Computer Aided Geometric Design 9 (1992) 219–239
North-Holland

COMAID 282

Surface compression *

Ronald A. DeVore, Björn Jawerth
Department of Mathematics, University of South Carolina, Columbia, SC 29208, USA

Bradley J. Lucier
Department of Mathematics, Purdue University, West Lafayette, IN 47907, USA

Received November 1990
Revised December 1991

Abstract

DeVore, R.A., B. Jawerth and B.J. Lucier, Surface compression, Computer Aided Geometric Design 9 (1992) 219–239.

We propose wavelet decompositions as a technique for compressing the number of a control parameters of surfaces that arise in Computer-Aided Geometric Design. In addition, we give a specific numerical algorithm for surface compression based on wavelet decompositions of surfaces into box splines.

Keywords. Compression of surfaces, box splines, computer-aided geometric design, nonlinear approximation

1. Introduction

Some surfaces in Computer-Aided Geometric Design can be described naively but quite accurately by using a large number of control parameters; these parameters can arise, for example, as measurements from a physical model. In order to effectively store and manipulate the computer representation of such surfaces, we wish to reduce the amount of data while maintaining accuracy, a process we will call *surface compression*. Previous work in this subject has been based on knot removal (see for example [Lyche & Mørken '87]). The purpose of the present paper is to give a new approach, developed from the ideas in [DeVore et al. '92], for compressing parametric surfaces by means of wavelet decompositions. We break the process into two steps: first, we approximate to the required accuracy a parametric surface, derived in any way, by a linear combination of translates of a single function called a *wavelet*, and, second, we derive a new, compressed, approximation to the surface, which will attain roughly the same accuracy as the first approximation, but with fewer control parameters.

A *wavelet decomposition* of a function f defined on $\mathbb{R}^d$ is an expression of the form

$$f = \sum_{k \in \mathbb{Z}} \sum_{j := (j_1, \dots, j_d) \in \mathbb{Z}^d} a_{j,k} \phi_{j,k}, \tag{1.1}$$

* The first author was supported by NSF Grant DMS-8620108, the second author by NSF Grant DMS-8803585, and the third author by NSF Grants DMS-8802734 and DMS-9006219. This research was supported in part by the Air Force Office of Scientific Research (contract 89-0455), the Office of Naval Research (contracts N00014-90-1343, N00014-91-J-1152, and N00014-91-J-1076), the Defense Advanced Research Projects Agency (AFOSR contract 89-0455), and the Army High Performance Computing Research Center.
Correspondence to: R.A. DeVore, Department of Mathematics, University of South Carolina, Columbia, SC 29208, USA.

where the coefficients $a_{j,k} := a_{j,k}(f)$ depend on f, and the functions $\phi_{j,k}$ are defined as

$$\phi_{j,k}(x) := \phi(2^k(x - j/2^k)),$$

the dyadic dilates (by 2^k) and translates (by $j/2^k$) of a single function ϕ called a *wavelet*. Higher values of k correspond to higher frequency or higher resolution features of f.

The decomposition (1.1) is particularly useful if the norm of f in some L_p space or smoothness class (such as a Sobolev space) can be determined solely by examining the size or decay of the coefficients $a_{j,k}$. Of course, not every choice of ϕ allows one to decompose general functions f as in (1.1); some examples of functions ϕ that *can* be used in (1.1) are the orthogonal wavelets of [Meyer '89] and [Daubechies '88], the ϕ transform of [Frazier & Jawerth '90], and various types of spline functions. Although methods for surface compression can be based on any of these wavelets, only for box splines [de Boor & Höllig '82] will we discuss in any detail how to calculate the representation (1.1).

For notational brevity, we shall sometimes index the j,k term of (1.1) by the dyadic cube $I := j2^{-k} + 2^{-k}\Omega$, where $\Omega := [0,1]^d$ is the unit cube in $\mathbb{R}^d$. We shall say that $j2^{-k}$ *corresponds* to I. We shall also denote by $\mathscr{D}_k$ the set of dyadic cubes I whose sidelength $l(I)$ is 2^{-k}, and by $\mathscr{D}$ the union of the $\mathscr{D}_k$, $k \in \mathbb{Z}$. Then, (1.1) can be rewritten

$$f = \sum_{I \in \mathscr{D}} a_I \phi_I. \tag{1.2}$$

The main idea of our compression algorithm is as follows. Suppose that the surface we wish to compress can be represented as $y = f(x)$, $x = (x_1, x_2)$, the graph of a function f defined on $\mathbb{R}^2$. We choose a wavelet function ϕ and view f as built up from its decomposition (1.2). To compress f we would like to replace the infinite sum (1.2) by a finite sum $S = \sum_I b_I \phi_I$ (the coefficients b_I are not necessarily the same as the a_I), while at the same time requiring that the distance between the two surfaces (which in our case can be bounded by $\| f - S \|_{L_\infty}$) be small. If we wish to do this compression as efficiently as possible, we are led to a problem of nonlinear approximation from the nonlinear manifold Σ_n of all functions $S = \sum a_I \phi_I$ with at most n of the coefficients $a_I \neq 0$. (This is a nonlinear problem because the sum of two functions in Σ_n is contained in Σ_{2n} but not, in general, in Σ_n). This approximation problem for approximation in the L_p metric, $0 < p < \infty$, was studied extensively in [DeVore et al. '92] and the algorithms presented in this paper are motivated by the results in [DeVore et al. '92].

In Section 2 we discuss how to calculate the decomposition (1.2) for box splines using quasi-interpolants. Then, in Section 3, we discuss the results of [DeVore et al. '92] that we find relevant to surface compression. While there are many meaningful choices for the function ϕ in the wavelet decomposition and the ensuing compression algorithms, we shall describe only one possibility in detail and give some numerical examples. This example is based on quartic box splines on a three-directional mesh and cardinal spline interpolation. Our algorithm for this special case is described in Section 4.

An a priori error bound for the compression, given in Section 5, shows that $\| f - S \|_{L_\infty} = O(n^{-\alpha/2})$ if, roughly speaking, f has α 'derivatives' in L_σ for some $\sigma > 2/\alpha$. This is to be contrasted with linear methods of approximations $I(f)$ for which $\| f - I(f) \|_{L_\infty} = O(n^{-\alpha/2})$ only if f has α 'derivatives' not in L_σ, but in L_∞, a much stricter requirement. Thus, for some functions f our compression algorithm achieves a much higher rate of approximation than is possible with linear approximation algorithms. We give examples in Section 6 that illustrate this phenomenon.

2. Box splines as wavelets

We briefly recall from [DeVore et al. '92] one way to obtain the decomposition (1.2) in the case of box splines.

Let $T := \{t_i\}_{i=1}^m$ be a set of vectors that span $\mathbb{R}^d$. Each vector t_i, which we assume has integer components, can appear several times in T. The box spline $M := M_T$ is the function defined by the distributional equation

$$\int_{\mathbb{R}^d} M(x) f(x)\, \mathrm{d}x = \int_{Q_m} f\left(\sum_{i=1}^m y_i t_i \right) \mathrm{d}y, \quad f \in C_0^\infty(\mathbb{R}^d), \tag{2.1}$$

where $Q_m := [-\frac{1}{2}, \frac{1}{2}]^m$ is the unit cube in $\mathbb{R}^m$. Then (see [de Boor & Höllig '82]) M is a piecewise polynomial of total order $r := m - d + 1$ (total degree $m - d$) which is supported on the set

$$\left\{ x\colon x = \sum_{i=1}^m y_i t_i;\ -\tfrac{1}{2} \leqslant y_i \leqslant \tfrac{1}{2},\ t_i \in T \right\}.$$

The box spline M is in $C^s(\mathbb{R}^d)$ where $s := s_0 - 2$ and s_0 is the smallest integer for which there are s_0 vectors in T whose removal from T results in a set of vectors which do not span $\mathbb{R}^d$ (see [de Boor & Höllig '82]). An important property of the box spline M is that there are constants c_j such that M can be refined as

$$M(x) := \sum_j c_j M(2x - j). \tag{2.2}$$

(When applied to box splines, property (2.2) has been known as subdivision, or refinement; computational and theoretical consequences of (2.2) have been developed in [Böhm '83, Cohen et al. '84, Dahmen & Micchelli '84, '85b].) It follows from (2.2) that any M_I, $I \in \mathscr{D}_k$, can be rewritten in terms of the box splines M_J, $J \in \mathscr{D}_m$, $m > k$, i.e., box splines at any finer dyadic level.

We shall be interested in box splines M whose integer translates are locally linearly independent. This is the case if and only if (see [Dahmen & Micchelli '85a] or [Jia '85])

$$|\det(Y_d)| = 1$$

for each $d \times d$ matrix Y_d whose columns are vectors from T that span $\mathbb{R}^d$.

Two useful examples of box splines in $\mathbb{R}^2$ are as follows. First, we can set $T := \{(1,0), (0,1), (1,1)\}$. The resulting continuous box spline will be linear between the *grid lines* $x_1 = i$, $x_2 = j$, and $x_1 - x_2 = k$, $i,j,k \in \mathbb{Z}$, and will have $M(0) = 1$ and $M(j) = 0$ for $0 \neq j \in \mathbb{Z}^2$. As a second example, we can take $T := \{(1,0), (1,0), (0,1), (0,1), (1,1), (1,1)\}$. The resulting box spline, which we will use in Section 4, will be C^2 and piecewise quartic; its third derivative will be discontinuous along the same grid lines as for the linear box spline; its integer translates will be locally linearly independent and will contain all polynomials of total degree three or less.

Associated with the box spline M, we have for each $k = 0, \pm 1, \ldots,$ the dilated spaces

$$\mathscr{S}_k := \operatorname{span}\{M(2^k x - j)\colon j \in \mathbb{Z}^d\}.$$

To create the wavelet decomposition (1.2), we first need to find a good approximant to f from the space $\mathscr{S}_k$ for each $k = 0, \pm 1, \ldots$. While there are many ways for doing this (such as cardinal spline interpolation which is described and used later in this paper), we shall for the moment discuss only the quasi-interpolant projectors Q_k which are defined for any function in L_1 as follows.

