

DESIGN OF HERMITE SUBDIVISION SCHEMES AIDED BY SPECTRAL RADIUS OPTIMIZATION*

BIN HAN[†], MICHAEL L. OVERTON[‡], AND THOMAS P.-Y. YU[§]

Abstract. We present a method for constructing multivariate refinable Hermite interpolants and their associated subdivision algorithms based on a combination of analytical and numerical approaches. As the limit of a linear iterative procedure, the critical L^2 Sobolev smoothness of a refinable Hermite interpolant is given by the spectral radius of a matrix dependent upon the *refinement mask*. The design question is, Given certain constraints (support size, symmetry type, refinement pattern, etc.), how can one choose the refinement mask so that the resulting refinable function has optimal smoothness? This question naturally gives rise to a spectral radius optimization problem. In general, the objective function is not convex and may not be differentiable, or even Lipschitz, at a local minimizer. Nonetheless, a recently developed robust solver for nonsmooth optimization problems may be applied to find local minimizers of the spectral radius objective function. In fact, we find that in specific cases that are of particular interest in the present context, the objective function is smooth at local minimizers and may be accurately minimized by standard techniques. We present two necessary mathematical tricks that make the method practical: (i) compression of matrix size based on symmetry and (ii) efficient computation of gradients of the objective function. We conclude by reporting some computational results.

Key words. subdivision schemes, smoothness analysis, spectral radius optimization, iterative methods

AMS subject classifications. 42C40, 39B12, 15A18, 46E35

DOI. 10.1137/S1064827502408608

1. Introduction to refinable Hermite interpolants. Subdivision algorithms are iterative methods for producing smooth curves and surfaces with a built-in multiresolution structure. They are now used rather extensively in computer-aided geometric design. They are also intimately connected to wavelet bases and their associated fast filter bank algorithms. Formally, a subdivision scheme is given by a linear operator $S := S_{a,M}$ of the form

$$(1.1) \quad Sv(\alpha) = \sum_{\beta \in \mathbb{Z}^s} v(\beta)a(\alpha - M\beta),$$

where $a \in [l^0(\mathbb{Z}^s)]^{m \times m}$ and M is an isotropic dilation matrix, i.e., an $s \times s$ integer matrix M with the properties that $\lim_{n \rightarrow \infty} M^{-n} = 0$ and there exists an invertible $s \times s$ matrix Λ such that

$$(1.2) \quad \Lambda M \Lambda^{-1} = \text{diag}(\sigma_1, \dots, \sigma_s),$$

with $|\sigma_1| = \dots = |\sigma_s|$. Note that S defined by (1.1) can be viewed as an operator on $[l(\mathbb{Z}^s)]^{m' \times m}$ for any m' .

*Received by the editors May 31, 2002; accepted for publication (in revised form) March 23, 2003; published electronically November 11, 2003.

<http://www.siam.org/journals/sisc/25-2/40860.html>

[†]Department of Mathematical and Statistical Sciences, University of Alberta, Edmonton, Alberta, Canada T6G 2G1 (bhan@ualberta.ca). The research of this author was partially supported by Natural Sciences and Engineering Research Council of Canada (NSERC Canada) grant G121210654.

[‡]Courant Institute of Mathematical Sciences, New York University, New York, NY 10012 (overton@cs.nyu.edu).

[§]Department of Mathematical Sciences, Rensselaer Polytechnic Institute, Troy, NY 12180-3590 (yut@rpi.edu). The research of this author was partially supported by NSF CAREER Award CCR 9984501.

A refinable function vector is a vector of functions (or, more generally, distributions) $\phi = [\phi_1, \dots, \phi_m]^T$ that satisfies a two-scale refinement equation of the form

$$(1.3) \quad \phi = \sum_{\alpha \in \mathbb{Z}^s} a(\alpha) \phi(M \cdot -\alpha).$$

When $m = 1$, we refer to the subdivision scheme (resp., refinement equation) as a scalar scheme (resp., equation). “Vector subdivision scheme” or “vector refinement equation” then refers to the case $m > 1$. The connection between subdivision scheme and refinement function is rather clear in the scalar case: Let δ be the Dirac sequence, and assume (for simplicity) that $\mathcal{I}_n S^n \delta := \sum_{\alpha} (S^n \delta)(\alpha) h(M^n \cdot -\alpha)$ (where h is the hat function) converges uniformly to a continuous function ϕ . Then one can verify that ϕ satisfies (1.3). The corresponding connection is somewhat more technical in the general vector case, but for refinable Hermite interpolants considered in this article, the relevant connection between refinable equation and subdivision algorithm is described by (1.5) and (1.6).

In this paper, we focus our attention on a specific class of refinement equation (1.3) with an extra Hermite interpolatory property, although the design principle ought to be applicable in a more general setting. Thus this paper can be viewed as a sequel to [HYP02]. This section reviews the materials necessary for what follows.

For refinable Hermite interpolants of order r , the multiplicity is the combinatorial number $m = \binom{r+s}{r} = \#\Lambda_r$, where Λ_r is the set of all multi-indices ν such that $|\nu| := \sum_{i=1}^s \nu_i \leq r$ and all $\nu_i, i = 1, \dots, s$, are nonnegative integers. We conveniently index the components of ϕ by the elements in Λ_r instead of $\{1, \dots, m\}$. A refinable Hermite interpolant of order r is then a function vector $(\phi_{\nu})_{\nu \in \Lambda_r}$ such that

$$(1.4) \quad \phi \in [C^r(\mathbb{R}^s)]^m \text{ and satisfies } \partial^{\mu} \phi_{\nu}(\alpha) = \delta_{\alpha, \mathbf{0}} \delta_{\mu, \nu}.$$

Such interpolants in the case $s = 2$ have applications in modeling surfaces of arbitrary topological type. However, in order to avoid getting into the technical details of free-form subdivision surfaces, we simply explain their application in the functional setting: Given a refinable Hermite interpolant ϕ , one can smoothly interpolate any prescribed Hermite data $\{v_{\nu}(\alpha) : \nu \in \Lambda_r, \alpha \in \mathbb{Z}^s\}$ by

$$(1.5) \quad \mathcal{I}_{\phi} v := \sum_{\alpha} \sum_{\nu} v_{\nu}(\alpha) \phi_{\nu}(\cdot - \alpha).$$

Moreover, the function $\mathcal{I}_{\phi} v$ can be conveniently computed to any desired resolution by iteratively applying (1.1):

$$(1.6) \quad D_n^{\leq r} \mathcal{I}_{\phi} v = S^n v \quad \forall n = 0, 1, 2, \dots$$

Here, the operator $D_n^{\leq r}$ reads out all the mixed directional derivatives, in directions $M^{-n} e_j, j = 1, \dots, s$, of its operand of order up to r at the lattice points $M^{-n} \alpha, \alpha \in \mathbb{Z}^s$. More formally, let $S(E) := [\partial^{\nu} (E \cdot)^{\mu}](0) / \mu!$ be the matrix that measures how Hermite data changes under a linear change of variable: $g = f(E \cdot) \Rightarrow \partial^{\leq r} g = \partial^{\leq r} f(E \cdot) S(E)$. For smooth f , define $\partial^{\leq r} f(x)$ to be the row vector of length $\#\Lambda_r$ with entries $\partial^{\mu} f(x), \mu \in \Lambda_r$ (Λ_r is ordered lexicographically). Define also $D_n^{\leq r} f \in [l(\mathbb{Z}^s)]^{1 \times m}$ by $D_n^{\leq r} f(\alpha) = \partial^{\leq r} f(M^{-n} \alpha) S(M^{-n})$.

