedures of section 8. Table 7 shows the numbers of coefficients remaining after thresholding with  $\varepsilon = .0001$  using the strategies  $T^0$  and  $T^{-1}$ . In this case the original 591,360 coefficients are reduced to 6703 and 2277, respectively, after seven decomposition steps. This corresponds to compression ratios of 88.22 and 259.71, respectively.

To illustrate the accuracy of the modified thresholding procedures, in Table 8 we list the maximum and average  $\ell_1$  errors in the coefficient vector of Example 2 after reconstruction using the thresholding strategy  $T^{-1}$ . As can be seen by comparing with Table 6, the accuracies are essentially the same as with restricted thresholding, but with much higher compression ratios. The errors corresponding to  $T^0$  thresholding are very similar.

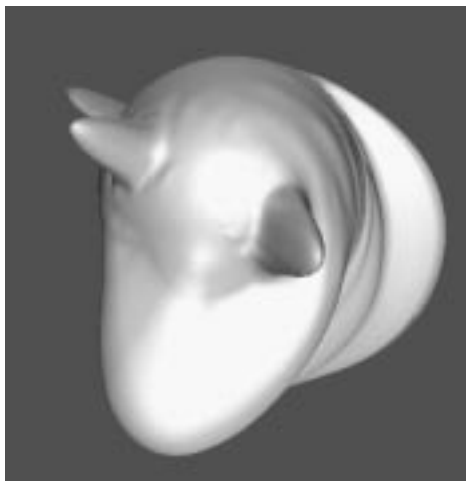
The reconstructed surfaces corresponding to  $T^0$  and  $T^{-1}$  thresholding look almost identical to those shown in Figure 1, with only small disturbances at the poles for larger values of  $\varepsilon$ . To get an idea of what such disturbances look like, in Figure 2 we show polar views of the reconstructed surface for Example 2 in the case  $\varepsilon = .001$ . The results for  $T^0$  and  $T^{-1}$  are shown on the left and right, respectively. The pole is located midway between the “ears” and is virtually invisible for  $T^0$  thresholding.

**10. Remarks.**

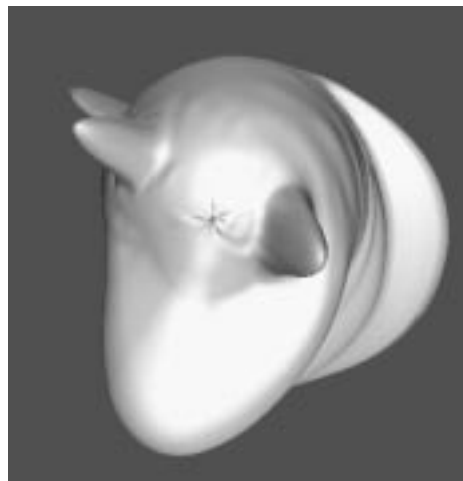
*Remark 10.1.* For convenience, we have elected to use orthogonal wavelets in both the polynomial and the trigonometric spline spaces. This leads to linear systems with very small band size which can be solved efficiently. The need to solve

TABLE 8  
Coefficient errors in Example 2 with  $T^{-1}$  thresholding.

$\varepsilon$	$nco$	$e_\infty$	$e_1$
0	591360	0	0
10 (-9)	103251	5.54 (-7)	2.77 (-8)
10 (-8)	66720	6.93 (-6)	2.37 (-7)
10 (-7)	37015	5.80 (-5)	1.82 (-6)
10 (-6)	17089	3.69 (-4)	1.42 (-5)
10 (-5)	6391	3.08 (-3)	8.31 (-5)
10 (-4)	2277	1.39 (-2)	4.86 (-4)
10 (-3)	776	5.85 (-2)	3.07 (-3)
10 (-2)	156	2.88 (-1)	1.78 (-2)
10 (-1)	48	6.21 (-1)	4.85 (-2)



$\varepsilon = .001$ ,  $nco = 5276$



$\varepsilon = .001$ ,  $nco = 776$

FIG. 2. Surfaces with  $T^0$  and  $T^{-1}$  thresholding.

such systems could be avoided altogether with the use of appropriate (but possibly complicated) biorthogonal bases. However, using our bases leads to a decomposition and reconstruction algorithm whose complexity is of the same order as the number of initial coefficients. Thus, no gain in complexity is possible regardless of what other bases one chooses.