Each $S \in \mathscr{S}_k$ has the representation

$$S = \sum_{I \in \mathscr{D}_k} \gamma_I(S) M_I$$

where the dual functionals γ_I are all dilates of a single functional

$$\gamma_I(S) := \gamma\big(S(2^{-k}(\cdot + j))\big)$$

when I corresponds to $j2^{-k}$. By the Hahn–Banach theorem, we can extend the functional γ to all of L_1 while preserving its norm as a functional on L_1. If we continue to denote the extension by γ and its dilates by γ_I, then the operator

$$Q_k(f) := \sum_{I\in\mathscr{D}_k} \gamma_I(f) M_I$$

is a projector from $L_1(\text{loc})$ to $\mathscr{S}_k$ that is bounded on L_p for each $1 \leqslant p \leqslant \infty$. ($L_1(\text{loc})$ is the space of functions that are integrable on any compact subset of $\mathbb{R}^2$.)

Instead of just invoking the Hahn–Banach theorem (which is non-constructive), one can find practical algorithms for extending the functional γ to all of L_1. Such formulas can be found in the paper [de Boor & Fix '73] (for the univariate case) and in the book [Chui '88]. Basically, one finds local projections from L_1 onto discontinuous piecewise polynomial spaces, and then projects in a separate step from discontinuous piecewise polynomials to the spline space $\mathscr{S}_k$. This gives a bounded projector on L_p for all $1 \leqslant p \leqslant \infty$. It is sometimes important to have bounded (nonlinear) projectors in L_q for $q < 1$ onto $\mathscr{S}_k$. Using results from [Brown & Lucier '92], one can show that starting with the *best* local projection from L_1 onto discontinuous piecewise polynomial spaces gives a final (nonlinear) projection Q_k that is bounded (and a locally near-best approximation) for all $0 < q \leqslant \infty$.

Using the fact that the ϕ_I are a partition of unity, it is easy to prove (see [DeVore et al. '92]) that for each $f \in L_p(\mathbb{R}^d)$, $1 \leqslant p < \infty$, and for each $f \in C(\mathbb{R}^d)$ we have

$$\| f - Q_k(f) \|_{L_p(\mathbb{R}^d)} \to 0 \quad \text{as } k \to \infty.$$

Therefore, for each k_0, we can write

$$f = Q_{k_0}(f) + \sum_{k=k_0}^{\infty} (Q_{k+1}(f) - Q_k(f)) = \sum_{k=k_0}^{\infty} \sum_{I\in\mathscr{D}_k} a_I M_I \tag{2.3}$$

with convergence in the L_p norm. Here in the last equality, we use the refinement equation (2.2) to rewrite $Q_{k+1}(f) - Q_k(f)$ in terms of ϕ_I, $I \in \mathscr{D}_{k+1}$. This gives the representation (1.2) for $\phi = M$. When $1 < p < \infty$, one can let $k_0 \to -\infty$ since $\| Q_{k_0}(f) \|_{L_p(\mathbb{R}^d)} \to 0$ as $k_0 \to -\infty$ (see [DeVore et al. '92]).

We should make clear that the general representation (1.2) is not unique. Indeed, any M_I can be rewritten as a linear combination of M_J at any finer dyadic level because of (2.2). So, even the function M_I does not have a unique decomposition. On the other hand, the decomposition that we have derived has uniquely determined coefficients once the extension of γ is chosen; once the operator Q_k is fixed, the decomposition is completely determined by (2.3). We could begin with another projector in place of quasi-interpolants and obtain yet another representation of f. Other possible projections Q_k include the L_2 spline projector and the cardinal spline interpolant. The former associates to $f \in L_2$ its best L_2 approximant $S_k(f)$ from $\mathscr{S}_k$ and we obtain our decomposition (1.2) for $f \in L_2(\mathbb{R}^d)$ by writing f again as a telescoping sum $f = \sum_k (S_{k+1}(f) - S_k(f))$. The cardinal interpolant is the spline $I_k(f) \in \mathscr{S}_k$ which interpolates $f \in C$ at the lattice points $j2^{-k}$, $j \in \mathbb{Z}^d$. This leads to a decomposition (1.2) for $f \in C(\mathbb{R}^d)$. We shall in fact use the cardinal interpolants in our algorithm of Section 4 and the analysis of Section 5.

3. Approximation from Σ_n

We recall some of the results of [DeVore et al. '92] which form the basis for the compression algorithm of the next section. Let ϕ be a function which allows the decomposition (1.2) for general functions f. We shall be interested in approximating functions f by elements from Σ_n. In order to measure the error of such an approximation in the metric of L_p, $0<p\leqslant\infty$, to a function f on a domain D, we introduce

$$\sigma_n(f)_p := \inf_{S\in\Sigma_n} \| f-S\|_{L_p(D)}.$$

The main goal of [DeVore et al. '92] is to characterize, under suitable conditions on ϕ that we will not relate here, the L_p order of approximation by the elements of Σ_n in terms of the smoothness of f in Besov spaces. We recall that a Besov space measures the smoothness of a function $f\in L_p(D)$, D a domain in $\mathbb{R}^d$, in terms of the modulus of smoothness

$$\omega_r(f,t)_p := \sup_{|h|\leqslant t} \|\Delta_h^r(f,\cdot)\|_{L_p(D(rh))},$$

where $D(h)$ is the set of x such that the line segment $[x, x+h]$ is contained in D. (The forward difference $\Delta_h^r(f,x)$ is defined iteratively by $\Delta_h^0(f,x):=f(x)$ and $\Delta_h^{k+1}(f,x):=\Delta_h^k(f,x+h)-\Delta_h^k(f,x)$, $k\geqslant 0$.) In what follows $D=\mathbb{R}^d$ or D is a cube in $\mathbb{R}^d$.

If $\alpha>0$ and $0<p,\ q\leqslant\infty$, the Besov space $B_q^\alpha(L_p(D))$ is the collection of all $f\in L_p(D)$ for which the following is finite:

$$|f|_{B_q^\alpha(L_p(D))} := \begin{cases} \left(\displaystyle\int_0^\infty [t^{-\alpha}\omega_r(f,t)_p]^q \frac{\mathrm{d}t}{t}\right)^{1/q}, & 0<q<\infty,\\ \displaystyle\sup_{t\geqslant 0} t^{-\alpha}\omega_r(f,t)_p, & q=\infty, \end{cases} \tag{3.1}$$

where $r-1\leqslant\alpha<r$. The expression in (3.1) is a semi-(quasi)norm for $B_q^\alpha(L_p(D))$ and we obtain the norm for this space by adding the $\|\cdot\|_{L_p(D)}$ norm to the semi-norm. There are several other equivalent semi-norms for $B_q^\alpha(L_p)$ which are of interest. We shall make use of the fact that for any cube D,

$$|f|_{B_q^\alpha(L_p(D))} \approx \begin{cases} \left(\displaystyle\sum_{k=0}^\infty [2^{k\alpha}\omega_r(f,2^{-k})_p]^q\right)^{1/q}, & 0<q<\infty,\\ \displaystyle\sup_{k\geqslant 0} 2^{k\alpha}\omega_r(f,2^{-k})_p, & q=\infty. \end{cases} \tag{3.2}$$

This follows by noting first that the integral can be taken over [0,1] and that $\omega_r(f,t)_p$ is an increasing, bounded function of t and then discretizing the integral in (3.1) at the points 2^{-k}, $k\geqslant 0$.

Of special importance for the L_p approximation by the elements of Σ_n are the spaces

$$B^\alpha := B_\tau^\alpha(L_\tau) \quad \text{where} \quad \tau := \tau(\alpha,p) := (\alpha/d+1/p)^{-1}.$$

Of course the spaces B^α also depend on p but in our applications p will be fixed and clearly understood.

The main result of [DeVore et al. '92] is the characterization

$$\sum_{n=1}^\infty [n^{\alpha/d}\sigma_n(f)_p]^\tau \frac{1}{n} < \infty \iff f\in B^\alpha \tag{3.3}$$

which holds for $D=\mathbb{R}^d$ and D a cube in $\mathbb{R}^d$; for $1\leqslant p<\infty$, and for all $\alpha<\min(r, s+1/p)$. Here s is associated with the smoothness of ϕ and r is related to how well we can approximate by the elements of $\mathscr{S}_k$. In the special case where ϕ is a box spline, $\phi=M$, then r is the order of the box spline ϕ and s is its smoothness ($M\in C^s$).

We make the following heuristic comments about the equivalence between approximation and regularity expressed by (3.3). The Besov space B^α measures smoothness of order α in L_τ,

$\tau = (\alpha/d + 1/p)^{-1}$. On the other hand the approximation order in L_p of (3.3) is, roughly speaking, $O(n^{-\alpha/d})$. In other words, (3.3) characterizes the functions f that have an order of approximation like $O(n^{-\alpha/d})$ in L_p as those having smoothness of order α in L_τ. The fact that $\tau < p$ (indeed $\tau < 1$ if $\alpha > d$) represents the gain in nonlinear methods over their linear counterparts.

There is in fact no simple characterization of functions with approximation order $\sigma_n(f)_p$ precisely $O(n^{-\alpha/d})$; however (3.3) is a close substitute. In one direction, it does follow from (3.3) that each $f \in B^\alpha$ can be approximated with order of approximation $\sigma_n(f)_p = O(n^{-\alpha/d})$:

$$\sigma_n(f)_p \leqslant C n^{-\alpha/d} | f |_{B^\alpha}. \tag{3.4}$$

Conversely, we know that any $f \in L_p$ with this approximation order is in $B^{\alpha-\epsilon}$ for all $\epsilon > 0$.

The proof of (3.4) given in [DeVore et al. '92] implicitly gives a numerical algorithm for determining 'good' approximations $S \in \Sigma_n$. Actually, there are two algorithms. The simplest of these chooses the n largest coefficients in the decomposition (1.2). This choice however results in constants in the bound of the error of approximation (3.4) that depend on p; they blow up as $p \to \infty$. A second algorithm presented in [DeVore et al. '92] results in constants independent of p but is much more complicated. So much so, that we shall use a modified algorithm for our compression in Section 4.

The results in [DeVore et al. '92] do *not* imply that (3.4) holds when $p = \infty$. Since the case $p = \infty$ is perhaps the most natural choice for surface compression, the numerical algorithm of Section 4 is based on this choice. We establish therefore in Section 5 a bound for $\sigma_n(f)_\infty$ which is only slightly worse than (3.4). The approximant providing the bound (3.4) for $p = \infty$ is constructed by our compression algorithm.

As noted in Section 1, an important feature of the decomposition (1.2) for f is that the norm of f in various function spaces can be described in terms of the coefficients a_I. In particular, it was proved in [DeVore et al. '92] that

$$| f |_{B^\alpha} \approx \left(\sum | a_I |^\tau | I |^{\tau/p} \right)^{1/\tau}. \tag{3.5}$$

We shall make no specific use of (3.5) in what follows.