Analysis from [HYP02], based on combining general results on refinement equations [JJ01, CJR02] and elementary properties of Hermite interpolation, shows that, for $\gamma > r + s/2$, equations (1.3) and (1.4) have a unique solution in the Sobolev space $[W_2^{\gamma}(\mathbb{R}^s)]^m (\subseteq C^r(\mathbb{R}^s))$ if and only if we have the following conditions:

(i) The refinement mask a satisfies the Hermite interpolatory condition

$$(1.7) \quad a(M\alpha) = S(M^{-1}) \delta_{\alpha, \mathbf{0}}.$$

(ii) The subdivision operator reproduces Π_{k-1} , i.e.,

$$(1.8) \quad S_{a,M} \left(D_0^{\leq r} p \right) = D_1^{\leq r} p \quad \forall p \in \Pi_{k-1}$$

for some $k \geq \lceil \gamma \rceil + 1$.

(iii) Let $a_n := S_a^n(\delta I_m)$, where $\delta I_m \in [l(\mathbb{Z}^s)]^{m \times m}$ is the matrix sequence which equals the identity matrix I_m at $\alpha = 0$ and the zero matrix elsewhere; then a_n satisfies

$$(1.9) \quad \max \left\{ \lim_{n \rightarrow \infty} \|a_n * v\|_2^{1/n} : v \in H_k \right\} < |\det(M)|^{1/2 - \gamma/s},$$

where $H_k := \{v \in [l^0(\mathbb{Z}^s)]^m : \sum_{\alpha \in \mathbb{Z}^s} (D_0^{\leq r} p)(-\alpha)v(\alpha) = 0 \quad \forall p \in \Pi_{k-1}(\mathbb{R}^s)\}$.

(iv) The matrix $J := \frac{1}{|\det M|} \sum_{\alpha} a(\alpha)$ has 1 as a simple eigenvalue and all the other eigenvalues have modulus less than $|\det M|^{-\lceil \gamma \rceil/s}$.

By carefully examining the arguments in the two papers [JJ01, CJR02], one sees that, in fact, condition (iv) is redundant, i.e., it is implied by (i)–(iii).

In surface modeling applications, we also need symmetry,

$$(1.10) \quad a(E\alpha) = S(M^{-1}EM)a(\alpha)S(E^{-1}) \quad \forall \alpha \in \mathbb{Z}^s, \forall E \in G,$$

where G is a *symmetry group with respect to a dilation matrix M* , i.e., G is a finite set of integer matrices with determinants equal to ± 1 and forms a group under matrix multiplication; moreover, $MEM^{-1} \in G$ for all $E \in G$ [Han, HYP02]. Of particular interest are

- the *hexagonal symmetry group* (also known as D_6) with respect to $M = 2I_2$,

$$(1.11) \quad D_6 = \left\{ \pm \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \pm \begin{bmatrix} 0 & -1 \\ 1 & -1 \end{bmatrix}, \pm \begin{bmatrix} -1 & 1 \\ -1 & 0 \end{bmatrix}, \right. \\ \left. \pm \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \pm \begin{bmatrix} 1 & -1 \\ 0 & -1 \end{bmatrix}, \pm \begin{bmatrix} -1 & 0 \\ -1 & 1 \end{bmatrix} \right\};$$

- the *square symmetry group* (also known as D_4) with respect to $M = 2I_2$,

$$(1.12) \quad D_4 = \left\{ \pm \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \pm \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, \pm \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \pm \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \right\};$$

- D_6 with respect to the $\sqrt{3}$ -dilation matrix $M = M_{\sqrt{3}} := \begin{bmatrix} 1 & -2 \\ 2 & -1 \end{bmatrix}$;
- D_4 with respect to the quincunx dilation matrix $M = M_{\text{Quincunx}} := \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$.

The reason why we work with L^2 Sobolev spaces is that the left-hand side of (1.9) equals the spectral radius of a linear operator restricted to a *finite-dimensional* invariant subspace. Consider the transition operator $\mathcal{T}_a : [l(\mathbb{Z}^s)]^{m \times m} \rightarrow [l(\mathbb{Z}^s)]^{m \times m}$:

$$(1.13) \quad \mathcal{T}_a u(\alpha) := \sum_{\beta, \gamma} a(M\alpha - \beta)u(\beta + \gamma)a(\gamma)^*.$$

For a matrix mask a , we denote $\text{supp}(a) := \{\beta \in \mathbb{Z}^s : a(\beta) \neq 0\}$. Define a set K_M^a by

$$(1.14) \quad K_M^a := \sum_{j=1}^{\infty} M^{-j}(\text{supp}(a) - \text{supp}(a)),$$

where for two sets S_1 and S_2 , $S_1 - S_2 := \{s_1 - s_2 : s_1 \in S_1, s_2 \in S_2\}$. We define $l(K_M^a) := \{u \in l(\mathbb{Z}^s) : u(\beta) = 0 \quad \forall \beta \in \mathbb{Z}^s \setminus K_M^a\}$. By results in the general theory of refinement equations (see, e.g., [Han02, JJ01]), conditions (i) and (ii) above guarantee that $W_k := \{w = u * v^* : u, v \in H_k\}$ is an invariant subspace of \mathcal{T}_a . Let the set K_M^a be defined in (1.14). In fact, $W_k \cap [l(K_M^a)]^{m \times m}$ has finite dimension and we have

$$(1.15) \quad \max \left\{ \lim_{n \rightarrow \infty} \|a_n * v\|_2^{1/n} : v \in H_k \right\} = \sqrt{\rho(\mathcal{T}_a|_{W_k \cap [l(K_M^a)]^{m \times m}})}.$$

