Journal of Computational Physics **175,** 363–397 (2002)
doi:10.1006/jcph.2001.6743, available online at http://www.idealibrary.com on IDEAL®

Separable and Nonseparable Multiwavelets in Multiple Dimensions

C. J. Tymczak, Anders M. N. Niklasson, and Heinrich Röder

Theoretical Division, Los Alamos National Laboratory, MS B221, Los Alamos, New Mexico 87545

Received May 24, 2000

We report on a method of constructing multidimensional biorthogonal interpolating multiwavelets. The approach is based upon polynomial interpolation in $\mathbf{R}^d$ (C. de Boor and A. Ron, *Math. Comput.* **58**, 198 (1997)) and an extension of the lifting scheme (J. Kovačević and W. Sweldens, *IEEE Trans. Image Process.* **9**, No. 3, 480 (2000)). The constructed wavelets have compact support, are nearly isotropic, and retain partial scale invariance leading to a fast and efficient multidimensional wavelet transform. We demonstrate an implementation for these wavelets of variable polynomial order up to four dimensions. Finally, we show that these wavelets have a much sparser representation of discontinuous functions as compared to tensor product wavelets, which allows for a more compact and efficient representation.

Key Words: multiwavelets; multidimensional; lifting scheme; polynonimal interpolation.

1. INTRODUCTION

Many problems in condensed matter theory, as well as in many other areas of physics, are solved most conveniently within a variational minimization approach. This requires the expansion of functions and operators in suitably chosen basis functions. In electronic structure calculations the choice of plane waves, atomic orbitals, muffin-tin orbitals or Gaussians [1, 4, 6, 24, 29] has led to the development of different schemes, with different properties regarding speed, accuracy, and simplicity.

In order to keep the representation of functions and operators small we want to minimize the size of the basis set used in the variational ansatz. This is possible if the basis set is closely adapted to the actual solution. Usually this is difficult to accomplish. In electronic structure calculations, one would like expansions that work equally well for the interstitial part of the states, which are best described by plane wave like functions, and the atomic core part of the states, which are best described by atomic orbital like functions, but poorly described by plane wave like functions. This problem is solved in various ways by the different electronic structure schemes. For example, plane wave methods rectify it by replacing the atomic core region with a suitably chosen, smooth, pseudo-potential [25]. The muffin-tin orbital methods address the problem by augmenting interstitial plane wave like functions outside the atomic core region with atomic like solutions inside the atomic core region [1]. Another promising approach which recently has gained attention, and directly addresses this problem, is the use of wavelets [5, 9, 19, 27, 36, 37]. The purpose of this report is to describe a new class of wavelets which are well adapted to the problems in electronic structure calculations. However, the constructed wavelets are quite general and should form a suitable basis for many different variational problems in physics.

A wavelet basis forms a basis for a multiresolution analysis (MRA). Since a wavelet basis exists on many scales of resolution it can naturally and easily represent both small scales with rapidly changing functions, like the atomic core orbitals or the core part of the atomic valence states, and large-scale structures like interstitial parts of the valence states. Another important feature of the wavelet basis is the existence of fast algorithms, the pyramid algorithm, and the lifting scheme [10, 21, 33, 34]. These algorithms allow the calculation of the expansion of functions to linear order, similar to the fast Fourier transform. Moreover, it has been shown that a large class of operators is sparse in the wavelet representation [3] (order-N representation) and can therefore be treated very efficiently within a variational wavelet ansatz.

A large body of work has been devoted to the construction of one-dimensional wavelets [11, 13, 14, 16] and it has been implicitly assumed that the extension to higher dimensions should be done via the tensor product of the one-dimensional functions. However, there are many problems associated with this approach. If one takes a tensor product representation of the higher dimensional scaling spaces via $\mathbf{V} = V^1 \otimes V^1 \otimes \ldots \otimes V^1$, then the number of different basis functions grows as 2^d [10], where d is the dimension, and, since the basis functions will be products of one-dimensional wavelets and scaling functions of the different scales, some of the multidimensional wavelets will be highly biased along coordinate directions, which is an undesirable property when expanding spherically symmetric functions such as the atomic core states. Another disadvantage of the tensor product basis is that all scales also get mixed in the representation of operators, rendering them less sparse and reducing their condition number [31]. This makes the implementation of the tensor product wavelet basis to higher dimensions undesirable.

In order to overcome these problems we would like to construct a nontensor product wavelet basis, as isotropic as possible, with liftable filters and compact support in multiple dimensions. Some nonseparable orthogonal wavelets have been constructed [2, 22, 28, 30], which, unfortunately, are not easy to use in a computationally efficient implementation. The lifting scheme devised by Kovačević and Sweldens [21] is able to construct nonseparable wavelets in higher dimensions. Within their implementation the sampling lattices on different scales are distinct; i.e., there is no scale invariance. Unfortunately, this renders a constructive numerical analysis of differential operators difficult. However, one does not need to give up scale invariance totally, and it is possible to construct filters that repeat after a finite number of N scales. This implies that the sampling lattices, the lattices of points around which the scaling functions are centered, also repeat after N steps. From a multiresolution point of view such a construction is reminiscent of multiwavelets [20, 26, 32], where a set of N scaling functions $\{\phi_j^{(i)} \mid i = 1, \ldots, N\}$ spans the scaling space V_j. In the tensor product representation one has one scaling function and $2^d - 1$ different wavelets functions in a dilation of 2^d. In the construction scheme which we will devise in the present paper we have N different types of scaling functions and N different types of wavelets, where typically $N = d$.

In this paper we use a restricted lifting scheme involving an N-periodic scale invariance to construct a nontensor product MRA analysis of $L^2(\mathbf{R}^d)$. In the first section, we give a short introduction to wavelets and the lifting scheme. Next we introduce an unrecognized relation between lifting and multiwavelets with a simple example in one dimension. Then we generalize the scheme to higher dimensions. We present examples for separable and nonseparable nontensor product wavelet basis up to four dimensions and discuss their isotropy and compression properties. Finally, we discuss the ramifications and future directions of the work. In the appendices, we present two practical algorithms, which are a natural outcome of the presented work (i) the in-place inverse wavelet transform which allows the calculation of the value of a function at a point, where the function is represented in the wavelet basis and (ii) the top–down algorithm which allows for the adaptive wavelet analysis of a function using a truncation algorithm similar to one devised by Zorin *et al.* [38], where as an example of the efficiently of the top–down algorithm, we present the wavelet transform of the potential generated by 32 arbitrarily placed point charges.

2. WAVELET THEORY

In this section, we first present a brief review of how to construct a multiresolution analysis, which generates a wavelet basis, in one dimension. Then, we present a brief review of the lifting scheme and how it is related to, and generates, a MRA. Finally, in the last section, we show how the lifting scheme can be used to construct multiwavelets, and we give a simple example. For a more detailed discussion on wavelets, we refer the reader to [10, 15, 23].