4. An algorithm for surface compression

In this section, we shall describe a specific algorithm (and some of its possible variants) for surface compression. In order to simplify the discussion that follows, we shall first describe how to construct a surface $z = S(x, y)$ that compresses a surface given by a function $z = f(x, y)$ defined on all of $\mathbb{R}^2$ (thus, for the remainder of the paper, $d = 2$). Compressing surfaces defined on all of $\mathbb{R}^2$ leads to computations with infinite matrices, etc. Of course, the actual implementation of our algorithm is for surfaces on finite domains. Therefore, at the end of this section we will show how to restrict attention to surfaces defined on a square $\Omega := [0,1]^2$. Obvious modifications allow for the construction of parametrized surfaces $S(u, v)$, $0 \leqslant u, v \leqslant 1$, with S a mapping of Ω into $\mathbb{R}^3$. Essentially, the only changes needed below to move to the parametric case is to treat the coefficients as vectors in $\mathbb{R}^3$.

We devote the next few paragraphs to a brief overview of our compression algorithm. Our algorithm is based on the quartic box spline M for a three-directional mesh described briefly in Section 2. At each dyadic level we will calculate the cardinal interpolant $I_k(f)$ to f. $I_k f$ is defined by the property

$$I_k f\left(\frac{j}{2^k}\right) = f\left(\frac{j}{2^k}\right), \quad j \in \mathbb{Z}^2.$$

We choose a dyadic level K at which the wavelet decomposition is truncated, i.e., for our purposes $I_K(f)$ will be a sufficiently good approximation to f. We obtain the decomposition

$$f \approx I_K(f) = I_1(f) + \sum_{k=2}^{K} (I_k(f) - I_{k-1}(f)) = \sum_{k=1}^{K} \sum_{I \in \mathscr{D}_k} a_I(f) M_I. \tag{4.1}$$

Let $T_k := I_k(f) - I_{k-1}(f)$, $k = 2, \ldots, K$ and $T_1 := I_1(f)$. We write $T_k = \sum_{I \in \mathscr{D}_k} a_I(f) M_I$. Our compressed approximation $S := S_1 + \cdots + S_K$ of f consists of terms $b_I M_I$ from each dyadic level k. If one wanted, one could simply choose for the coefficients of S_k all coefficients $b_I = a_I(f)$, $I \in \mathscr{D}_k$, in the decomposition (4.1) bigger than some specified tolerance ϵ.

In our method, we modify this simple strategy in two ways. First, our tolerance depends on the set $\mathscr{D}_k$: at the dyadic level k, the tolerance is set to $\epsilon_k := k\epsilon/2K$. We make the selection of coefficients at coarse levels more likely for two reasons: (1) because each M_I has large support, an added coefficient will decrease the error over a large area, and (2) there are many fewer coefficients at coarse levels than at fine levels. A second modification in our algorithm is that we pass along to the next finer level (by rewriting) any term $b_I M_I$ which is not put into our approximation S. The effect of this is to retain until the finest level all information about f (that is an exact realization of f up to the finest dyadic level).

At the coarsest level, we let Λ_1 denote the collection of all cubes $I \in \mathscr{D}_1$ such that $|a_I(f)| > \epsilon/2K$ and set $S_1 = \sum_{I \in \Lambda_1} a_I(f) M_I$. The remaining terms $a_I(f) M_I$ with $I \in \mathscr{D}_1$ but $I \notin \Lambda_1$ are rewritten at the next dyadic level and are added to T_2. This gives $T_2' = \sum_{I \in \mathscr{D}_2} d_I(f) M_I$. Here $d_I(f) = a_I(f) + a_I'(f)$ where a_I are the coefficients in T_2 and the a_I' arise by rewriting. We now test the coefficients d_I. If $|d_I(f)| > 2\epsilon/2K$, then I is in Λ_2 otherwise $d_I(f) M_I$ is rewritten. We let $S_2 := \sum_{I \in \Lambda_2} d_I(f) M_I$. We continue in this way and arrive at our final approximation $S = S_1 + \cdots + S_K$ to f.

We now give a more detailed numerical description of the algorithm by explaining the various steps in the construction.

4.1. Preliminaries

In order to specify an algorithm one must first choose a wavelet ϕ, a norm in which to measure the error, and the highest and lowest dyadic level of refinement. We discuss these preliminary topics in this section.

As noted above, there are many possible choices of ϕ. Whereas our algorithm takes ϕ to be the quartic box spline M on a three-directional mesh, one could just as well choose various box splines or tensor product B-splines or the wavelets obtained by multi-resolution. The box splines on a three-dimensional mesh have been extensively studied in [de Boor & Höllig '82] (see also the monograph [Chui '88]). We recall some of their important properties.

Let $T := \{t_i\}_1^6$ with $t_1 := t_4 := (1,0)$, $t_2 := t_5 := (0,1)$, and $t_3 := t_6 := (1,1)$. The box spline $M := M_T$ is the function defined by the distributional equation (2.1). It is a piecewise quartic (total degree 4; $r = 5$) polynomial on the mesh consisting of the lines ut_i, $u \in \mathbb{R}$, $i = 1, 2, 3$ and their integer translates; it has smoothness C^2. The support of M is the set

$$\left\{ x \colon x = \sum_{i=1}^{6} y_i t_i;\ -\tfrac{1}{2} \leqslant y_i \leqslant \tfrac{1}{2},\ t_i \in T \right\}.$$

The refinement identity (2.2) for M is easily derived from its Fourier transform

$$\hat{M}(x) = \prod_{i=1}^{6} \frac{\sin(t_i \cdot x/2)}{t_i \cdot x/2},$$

where $x \cdot y$ is the scalar product of the two vectors x and y. Taking the Fourier transform of both sides of (2.2) leads to a polynomial identity for the coefficients c_j in (2.2). In this way, we obtain

$$M(x) := \sum_j c_j M(2x-j), \tag{4.2}$$

where $c_j = 10/16$, $j=0$; $c_j = 6/16$, $j = \pm(1{,}0)$, $\pm(0{,}1)$, $\pm(1{,}1)$; $c_j = 2/16$, $j = \pm(1,-1)$, $\pm(1{,}2)$, $\pm(2{,}1)$; $c_j = 1/16$, $j = \pm(2{,}0)$, $\pm(0{,}2)$, $\pm(2{,}2)$; and $c_j = 0$ otherwise.

For surface compression, it seems most natural to measure the error of approximation in the L_∞ norm. We shall assume that the surface we wish to compress is continuous. It is important to note that there is a simple relationship between the L_∞ norm of a spline $S \in \mathscr{S}_k$ and its coefficients s_I in the box spline representation $S = \sum_{I \in \mathscr{D}_k} s_I M_I$; namely, for some constant C_0,

$$C_0^{-1} \max_{I \in \mathscr{D}_k} |s_I| \leqslant \| S \|_{L_\infty(\mathbb{R}^d)} \leqslant \max_{I \in \mathscr{D}_k} |s_I|.$$

The upper inequality follows from the fact that the $\{M_I\}$ are a partition of unity. We will not need the numerical value of the constant in the lower inequality.

Our algorithm requires one to prescribe the allowable error ϵ of the compressed approximation to the surface. As noted above, the error of approximation is measured in the L_∞ norm. The algorithm guarantees the approximation error does not exceed ϵ provided the initial numerical realization of f has an L_∞ error in approximating f which does not exceed $\epsilon/2$.

We arbitrarily choose the lowest dyadic level to be 1, which corresponds to a dyadic grid size of $1/2$. It is also necessary to choose a highest dyadic level K at which the wavelet decomposition of f is to be truncated for its numerical representation. We denote by

$$T := \sum_{k \leqslant K} \sum_{I \in \mathscr{D}_k} a_I M_I$$

the truncation of the wavelet decomposition of f at level K. The level K should be chosen so that $\| f - T \|_{L_\infty} \leqslant \epsilon/2$. There is a simple a priori bound for $\| f - T \|_{L_\infty}$ in terms of the smoothness of f. Namely, if f is in the Lipschitz space Lip δ, then the following bound was given in [DeVore et al. '92]:

$$\| f - T \|_{L_\infty} \leqslant C 2^{-K\delta}$$

with C depending on δ. K should be chosen so that $C 2^{-K\delta} < \epsilon/2$. We discuss in more detail a priori bounds for truncation in Section 5.

4.2. Rewriting splines

The refinement equation (4.2) allows us to rewrite any spline $S \in \mathscr{S}_k$ at a finer dyadic level:

$$S = \sum_{j \in \mathbb{Z}^2} s_j M(2^k x - j) = \sum_{j \in \mathbb{Z}^2} s_j' M(2^{k+1} x - j); \quad s_j' = \sum_{\mu + 2\nu = j} c_\mu s_\nu. \tag{4.3}$$

The subroutine Rewrite(A) uses (4.3) to rewrite a spline $S \in \mathscr{S}_k$ at the next finer dyadic level. Rewrite(A) takes the matrix $A = (s_j)_{j \in \mathbb{Z}^2}$ which corresponds to coefficients of a spline $S \in \mathscr{S}_k$, at some dyadic level k, and returns the matrix $A' := (s_j')$ corresponding to the coefficients of S with respect to the basis M_I, $I \in \mathscr{D}_{k+1}$.

4.3. Calculating the cardinal interpolant

While any projector onto $\mathscr{S}_0$ that is defined for continuous functions would be a possible choice in our algorithm, we shall use the cardinal interpolant, which we now describe.

If $y := (y_j)_{j \in \mathbb{Z}^2}$ is a collection of real numbers, then the spline $S \in \mathscr{S}_0$ which satisfies

$$S(j) = y_j, \quad j \in \mathbb{Z}^2, \tag{4.4}$$

is called the cardinal interpolant to y. It has been shown in [de Boor et al. '85] that there is a unique solution $S \in C(\mathbb{R}^2)$ satisfying (4.4) whenever $y \in l_\infty$. If we write S in its B-spline series $S = \sum_{j \in \mathbb{Z}^2} s_j M(x - j)$, then the coefficients $s := (s_j)$ of the cardinal interpolant S can be found by inverting the Toeplitz operator $\mathscr{T}$:

$$(\mathscr{T}b)_i := \sum_{j \in \mathbb{Z}^2} M(i-j) b_j, \quad b := (b_j)_{j \in \mathbb{Z}^2}.$$

Namely,

$$s = \mathscr{T}^{-1} y.$$

To assemble the Toeplitz operator, one needs the values of M at the integers: $M(0) = 1/2$, $M(j) = 1/12$, if $j = \pm(1,0)$, $\pm(0,1)$, $\pm(1,1)$ and otherwise $M(j) = 0$, $j \in \mathbb{Z}^2$.