If the mask a does not reproduce Π_k and (1.9) holds for some $\gamma > r + s/2$, then the critical L^2 Sobolev smoothness of ϕ equals

$$(1.16) \quad \frac{s}{2} \left(2 - \log_{|\det M|} \rho(\mathcal{T}_a|_{W_k \cap [l(K_M^a)]^{m \times m}}) \right).$$

2. Accuracy order versus smoothness: A computational design principle based on spectral radius minimization. The basic theory reviewed in section 1 suggests the following recipe for constructing refinable Hermite interpolants for a given dilation matrix M and symmetry group G with respect to M :

- (I) Pick a finite G -symmetric support of the mask a , i.e., $\text{supp}(a) \subseteq \mathbb{Z}^s$ such that $\alpha \in \text{supp}(a)$ implies $E\alpha \in \text{supp}(a)$ for all $E \in G$.
- (II) Pick a target polynomial reproduction order k .
- (III) Solve the system of *linear* equations determined by (1.7), (1.8), and (1.10):
 - if no solution is found, enlarge $\text{supp}(a)$ or reduce k , and return to (III)
 - otherwise $a = a(\mathbf{t})$, where $\mathbf{t} \in \mathbb{R}^L$ and L equals the nullity of the linear system in (III).
- (IV) Solve

$$(2.1) \quad \min_{\mathbf{t}} \rho(\mathcal{T}_{a(\mathbf{t})}|_{W_k \cap [l(K_M^a)]^{m \times m}}).$$

Steps (I)–(III) had been used in [HYP02] to construct refinable Hermite interpolants with small support. The final optimization step (IV), as we shall see in section 5, becomes useful when we consider schemes with support of moderate sizes.

It is known to waveleticians that the *approximation order* and *smoothness* of a refinable function have a subtle relationship, except in very special cases (e.g., when ϕ is a spline, where the two properties go hand-in-hand [CDR96]). Under certain conditions, a C^k refinable function vector guarantees to provide approximation order $k + 1$. In the case of refinable Hermite interpolants, the latter condition is equivalent to (1.8).

In short, *smooth refinable functions provide good approximation orders* [Ron97]; the converse implication, however, is not true. One consequence is that when one is interested in optimal smoothness, one should not necessarily optimize approximation order. (The latter is a general practice in construction of wavelet and related systems, because approximation orders typically boil down to simple algebraic conditions on the refinement mask and are easier to work with.) To put it in a different way, within a certain regime one can *trade off some approximation orders for smoothness*. The goal of this article is to discuss how one can capitalize on this observation to construct smooth refinable Hermite interpolants.

The above computational optimization principle has been used in other settings by Daubechies and by Dyn et al. In these applications, however, both the spatial dimension and the multiplicity is 1; the number of parameters is quite small (one or two), so relatively ad hoc optimization methods suffice. For example, in Daubechies [Dau93], an easily computable upper bound, rather than the honest Hölder exponent,

was optimized, and it happened that a smoother wavelet was discovered. For our multivariate problems with typically at least half a dozen variables, we advocate an optimization objective based on the L^2 Sobolev exponent and a robust solver for nonsmooth optimization problems.

3. Reducing matrix size using symmetry. In principle, the right-hand side of (1.15) is just the spectral radius of a finite matrix. Nevertheless it seems computationally troublesome to explicitly form a matrix representation of $\mathcal{T}_a|_{W_k \cap [l(K_M^a)]^{m \times m}}$, where the set K_M^a is defined in (1.14). Jia and Jiang [JJ01] provide the following simple formula for computing $\rho(\mathcal{T}_a|_{W_k \cap [l(K_M^a)]^{m \times m}})$. Let $b(\alpha) := \sum_{\beta} \overline{a(\beta)} \otimes a(\alpha + \beta) / |\det(M)|$. Consider the size $m^2(\#K_M^a)$ matrix

$$(3.1) \quad F := (b(M\alpha - \beta))_{\alpha, \beta \in K_M^a}.$$

If $J := \sum_{\alpha} a(\alpha) / |\det M|$ satisfies $\text{spec}(J) = \{\eta_1 = 1, \eta_2, \dots, \eta_m\}$, $|\eta_j| < 1$ for $j > 1$, then

$$(3.2) \quad \max \left\{ \lim_{n \rightarrow \infty} \|a_n * v\|_2^{1/n} : v \in H_k \right\} = \sqrt{\rho(\mathcal{T}_a|_{W_k \cap [l(K_M^a)]^{m \times m}})} = \sqrt{\rho_k |\det M|},$$

where $\rho_k = \max\{|\nu| : \nu \in \text{spec}(F) \setminus E_k\}$, where

$$(3.3) \quad E_k = \{\eta_j \overline{\sigma^{-\mu}}, \overline{\eta_j} \sigma^{-\mu} : |\mu| < k, j = 2, \dots, m\} \cup \{\sigma^{-\mu} : |\mu| < 2k\}.$$

In fact, if the mask a does not reproduce Π_k and the shifts of ϕ are stable, then by (1.16) the critical L^2 Sobolev smoothness of ϕ_a is equal to

$$(3.4) \quad -(\log_{|\det M|} \rho_k) s / 2.$$

Despite the simplicity of Jia and Jiang’s algorithm, the matrix (3.1) is exceedingly large for the bivariate examples we are interested in in section 5. It had already been shown in Han [Han01] that, in the scalar case, one can substantially reduce the size of the matrix by making use of *symmetry*. We find that the computational savings are even more significant (see section 5) in the vector case. We shall follow along the lines developed in [Han01] to generalize the algorithm in the scalar case there to the vector case.

Let G be a symmetry group with respect to a dilation matrix M . We define the operators $\Theta, \Theta^* : [l(\mathbb{Z}^s)]^{m \times m} \mapsto [l(\mathbb{Z}^s)]^{m \times m}$ by

$$\Theta(u)(\alpha) := \frac{1}{\#G} \sum_{E \in G} S(E)u(E^{-1}\alpha)S(E)^T$$

and

$$\Theta^*(u)(\alpha) := \frac{1}{\#G} \sum_{E \in G} S(E)^T u(E\alpha)S(E).$$

Again the transition operator $\mathcal{T}_a : [l(\mathbb{Z}^s)]^{m \times m} \mapsto [l(\mathbb{Z}^s)]^{m \times m}$ is given by

$$\mathcal{T}_a u(\alpha) := \sum_{\beta, \gamma \in \mathbb{Z}^s} a(M\alpha - \beta)u(\beta + \gamma)a(\gamma)^*.$$