2.1. Multiresolution Analysis

Let us start by considering the decomposition of $L_2(\mathbf{R})$ into a set of nested function subspaces

$$\ldots V_{j-1} \subset V_j \subset V_{j+1} \ldots \quad j \in \mathcal{Z}, \tag{1}$$

where we associate with each subspace V_j, a set of points γ_j. These subspaces form a multiresolution analysis with the following properties:

1. $f(x) \in V_j \Leftrightarrow f(2x) \in V_{j+1}$.
2. $f(x) \in V_j \Leftrightarrow f(x+k) \in V_j : \forall k \in \gamma_j$.
3. $\cup_j V_j$ is dense in $L_2(\mathbf{R})$ and $\cap_j V_j = \{\emptyset\}$.
4. There exists for the scaling space V_j a *scaling function* $\phi_j(x) \in V_j$ such that the collection

$$\phi_j(x+k) : \forall k \in \gamma_j \tag{2}$$

forms a Riesz basis of V_j,

$$V_j = \operatorname{span}\{\phi_j(x+k) : k \in \gamma_j\}. \tag{3}$$

There also exists a *wavelet function* $\psi_j(x)$ which spans the detail space W_j, the complement of $V_j \in V_{j+1}$; i.e.,

$$V_{j+1} = W_j \oplus V_j, \quad V_j \perp W_j, \tag{4}$$

and

$$\bigoplus_j W_j = L_2(\mathbf{R}). \tag{5}$$

In the case of a biorthogonal basis, these properties also hold for the dual spaces, $\tilde{V}_j$ and $\tilde{W}_j$, but with the biorthogonality conditions [23],

$$\tilde{V}_j \perp W_j \quad \text{and} \quad V_j \perp \tilde{W}_j. \tag{6}$$

Because of the properties of the multiresolution analysis, it is possible to show that the scaling and dual scaling functions obey a two-scale relation,

$$\begin{cases} \phi(x) = \sum_k h_k \phi(2x - k) \\ \tilde{\phi}(x) = \sum_k \tilde{h}_k \tilde{\phi}(2x - k) \end{cases} \tag{7}$$

and for the wavelet and dual wavelets we have the relations,

$$\begin{cases} \psi(x) = \sum_k g_k \phi(2x - k) \\ \tilde{\psi}(x) = \sum_k \tilde{g}_k \tilde{\phi}(2x - k) \end{cases}, \tag{8}$$

with the biorthogonality condition,

$$\int dx \begin{bmatrix} \tilde{\phi}_{jk}(x) \\ \tilde{\psi}_{jk}(x) \end{bmatrix} \cdot [\phi_{j'k'}(x) \; \psi_{j'k'}(x)] = \begin{bmatrix} \delta_{jj'}\delta_{kk'} & 0 \\ 0 & \delta_{jj'}\delta_{jj'} \end{bmatrix}. \tag{9}$$

The h_k, g_k, $\tilde{h}_k$, and $\tilde{g}_k \in l_2(\mathcal{Z})$ are the dual and nondual filter coefficients.

A wavelet transform of a function can be constructed via the pyramid algorithm [10]. Consider a function expanded on an arbitrary resolution scale J and $x \in [0, 1]$,

$$f(x) = \sum_k f^{\phi}_{Jk} \phi(2^J x - k). \tag{10}$$

The wavelet transform of this function can be computed by successive applications of Eqs. (7) and (8) using the dual filters, and the inverse transform can be computed by the successive application of the nondual filters. The application of the filters is depicted in Fig. 1, where we are showing one step of the pyramid algorithm.

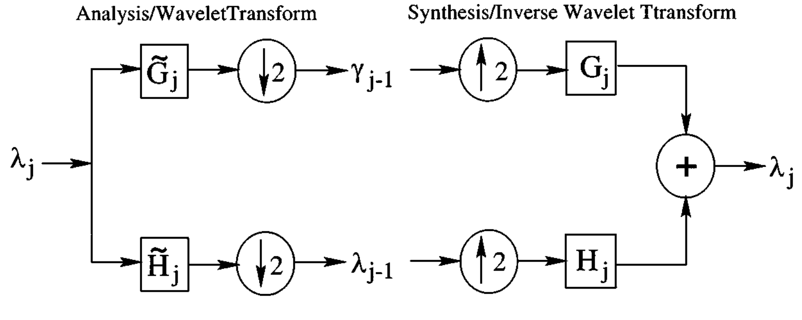

FIG. 1. A diagram of the two-channel filter bank, which represents one step of the pyramid algorithm for a biorthogonal MRA.

2.2. The Lifting Scheme

One of the more elegant ways of generating a biorthogonal MRA is via the lifting scheme [21]. This method is straightforward to implement and can easily be adapted to multiple dimensions. The lifting scheme is constructed via three stages: *split, predict,* and *update*. First, let us start by considering a data set λ_0. In the first stage we split the data set into two smaller disjoint subsets λ_{-1} and γ_{-1}, where $\lambda_0 = \gamma_{-1} \cup \lambda_{-1}$ and $\gamma_{-1} \cap \lambda_{-1} = \{\emptyset\}$. The simplest way of splitting the data is into the even and odd numbers. This is sometimes referred to as the lazy wavelet transform [21]. Next, let us use the data set, λ_{-1}, to predict the values of the other data set, γ_{-1}, i.e.,

$$\gamma_{-1} = P(\lambda_{-1}), \tag{11}$$

where P is the predict operator. This would allow us to replace the original data set with the subset λ_{-1}. In practice, it is usually not possible to construct a predict operator which exactly predicts γ_{-1} from λ_{-1}. Let us instead replace γ_{-1} with the difference,

$$\gamma_{-1}^{w} = \gamma_{-1} - P(\lambda_{-1}). \tag{12}$$

The subset γ_{-1}^{w} now encodes how much γ_{-1} deviates from the model on which the prediction was constructed, a set of details. It is also a simple matter to recover the original data set by reversing the sign of Eq. (12). In many cases we would also like to preserve some of the properties of the original data set, λ_0, into the new data set, λ_{-1}. We can do this by devising a new operator which uses γ_{-1} to update λ_{-1},

$$\lambda_{-1}^{s} = \lambda_{-1} + U(\gamma_{-1}), \tag{13}$$

where U is the update operator. One nice property of the lifting scheme is that the inverse process is easy to accomplish just by reversing the signs of Eqs. (12) and (13), which is illustrated in Fig. 2. The process depicted in Fig. 2 can also be interpreted as one step of the pyramid algorithm, where the λ_j^s's are the scaling function expansion coefficients, and the γ_j^w's are the wavelet functions expansion coefficients. This can be seen by realizing that the process depicted in Fig. 2 is also a biorthogonal MRA, as was depicted in Fig. 1, where the λ_j^s's form the resolution spaces and the γ_j^w's form the detail spaces.

Another property of the lifting scheme is that the operations can be chained together. This allows the lifting scheme to be used to modify any existing wavelet transforms. Consider

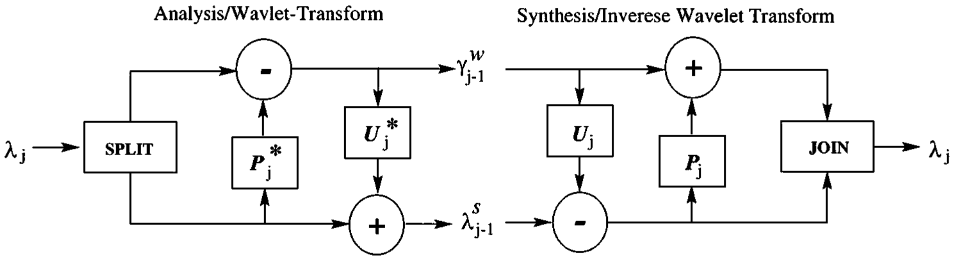

FIG. 2. A diagram of the predict and update operations of the lifting scheme.

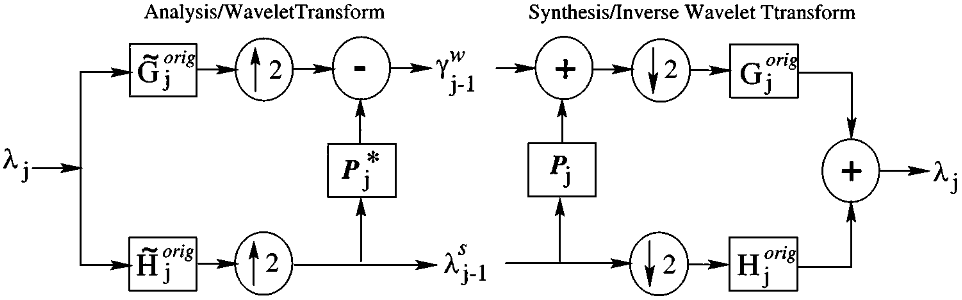
Analysis/WaveletTransform
Synthesis/Inverse Wavelet Ttransform
λj
G̃j orig
H̃j orig
2
2
-
P*j
γw j-1
λs j-1
+
Pj
2
2
Gj orig
Hj orig
+
λj

FIG. 3. A diagram of the modification of the filters using a predict operation.