One can find to any desired accuracy a finite number of coefficients of the cardinal interpolant by creating a finite matrix which is an approximation to $\mathscr{T}^{-1}$. The operator $\mathscr{T}^{-1}$ is also Toeplitz, $\mathscr{T}^{-1} := (\alpha(i-j))$, and the coefficients $\alpha := (\alpha(j))$ can be found formally by inverting the symbol of $\mathscr{T}$. To numerically find these coefficients, we solve the equation $\mathscr{T}\alpha = \delta$ with $\delta := (\delta(j))$ the Kronecker sequence $\delta(0) := 1$ and $\delta(j) := 0$, $j \neq 0$. Since the coefficients in $\mathscr{T}^{-1}$ are known (from the inverse of the symbol) to decay exponentially, it is sufficient to write $\mathscr{T}y = \delta$ as a system of equations and take a large enough block of this system corresponding to the indices $|i| \leqslant m$, with m sufficiently large. The integer m is chosen depending on the desired accuracy of the approximation.

The subroutine Interpolate(B) generates the approximate cardinal interpolant to the entries of a given matrix $B = (b_j)$ (each entry b_j is associated to the point j). In our program we did this by applying the cardinal mask of our approximation of $\mathscr{T}^{-1}$ to B. Namely, the coefficients a_j of the cardinal interpolant are given by

$$a_j := \sum_{|i| \leqslant m} \alpha(i) b_{j-i}.$$

Thus, if we begin with the matrix $B = F_k := (f(j/2^k))_{j \in \mathbb{Z}^2}$ then Interpolate returns the coefficients of the cardinal interpolant $I_k(f)$ of f.

4.4. Constructing the compressed approximant

The algorithm Compress uses the two subroutines Rewrite and Interpolate to produce $B := (b(i, k))$, the non-zero coefficients of the compressed approximant

$$S = \sum_{k=1}^{K} \sum_{i \in \Lambda_k} b(i, k) M(2^k x - i)$$

to the surface $z = f(x, y)$. Here Λ_k denotes the set of those indices i such that $b(i, k) := B_k(i) \neq 0$. Compress assumes that f can be evaluated at any point in $\mathbb{R}^2$ and that K, the number of dyadic levels, and ϵ, the error tolerance, have been provided by the user.

```
COMPRESS:
    for k = 1 to K do
        A_k = INTERPOLATE((f(j/2^k))_{j ∈ Z^2})
    next k
    for k = K down to 2 do
        A_k = A_k − REWRITE(A_{k−1})
    next k
    for k = 1 to K do
        B_k = 0
        for j ∈ Z^2 do
            if |A_k(j)| ≥ kε/2K then
                B_k(j) = A_k(j); A_k(j) = 0
            end if
        next j
        if k < K then
            A_{k+1} = A_{k+1} + REWRITE(A_k)
        end if
    next k
end COMPRESS
```

4.5. Operating on a finite domain

In this section, we derive the finite subsets of $\mathbb{Z}^2$ that enter into the computation when we wish to compress a surface over the unit square $\Omega := [0,1]^2$. A similar analysis can be carried out for other domains.

First of all, we must keep only the coefficients $B_k(j)$ that contribute a nonzero amount to the surface on Ω. We refer to the support of M to see that we need B_k defined for j in the set

$$J_k^B := \{ j = (j_1, j_2) \in \mathbb{Z}^2 \mid -1 \leqslant j_1 \leqslant 2^k + 1,\ -1 \leqslant j_2 \leqslant 2^k + 1 \},$$

which contains one 'strip' of coefficients outside of Ω. To calculate

$$B_{k+1}(j) := A_{k+1}(j) - \text{REWRITE}(A_k)(j)$$

for j in one strip outside of Ω, we need $A_k(j)$ to be defined on the set

$$J_k^A := \{ j = (j_1, j_2) \in \mathbb{Z}^2 \mid -2 \leqslant j_1 \leqslant 2^k + 2,\ -2 \leqslant j_2 \leqslant 2^k + 2 \},$$

which has two strips outside of Ω. If the coefficients $A_k(j)$ are calculated using the finite approximation with $|i| \leqslant m$ to $\mathscr{T}^{-1}$ then we need values of $f(j/2^k)$ for the set

$$J_k^F := \{ j = (j_1, j_2) \in \mathbb{Z}^2 \mid -2-m \leqslant j_1 \leqslant 2^k + 2 + m,\ -2-m \leqslant j_2 \leqslant 2^k + 2 + m \}.$$

A value of $m = 12$ gives a relative error of about 10^{-7} in L^∞.

If the surface we are trying to compress is not a mathematical surface but rather is described only by scattered data, then one needs to begin with a mathematical approximation to this data. One possibility is to use one of the standard methods for scattered data interpolation. In the event that the data is densely distributed, a triangulation of the data set and piecewise linear interpolation may be sufficient.

If f is defined only on Ω then, in order to evaluate it at the points in $J_k^F/2^k$, it is necessary to extend f to a larger domain while at the same time maintaining its smoothness. (One could always assume that f is identically zero outside Ω, but this would introduce artificial discontinuities in f along the boundary of Ω.) We shall use the following method of Whitney to extend a function f from Ω to the larger cube $\Omega_0 := [-1,2]^2$. We first describe a univariate operator which extends a function f from $[0,1]$ to $[-1,1]$. Let β_i, $0 \leqslant i \leqslant 3$, satisfy

$$\sum_{i=0}^{3} \beta_i(-2^{-i})^j = 1, \quad j = 0,\dots,3. \tag{4.5}$$

The system of equations (4.5) is a vanderMonde system and hence has a unique solution. This choice of β_i guarantees that the following univariate extension operator E_0 extends cubic polynomials exactly:

$$E_0 f(x) := \begin{cases} f(x), & 0 \leqslant x \leqslant 1, \\ \sum_i \beta_i f(-2^{-i}x), & -1 \leqslant x \leqslant 0. \end{cases} \tag{4.6}$$

To obtain a multivariate extension of f defined on Ω, we first extend, for each fixed y, the function $f(\cdot, y)$ to the rectangle $[-1,1] \times [0,1]$ by using (4.6). Using obvious modification of (4.6), we can now extend the function to $[-1,1]^2$, to $[-1,2] \times [-1,1]$ and finally to Ω_0. If necessary, we could repeat this extension procedure and thereby extend f to any larger finite cube. This results in a function Ef defined on Ω_0 which agrees with f on Ω and satisfies

$$\omega_4(Ef, t)_{L_p(\Omega_0)} \leqslant C\omega_4(f, t)_{L_p(\Omega)}, \quad t > 0. \tag{4.7}$$

If a function f must be extended from a cube Q to a cube Q' that is much larger than Q, the extension operators of (4.6) must be applied a large number of times. In this case, it may be useful to multiply the extension by a smooth cutoff function that is one on Q and vanishes outside of a neighborhood of Q. For example, the function f could be extended to Ω_0 and then be multiplied by a cutoff function that is one on Ω but vanishes outside of $[-1,2]^2$. Multiplication by a smooth cutoff function does not preserve (4.7), but it does preserve membership in the regularity spaces that interest us, which is sufficient for our purposes.

For projections other than cardinal interpolants, it may be that the problem of extending f can be avoided. For example, if we use quasi-interpolants, we can restrict the support of the coefficient functional γ_I to lie in Ω whenever the support of M_I intersects Ω nontrivially. In this way there is no need to extend f to a larger set.

5. Error bounds

In this section, we shall give an error bound for the approximation of f by the compressed wavelet decompositions of the algorithm of the previous section. Since the approximation takes place in the L_∞ norm and for an adaptive choice of coefficients, the error bound we shall give is not covered by the results in [DeVore et al. '92]. Although we continue to formulate our results solely for the case of the quartic box spline M, it will be clear from our proof that our results apply to more general ϕ. We shall assume that all computations are done exactly, including the computation of the coefficients of the cardinal spline interpolants. A finer analysis could take into consideration errors in these computations.

Let $f \in C(\Omega)$ be the function which determines the mathematical surface that is being compressed. We first want to observe that we can assume that f has support in the larger cube $\Omega'' := [-2,3]^2$. Indeed, if f is not defined outside of Ω then we extend f to the larger cube $[-2,3]^2$ by using the Whitney extension theorem with a preservation of the modulus of smoothness in the sense of (4.7). We can then multiply f by a smooth function which is one on $[-1,2]^2$ and has support on $[-2,3]^2$. In this way, we extend f from Ω to all of $\mathbb{R}^2$ with the extended function supported on Ω''. Moreover, if f is in a Besov space $B_q^\alpha(L_p(\Omega))$, $\alpha < 4$, then, because of (4.7), the extension is in $B_q^\alpha(L_p(\mathbb{R}^2))$ with comparable norm. We continue to denote the extended function by f.

We have in (3.3) described the connection between the smoothness of the function f and its order of approximation in L_p, $0 < p < \infty$, by the compressed wavelet decomposition. For example, (3.3) shows that if $f \in B^\alpha(\mathbb{R}^d)$, then f is approximated with order $O(n^{-\alpha/d})$ with n the number of terms in the compressed approximant. We shall see that (3.3) holds in the case $p = \infty$, $d = 2$, as well, provided we assume slightly more smoothness for f. In fact our compression algorithm will provide this approximation. We let $X^\alpha := B_\sigma^\alpha(L_\sigma(\mathbb{R}^2))$ with $\sigma > \tau := 2/\alpha$. The number $\delta := 2/\tau - 2/\sigma$, which we call the 'discrepancy', measures how much membership in X^α differs from membership in B^α.

The space X^α consists of functions which have smoothness of order α but measured in L_σ. Since $\sigma > \tau$, X^α is a smaller space than B^α: $X^\alpha \subset B^\alpha$.

From the Sobolev embedding theorem for Besov spaces, we have for any cube I, the continuous embedding: $X^\alpha(I) \subset B_\infty^\delta(L_\infty(I)) = \mathrm{Lip}(\delta, I)$ and

$$|f|_{\mathrm{Lip}(\delta,I)} := \sup_{x \neq y} \frac{|f(x) - f(y)|}{|x-y|^\delta} \leqslant C \| f \|_{X^\alpha(I)}. \tag{5.1}$$

For $\sigma \geqslant 1$, this embedding can be found in any of the standard treatments of Besov spaces (see e.g. [Peetre '76]). The case $\sigma \leqslant 1$ follows from Theorem 9.6 in [DeVore & Sharpley '84]. We say more about this embedding later.