Define the convolved subdivision operator $\mathcal{S}_a : [l(\mathbb{Z}^s)]^{m \times m} \mapsto [l(\mathbb{Z}^s)]^{m \times m}$ by

$$\mathcal{S}_a u(\alpha) := \sum_{\beta, \gamma \in \mathbb{Z}^s} a(M\beta - \alpha + \gamma)^* u(\beta)a(\gamma),$$

where A^* denotes the complex conjugate of the transpose of A . Define an inner product on $[l^0(\mathbb{Z}^s)]^{m \times m}$ as follows:

$$\langle u, v \rangle := \text{trace} \left(\sum_{\beta \in \mathbb{Z}^s} u(\beta)v(\beta)^* \right), \quad [l^0(\mathbb{Z}^s)]^{m \times m}.$$

PROPOSITION 3.1. *Let a be a mask satisfying (1.10). Then $\Theta(\mathcal{T}_a u) = \mathcal{T}_a \Theta(u)$ for all $u \in [l^0(\mathbb{Z}^s)]^{m \times m}$. Moreover, for any $u, v \in [l^0(\mathbb{Z}^s)]^{m \times m}$, we have $\langle \mathcal{T}_a u, v \rangle = \langle u, \mathcal{S}_a v \rangle$ and $\Theta^*(\mathcal{S}_a u) = \mathcal{S}_a \Theta^*(u)$.*

Proof. We have

$$\begin{aligned} \mathcal{T}_a \Theta(u)(\alpha) &= \sum_{\beta, \gamma \in \mathbb{Z}^s} a(M\alpha - \beta)\Theta(u)(\beta + \gamma)a(\gamma)^* \\ &= \frac{1}{\#G} \sum_{E \in G} \sum_{\beta, \gamma \in \mathbb{Z}^s} a(M\alpha - \beta)S(E)u(E^{-1}\beta + E^{-1}\gamma)S(E)^T a(\gamma)^* \\ &= \frac{1}{\#G} \sum_{E \in G} \sum_{\beta, \gamma \in \mathbb{Z}^s} S(M^{-1}EM)a(E^{-1}M\alpha - E^{-1}\beta) \\ &\quad u(E^{-1}\beta + E^{-1}\gamma)a(E^{-1}\gamma)^* S(M^{-1}EM)^T \\ &= \frac{1}{\#G} \sum_{E \in G} S(M^{-1}EM) \\ &\quad \sum_{\beta, \gamma \in \mathbb{Z}^s} a(MM^{-1}E^{-1}M\alpha - \beta)u(\beta + \gamma)a(\gamma)^* S(M^{-1}EM)^T \\ &= \frac{1}{\#G} \sum_{E \in G} S(E) \sum_{\beta, \gamma \in \mathbb{Z}^s} a(ME^{-1}\alpha - \beta)u(\beta + \gamma)a(\gamma)^* S(E)^T \\ &= \frac{1}{\#G} \sum_{E \in G} S(E)\mathcal{T}_a u(E^{-1}\alpha)S(E)^T \\ &= \Theta(\mathcal{T}_a u). \end{aligned}$$

The other claims can be proved by similar computation. \square

The following result generalizes the algorithm in [Han01] to the vector case and is the main result in this section.

THEOREM 3.2. *Let a be a Hermite interpolatory mask such that (1.8) and (1.10) are satisfied. Let K_M^a be the set defined in (1.14). Then we have*

$$(3.5) \quad \rho(\mathcal{T}_a|_{W_k \cap [l(K_M^a)]^{m \times m}}) = \rho(\mathcal{T}_a|_{\Theta(W_k \cap [l(K_M^a)]^{m \times m})}).$$

Moreover,

$$(3.6) \quad \text{spec}(\mathcal{T}_a|_{\Theta(W_k \cap [l(K_M^a)]^{m \times m})}) = \text{spec}(\mathcal{T}_a|_{\Theta([l(K_M^a)]^{m \times m})}) \setminus \text{spec}(\mathcal{S}_a|_{\Theta^*(U_k)}),$$

where the space U_k is defined to be

$$\begin{aligned} U_k := \text{span} \left\{ v D_0^{\leq r} p, [v D_0^{\leq r} p]^T, \left(\sum_{\mu \in \mathbb{Z}_+^s} \sum_{0 \leq \nu \leq \mu, \nu \in \Lambda_r, \mu - \nu \in \Lambda_r} p^{(\mu)}(\beta) E_{\nu, \mu - \nu} \right)_{\beta \in \mathbb{Z}^s} \right. \\ \left. : v \in [\mathbb{C}]^{m \times 1}, p \in \Pi_{k-1} \right\}, \end{aligned}$$

where $E_{\alpha,\beta}$ is the $\#\Lambda_r \times \#\Lambda_r$ matrix with the (α, β) entry equal to 1 and all other entries equal to 0.

The proof of the above result is largely similar to that given in [Han01] for the scalar case, with Proposition 3.1 being a key step; we omit the details in this computational paper.

A finer analysis, based on combining the ideas in [Han01] and [JJ01], shows that $\text{spec}(\mathcal{S}_a|_{\Theta^*(U_k)})$ is a subset of $|\det M|E_k$, where the set E_k is defined in (3.3). More specifically, we can compute $\text{spec}(\mathcal{S}_a|_{\Theta^*(U_k)})$ by the following simple procedure:

$$(3.7) \quad \begin{aligned} \text{spec}(\mathcal{S}_a|_{\Theta^*(U_k)}) &= \text{spec}(\tau_{M,J}|_{\Pi_{k-1,m,\Theta^*}}) \cup \overline{\text{spec}(\tau_{M,J}|_{\Pi_{k-1,m,\Theta^*}})} \\ &\cup \text{spec}(\tau_M|_{\Theta^*(\Pi_{2k-1} \setminus \Pi_{k-1})}) \setminus \text{spec}(\tau_M|_{\Theta^*(\Pi_{k-1})}), \end{aligned}$$

where $J := \sum_{\beta \in \mathbb{Z}^s} a(\beta)/|\det M|$, $\Pi_{k-1,m,\Theta^*} = \{\sum_{E \in G} S(E)^T p(Ex) : p \in (\Pi_{k-1})^{m \times 1}\}$, and

$$\tau_{M,J}(p)(x) := |\det M|J^*p(M^{-1}x), \quad p \in \Pi_{k-1,m,\Theta^*},$$

and

$$\tau_M(p)(x) := |\det M|p(M^{-1}x), \quad p \in \Pi_{2k-1}.$$

In order to compute $\text{spec}(\mathcal{S}_a|_{\Theta^*(U_k)})$ by the above procedure, we see that the procedure only depends on J , the dilation matrix M , and the symmetry group G . So, in general, we need to compute the set $\text{spec}(\mathcal{S}_a|_{\Theta^*(U_k)})$ only once.