Fig. 3, the predict step will modify the original filters by

$$\begin{aligned} \tilde{H}_j &= \tilde{H}_j^{orig} & H_j &= H_j^{orig} + P_j\, G_j^{orig} \\ \tilde{G}_j &= \tilde{G}_j^{orig} - P_j^* \tilde{H}_j^{orig} & G_j &= G_j^{old} \end{aligned} \tag{14}$$

and similarly for an update step the original filters get modified by (not in figure)

$$\begin{aligned} \tilde{H}_j &= \tilde{H}_j^{orig} + U_j \tilde{G}_j^{orig} & H_j &= H_j^{orig} \\ \tilde{G}_j &= \tilde{G}_j^{orig} & G_j &= G_j^{orig} - U_j^* H_j^{orig}. \end{aligned} \tag{15}$$

For the case where the original filters are the lazy wavelet transform (just splitting the data sets), we can use Eqs. (14) and (15) to deduce the new biorthogonal filters as

$$\begin{aligned} \tilde{H}_j &= 1 + U_j(1 - P_j^*) & H_j &= 1 + P_j \\ \tilde{G}_j &= 1 - P_j^* & G_j &= 1 - U_j^*(1 + P_j). \end{aligned} \tag{16}$$

The asterisks denote the complex conjugation of the filters, which is included for the general case of complex coefficients. The basis functions associated with some γ_{jk}^w or λ_{jk}^s of the lifting scheme can easily be generated by the inverse pyramid algorithm, where $\{\gamma^w\}_{j'k'} = \delta_{jj'}\delta_{kk'}$ generates the wavelet $\psi_j(x-k)$, and $\{\lambda^s\}_{j'k'} = \delta_{jj'}\delta_{kk'}$ generates the scaling function $\phi_j(x-k)$. In the case where there is no update, the dual scaling functions correspond to Dirac delta functions. The dual wavelet functions are sums of Dirac delta functions, and the nondual functions correspond to polynomial interpolating wavelets [8], where the wavelets and scaling functions are related by

$$\psi(x) = \phi(2x-1). \tag{17}$$

Notice, that without the update the dual scaling and wavelet functions do not form a proper basis of $L_2(\mathbf{R})$, since Dirac delta functions are not square integrable. However, the primal set forms a basis of $L_2(\mathbf{R})$ and we will therefore still use the expression biorthogonal basis, even if it is not strictly correct.

In the next section we show how the lifting scheme can be used to generate multiwavelets, a connection which we have found to be very informative in our analysis.

2.3. The Lifting Scheme and Multiwavelets

In this section we present a multiwavelet derived from lifting in one dimension. We construct this by performing two consecutive subdivision and predict steps, where we may choose the order of the two predict polynomials to be different. As usual in the lifting scheme we start by defining a subdivision. The original lattice is $\Gamma_0 = \mathcal{Z}$, and we subdivide into the scaling set $\Gamma_s^{(1)} = 2\Gamma_0$ and the wavelet set $\Gamma_w^{(1)} = 2\Gamma_0 + 1$. This corresponds to splitting the set of integers into even and odd numbers. Next, we split the lattice again into the new scaling set $\Gamma_s^{(2)} = 2\Gamma_1 = 4\Gamma_0$ and the new wavelet set $\Gamma_w^{(1)} = 2\Gamma_1 + 2 = 4\Gamma_0 + 2$. The corresponding two-scale relations for the scaling functions are

$$\begin{aligned} \phi^{(1)}(x) &= \phi^{(2)}(2x) + \sum_{m\in\Gamma_w^{(1)}} P_m^{(1)}\phi^{(2)}(2x-m) \\ \phi^{(2)}(x) &= \phi^{(1)}(2x) + \sum_{m\in\Gamma_w^{(2)}} P_m^{(2)}\phi^{(1)}(2x-m), \end{aligned} \tag{18}$$

where we have two separate scaling functions ($\phi^{(1),(2)}$) on different scales. The $P_m^{(1),(2)}$ are the prediction coefficients of the lifting scheme. We can decouple Eqs. (18)

$$\begin{aligned} \phi^{(1)}(x) &= \phi^{(1)}(4x) + \sum_{k\in\Gamma_w^{(2)}} P_m^{(2)}\phi^{(1)}(4x-2k) + \sum_{k\in\Gamma_w^{(1)}} P_m^{(1)}\phi^{(1)}(4x-2k) \\ &\quad + \sum_{k\in\Gamma_w^{(1)};k'\in\Gamma_w^{(2)}} P_m^{(1)}P_{m'}^{(2)}\phi^{(1)}(4x-2k-k') \\ \phi^{(2)}(x) &= \phi^{(2)}(4x) + \sum_{k\in\Gamma_w^{2}} P_m^{(2)}\phi^{(2)}(4x-2k) + \sum_{k\in\Gamma_w^{(1)}} P_m^{(1)}\phi^{(2)}(4x-2k) \\ &\quad + \sum_{k\in\Gamma_w^{(1)};k'\in\Gamma_w^{(2)}} P_m^{(1)}P_{m'}^{(2)}\phi^{(2)}(4x-2k'-k), \end{aligned} \tag{19}$$

which can be phrased in a typical multiwavelet form,

$$\mathbf{\Phi}(x) = \begin{pmatrix} \phi^{(1)}(x) \\ \phi^{(2)}(x) \end{pmatrix} = \sum_k \mathbf{H}_k \begin{pmatrix} \phi^{(1)}(4x-k) \\ \phi^{(2)}(4x-k) \end{pmatrix} = \sum_k \mathbf{H}_k\mathbf{\Phi}(4x-k). \tag{20}$$

The matrices $\mathbf{H}_k$ are of the form

$$\mathbf{H}_k = \begin{pmatrix} h_k^{(1)} & 0 \\ 0 & h_k^{(2)} \end{pmatrix},$$

where

$$h_k^{(1),(2)} = \begin{cases} 1 & k = 0 \\ P_k^{(2),(1)} & k \in \Gamma_w^{(2),(1)} \\ P_k^{(1),(2)} + \sum_{k'\in\Gamma_w^{(1),(2)}} P_{k'}^{(1),(2)}P_{k+k'}^{(2),(1)} & k \in \Gamma_w^{(1),(2)} \end{cases}. \tag{21}$$

In a multiwavelet setting the functions $\phi^{(1),(2)}$ span the scaling spaces. The wavelets are chosen from the interpolating algorithm as $\psi^{(1)}(x) = \phi^{(1)}(4x-3)$ and $\psi^{(2)}(x) = \phi^{(2)}(4x-3)$, and their two-scale relation in a multiwavelet form becomes

$$\Psi(x) = \begin{pmatrix} \psi^{(1)}(x) \\ \psi^{(2)}(x) \end{pmatrix} = \sum_k \mathbf{G}_k \mathbf{\Phi}(4x - k), \tag{22}$$

where

$$\mathbf{G}_k = \begin{pmatrix} \delta_{0k+3} & 0 \\ 0 & \delta_{0k+3} \end{pmatrix}. \tag{23}$$

The subdivision process above could have been continued in N steps with N different filters. The above analysis would then lead to multiwavelets of multiplicity N. The fully generalized lifting scheme (no scale invariance of the filters) can then be viewed as a multiwavelet with infinitely many components.