Now let $I_k(f) = \sum_{I \in \mathscr{D}_k} s_I(f) M_I$ denote the cardinal spline interpolant to f. Let $\mathscr{D}_k'$ denote the collection of $I \in \mathscr{D}_k$ for which M_I is not identically zero on Ω. Then, on Ω, $I_k(f) = \sum_{I \in \mathscr{D}_k'} s_I(f) M_I$. From our remarks in Section 4 concerning the Toeplitz structure of I_k, it follows that the coefficient functionals a_I are uniformly bounded as mappings from $C(\mathbb{R}^d)$ into l_∞. Moreover, from the exponential decay of the entries in Toeplitz representation of I_k, we have that for some $0 < \eta < 1$,

$$|s_I(f)| \leqslant C \sum_{j \in \mathbb{Z}^2} \eta^{|i-j|} \left| f\left(\frac{j}{2^k}\right) \right|, \tag{5.2}$$

with C an absolute constant. Here and later, we shall use the convention that cubes I correspond to $i/2^k$ and cubes J correspond to $j/2^k$.

Our first result limits the number of dyadic levels that need to be considered in the decomposition of f.

Lemma 5.1. *If $f \in X^\alpha(\mathbb{R}^2)$, then for $k = 1, 2, \ldots,$*

$$\| f - I_k(f) \|_{L_\infty(\mathbb{R}^2)} \leqslant C 2^{-k\delta} \| f \|_{X^\alpha(\mathbb{R}^2)}$$

with C depending only on δ and α.

Proof. Since $|s_I(f)| \leqslant C \| f \|_{L_\infty(\mathbb{R}^2)}$, for all $I \in \mathscr{D}_k$, and since the M_I, $I \in \mathscr{D}_k$, are nonnegative and form a partition of unity, we obtain

$$\| I_k(f) \|_{L_\infty(\mathbb{R}^2)} \leqslant \| (s_I(f))_{I \in \mathscr{D}_k} \|_{l_\infty(\mathbb{Z}^2)} \leqslant C \| f \|_{L_\infty(\mathbb{R}^2)}.$$

Hence, I_k is a bounded mapping from $C(\mathbb{R}^2)$ into itself with norm independent of k. Since I_k is also a linear projector onto $\mathscr{S}_k$, we have $f - I_k(f) = f - S + I_k(S - f)$, for each $S \in \mathscr{S}_k$. Therefore,

$$\| f - I_k(f) \|_{L_\infty(\mathbb{R}^2)} \leqslant (1 + \| I_k \|) \operatorname{dist}(f, \mathscr{S}_k)_{L_\infty(\mathbb{R}^2)} \leqslant C \operatorname{dist}(f, \mathscr{S}_k)_{L_\infty(\mathbb{R}^2)}. \tag{5.3}$$

We have given in [DeVore et al. '92] the following bound

$$\operatorname{dist}(f, \mathscr{S}_k)_{L_\infty(\mathbb{R}^2)} \leqslant C\ \omega(f, 2^{-k})_\infty \tag{5.4}$$

where ω is the modulus of continuity of f. From the embedding inequality (5.1), it follows that $\omega(f, t) \leqslant Ct^\delta \| f \|_{X^\alpha(\mathbb{R}^2)}$. The lemma now follows from (5.3) and (5.4). □

The significance of Lemma 5.1 is that it gives us an a priori bound for the number of dyadic levels needed in the representation of f if we are to have an error $\leqslant \epsilon$. Namely, it is enough to take K so that $C2^{-K\delta} \| f \|_{X^\alpha} \leqslant \epsilon$ where C is the constant of Lemma 5.1. That is, K need only be larger than $\log_2 C \| f \|_{X^\alpha}/\epsilon$.

We now recall the basic steps of the algorithm of Section 4 when applied to functions $f \in X^\alpha$. We shall at first consider only functions f with $\| f \|_{X^\alpha} \leqslant 1$. If we are given an error tolerance $\epsilon > 0$, we choose K as above so that

$$\| f - I_K(f) \|_{L_\infty(\Omega)} \leqslant \epsilon/2.$$

Because of our restriction on $\| f \|_{X^\alpha} \leqslant 1$, K can be chosen independent of f, i.e., K depends only on δ. For each $k = 2, \dots, K$, let $T_k := I_k(f) - I_{k-1}(f)$ and $T_1 := I_1(f)$. On Ω, we can write $T_k(f) = \sum_{I \in \mathscr{D}'_k} a_I(f) M_I$ where, as before, $\mathscr{D}'_k$ consists of all cubes in $\mathscr{D}_k$ such that M_I does not vanish identically on Ω. Then, $T_1 + T_2 + \cdots + T_K$ is our (approximate) wavelet decomposition of f.

To construct our approximation to f, we examine the coefficients of T_1. We let Λ_1 be the collection of those $I \in \mathscr{D}'_1$ for which $|a_I(f)| \geqslant \epsilon/2K$ and Λ'_1 the remaining cubes in $\mathscr{D}_1$. Then, $S_1 := \sum_{I \in \Lambda_1} a_I(f) M_I$, is our initial approximation to f. We let $T'_2 := \sum_{I \in \Lambda'_1} a_I(f) M_I$ and rewrite T'_2 at the next finer dyadic level: $T'_2 = \sum_{I \in \mathscr{D}'_2} a'_I(f) M_I$.

In general, if T'_k is the current rewrite (obtained by rewriting terms from level $k-1$ at level k), then we examine the coefficients of $T_k + T'_k$ to determine which terms will be kept in our approximation. Such a coefficient is $d_I(f) = a_I(f) + a'_I(f)$ where $a_I(f)$ is the original coefficient which appears in T_k and $a'_I(f)$ is the coefficient obtained by rewriting T'_k. Let Λ_k denote the collection of all cubes $I \in \mathscr{D}'_k$ for which $|d_I| \geqslant k\epsilon/2K$ and let $S_k := \sum_{I \in \Lambda_k} d_I(f) M_I$. Then $S_1 + \cdots + S_k$ is our updated approximation to f and $T'_{k+1} := \sum_{I \in \Lambda'_k} d_I(f) M_I$ where the set Λ'_k is the collection of all cubes in $\mathscr{D}'_k$ which are not in Λ_k. We can rewrite T'_{k+1} at the level $k+1$:

$$T'_{k+1} = \sum_{I \in \mathscr{D}'_{k+1}} a'_I(f) M_I.$$

Now $S := S_1 + S_2 + \cdots + S_K$ is our final approximation to f. The coefficients of the error $I_K(f) - S = \sum_{I \in \Lambda'_K} d_I(f) M_I$ all satisfy $|d_I(f)| \leqslant \epsilon/2$. Since the M_I are a partition of unity of nonnegative functions, we have $\| I_K(f) - S \|_{L_\infty(\Omega)} \leqslant \epsilon/2$. Therefore,

$$\| f - S \|_{L_\infty(\Omega)} \leqslant \| f - I_K(f) \|_{L_\infty(\Omega)} + \| I_K(f) - S \|_{L_\infty(\Omega)} \leqslant \epsilon. \tag{5.5}$$

Our main result, Theorem 5.3, will count the number of cubes N in $\Lambda := \bigcup_{k=1}^K \Lambda_k$; then N is also the number of terms in S. Before doing this, we give a bound for the size of the coefficients $a_I(f)$. For this, we shall use the quasi-interpolant operators Q_k for $\mathscr{S}_k$ introduced in Section 1. For each $f \in L_1(\mathbb{R}^2)$,

$$Q_k(f) = \sum_{I \in \mathscr{D}_k} \gamma_I(f) M_I$$

where γ_I are linear functionals on $L_1(\mathbb{R}^2)$ all obtained from one functional γ by dilation and translation. Actually, there are many possible choices for γ; each is obtained by extending the dual basis functionals (for the basis $(M_I)_{I \in \mathscr{D}_k}$) from $\mathscr{S}_k$ to $L_1(\mathbb{R}^2)$. In particular, γ can be chosen so that each γ_I is supported on I (see [DeVore et al. '92]). It follows that the Q_k are bounded projectors from $L_p(\mathbb{R}^2)$ onto $\mathscr{S}_k$, $1 \leqslant p < \infty$, from $C(\mathbb{R}^2)$ onto $\mathscr{S}_k$ for $p = \infty$. Using the fact that $\mathscr{S}_k$ contains all polynomials of total degree 3, and that Q_k is a projector onto $\mathscr{S}_k$, it is easy to prove that (see [DeVore et al. '92])

$$\| f - Q_k(f) \|_{L_\infty(J)} \leqslant C\, E(f, J^*), \quad J \in \mathscr{D}_k \tag{5.6}$$

where $E(f, J^*)$ is the error in approximating f by polynomials of total degree $\leqslant 3$ in the norm of $C(J^*)$ and J^* is the cube with the same center as J and sidelength $l(J^*) = 16 l(J)$.

We also will need a bound for $E(f, I^*)$ which is related to the embedding inequality (5.1).

Lemma 5.2. *If* $0 < \alpha < 4$, *then for each* $f \in X^\alpha(I^*)$ *and each* $I \in \mathscr{D}_k$, *we have*

$$E(f, I^*) \leqslant C \,|\, I \,|^{\delta/2} \,|\, f \,|_{X^\alpha(I^*)} \tag{5.7}$$

with the constant C *depending only on* δ.