In the particular case $G = \{I_s\}$ (that is, no symmetry is assumed), we have $\Pi_{k-1,m,\Theta^*} = (\Pi_{k-1})^{m \times 1}$ and $\Theta^*(\Pi_{2k-1}) = \Pi_{2k-1}$. By (1.2), $M = \Lambda D \Lambda^{-1}$, where $D = \text{diag}(\sigma_1, \dots, \sigma_s)$. A simple argument shows that $\text{spec}(\tau_M|_{\Pi_{2k-1}}) = \text{spec}(\tau_D|_{\Pi_{2k-1}}) = \{|\det M|\sigma^{-\mu} : |\mu| < 2k\}$. On the other hand, one can easily verify that

$$\begin{aligned} \text{spec}(\tau_{M,J}|_{(\Pi_{k-1})^{m \times 1}}) &= \{\eta\lambda : \eta \in \text{spec}(J^*), \lambda \in \text{spec}(\tau_M|_{\Pi_{k-1}})\} \\ &= \{|\det M|\eta_j\sigma^{-\mu} : j = 1, \dots, s, |\mu| < k\}. \end{aligned}$$

Hence, in the special case $G = \{I_s\}$, we have $\text{spec}(\mathcal{S}_a|_{U_k}) = |\det M|E_k$, which recovers Jia and Jiang's result in (3.3).

When $M = 2I_s$, one has

$$(3.8) \quad \begin{aligned} \text{spec}(\mathcal{S}_a|_{\Theta^*(U_k)}) &= \{\eta_j 2^{1-2r} \text{ with multiplicity } m_G(2r) \quad : \quad j = 2, \dots, m; r = 0, \dots, k/2 - 1\} \\ &\cup \{2^{-2r} \text{ with multiplicity } m_G(2r) \quad : \quad r = 0, \dots, k - 1\}, \end{aligned}$$

where the quantity $m_G(2r), r \in \mathbb{N}$, is defined in [Han01].

The following procedure tells us how to compute the set K_M^a in (1.14).

PROPOSITION 3.3. *Let a be a mask. Let K_0 be a finite subset of \mathbb{Z}^s such that $K_M^a \subset K_0$ (that is, K_0 is a guessing set which is large enough to contain K_M^a). Define*

$$K_j = K_{j-1} \cap M^{-1}(K_{j-1} + (\text{supp } a - \text{supp } a)), \quad j \in \mathbb{N}.$$

Then a positive integer j exists such that $K_j = K_{j-1}$ and consequently $K_j = K_M^a$.

Proof. Clearly, $K_j \subset K_{j-1}$ for all $j \in \mathbb{N}$. Since K_0 is a finite subset, there must exist a positive integer j such that $K_{j-1} = K_j$.

By the definition of K_M^a and $K_M^a \subset K_0$, by induction, one can easily prove that $K_M^a \subset K_j$ for all $j \in \mathbb{N}$. Therefore, $K_M^a \subset K_j$. To complete the proof, one needs to show that $K_M^a \supset K_j$. By the definitions of K_j and K_M^a , one can deduce that

$$\bigcap_{k=j}^{\infty} K_k \subset K_M^a.$$

Since $K_j = K_{j+1} = \dots$, we deduce that $K_j = \bigcap_{k=j}^{\infty} K_k \subset K_M^a$. \square

The set K_M^a plays an important role in the analysis of various properties of subdivision schemes. For example, if W is a subspace of $[l^0(\mathbb{Z}^s)]^{m \times m}$ and $\mathcal{T}_a W \subset W$, then it was proved in [HJ98] that $\text{spec}(\mathcal{T}_a|_W) \cup 0 = \text{spec}(\mathcal{T}_a|_{W \cap [l(K_M^a)]^{m \times m}}) \cup 0$. For more detail and discussion on the set K_M^a and the eigenvalue property of $\mathcal{T}_a|_{[l(K_M^a)]^{m \times m}}$, the reader is referred to Han and Jia [HJ98].

Finally, we point out that a basis for $\Theta([l(K_M^a)]^{m \times m})$ can be easily constructed using the definition of the operator Θ and the standard basis for $[l(K_M^a)]^{m \times m}$. Under such a basis for the space $\Theta([l(K_M^a)]^{m \times m})$, the representation matrix for the transition operator \mathcal{T}_a can be easily obtained and in fact is a “folded version” of the matrix F in (3.1).

4. Minimization of the spectral radius. In step IV of the algorithm given in section 2, we need to solve a spectral radius minimization problem. First consider the general problem of finding minimizers of

$$(4.1) \quad f(\mathbf{t}) := \rho(A(\mathbf{t})),$$

where the $n \times n$ matrix A depends smoothly on $\mathbf{t} \in \mathbb{R}^L$ and ρ denotes spectral radius. The function f is nonconvex, nonsmooth, and, in fact, non-Lipschitz. We cannot expect to find an efficient algorithm to find global minimizers of f ; indeed, it is known that related spectral minimization problems are NP-hard [BT97, Nem93]. However, we may search for local minimizers. A key point is that very often, local minimizers of spectral functions such as f are points at which the associated matrix has multiple eigenvalues and the spectral function is not differentiable [BLO01b, BLO01a]. Consequently, standard optimization methods, such as Newton’s method, BFGS, or conjugate gradient, are not generally effective tools for finding local minimizers of spectral functions. Other more ad hoc methods, such as the Nelder–Mead simplex method, are typically very slow and do not produce accurate approximations to local minimizers. However, a robust minimization method for approximating local minimizers of general nonsmooth functions, specifically including spectral functions, has been recently developed [BLO02]. We have used this method to find local minimizers of f , and in our early investigations using somewhat differently defined spectral radius objective functions we indeed found minimizers at which the objective function was not smooth. However, for the problems of greatest interest in the present context, we find local minimizers where f is, in fact, differentiable. Consequently the minimization is, somewhat to our surprise, amenable to standard techniques such as BFGS.