3. MULTIWAVELETS IN MULTIPLE DIMENSIONS

In this section we extend the scheme from the previous section to higher dimensions. This allows us to construct a nontensor product MRA in $L_2(\mathbf{R}^d)$. We start by generalizing the MRA to $\mathbf{R}^d$. Let us consider the decomposition of $L_2(\mathbf{R}^d)$ into a set of nested subspaces [23, 34].

$$\ldots \mathbf{V}_{j-1} \subset \mathbf{V}_j \subset \mathbf{V}_{j+1} \ldots \quad j \in \mathcal{Z}, \tag{24}$$

where we associate with each subspace $\mathbf{V}_j$, a lattice $\Gamma_j \in \mathcal{Z}^d$ defined as $\Gamma_j = \mathbf{L}_j \mathcal{Z}^d$, where $\mathbf{L}_j$ is a nonsingular $d \times d$ matrix which generates the lattice, and D_{j+1} is a nonsingular $d \times d$ dilation matrix which projects us from the lattice associated with $\mathbf{V}_{j+1}$ to the lattice associated with $\mathbf{V}_j$; i.e.,

$$\Gamma_j = \mathbf{D}_{j+1} \Gamma_{j+1}. \tag{25}$$

The subspaces $\mathbf{V}_J$ form a multiresolution analysis with the following properties:

1. $f(\mathbf{x}) \in \mathbf{V}_j \Leftrightarrow f(\mathbf{D}_{j+1}\mathbf{x}) \in \mathbf{V}_{j+1}$.
2. $f(\mathbf{x}) \in \mathbf{V}_j \Leftrightarrow f(\mathbf{x}+\mathbf{k}) \in \mathbf{V}_j : \forall \mathbf{k} \in \Gamma_j$.
3. $\cup_j \mathbf{V}_j$ is dense in $L_2(\mathbf{R}^d)$ and $\cap_j \mathbf{V}_j = \{\emptyset\}$.
4. There exists for the scaling space $\mathbf{V}_j$ a *scaling function* $\phi_j(\mathbf{x}) \in \mathbf{V}_j$ such that the collection

$$\phi_j(\mathbf{x}+\mathbf{k}) : \forall k \in \Gamma_j \tag{26}$$

forms a Riesz basis of $\mathbf{V}_j$;

$$\mathbf{V}_j = \operatorname{span}\{\phi_j(\mathbf{x}+\mathbf{k}) : \mathbf{k} \in \Gamma_j\}. \tag{27}$$

There also exists a *wavelet function* $\psi_j(\mathbf{x})$ which spans the detail space $\mathbf{W}_j$, the complement of $\mathbf{V}_j \in \mathbf{V}_{j+1}$ such that

$$\mathbf{V}_{j+1} = \mathbf{W}_j \oplus \mathbf{V}_j, \quad \mathbf{W}_j \perp \mathbf{V}_j, \tag{28}$$

and

$$\bigoplus_j \mathbf{W}_j = L_2(\mathbf{R}^d). \tag{29}$$

Again, in the case of a biorthogonal basis, these properties also hold for the dual spaces, $\tilde{\mathbf{V}}_j$ and $\tilde{\mathbf{W}}_j$, but with the biorthogonality conditions,

$$\tilde{\mathbf{V}}_j \perp \mathbf{W}_j \quad \text{and} \quad \mathbf{V}_j \perp \tilde{\mathbf{W}}_j. \tag{30}$$

The lattices Γ_j associated with the nested subspaces V_j can be constructed via a subdivision of sublattices. A lattice Γ_j is decomposed into m disjoint sublattices via

$$\Gamma_j = \bigcup_{i=1}^{M_j} \left(\mathbf{D}_j \Gamma_j + \mathbf{t}_i^{(j)}\right), \tag{31}$$

where $\mathbf{t}_i^{(j)}$ are translation vectors in Γ_j, and $M_j = \det[\mathbf{D}_j]$. We will make major use of Eq. (31) in our multidimensional lattice decomposition algorithm which is described in more detail in Section 3.3. The arguments of the scaling and wavelet functions in the two-scale relation are now scaled by the dilation matrix $\mathbf{D}_j$,

$$\begin{cases} \phi_{j\mathbf{k}}(\mathbf{x}) = M_j^{-\frac{1}{2}} \sum_{\mathbf{k}'\in\Gamma_j} h_{j\mathbf{k}'} \phi_{j\mathbf{k}}(\mathbf{D}_j\mathbf{x} - \mathbf{k}') \\ \tilde{\phi}_{j\mathbf{k}}(\mathbf{x}) = M_j^{-\frac{1}{2}} \sum_{\mathbf{k}'\in\tilde{\Gamma}_j} \tilde{h}_{j\mathbf{k}'} \phi_{j\mathbf{k}}(\mathbf{D}_j\mathbf{x} - \mathbf{k}') \end{cases}, \tag{32}$$

and also for the wavelets,

$$\begin{cases} \psi_{j\mathbf{k}}(\mathbf{x}) = M_j^{-\frac{1}{2}} \sum_{\mathbf{k}'\in\tilde{\Gamma}_j} g_{j\mathbf{k}'} \phi_{j\mathbf{k}}(\mathbf{D}_j\mathbf{x} - \mathbf{k}') \\ \tilde{\psi}_{j\mathbf{k}}(\mathbf{x}) = M_j^{-\frac{1}{2}} \sum_{\mathbf{k}'\in\tilde{\Gamma}_j} \tilde{g}_{j\mathbf{k}'} \tilde{\phi}_{j\mathbf{k}}(\mathbf{D}_j\mathbf{x} - \mathbf{k}') \end{cases}, \tag{33}$$

which, for a biorthogonal basis, obey the biorthogonality condition:

$$\int d\mathbf{x} \begin{bmatrix} \tilde{\phi}_{j\mathbf{k}}(\mathbf{x}) \\ \tilde{\psi}_{j\mathbf{k}}(\mathbf{x}) \end{bmatrix} [\phi_{j'\mathbf{k}'}(\mathbf{x}) \quad \psi_{j'\mathbf{k}'}(\mathbf{x})] = \begin{bmatrix} \delta_{jj'}\delta_{\mathbf{k}\mathbf{k}'} & 0 \\ 0 & \delta_{jj'}\delta_{\mathbf{k}\mathbf{k}'} \end{bmatrix}. \tag{34}$$

Here $h_{j\mathbf{k}}$, $g_{\nu;j\mathbf{k}}$, $\tilde{h}_{j\mathbf{k}}$, and $\tilde{g}_{\nu;j\mathbf{k}} \in l_2(\Gamma_j)$ are the dual and nondual filter coefficients.