Proof. If $I \in \mathscr{D}_k$, let $\tilde{\mathscr{D}}_j$, $j = 0, 1, \ldots,$ be the dyadic partition relative to I^* that is obtained by 'quartering' I^* and so on. Let $s_j(f)_p := s_j(f, I^*)_p$ denote the error in approximating f in the $L_p(I)$ norm by piecewise polynomials of total degree $\leqslant 3$ on the partition $\tilde{\mathscr{D}}_j$, $j = 0, 1, \ldots.$ Then $E(f, I^*) = s_0(f, I^*)_\infty$. As was shown in [DeVore & Popov '88], for the analogous case of piecewise polynomials of coordinate degree, we have for all $0 < p \leqslant \infty$,

$$s_j(f)_p \leqslant C\, \omega_4(f, 2^{-k-j})_p, \quad j = 0, 1, \ldots \tag{5.8}$$

From (5.8) for $p = \sigma$ and from the discretization (3.2) of the Besov space semi-norm $|\cdot|_{X^\alpha}$, we have for $0 < \alpha < 4$,

$$\sum_{j=0}^{\infty} 2^{(j+k)\alpha\sigma} s_j(f)_\sigma^\sigma \leqslant C \,|\, f \,|_{X^\alpha}^\sigma.$$

Now, let R_j denote the best piecewise cubic approximation in $L_\sigma(I^*)$ to f on the partition $\tilde{\mathscr{D}}_j$: $\| f - R_j \|_{L_\sigma(I^*)} = s_j(f)_\sigma$. Then, $f = R_0 + (R_1 - R_0) + \cdots$ with convergence in $L_\sigma(I^*)$ and $R_j - R_{j-1}$ is a piecewise polynomial on the finer partition $\tilde{\mathscr{D}}_j$. The cubes in $\tilde{\mathscr{D}}_j$ have side length $C 2^{-k-j}$. Therefore, from elementary inequalities for polynomials, we have

$$\| R_j - R_{j-1} \|_{L_\infty(I^*)} \leqslant C 2^{2(k+j)/\sigma} \| R_j - R_{j-1} \|_{L_\sigma(I^*)}.$$

It follows that

$$\| f - R_0 \|_{L_\infty(I^*)} \leqslant C \sum_{j=0}^{\infty} 2^{2(k+j)/\sigma} \| R_j - R_{j-1} \|_{L_\sigma(I^*)}.$$

Since $2/\sigma = 2/\tau - \delta = \alpha - \delta$ and $2^{-(k+j)\delta} \leqslant 2^{-k\delta}$ for $j \geqslant 0$, if $\sigma \leqslant 1$, we obtain from the previous inequality that

$$\begin{aligned} \| f - R_0 \|_{L_\infty(I^*)} &\leqslant C 2^{-k\delta} \sum_{j=0}^{\infty} 2^{(j+k)\alpha} \| R_j - R_{j-1} \|_{L_\sigma(I^*)} \\ &\leqslant C 2^{-k\delta} \left(\sum_{j=0}^{\infty} 2^{(j+k)\alpha\sigma} \| R_j - R_{j-1} \|_{L_\sigma(I^*)}^\sigma \right)^{1/\sigma}. \end{aligned} \tag{5.9}$$

Now, $R_j - R_{j-1} = f - R_{j-1} - (f - R_j)$ and so on the right side of (5.9), $\| R_j - R_{j-1} \|_{L_\sigma(I^*)}^\sigma$ can be replaced by $s_j(f)_\sigma^\sigma + s_{j-1}(f)_\sigma^\sigma$. In this way, we see that the right side of (5.9) does not exceed $C 2^{-k\delta} |\, f \,|_{X^\alpha}$ which gives (5.7). The case $\sigma > 1$ is proved similarly using Hölder's inequality. □

The operators I_k are also bounded projectors from $C(\mathbb{R}^2)$ onto S_k and $I_k(S)-I_{k-1}(S)=0$, for all $S \in \mathscr{S}_{k-1}$. From this, it follows that $a_I(S)=0$, $I \in \mathscr{D}_k$, for all $S \in \mathscr{S}_{k-1}$. Also, for each $I \in \mathscr{D}_k$, $a_I(f)$ is a linear combination of $s_I(f)$ and (by rewriting) of $s_J(f)$ for cubes $J \in \mathscr{D}_{k-1}$ with supp $M_J \cap$ supp $M_I \neq \emptyset$. Therefore, from (5.2), we obtain

$$|a_I(f)| \leqslant C \sum_{j \in \mathbb{Z}^2} \eta^{|i-j|} \left| f\left(\frac{j}{2^k}\right) \right| \tag{5.10}$$

with C an absolute constant. As before, we use the notation that I corresponds to $i/2^k$ and J corresponds to $j/2^k$.

In inequality (5.10), we can replace f by $f-Q_{k-1}(f)$ and use (5.6) to obtain

$$|a_I(f)| = |a_I(f-Q_{k-1}(f))| \leqslant C \sum_{J \in \mathscr{D}_k} \eta^{|i-j|} E(f, J^*). \tag{5.11}$$

We recall σ used in the definition of X^α. If $\sigma \leqslant 1$, we use the fact that an l_σ norm is greater than an l_1 norm to obtain from (5.11):

$$|a_I(f)|^\sigma \leqslant C \sum_{J \in \mathscr{D}_k} \eta^{|i-j|} E(f, J^*)^\sigma. \tag{5.12}$$

Here $0<\eta<1$ is not necessarily the same constant as in (5.11). When $\sigma>1$ this also holds (with a new value of $0<\eta<1$) because of Hölder's inequality.

We next give a bound for the number of terms N in our compressed approximation to f needed to attain an error ϵ.

Theorem 5.3. *Let $\delta>0$ be fixed and let $0<\alpha<4$. Given $\epsilon>0$ and $f \in X^\alpha$, with $\|f\|_{X^\alpha} \leqslant 1$, the algorithm of the previous section will give a compressed approximation $S=\sum_{I \in \Lambda} b_I M_I$ to f such that*

$$\|f-S\|_{L_\infty(\Omega)} \leqslant \epsilon \tag{5.13}$$

and

$$N := |\Lambda| \leqslant C_0 \epsilon^{-2/\alpha} \tag{5.14}$$

with C_0 depending only on δ.

Proof. We have already observed that (5.13) holds (see (5.5)). We need therefore only count the number of terms N in $\Lambda := \bigcup_1^K \Lambda_k$. Let $N_k := |\Lambda_k|$. A trivial bound is

$$N_k \leqslant C 2^{2k}, \tag{5.15}$$

which follows from the fact that there are at most $C2^{2k}$ cubes $I \in \mathscr{D}_k'$. We shall now give another bound for N_k.

Let $I \in \Lambda_k$ and consider the coefficient $d_I(f)=a_I(f)+a_I'(f)$. Since all the coefficients $d_J(f)$, $J \in \Lambda_{k-1}'$, by definition, satisfy $|d_J(f)| \leqslant (k-1)\epsilon/2K$ and since the refinement weights c_μ in (4.3), which are active in determining $a_I'(f)$, are nonnegative and sum to one, we have $|a_I'(f)| \leqslant (k-1)\epsilon/2K$, for all $I \in \Lambda_k$. On the other hand, by the definition of Λ_k, $|d_I(f)| \geqslant k\epsilon/2K$. This gives

$$|a_I(f)| \geqslant \epsilon/2K, \quad I \in \Lambda_k. \tag{5.16}$$

We raise both sides of (5.16) to the power σ and replace $|a_I(f)|^\sigma$ by its upper bound (5.12) and then sum over all $I \in \Lambda_k$ to obtain

$$N_k \epsilon^\sigma \leqslant C K^\sigma \sum_{I \in \Lambda_k} \sum_{J \in \mathscr{D}_k} \eta^{|i-j|} E(f, J^*)^\sigma \leqslant C K^\sigma \sum_{J \in \mathscr{D}_k} E(f, J^*)^\sigma$$

where we use our previous notation that $i/2^k$ corresponds to I and $j/2^k$ corresponds to J. Here, the constant C depends only on δ.

Now, we use our bound (5.7) of Lemma 5.2 in (5.12) to obtain

$$N_k \leqslant CK^{\sigma}\epsilon^{-\sigma}2^{-k\delta\sigma} \sum_{J\in\mathscr{D}_k} |f|^{\sigma}_{X^{\alpha}(J^*)}. \tag{5.17}$$

The semi-norm on X^{α} is set subadditive in the sense that if $\{A_j\}$ is a family of sets and $A=\bigcup A_j$ then

$$|f|^{\sigma}_{X^{\alpha}(A)} \leqslant C\sum_j |f|^{\sigma}_{X^{\alpha}(A_j)}$$

where C is the maximum number of times a point $x\in A$ appears in different of the A_j (see [DeVore & Popov '88]). In our case, a point $x\in\mathbb{R}^2$ appears in at most C of the sets J^* with C an absolute constant. Therefore, (5.17) gives

$$N_k \leqslant CK^{\sigma}\epsilon^{-\sigma}2^{-k\delta\sigma}|f|^{\sigma}_{X^{\alpha}(\mathbb{R}^2)} \leqslant CK^{\sigma}\epsilon^{-\sigma}2^{-k\delta\sigma}. \tag{5.18}$$

This is the other bound we wanted for N_k.

From (5.15) and (5.18), we find for any integer m:

$$N \leqslant C\left\{\sum_{k=1}^{m} 2^{2k} + \sum_{k=m}^{K} \epsilon^{-\sigma}2^{-k\delta\sigma}\right\} \leqslant C^{\sigma}\{2^{2m}+\epsilon^{-\sigma}2^{-m\delta\sigma}\}. \tag{5.19}$$

If we choose m so that $2^{2m}\approx\epsilon^{-\sigma}2^{-m\delta\sigma}$, the two terms on the right side of (5.19) will be of the same order. Also, $2^{2m(1+\delta\sigma/2)}\approx\epsilon^{-\sigma}$ and hence $2^{2m}\approx\epsilon^{-\sigma/(1+\delta\sigma/2)}=\epsilon^{-1/\tau}=\epsilon^{-2/\alpha}$, as desired. □

Corollary 5.4. *If $f\in X^{\alpha}$, $0<\alpha<4$, and n are given, then by choosing ϵ and K appropriately the algorithm of Section 4 generates a compressed approximant S with at most n terms and*

$$\| f-S\|_{L_\infty(\Omega)} \leqslant C_0^{\alpha/2}|f|_{X^{\alpha}}n^{-\alpha/2} \tag{5.20}$$

with C_0, given in Theorem 5.3, depending only on δ.

Proof. Let $\lambda:=|f|_{X^{\alpha}}$ and $\epsilon:=C_0^{\alpha/2}\ n^{-\alpha/2}\lambda$ with C_0 the constant in (5.14). The algorithm when applied to f and ϵ gives an approximant S. If we apply the algorithm to $g:=f/\lambda$ and ϵ/λ, the algorithm gives S/λ as the approximant to g. Hence, from Theorem 5.3,

$$\| f-S\|_{L_\infty(\Omega)} = \lambda\| g-S/\lambda\|_{L_\infty(\Omega)} \leqslant \epsilon,$$

which is (5.20). On the other hand, the number of terms in S by (5.14) does not exceed $C_0(\epsilon/\lambda)^{-2/\alpha}=n$. □

6. Computational examples

We now discuss the application of the compression algorithm of Section 4 to three surfaces $z=f(x, y)$ defined on the domain $[-1,1]^2$ (in place of the domain $[0,1]^2$ used in our mathematical analysis of Section 5). These surfaces were chosen to display various singularities. The first example is the 'sombrero' function

$$f_1(x, y) := \sin(2\pi r^2)/r^2, \quad r:=\sqrt{x^2+y^2},$$

which has a removable singularity at 0. The second is the 'ridge' function

$$f_2(x, y) := \mathrm{e}^{-|x-y|}$$

which has singularities on the line $y = x$. The third function

$$f_3(x, y) := r^{1/2}$$

has a cusp singularity at the origin.