In order to apply either the method of [BLO02] or the more standard BFGS method, we need a formula for the gradient of f where it is defined. Let us consider a fixed parameter vector $\hat{\mathbf{t}}$ and allow only its j th component to vary, say $t_j = (\hat{\mathbf{t}})_j + s$, where s is a single real parameter. We now consider the variation of the matrix and its spectral radius as a function of s around 0, say

$$(4.2) \quad \tilde{A}(s) = A(\hat{\mathbf{t}} + s\mathbf{e}), \quad \tilde{f}(s) = \rho(\tilde{A}(s)),$$

where \mathbf{e} is the j th unit vector in \mathbb{R}^L . Let $\hat{\lambda} \in \mathbb{C}$ be a given eigenvalue of $\tilde{A}(0)$, and suppose that it is simple, i.e., its algebraic multiplicity is one. Because roots of polynomials are continuous functions of their coefficients, it is clear that for all $\delta > 0$, there exists an $\epsilon > 0$ such that for all $|s| < \epsilon$, $\tilde{A}(s)$ has only one eigenvalue within δ of $\hat{\lambda}$. Let us denote this eigenvalue (which is a function of s and well defined near $s = 0$) by $\lambda(s)$, with $\lambda(0) = \hat{\lambda}$. Since $\tilde{A}(s)$ is differentiable, and $\hat{\lambda}$ is a simple eigenvalue, it is known [HJ85, Theorem 6.3.12] that λ is differentiable as a function of s and that its derivative is given by

$$(4.3) \quad \lambda'(s)|_{s=0} = u^* \left(\tilde{A}'(s) \Big|_{s=0} \right) v,$$

where v and u , respectively, denote right and left eigenvectors of $\tilde{A}(0)$, defined by

$$(4.4) \quad \tilde{A}(0)v = \hat{\lambda}v, \quad u^* \tilde{A}(0) = \hat{\lambda}u^*,$$

assuming the normalization $u^*v = 1$. (We note that u^*v cannot be zero since $\hat{\lambda}$ is a simple eigenvalue; see [HJ85, Lemma 6.3.10].) Since λ is complex-valued, its derivative is also, but the derivative is with respect to the real parameter s .

Now we turn our attention to variational properties of the spectral radius function $\tilde{f}(s)$ at $s = 0$. Clearly only the eigenvalues with maximum modulus, which we denote the *active eigenvalues*, are relevant. In fact, it is not difficult to see that \tilde{f} is differentiable at $s = 0$ if either exactly one real eigenvalue of $\tilde{A}(0)$ is active or one complex conjugate pair of eigenvalues of $\tilde{A}(0)$ are active (because $\tilde{A}(s)$ is real, its nonreal eigenvalues must occur in conjugate pairs). In the first case, let us denote the active real eigenvalue by $\hat{\lambda}$, and in the second case, let us denote the active conjugate pair by $(\hat{\lambda}, \bar{\hat{\lambda}})$. Note that $\hat{\lambda}$ cannot be zero except in the trivial case $n = 1$, which we exclude. With $\hat{\lambda}$ defined, the function $\lambda(s)$ converging to $\hat{\lambda}$ is also well defined near $s = 0$. Now consider the complex modulus function $\omega : \mathbb{C} \rightarrow \mathbb{R}$, defined by $\omega(z) = |z|$. This function does not have a *complex* derivative, but if we identify \mathbb{C} with \mathbb{R}^2 and consider its real derivative as a function mapping \mathbb{R}^2 to \mathbb{R} , we find that

$$(4.5) \quad \omega'(z) = \frac{z}{|z|}$$

provided $z \neq 0$. (This may be interpreted as a real gradient vector, expressed using complex notation.) Now we use the ordinary chain rule to combine (4.3) and (4.5) and obtain

$$(4.6) \quad \tilde{f}'(s) \Big|_{s=0} = \operatorname{Re} \left(\frac{\bar{\hat{\lambda}}}{|\hat{\lambda}|} u^* \tilde{A}'(s) \Big|_{s=0} v \right).$$

The complex conjugate arises because the inner product on \mathbb{C} used in the chain rule is defined by $\langle y, z \rangle = \operatorname{Re}(\bar{y}z)$; alternatively, we may avoid this confusion by carrying out the chain rule using \mathbb{R}^2 instead of \mathbb{C} . Notice that, if we replace $\hat{\lambda}$ in this derivation by its conjugate, we must also conjugate the eigenvectors v and u defined in (4.4), so the result remains unchanged. Also note that the scaling of either u or v is arbitrary, but once one is determined, the other is also, because of the normalization $u^*v = 1$.

Consistently normalized eigenvectors corresponding to a simple eigenvalue are locally continuous functions of the associated matrix, so the directional derivative (4.6) is locally continuous as $\hat{\mathbf{t}}$ is varied. We therefore conclude that the spectral

radius function $f(\mathbf{t})$ is differentiable at $\hat{\mathbf{t}}$ provided that $A(\hat{\mathbf{t}})$ has exactly one real active eigenvalue or one complex conjugate pair of active eigenvalues and that its gradient may be computed from (4.6).

We note in passing that if two or more eigenvalues of $A(\hat{\mathbf{t}})$ with distinct imaginary part are active, i.e., have modulus equal to the maximum modulus, the spectral radius function f is typically not differentiable at $\hat{\mathbf{t}}$, but if each of the active eigenvalues is simple (has multiplicity one), then f may be represented locally near $\hat{\mathbf{t}}$ as the maximum of two or more smooth functions that coincide at $\hat{\mathbf{t}}$. However, if any active eigenvalue has multiplicity greater than one, the variational properties of f at $\hat{\mathbf{t}}$ are considerably more complicated; a detailed analysis is given by Burke and Overton [BO01].

Finally, we consider the matrix function defined in section 2 whose spectral radius is to be minimized, i.e., $\mathcal{T}_{a(\mathbf{t})}|_{W_k \cap [l(K_M^a)]^{m \times m}}$ in (2.1). A key point is that the subspace W_k is independent of \mathbf{t} . Just as there is no need to construct a basis for W_k to compute the spectral radius of this matrix (see (3.2)), there is no need to construct a basis to compute its gradient. Since the eigenvalues of $\mathcal{T}_{a(\mathbf{t})}|_{W_k \cap [l(K_M^a)]^{m \times m}}$ are precisely the eigenvalues of F (see (3.1)) that are not in the set E_k , we simply compute the eigenvalues of F , remove those in the set E_k , find the largest in absolute value that remains, and construct the gradient via (4.6) using the corresponding left and right eigenvectors of F for u and v and the partial derivative $\partial F / \partial \mathbf{t}_j$ for $\tilde{A}'(0)$ for $j = 1, \dots, L$.

The same principle above can be applied when we compress matrix size based on the formulas (3.5), (3.6), and (3.7).

5. Computational results. It was proved in [Han00] that no C^2 interpolating dyadic refinable function can be supported on $[-3, 3]^s$ and no C^1 dyadic refinable function can be supported on $[-1, 1]^s$ in any dimension s . We use our computational optimization approach to explore how much smoothness one can gain by increasing r from 0 to 1.

We consider here D_6 -symmetric Hermite interpolatory schemes with the following (D_6 -symmetric) supports; the corresponding ϕ 's are supported on $[-3, 3]^2$.