The generalization to multiple dimensions of a multiresolution analysis usually proceeds via a tensor product scheme. The tensor product representation of a higher dimensional scaling space $\mathbf{V}_J$ using one-dimensional scaling spaces V_J^1 is created via

$$\mathbf{V}_J = V_J^1 \otimes V_J^1 \otimes \cdots \otimes V_J^1, \tag{35}$$

where J denotes scaling spaces separated by d scales; i.e., $\mathbf{V}_{J-1}$ is a space which is dilated by a factor of two in all dimensions as compared to $\mathbf{V}_J$. The dual spaces follow in analogy. These spaces can be projected into the detail spaces using Eq. (4),

$$\mathbf{V}_J = \left\{V_{J-1}^1 \oplus W_{J-1}^1\right\} \otimes \left\{V_{J-1}^1 \oplus W_{J-1}^1\right\} \otimes \cdots \otimes \left\{V_{J-1}^1 \oplus W_{J-1}^1\right\} \tag{36}$$

$$= \left\{V_0^1 \bigoplus_{j=0}^{J-1} W_j^1\right\} \otimes \left\{V_0^1 \bigoplus_{j=0}^{J-1} W_j^1\right\} \otimes \cdots \otimes \left\{V_0^1 \bigoplus_{j=0}^{J-1} W_j^1\right\} \tag{37}$$

$$= \mathbf{V}_0 \bigoplus_{j=0}^{J-1} \mathbf{W}_j, \tag{38}$$

where

$$\mathbf{W}_J = \bigoplus_{\{j_1, j_2 \ldots j_d\}} W^1_{j_1} \otimes W^1_{j_2} \otimes \cdots \otimes W^1_{j_d} \bigoplus_{\{j_2 \ldots j_d\}} V_0 \otimes W^1_{j_2} \otimes \ldots \otimes W^1_{j_d} \bigoplus \cdots. \tag{39}$$

Here we immediately see the problem which develops. The multidimensional wavelet functions will be the products of one-dimensional wavelet functions of different resolutions scales such as

$$\psi_{J\mathbf{k}}(\mathbf{x}) = \psi_{j_1 k_1}(x_1)\psi_{j_2 k_2}(x_2) \ldots \psi_{j_d k_d}(x_d), \tag{40}$$

where the $\{j_n\} \in [0, J-1]$. Some of these wavelets mix small and large resolution scales and will therefore be highly anisotropic. This may lead to a large degradation of their performance at compressing functions, which will be shown in Section 4. The new multiwavelets are not constructed in this fashion, instead, the Jth level scaling space is decomposed into N scaling spaces via

$$\ldots \mathbf{V}^{(N)}_{J-1} \subset \mathbf{V}^{(1)}_J \subset \mathbf{V}^{(2)}_J \subset \cdots \subset \mathbf{V}^{(N)}_J \subset \mathbf{V}^{(1)}_{J+1} \cdots, \tag{41}$$

with the corresponding detail spaces $\mathbf{W}^{(n)}_J$ given by the relations,

$$\mathbf{V}^{(n)}_J = \mathbf{V}^{(n-1)}_J \oplus \mathbf{W}^{(n-1)}_J \quad \text{and} \quad \mathbf{V}^{(1)}_J = \mathbf{V}^{(N)}_{J-1} \oplus \mathbf{W}^{(N)}_{J-1}. \tag{42}$$

Here J refers to scaling spaces separated by N scales. Using Eqs. (42) we can project $\mathbf{V}_J$ into the detail spaces,

$$\begin{aligned} \mathbf{V}_J &= \mathbf{V}^{(N)}_J && (43)\\ &= \mathbf{V}^{(N-1)}_J \oplus \mathbf{W}^{(N-1)}_J && (44)\\ &= \mathbf{V}_{J-1} \oplus \left\{\mathbf{W}^{(1)}_J \oplus \cdots \oplus \mathbf{W}^{(N-1)}_J \oplus \mathbf{W}^{(N)}_{j-1}\right\} && (45)\\ &= \mathbf{V}_0 \left\{ \bigoplus_{j=1}^{J} \left\{\mathbf{W}^{(1)}_j \oplus \cdots \oplus \mathbf{W}^{(N-1)}_j \oplus \mathbf{W}^{(N)}_{j-1}\right\} \right\}, && (46) \end{aligned}$$

where we define the multiwavelet space

$$\mathbf{W}_{J-1} = \mathbf{W}^{(1)}_J \oplus \mathbf{W}^{(2)}_J \oplus \cdots \oplus \mathbf{W}^{(N)}_{J-1}; \tag{47}$$

i.e.,

$$\mathbf{V}_J = \mathbf{V}_0 \bigoplus_{j=0}^{J-1} \mathbf{W}_j.$$

There is no mixing (products) of scales in this decomposition, instead there is a sum of N distinct detail spaces per scale, which forms a direct link to the multiwavelet framework. This will be discussed more in Section 3.2.

3.1. Polynomial Interpolation on Lattices

Given the sublattice $\Omega_i^{(j)} = \mathbf{D}_j \Gamma_j + \mathbf{t}_j^{(i)}$, we can construct a polynomial interpolation, $\Pi(\Omega_i^{(j)})$, upon a restricted set of points. This polynomial interpolation can be used to predict an approximation of the function, which, via the lifting scheme devised by Sweldens *et al*. [21, 33, 34], will allow the generation of a multiresolution analysis [23]. Building polynomial interpolating filters in d dimensions of order P requires that we can construct a polynomial interpolation in $\mathbf{R}^d$. In one dimension, polynomial interpolation always has a Vandermonde matrix which is invertible [31]. However, in higher dimensions, for a polynomial interpolation of a given order, the Vandermonde matrix is not always invertible and the system is either over or under determined. In this case, one may need either fewer or more interpolation points. However, the order of the polynomial interpolation is determined not only by the number of interpolation points, but also by their configuration. For example, in general, three points in $\mathbf{R}^2$ define a linear interpolation, but if the three points are all co-linear, they define a quadratic interpolation. An elegant solution to the problem of polynomial interpolation in higher dimensions is provided by de Boor and Ron [17]. First, they specify the configuration of f points in $\mathbf{R}^d$, and then they find the minimal polynomial space which spans it. The de Boor–Ron algorithm has been used throughout this work in the construction of the polynomial interpolating filters in $\mathbf{R}^d$.

3.2. Construction of the Filters

Following the procedure outlined above, we can construct polynomial interpolating multiwavelets from a nonstationary lifting scheme. We subdivide the scaling space (lattice) into two subspaces (sublattices), use a suitable polynomial interpolation for the predict step of the lifting scheme, and subdivide the scaling space again, with possibly a different subdivision or polynomial interpolation. The filters are constructed via the following procedure:

- **Separate** the lattice (scaling space) on scale j into two sublattices

$$\begin{aligned} \Omega_s^{(j)} &= \mathbf{D}_j \Gamma_j \\ \Omega_w^{(j)} &= \mathbf{D}_j \Gamma_j + \mathbf{t}_j \\ \Gamma_j &= \Omega_s^{(j)} \cup \Omega_w^{(j)}. \end{aligned} \tag{48}$$

- **Construct** a polynomial interpolation, $P_N(\mathbf{x})$, around a point $\mathbf{p}_0 \in \Omega_w^{(j)} = \mathbf{D}_j \Gamma_j + \mathbf{t}_j$ for a restricted set of points on the disjoint sublattice $\Omega_s^{(j)} = \mathbf{D}_j \Gamma_j$.
- **Predict** the value of the function on point $\mathbf{p}_0 \in \Omega_w$ using the polynomial interpolation on Ω_s. Since the filters are translationally invariant (apart from boundaries), the same polynomial interpolation can be used for every point on the lattice $\Omega_w^{(j)}$. The scaling and wavelet filters are given by

$$\begin{array}{ll} \tilde{h}_{j\mathbf{k}} = \delta_{\mathbf{k},0} & h_{j\mathbf{k}} = \delta_{\mathbf{k},0} + P_{j\mathbf{k}} \\ \tilde{g}_{j\mathbf{k}} = \delta_{\mathbf{k},0} - P_{j\mathbf{k}} & g_{j\mathbf{k}} = \delta_{\mathbf{k},0} \end{array}, \tag{49}$$

where $P_{j\mathbf{k}}$ are the polynomial interpolating coefficients (PIC's).

- **Calculate** the difference of the predicted value from the actual value of the function at point $\mathbf{p}_0$, i.e., the wavelet coefficient (via the $\tilde{g}_{j\mathbf{k}}$ filter).
- **Repeat** this procedure for the lattice $\Gamma_{j-1} = \Omega_s^{(j)}$.