For surfaces $z = f(x, y)$ we can compare the error in our compressed, nonlinear, approximation S and the linear approximation $I_k(f)$ to f. In particular, if we have n coefficients in each approximation,

$$\| f - S \|_{L_\infty} \leqslant C n^{-\alpha/2} \| f \|_{X^\alpha}, \quad \alpha < 4, \tag{6.1}$$

whereas

$$\| f - I_k(f) \|_{L_\infty} \leqslant C n^{-\alpha/2} \| f \|_{W^{\alpha,\infty}(\Omega)}, \quad n = 2^k, \ \alpha < 4, \tag{6.2}$$

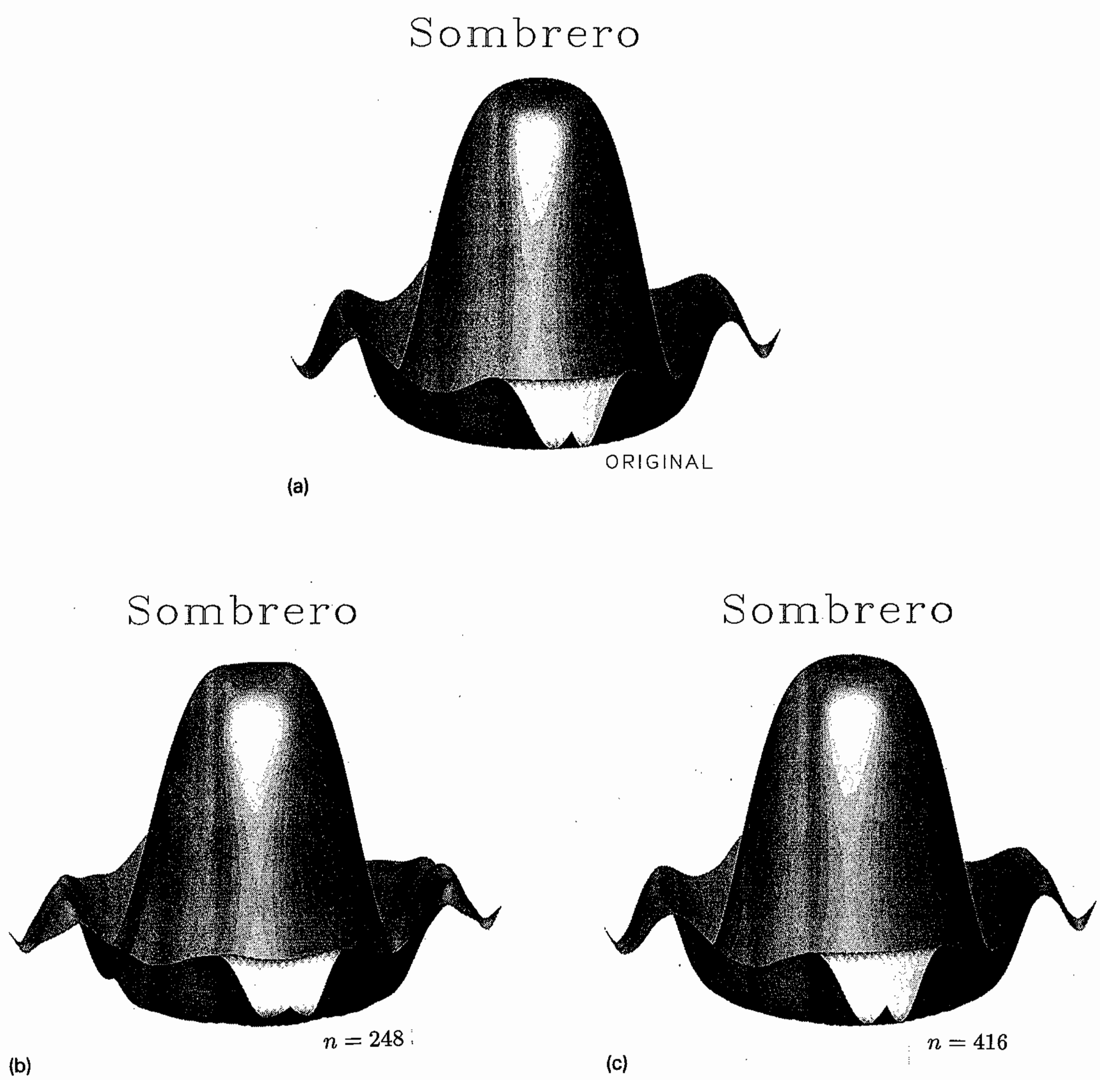

Fig. 1.

whenever the right sides of (6.1) and (6.2) are finite. In (6.2) we define $\| f \|_{W^{\alpha,\infty}(\Omega)}$ to be $\sup_{t>0} t^{-\alpha}\omega_4\ (f,\ t)_\infty$. Roughly speaking, the right side of (6.1) will be finite if f has α 'derivatives' in L_σ, $1/\sigma = \alpha/2 - \delta$, while the right side of (6.2) is finite when f has α derivatives in L_∞. It can often happen that the right side of (6.1) is finite for values of α for which the right side of (6.2) is infinite. Thus, we can get higher order approximation with our nonlinear method than with the typical linear method. The space

$$X^\alpha := B^\alpha_\sigma(L_\sigma(\Omega)),\quad \frac{1}{\sigma} = \frac{\alpha}{2} - \delta = \frac{1}{\tau} - \delta,$$

is close to the space $B^\alpha := B^\alpha_\tau(L_\tau(\Omega))$, so we will base our discussion on the spaces B^α.

Because the fourth derivatives of f_1, which is in C^∞, are all bounded, it follows that $\| f_1 \|_{W^{\alpha,\infty}(\Omega)}$ is comparable to $\| f_1 \|_{B^\alpha_\sigma(L_\sigma(\Omega))}$ for all $\sigma \geqslant 0$. Consequently, for a fixed number of

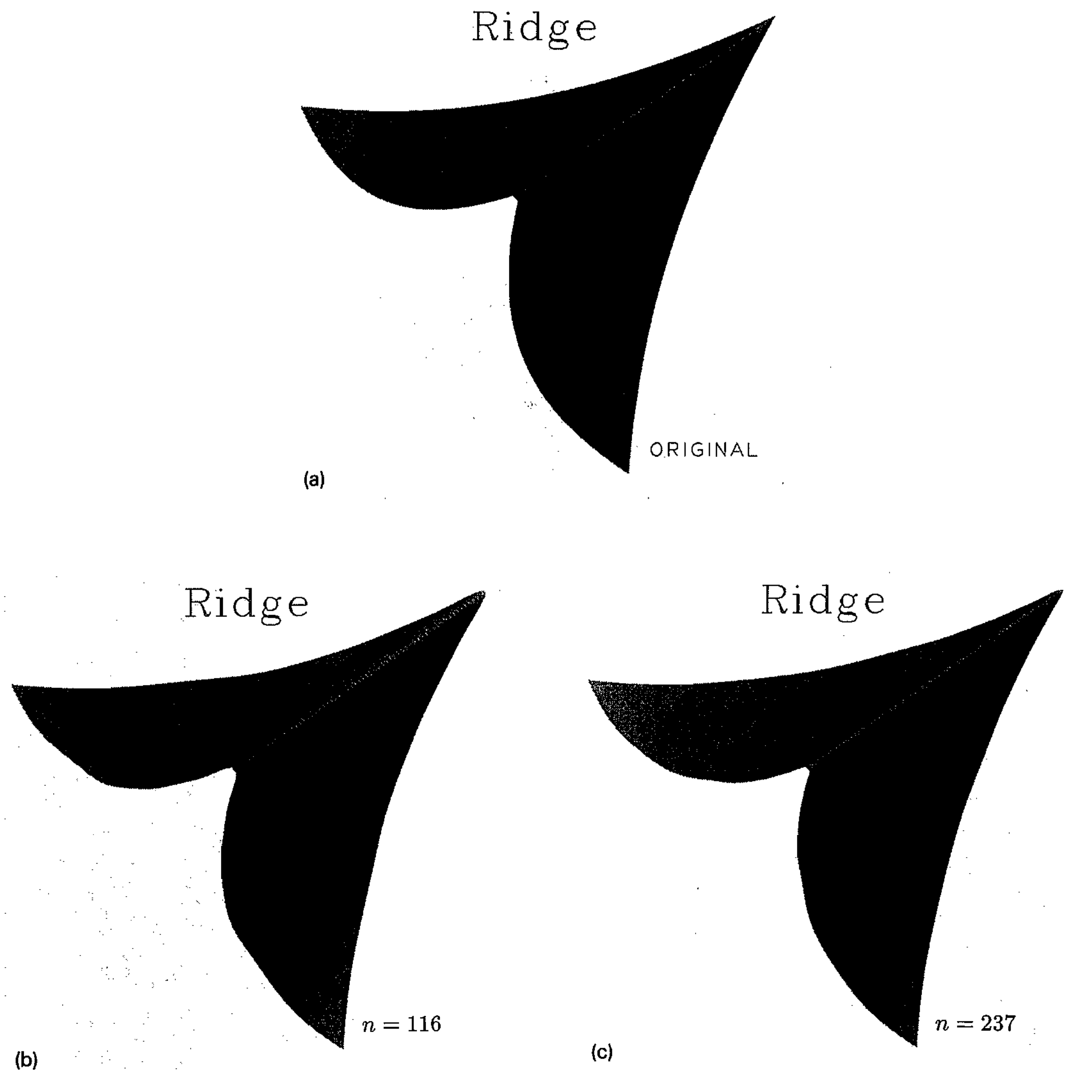

Fig. 2.

coefficients n, the error in our wavelet approximation scheme and the error in the associated linear scheme will be comparable. In fact, the fourth derivative of $\sin(2\pi r^2)/r^2$ is greater than 3000 when r is in a neighborhood of 1, so, estimating $\| f_1 \|_{X^\alpha}$ accordingly and taking $\alpha = 4$, we see that the error in our approximation is about $3000Cn^{-2}$ for some unknown (but presumably benign) constant C. So, even though this function is smooth, its fourth derivatives are very big, and the approximation requires more coefficients than for the 'ridge' and 'cusp' surfaces, which are in some sense more singular. Nonetheless, as few as 416 coefficients will give a compressed surface which approximates the original to within 1.48×10^{-2}. The approximant differs little from the original; however, the two surfaces can be distinguished in Fig. 1 by examining contour lines.