- TWO-POINT SCHEME. $\text{supp}(a) = \{(0, 0), \pm(1, 0), \pm(0, 1), \pm(1, 1)\}$,
- DIAMOND SCHEME. $\text{supp}(a) = \{(0, 0), \pm(1, 0), \pm(0, 1), \pm(1, 1), \pm(-1, 1), \pm(2, 1), \pm(1, 2)\}$,
- BUTTERFLY SCHEME. $\text{supp}(a) = \{(0, 0), \pm(1, 0), \pm(0, 1), \pm(1, 1), \pm(-1, 1), \pm(2, 1), \pm(1, 2), \pm(-1, 2), \pm(-2, 1), \pm(1, 3), \pm(3, 1), \pm(3, 2), \pm(2, 3)\}$,
- TEN-POINT SCHEME. $\text{supp}(a) = \{(0, 0), \pm(1, 0), \pm(0, 1), \pm(1, 1), \pm(-1, 1), \pm(2, 1), \pm(1, 2), \pm(-1, 2), \pm(-2, 1), \pm(1, 3), \pm(3, 1), \pm(3, 2), \pm(2, 3), \pm(0, 3), \pm(3, 0), \pm(3, 3)\}$.

(For justification of the names of the schemes above, see [Yu01, Figure 1].) One finds by computation that the numbers of parameters in the refinement mask (= nullity in the linear systems in step (III) of the procedure in section 2) are given by Table 5.1.

Here we are interested in bivariate interpolatory Hermite subdivision schemes with $r = 1$ which produce limit surfaces which are at least twice continuously differentiable. To guarantee that $\phi \in [C^2(\mathbb{R}^2)]^3$, it suffices, thanks to a Sobolev embedding theorem, to construct ϕ such that $\phi \in [W_2^\gamma(\mathbb{R}^2)]^3$ and $\gamma > 3$. The latter condition implies that the associated subdivision operator must reproduce Π_3 . Therefore, we apply our optimization method to the families DIAMOND($k = 3$), BUTTERFLY($k = 3, 4$), and TEN-POINT($k = 3, 4$).

TABLE 5.1

Number of parameters in the symmetric refinement mask reproducing Π_k . The first column corresponds to refinement masks which satisfy only symmetry but no polynomial reproducibility of any degree. (n.s. stands for no solution in the associated linear system.)

k		0	1	2	3	4	5	6
TWO-POINT	5	4	2	1	n.s.	n.s.	n.s.	n.s.
DIAMOND	10	9	7	5	1	n.s.	n.s.	n.s.
BUTTERFLY	19	18	16	14	10	7	1	n.s.
TEN-POINT	24	23	21	19	15	12	6	3

DIAMOND($k = 3$). We plot the critical L^2 Sobolev exponent (1.16) versus the single parameter t on which the mask is dependent. In Figure 5.1, we show the objective function in the regions $t \in [-0.5, 0.5]$, $t \in [-0.05, 0.05]$, and $t \in [-0.005, 0.005]$; these plots strongly suggest that the objective function is globally smooth (differentiable at least) and has a unique maximizer (recall that minimizing the spectral radius (1.16) is the same as maximizing smoothness) near $t = 0$. One sees that in this parametric family, the highest L^2 Sobolev smoothness found is less than 2.75, failing to infer the existence of a C^2 scheme.

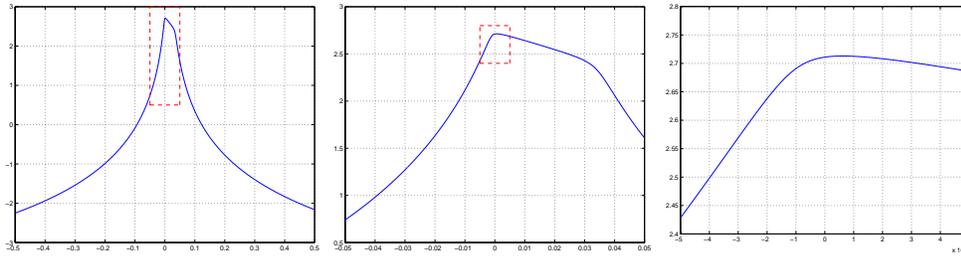


FIG. 5.1.

BUTTERFLY($k = 3, 4$). For the 10 and 7 parameter families the following masks occur to give optimal L^2 Sobolev smoothness within each family:

$$\begin{aligned}
 a(1, 0) &= \begin{bmatrix} 0.4960 & -0.8380 & 0.4190 \\ 0.1370 & -0.1974 & 0.1701 \\ 0 & 0 & 0.1428 \end{bmatrix}, \\
 a(1, 2) &= \begin{bmatrix} 0.0240 & 0 & -0.2651 \\ -0.0015 & -0.0011 & -0.0437 \\ -0.0030 & 0 & -0.0885 \end{bmatrix}, \\
 a(-1, 2) &= \begin{bmatrix} -0.0100 & 0.0113 & 0.0465 \\ 0.0095 & 0.0070 & -0.0032 \\ 0.0025 & 0.0071 & 0.0139 \end{bmatrix}.
 \end{aligned}$$

Critical L^2 Sobolev exponent = 3.5227.

$$\begin{aligned}
 a(1, 0) &= \begin{bmatrix} 0.5000 & -0.8529 & 0.4265 \\ 0.1338 & -0.1961 & 0.1591 \\ 0 & 0 & 0.1221 \end{bmatrix}, \\
 a(1, 2) &= \begin{bmatrix} 0.0181 & 0 & -0.2809 \\ -0.0022 & 0.0016 & -0.0458 \\ -0.0044 & 0 & -0.0899 \end{bmatrix}, \\
 a(-1, 2) &= \begin{bmatrix} -0.0090 & 0.0034 & 0.0428 \\ 0.0078 & 0.0081 & -0.0057 \\ 0.0022 & 0.0049 & 0.0105 \end{bmatrix}.
 \end{aligned}$$

Critical L^2 Sobolev exponent = 3.4031.

The other entries of the mask are given by symmetry relation (1.10). Notice the following:

- C^2 schemes are found in both families.
- Smoothness/accuracy order trade-off: The smoothest scheme in BUTTERFLY ($k = 3$) has accuracy order 4 and is smoother than the smoothest scheme in BUTTERFLY ($k = 4$), which has accuracy order 5.
- In this case, $|K_M^a| = 109$, so a direct use of Jia and Jiang's algorithm requires computing the spectrum of matrices of size $109 \times 3^2 = 981$; our method based on (3.6) reduces the matrix size down to 87.