A MRA of an object, say a function of $\mathbf{R}^d \to \mathbf{R}$, using these filters can be performed via the lifting scheme [21, 33, 34]. Running the above process backward will reconstruct the function; i.e., it performs the inverse wavelet transform. A similar scheme was used by Kovačević and Sweldens [21]. In their scheme each lattice Γ_j is usually different; i.e., there is no scale invariance. However, it is possible to regain partial scale invariance, not as $j \to j+1$, but as previously mentioned, $j \to j+N$. This condition requires that

$$\prod_{i=j}^{j+N} \mathbf{D}_i = 2\mathbf{I}, \tag{50}$$

which implies $\mathbf{D}_j = \mathbf{D}_{j+N}$ and $\Gamma_{j+N} = \frac{1}{2}\mathbf{I}\Gamma_j$; i.e., after N dilation steps we return to the starting lattice, but with all the directions dilated by a factor of 2. Instead of scale invariance, we have an N-scale invariance. Given that we split our lattices by a factor of 2, the minimal N is the dimension d of $\mathbf{R}^d$. However, it should be noted that in special cases it can be true that some subsets of the dilation matrices are the same.

To make the connection with multiwavelets we write down the two-scale relations between the scaling functions on each subsequent sublattice Γ_j,

$$\begin{aligned}
\phi^{(1)}(\mathbf{x}) &= \sum_{\mathbf{k}\in\Gamma_0+\mathbf{t}_0} h_{1\mathbf{k}}\phi^{(2)}(\mathbf{D}_1\mathbf{x}-\mathbf{k})\\
\phi^{(2)}(\mathbf{x}) &= \sum_{\mathbf{k}\in\Gamma_1+\mathbf{t}_1} h_{2\mathbf{k}}\phi^{(3)}(\mathbf{D}_2\mathbf{x}-\mathbf{k})\\
\vdots \qquad & \qquad\quad \vdots\\
\phi^{(N)}(\mathbf{x}) &= \sum_{\mathbf{k}\in\Gamma_{N-1}+\mathbf{t}_{N-1}} h_{N\mathbf{k}}\phi^{(1)}(\mathbf{D}_N\mathbf{x}-\mathbf{k}).
\end{aligned} \tag{51}$$

By back substituting each consecutive scaling function, Eqs. (51) can easily be cast into the multiwavelet from of the two-scale relation,

$$\Phi(\mathbf{x}) = \sum_{\mathbf{k}\in\Gamma_0} \mathbf{H}_{\mathbf{k}}\Phi(2\mathbf{I}\mathbf{x}-\mathbf{k}), \tag{52}$$

with the corresponding wavelet relation,

$$\Psi(\mathbf{x}) = \sum_{\mathbf{k}\in\Gamma_0} \mathbf{G}_{\mathbf{k}}\Phi(2\mathbf{I}\mathbf{x}-\mathbf{k}). \tag{53}$$

Here $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ are the vectors of N different scaling and wavelet functions

$$\boldsymbol{\Phi} = \left(\phi^{(1)}, \phi^{(2)}, \ldots\ \phi^{(N)}\right)^{\mathrm{T}} \quad \text{and} \quad \boldsymbol{\Psi} = \left(\psi^{(1)}, \psi^{(2)}, \ldots \psi^{(N)}\right)^{\mathrm{T}},$$

and the filters are

$$\mathbf{H}_{\mathbf{k}} = \begin{pmatrix} H_{\mathbf{k}}^{(1)} & \cdots & 0\\ \vdots & \ddots & \vdots\\ 0 & \cdots & H_{\mathbf{k}}^{(N)} \end{pmatrix} \quad \text{and} \quad \mathbf{G}_{\mathbf{k}} = \begin{pmatrix} \delta_{0,\mathbf{k}+\mathbf{t}_{vec}} & \cdots & 0\\ \vdots & \ddots & \vdots\\ 0 & \cdots & \delta_{0,\mathbf{k}+\mathbf{t}_{vec}} \end{pmatrix}, \tag{54}$$

where

$$H_{\mathbf{k}}^{(n)} = \sum_{\{\mathbf{k}_1,\mathbf{k}_2\ldots\mathbf{k}_N\}} h_{1\mathbf{k}_1}\ldots h_{N\mathbf{k}_N}\delta(2\mathbf{k}-\mathbf{k}_n-\mathbf{D}_n\mathbf{k}_{n+1}-\mathbf{D}_n\mathbf{D}_{n+1}\mathbf{k}_{n+2}\ldots) \tag{55}$$

and

$$\mathbf{t}_{vec} = \sum_{i=0}^{N-1} \mathbf{t}_i, \tag{56}$$

i.e., as a sum of translation vectors. The dual relations follow in analogy.

3.3. Decomposition of the Lattices

In a wavelet subdivision scheme one might intuitively start from a fine set and proceed to split the set into wavelet and scaling subsets. However, in the multidimensional case this is not necessarily the most straightforward approach when constructing a MRA via the lifting scheme. In the present case, we start from a course set and proceed via a series of lattice translates and unions to a finer set, and then, after generating N lattices from N translates, deduce the dilation matrixes by reversing the procedure. Our procedure is based on Eq. (31) with the following lattice subdivision conditions:

- The integer translate, $\mathbf{t}_p$, are always at the bounds of $[0, 1]^d$ cube. This restricts the set of translation vectors to those which point from the origin to each corner of a d-dimensional cube.
- Points are never mapped into points; i.e., $\Omega_s^{(j)} \cap \Omega_w^{(j)} = \{\emptyset\}$.
- The number of elements in the new lattice doubles at each step, which is automatically obeyed if the first two conditions are met. This insures that

$$\det[\mathbf{D}_j] = M_j = 2. \tag{57}$$

A lattice construction obeying these conditions will create a tree-like structure of allowable translates. Our algorithm for constructing the lattice subdivision/union in $\mathbf{R}^d$ is as follows:

```
ALGORIHM 1.
Start with a simple cubic lattice, $\Gamma_0 = 2\mathbf{I}\mathcal{Z}^d$, collect this lattice into the set of allowable
    lattices $\{\Gamma_{j=0}\}$.
Loop from $j = 1$ to $j = d$ and do steps (a) and (b).
    (a) For each lattice in $\{\Gamma_j\}$, translate this lattice by each translate vector $\mathbf{t}_p$ of the set
        of translate vectors to obtain a new lattice $\Gamma_j^{(n,p)} = \Gamma_j^{(n)} + \mathbf{t}_p$, where $\Gamma_j^{(n)}$ is the $n$th
        lattice in the set $\{\Gamma_j\}$.
    (b) If $\Gamma_j^{(n)} \cap \Gamma_j^{(n,p)} = \{\emptyset\}$ add $\Gamma_j^{(n)} \cup \Gamma_j^{(n,p)}$ to $\{\Gamma_{j+1}\}$, else discard.
```

Each "translate chain" will terminate at the simple cubic lattice $\Gamma_d = \mathbf{I}\mathcal{Z}^d$, where a translate chain is the set of allowable translates which take us from Γ_0 to $\Gamma_d = \frac{1}{2}\Gamma_0$. The dilation matrices can then be deduced from Eq. (25). Notice that without an automatic procedure as described above a construction of a lattice decomposition in higher dimensions is impossible in practice.

3.4. Examples

We will now show some examples of how algorithm 1 can be used to construct different lattice decompositions. In the first example we show how to construct a set of decompositions which leads to a separable multiwavelet basis in three dimensions. In the next two examples, we show how to construct sublattices which lead to nonseparable multiwavelets in three and four dimensions.