An easy computation for the ridge function f_2 shows that for h orthogonal to the line $y = x$, we have $| \Delta_h^4(f_2, x, y)| \approx | \Delta_h^2(f_2, x, y)| \approx |h|$ in a band of width $|h|$ about this line. It

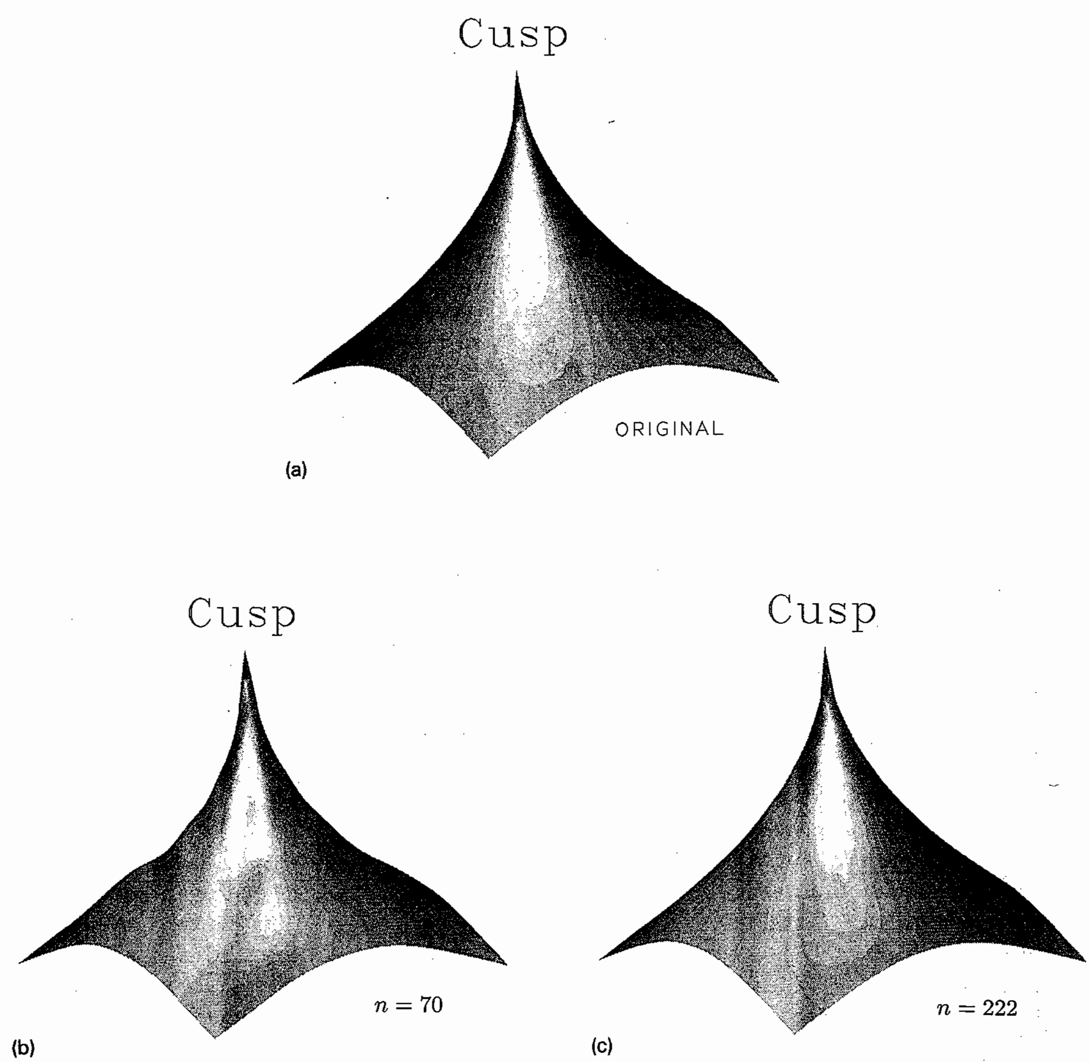

Fig. 3.

Table 1
Errors and coefficients for computational examples

Example	Error	Number of coefficients
Sombrero	1.30×10^{-5}	6485
	2.19×10^{-5}	4794
	1.07×10^{-4}	2500
	2.14×10^{-4}	1878
	8.66×10^{-4}	1077
	1.39×10^{-3}	881
	7.08×10^{-3}	541
	1.48×10^{-2}	416
	3.59×10^{-2}	306
	6.00×10^{-2}	248
Ridge	1.60×10^{-3}	1934
	6.49×10^{-3}	482
	1.30×10^{-2}	237
	2.59×10^{-2}	116
	4.22×10^{-2}	80
Cusp	1.38×10^{-3}	618
	2.76×10^{-3}	463
	1.38×10^{-2}	222
	2.69×10^{-2}	163
	7.13×10^{-2}	91
	1.05×10^{-1}	70

follows that

$$\omega_4(f_2, t)_\tau \approx t^{1+1/\tau}, \quad 0<t<1 \text{ and } 0<\tau\leqslant\infty.$$

Therefore, $f \in B^\alpha$ provided $\alpha < 2$. The a priori error bounds of Section 5 show that achieving an error of $O(n^{-\alpha/2})$ for any $0<\alpha<2$ requires at most $O(n)$ nonzero coefficients. This should be compared to the linear approximant $I_K(f_2)$, which also achieves an error of $O(n^{-\alpha/2})$ with $O(n)$ coefficients, but for which $0<\alpha<1$, since $\omega_4(f, t)_\infty \approx t$, $0<t<1$.

Roughly speaking, the compressed approximant to f_2 has a single row of cubes from the highest dyadic level along the line $y=x$. Cubes from coarser dyadic levels appear in wider bands about the singularity line. Fig. 2 presents the original surface and two compressed surfaces. The compressed surface with 237 coefficients is already virtually indistinguishable from the original.

As expected, the greatest advantage of the nonlinear compression algorithm of this paper appears when approximating a function with a point singularity such as the cusp f_3. The modulus of smoothness $\omega_4(f_3, t)_\tau \approx t^{1/2+2/\tau}$. Therefore, this function is in all of the B^α spaces even though the smoothness in L_∞ is $\omega_4(f_3, t)_\infty \approx t^{1/2}$. The error bounds of Section 5 show that our compression algorithm produces a surface with an error $O(n^{-\alpha/2})$ with $O(n)$ coefficients for any $\alpha < 4$. On the other hand, a typical linear method such as $I_K(f_3)$ achieves an error no smaller than $O(n^{-1/4})$ with $O(n)$ coefficients. We present the original surface with two compressed surfaces in Fig. 3. As few as 70 coefficients produce a compressed surface which is a good approximation to the original, while a compressed surface with 222 coefficients is virtually indistinguishable from the original.

In Table 1, we present some examples of the error of approximation produced by the compression algorithm for various numbers of coefficients. It is interesting to note that the error for the the cusp and ridge functions behaves like $604n^{-2.00}$ and $3.36n^{-1.01}$, respectively, in the number of coefficients n, as expected from our theory. On the other hand, the error for the sombrero function behaves like $860577n^{-2.97}$ for n between 248 and 881, like $24058n^{-2.46}$ for n between 881 and 4794, and like $49.4n^{-1.73}$ for n larger than 4794. This apparently is due to the fact that a higher order asymptotic expansion of the error is of the form

$$C_0 n^{-2} + C_1 n^{-5/2} + C_2 n^{-3},$$

where C_j depends on the $j+4$th derivatives of f_1:

$$\| f_1^{(iv)} \|_\infty \approx 40{,}000, \qquad \| f_1^{(v)} \|_\infty \approx 850{,}000, \quad \text{and} \quad \| f_1^{(vi)} \|_\infty \approx 12{,}000{,}000.$$

The relative sizes of C_0, C_1, and C_2 make the last term dominate for small n, the second term dominate for intermediate n, and the first term dominate for large n.

Acknowledgements

We thank Bert Still for conducting the calculations and rendering the figures.

References

Böhm, W. (1983), Subdividing multivariate splines, Computer-Aided Design 15, 345–352.

de Boor, C. and Fix, G.J. (1973), Spline approximation by quasi-interpolants, J. Approx. Theory 8, 19–45.

de Boor, C. and Höllig, K. (1982), B-splines from parallelpipeds, J. Analyse Math. 42, 99–115.

de Boor, C., Höllig, K. and Riemenschneider, S. (1985) Bivariate cardinal interpolation by splines on a three-direction mesh, Illinois J. Math. 29, 533–566.

Brown, L. and Lucier, B. (1992), Best approximations in L^1 are near best in L^p, $p<1$, Proc. Amer. Math. Soc., to appear.

Chui, C.K., (1988), *Multivariate Splines*, CBMS-NSF Conference Series, Providence, RI.

Cohen, E., Lyche, T. and Riesenfeld, R. (1984), Discrete box splines and refinement algorithms, Computer Aided Geometric Design 1, 131–141.

Dahmen, W. and Micchelli, C. (1984), Subdivision algorithms for the generation of box-spline surfaces, Computer Aided Geometric Design 1, 115–129.

Dahmen, W. and Micchelli, C. (1985a), Local linear independence of translates of a box spline, Studia Math. 82, 243–262.

Dahmen, W. and Micchelli, C. (1985b), Line average algorithm: a method for the computer generation of smooth surfaces, Computer Aided Geometric Design 2, 77–85.

Daubechies, I. (1988), Orthonormal basis of compactly supported wavelets, Comm. Pure Appl. Math. 41, 909–996.

DeVore, R., Jawerth, B. and Popov, V. (1992), Compression of wavelet decompositions, Amer. J. Math., to appear.

DeVore, R. and Popov, V. (1988), Interpolation of Besov spaces, Trans. Amer. Math. Soc. 305, 397–414.

DeVore, R.A. and Sharpley, R. (1984), Maximal functions measuring smoothness, Mem. Amer. Math. Soc. 283.

Frazier, M. and Jawerth, B. (1990), A discrete transform and decompositions of distribution spaces, J. Funct. Anal. 93, 34–170.

Jia, R-Q. (1985), Local linear independence of the translates of a box spline, Constr. Approx. 1, 175–182.

Lyche, T. and Mørken, K. (1987), Knot removal for parametric B-spline curves and surfaces, Computer Aided Geometric Design 4, 217–230.

Meyer, Y. (1989), Wavelets and operators, in: E. Berkson, T. Peck, and J. Uhl, eds., *Analysis at Urbana, Vol. I*, London Math. Soc. Lecture Note Series, Cambridge University Press, Cambridge, UK, 256–365.

Peetre, J. (1976), New Thoughts on Besov spaces, Mathematics Department, Duke University, Durham, NC.