TEN-POINT($k = 4$). The smoothest mask found is

$$\begin{aligned}
 a(1, 0) &= \begin{bmatrix} 0.4659 & -0.9156 & 0.4578 \\ 0.1651 & -0.2836 & 0.2613 \\ 0 & 0 & 0.2389 \end{bmatrix}, \\
 a(1, 2) &= \begin{bmatrix} 0.0007 & 0 & -0.1531 \\ 0.0007 & -0.0387 & -0.0064 \\ 0.0013 & 0 & -0.0516 \end{bmatrix}, \\
 a(-1, 2) &= \begin{bmatrix} -0.0004 & 0.0382 & 0.0532 \\ -0.0008 & -0.0034 & 0.0079 \\ -0.0007 & 0.0062 & 0.0213 \end{bmatrix}, \\
 a(3, 0) &= \begin{bmatrix} 0.0341 & -0.0421 & 0.0211 \\ 0.0085 & -0.0120 & 0.0087 \\ 0 & 0 & 0.0054 \end{bmatrix}.
 \end{aligned}$$

Critical L^2 Sobolev exponent = 4.0321.

This implies the existence of a C^3 scheme in the TEN-POINT($k = 4$) family. In this case $|K_M^a| = 127$, so a direct use of Jia and Jiang's algorithm requires computing the spectrum of matrices of size $127 \times 3^2 = 1143$; our method based on (3.6) reduces the matrix size down to 101.

As far as smoothness is concerned there is now no reason to consider the family TEN-POINT($k = 3$). This is because any scheme in TEN-POINT($k = 3$) \setminus TEN-POINT($k = 4$), which has accuracy order exactly 4, cannot have a L^2 Sobolev smoothness > 4 .

MATLAB programs for reproducing the results in this section can be obtained by contacting the authors through email.

The spectral radius optimization problems we encounter in this paper turn out to be simpler than one would expect. For one thing, we don't have trouble finding apparently global minimizers; the number of local minimizers seems to be small in all the examples we considered. For each of the problems above, running BFGS using most of the randomly selected initial guesses arrives at the same—empirically global—minimizer.

We notice also that, in each of the problems considered in the previous section, the objective function is *differentiable* at the computed minimizer. This follows from the computational observation that at the minimizer \mathbf{t} , the active eigenvalue of $\mathcal{T}_a(\mathbf{t})|_{W_k \cap [l(K_M^a)]^{m \times m}}$ is *simple*. This is atypical for spectral-related functions in a sense made precise in [BLO01a]. For instance, consider the function

$$(t_1, t_2) \mapsto \rho \left(\begin{bmatrix} 1 - t_1 & 1 & 0 \\ t_1 & 1 & 1 \\ t_2 & 0 & 1 \end{bmatrix} \right).$$

The global minimizer is $(t_1, t_2) = (0, 0)$, at which the active eigenvalue 1 is *not* simple and the spectral radius function is not differentiable. For such problems, the use of a robust solver like that in [BLO02] is necessary. For the problems considered here, a more traditional solver such as BFGS seems to suffice.

It is not clear to us whether the observed smoothness in our objective functions is a general feature of our problem.

Acknowledgments. We thank Adrian Lewis for a key observation regarding the gradient computation.

REFERENCES

- [BLO01a] J. V. BURKE, A. S. LEWIS, AND M. L. OVERTON, *Optimal stability and eigenvalue multiplicity*, *Found. Comput. Math.*, 1 (2001), pp. 205–225.
- [BLO01b] J. V. BURKE, A. S. LEWIS, AND M. L. OVERTON, *Optimizing matrix stability*, *Proc. Amer. Math. Soc.*, 129 (2001), pp. 1635–1642.
- [BLO02] J. V. BURKE, A. S. LEWIS, AND M. L. OVERTON, *Two numerical methods for optimizing matrix stability*, *Linear Algebra Appl.*, 351/352 (2002), pp. 117–145.
- [BO01] J. V. BURKE AND M. L. OVERTON, *Variational analysis of non-Lipschitz spectral functions*, *Math. Program.*, 90 (2001), pp. 317–352.
- [BT97] V. BLONDEL AND J. N. TSITSIKLIS, *NP-hardness of some linear control design problems*, *SIAM J. Control Optim.*, 35 (1997), pp. 2118–2127.
- [CDR96] A. COHEN, I. DAUBECHIES, AND A. RON, *How smooth is the smoothest function in a given refinable space?*, *Appl. Comput. Harmon. Anal.*, 3 (1996), pp. 87–89.
- [CJR02] D.-R. CHEN, R. Q. JIA, AND S. D. RIEMENSCHNEIDER, *Convergence of vector subdivision schemes in Sobolev spaces*, *Appl. Comput. Harmon. Anal.*, 12 (2002), pp. 128–149.
- [Dau93] I. DAUBECHIES, *Orthonormal bases of compactly supported wavelets II. Variations on a theme*, *SIAM J. Math. Anal.*, 24 (1993), pp. 499–519.
- [Han00] B. HAN, *Analysis and construction of optimal multivariate biorthogonal wavelets with compact support*, *SIAM J. Math. Anal.*, 31 (1999), pp. 274–304.
- [Han01] B. HAN, *Computing the smoothness exponent of a symmetric multivariate refinable function*, *SIAM J. Matrix Anal. Appl.*, 24 (2003), pp. 693–714.
- [Han02] B. HAN, *Vector cascade algorithms and refinable function vectors in Sobolev spaces*, *J. Approx. Theory*, 124 (2003), pp. 44–88.
- [Han] B. HAN, *Symmetry property and construction of wavelets with a general dilation matrix*, *Linear Algebra Appl.*, 353 (2002), pp. 207–225.
- [HJ85] R. A. HORN AND C. R. JOHNSON, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.
- [HJ98] B. HAN AND R.-Q. JIA, *Multivariate refinement equations and convergence of subdivision schemes*, *SIAM J. Math. Anal.*, 29 (1998), pp. 1177–1199.

- [HYP02] B. HAN, T. P.-Y. YU, AND B. PIPER, *Multivariate refinable Hermite interpolants*, Math. Comp., to appear.
- [JJ01] R. Q. JIA AND Q. T. JIANG, *Spectral analysis of the transition operator and its applications to smoothness analysis of wavelets*, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 1071–1109.
- [Nem93] A. NEMIROVSKII, *Several NP-hard problems arising in robust stability analysis*, Math. Control Signals Systems, 6 (1993), pp. 99–105.
- [Ron97] A. RON, *Smooth refinable functions provide good approximation orders*, SIAM J. Math. Anal., 28 (1997), pp. 731–748.
- [Yu01] T. P.-Y. YU, *Approximation order/smoothness tradeoff in Hermite subdivision schemes*, in Wavelets: Applications in Signal and Image Processing IX, A. F. Laine, M. A. Unser, and A. Aldroubi, eds., Proc. SPIE 4478, SPIE, Bellingham, WA, 2001, pp. 117–128.