3.4.1. *Separable, Three-Dimensional, Polynomial Interpolating Multiwavelets*

In $\mathbf{R}^d$ the simplest lattice subdivision/union scheme is to use a set of orthogonal translate vectors. We will follow the procedure outline in algorithm 1, but for the specific choice of a translate chain,

$$\{\mathbf{t}_0 = (0,\ 0,\ 1),\ \mathbf{t}_1 = (0,\ 1,\ 0),\ \mathbf{t}_2 = (1,\ 0,\ 0)\}. \tag{58}$$

To elaborate, let us start by the lattice defined by the lattice generator,

$$\mathbf{L}_0 = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}. \tag{59}$$

This is the simple cubic lattice, $\Gamma_0 = \mathbf{L}_0 \mathcal{Z}$. Let us translate this lattice by $\mathbf{t}_0 = (0, 0, 1)$ and then generate the union of Γ_0 with the translated version of itself. This gives us the lattice Γ_1 and its generator $\mathbf{L}_1$,

$$\Gamma_1 = \Gamma_0 \cup (\Gamma_0 + \mathbf{t}_0), \quad \mathbf{L}_1 = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{60}$$

This is equivalent to subdividing Γ_1 into the two disjoint sublattices,

$$\begin{aligned} \Omega_s &= \mathbf{D}_1 \Gamma_1 = \Gamma_0 \\ \Omega_w &= \mathbf{D}_1 \Gamma_1 + \mathbf{t}_0, \end{aligned} \tag{61}$$

where

$$\mathbf{D}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix}. \tag{62}$$

Let us repeat this process for the next lattice Γ_1 by translating by $\mathbf{t}_1 = (0, 1, 0)$. This give us

$$\Gamma_2 = \Gamma_1 \cup (\Gamma_1 + \mathbf{t}_1), \quad \mathbf{L}_2 = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{63}$$

which is the same as subdividing Γ_2 into the two disjoint sublattices,

$$\begin{aligned} \Omega_s &= \mathbf{D}_2 \Gamma_2 = \Gamma_1 \\ \Omega_w &= \mathbf{D}_2 \Gamma_2 + \mathbf{t}_1, \end{aligned} \tag{64}$$

where

$$\mathbf{D}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{65}$$

Repeating this process a final time for Γ_2 by translating by $\mathbf{t}_2 = (1,\ 0,\ 0)$, we get

$$\Gamma_3 = \Gamma_2 \cup (\Gamma_2 + \mathbf{t}_2), \quad \mathbf{L}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{66}$$

which is the same as subdividing Γ_3 into the two disjoint sublattices,

$$\begin{aligned} \Omega_s &= \mathbf{D}_3\Gamma_3 = \Gamma_2 \\ \Omega_w &= \mathbf{D}_3\Gamma_3 + \mathbf{t}_2, \end{aligned} \tag{67}$$

where

$$\mathbf{D}_3 = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{68}$$

We have now returned to the original lattice contracted by a factor of 2 in each direction; i.e., $\Gamma_3 = \frac{1}{2}\mathbf{\Gamma}_0$. The filters obtained from this subdivision scheme are all one-dimensional, aligned along coordinate directions. The multiwavelets are therefore products of one-dimentional functions, e.g., for the scaling functions,

$$\Phi_{J\mathbf{k}}(\mathbf{x}) = \begin{pmatrix} \phi^{(1)}_{J,\mathbf{k}}(\mathbf{x}) \\ \phi^{(2)}_{J,\mathbf{k}}(\mathbf{x}) \\ \phi^{(3)}_{J,\mathbf{k}}(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} \phi^{(a)}_{J,k_x}(x)\phi^{(b)}_{J,k_y}(y)\phi^{(c)}_{J,k_z}(z) \\ \phi^{(a)}_{J,k_x}(x)\phi^{(b)}_{J,k_y}(y)\phi^{(c)}_{J,k_z}(2z) \\ \phi^{(a)}_{J,k_x}(x)\phi^{(b)}_{J,k_y}(2y)\phi^{(c)}_{J,k_z}(2z) \end{pmatrix}, \tag{69}$$

and similarly for the wavelets and the dual functions. These separable multiwavelets are not tensor product wavelets; they do not mix scales and are easily generalized to any dimension.

3.4.2. *Nonseparable Three-Dimensional Polynomial Interpolating Multiwavelets*

In the case of nonseparable multiwavelets, our implementation is more difficult. Let us again consider in three dimensions a simple cubic lattice $\Gamma_0 = \mathbf{L}_0\mathcal{Z}^3$, as seen in Fig. 4a, with the lattice generator

$$\mathbf{L}_0 = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}. \tag{70}$$

Translate Γ_0 by $\mathbf{t}_0 = (1, 1, 1)$ and generate the union of this new lattice with itself, as shown in Fig. 4b. This gives us a body centered cubic (BCC) lattice and lattice generator,

$$\Gamma_1 = \Gamma_0 \cup (\Gamma_0 + \mathbf{t}_0), \quad \mathbf{L}_1 = \begin{bmatrix} -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{bmatrix}. \tag{71}$$

This is equivalent to subdividing Γ_1 into the two disjoint sublattices,

$$\begin{aligned} \Omega_s &= \mathbf{D}_1\Gamma_1 = \Gamma_0 \\ \Omega_w &= \mathbf{D}_1\Gamma_1 + \mathbf{t}_0, \end{aligned} \tag{72}$$

where

$$\mathbf{D}_1 = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}. \tag{73}$$

We now translate Γ_1 by $\mathbf{t}_1 = (0,\ 1,\ 1)$ and generate the union, as shown in Fig. 4c. This gives us a stretched body centered cubic lattice (sBCC) with lattice generator,

$$\Gamma_2 = \Gamma_1 \cup (\Gamma_1 + \mathbf{t}_1), \quad \mathbf{L}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & -1 \end{bmatrix}, \tag{74}$$

which is the same as subdividing Γ_2 into the two disjoint sublattices,

$$\begin{aligned} \Omega_s &= \mathbf{D}_2\Gamma_2 = \Gamma_1 \\ \Omega_w &= \mathbf{D}_2\Gamma_2 + \mathbf{t}_1, \end{aligned} \tag{75}$$

where

$$\mathbf{D}_2 = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & -1 \\ 1 & 1 & -1 \end{bmatrix}. \tag{76}$$

Finally, we translate Γ_2 by $\mathbf{t}_1 = (0, 0, 1)$ and then generate the union to obtain the lattice shown in Fig. 4d. This gets us back to a simple cubic lattice with lattice generator,

$$\Gamma_3 = \Gamma_2 \cup (\Gamma_2 + \mathbf{t}_2), \quad \mathbf{L}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{77}$$

which is the same as subdividing Γ_3 into the two disjoint sublattices,

$$\begin{aligned} \Omega_s &= \mathbf{D}_3\Gamma_3 = \Gamma_2 \\ \Omega_w &= \mathbf{D}_3\Gamma_3 + \mathbf{t}_2, \end{aligned} \tag{78}$$

where

$$\mathbf{D}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & -1 \end{bmatrix}. \tag{79}$$

We have now returned to the original lattice, i.e., $\Gamma_3 = \frac{1}{2}\Gamma_0$, contracted by a factor of 2 in each direction, and with $\mathbf{D}_1\mathbf{D}_2\mathbf{D}_3 = 2\mathbf{I}$. Tables I–III give the PIC's for each lattice refinement in three dimensions, and Fig. 5 shows the placement of the points for the polynomial interpolation. The number in parentheses is the interpolation point, and the number before the parentheses is the number of interpolation points.