*Remark 10.2.* An alternative way of making sure that tangent plane continuity is maintained at the poles is to decompose the original tensor product function  $s$  into two parts  $s_H$  and  $s_P$ , where

$$s_H := \sum_{i=3}^{m_k-2} \sum_{j=1}^{\tilde{m}_\ell} c_{ij} \varphi_{k,i} \tilde{\varphi}_{\ell,j}$$

and  $s_P := s - s_H$ . Then decomposition, thresholding, and reconstruction can be performed on  $s_H$ . After adding  $s_P$ , the reconstructed spline possesses tangent plane continuity at the poles. Our implementation of this method exhibits essentially the same performance in terms of compression and accuracy as the method described

here, but for higher compression ratios, it produces surfaces that are not quite as visually pleasing near the poles.

*Remark 10.3.* The method described here can be extended to the case of nonuniform knots in both the  $\theta$  and  $\phi$  variables. This can be advantageous when the surface is given on nonuniform knots to start with. There is some computational cost in using nonuniform knots, however, since the various matrices appearing in the decomposition and reconstruction processes can no longer be precomputed and stored.

*Remark 10.4.* In section 4 we have presented the details of the tensor-product decomposition and reconstruction algorithms assuming that the initial function  $f_{k,\ell}$  lies in the space  $\mathcal{V}_k \times \tilde{\mathcal{V}}_\ell$ , with  $k$  and  $\ell$  not necessarily the same. Since these spaces can always be reindexed, this is not strictly necessary in the abstract setting, but was convenient for our application, where there is a natural indexing for our spaces.

*Remark 10.5.* In computing the coefficients needed in sections 5 and 6, we found it convenient to use Mathematica.

*Remark 10.6.* There are several methods for computing approximations of the form (1.1). An explicit quasi-interpolation method using data on a regular grid (along with derivatives at the north and south poles) can be found in [13]. The same paper also describes a two-stage method that can be used to interpolate scattered data and a least squares method that can be used to fit noisy data. A general theory of quasi-interpolation operators based on trigonometric splines can be found in [9].

*Remark 10.7.* A closed, bounded, connected set  $U$  in  $\mathbb{R}^3$  which is topologically equivalent to a sphere is called a *sphere-like surface*. This means that there exists a one-to-one mapping of  $U$  onto the unit sphere  $S$ . Moreover, there exists a point  $O$  inside the volume surrounded by  $U$ , such that every point on the surface  $U$  can be seen from  $O$ . Such surfaces are also called *star-like*. For applications, we can focus on the class of sphere-like surfaces of the form

$$U = \{v\rho(v) : v \in S\},$$

where  $\rho$  is a smooth function defined on  $S$ . Then each function  $f$  defined on  $U$  is just the composition  $f(\cdot) = g(\rho(\cdot))$  with  $\rho$  of a function  $g$  defined on  $S$ .

*Remark 10.8.* As indicated in [11], compression methods on the sphere can be adapted to the problem of creating multiresolution representations of *bidirectional reflection distribution functions* (BRDFs), although the basic domain for such functions is actually a hemisphere. We are currently exploring the use of our method for this purpose.

*Remark 10.9.* It is well known that the polynomial B-splines are stable. In particular, for quadratic B-splines  $(\varphi_i)$  with general knots

$$\frac{1}{3}\|c\|_\infty \leq \left\| \sum c_i \varphi_i \right\|_{L_\infty} \leq \|c\|_\infty$$

for all coefficient vectors  $c$ . The same bounds hold for trigonometric splines since the linear functionals

$$\lambda_i f := \left[ -f(\tilde{x}_{i+1}) + 2(1 + \sigma_i) f\left(\frac{\tilde{x}_{i+1} + \tilde{x}_{i+2}}{2}\right) - f(\tilde{x}_{i+2}) \right] / 2$$

introduced in [13] are dual to the  $\tilde{\varphi}_i$ , i.e.,

$$\lambda_i \tilde{\varphi}_j = \delta_{ij}, \quad i, j = 1, \dots, \tilde{m},$$

where  $\sigma_i := \cos((\tilde{x}_{i+2} - \tilde{x}_{i+1})/2)$ . Analogous stability results hold for general  $p$ -norms with appropriately normalized B-splines.

## REFERENCES

- [1] C. K. CHUI, *An Introduction to Wavelets*, Academic Press, Boston, 1992.
- [2] S. DAHLKE, W. DAHMEN, I. WEINREICH, AND E. SCHMITT, *Multiresolution analysis and wavelets on  $S^2$  and  $S^3$* , Numer. Funct. Anal. Optim., 16 (1995), pp. 19–41.
- [3] P. DIERCKX, *Algorithms for smoothing data on the sphere with tensor product splines*, Computing, 32 (1984), pp. 319–342.
- [4] W. FREEDEN AND U. WINDHEUSER, *Spherical wavelet transform and its discretization*, Adv. Comput. Math., 5 (1996), pp. 51–94.
- [5] J. GOETTELMANN, *Construction of Splines and Wavelets on the Sphere and Numerical Solutions to the Shallow Water Equations of Global Atmospheric Dynamics*, dissertation, Johannes Gutenberg Univ., Mainz, Germany, 1998.
- [6] M. LOUNSBERY, T. D. DEROSE, AND J. WARREN, *Multiresolution analysis for surfaces of arbitrary topological type*, ACM Trans. Graphics, 16 (1997), pp. 34–73.
- [7] T. LYCHE AND K. MØRKEN, *Spline-wavelets of minimal support*, in Numerical Methods in Approximation Theory, D. Braess and L. L. Schumaker, eds., Birkhäuser, Basel, 1992, pp. 177–194.
- [8] T. LYCHE AND L. SCHUMAKER, *L-spline wavelets*, in Wavelets: Theory, Algorithms, and Applications, C. Chui, L. Montefusco, and L. Puccio, eds., Academic Press, New York, 1994, pp. 197–212.
- [9] T. LYCHE, L. SCHUMAKER, AND S. STANLEY, *Quasi-interpolants based on trigonometric splines*, J. Approx. Theory, 95 (1998), pp. 280–309.
- [10] D. POTTS AND M. TASCHÉ, *Interpolatory wavelets on the sphere*, in Approximation Theory VIII, Vol. 2: Wavelets, C. K. Chui and L. L. Schumaker, eds., World Scientific, Singapore, 1995, pp. 335–342.
- [11] P. SCHRÖDER AND W. SWELDENS, *Spherical wavelets: Efficiently representing functions on the sphere*, in Computer Graphics Proceedings, Annual Conference Series, 1995, ACM SIGGRAPH, pp. 161–172.
- [12] L. L. SCHUMAKER, *Spline Functions: Basic Theory*, Wiley, New York, 1981.
- [13] L. L. SCHUMAKER AND C. TRAAS, *Fitting scattered data on spherelike surfaces using tensor products of trigonometric and polynomial splines*, Numer. Math., 60 (1991), 133–144.
- [14] E. STOLLNITZ, T. D. DEROSE, AND D. SALESIN, *Wavelets for Computer Graphics*, Morgan Kaufmann, San Francisco, 1996.
- [15] I. WEINREICH, *Biorthogonale Wavelets auf der Sphäre*, dissertation, Tech. Univ. Aachen, Germany, 1997.