TABLE I
The PIC's for the Three-Dimensional Filters: Cubic → BCC

Order	No. points	Points	$P_{\Gamma_0} + t_1$
1	8	{8(1)}	$\left\{-\frac{1}{8}\right\}$
3	32	{8(1), 24(2)}	$\left\{\frac{11}{64}, -\frac{1}{64}\right\}$
5	56	{8(1), 24(2), 24(3)}	$\left\{\frac{91}{512}, -\frac{10}{512}, \frac{1}{512}\right\}$
7	88	{8(1), 24(2), 24(3), 8(5)}	$\left\{\frac{801}{4096}, -\frac{117}{4096}, \frac{12}{4096}, -\frac{1}{4096}\right\}$

TABLE II
The PIC's for the Three-Dimensional Filters: BCC → sBCC

Order	No. points	Points	$P_{\Gamma_1} + t_2$
1	6	{2(1), 4(2)}	$\left\{\frac{1}{3}, \frac{1}{12}\right\}$
2	14	{2(1), 4(2), 8(3)}	$\left\{\frac{1}{4}, \frac{1}{4}, -\frac{1}{16}\right\}$
3	22	{2(1), 4(2), 8(3), 8(4)}	$\left\{\frac{76}{176}, \frac{20}{176}, -\frac{4}{176}, -\frac{3}{176}\right\}$
4	32	{2(1), 4(2), 8(3), 8(4), 8(5)}	$\left\{\frac{156}{384}, \frac{48}{384}, -\frac{6}{384}, -\frac{8}{384}, -\frac{1}{384}\right\}$
5	40	{2(1), 4(2), 8(3), 8(4), 8(5), 2(6), 8(7)}	$\left\{\frac{6486}{16064}, \frac{2040}{16064}, -\frac{267}{16064}, -\frac{121}{16064}, \frac{267}{16064}, -\frac{8}{16064}, \frac{4}{16064}\right\}$

TABLE III
The PIC's for the Three-Dimensional Filters: sBCC → Cubic

Order	No. points	Points	$P_{\Gamma_2} + t_3$
1	4	{4(1)}	$\left\{\frac{1}{4}\right\}$
3	12	{4(1), 8(2)}	$\left\{\frac{10}{32}, -\frac{1}{32}\right\}$
5	16	{4(1), 8(2), 4(3)}	$\left\{\frac{81}{256}, -\frac{9}{256}, \frac{1}{256}\right\}$
7	32	{4(1), 8(2), 4(3), 8(4), 8(5)}	$\left\{\frac{1404}{4096}, -\frac{231}{4096}, \frac{34}{4096}, \frac{27}{4096}, -\frac{3}{4096}\right\}$

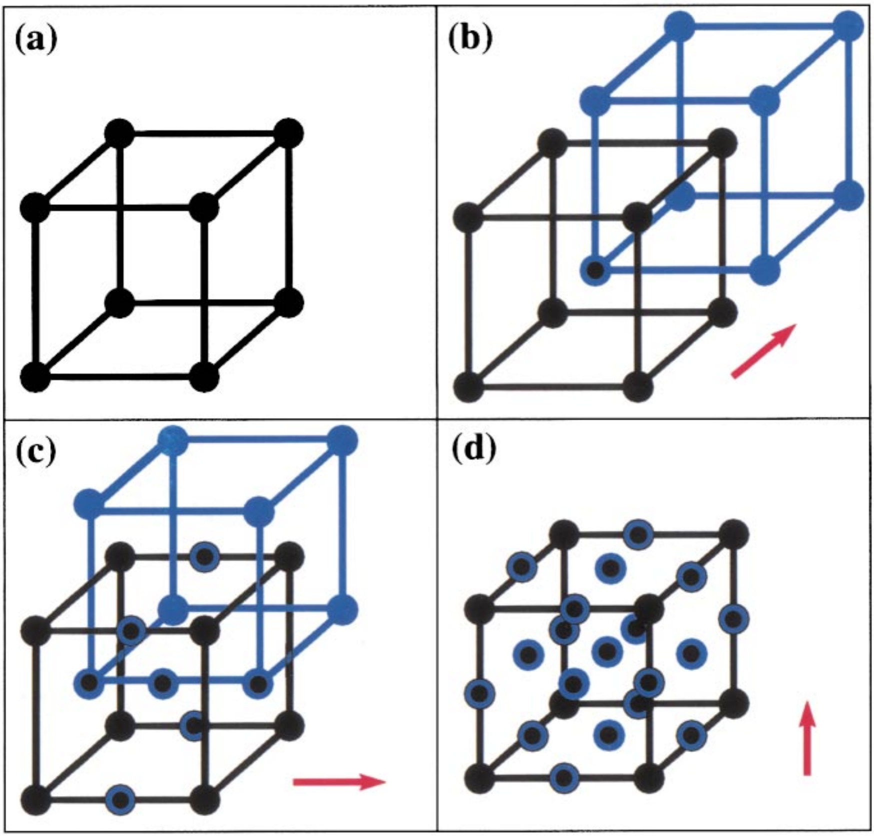

FIG. 4. The lattice translates used in constructing the three-dimensional, nonseparable polynomial interpolating multiwavelet.

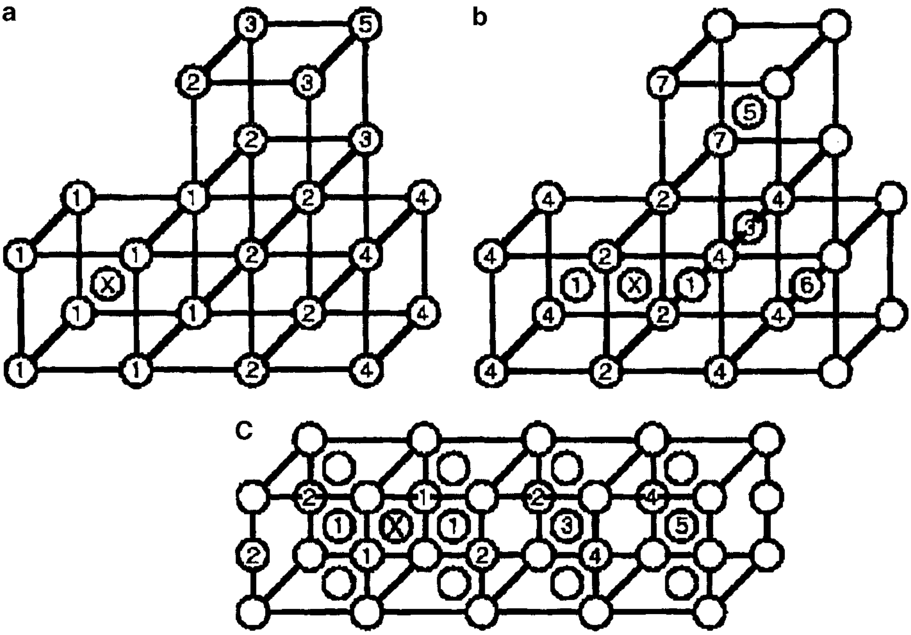

FIG. 5. Placement of the points for the polynomial interpolation. (a) Points for the BCC filter, (b) points for the SBCC filter, and (c) points for the cubic filter.

3.4.3. *Nonseparable, Four-Dimensional, Polynomial Interpolating Multiwavelets*

The procedure that we have devised in the previous section can be extended to any dimension. To show this we also implement our construction in four dimensions. We emphasize that the only constraints are that the lattice translate we choose cannot map lattice points into lattice points, and that we double the number of points at each step. This was easy to visualize in three dimensions, but it is difficult in four dimensions. However, by using algorithm 1 outlined in the previous section, it is straightforward to construct lattice decompositions for the multiresolution analysis. Let us start by the lattice $\Gamma_0 = \mathbf{L}_0\mathcal{Z}^4$ defined by the lattice generator,

$$\mathbf{L}_0 = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}. \tag{80}$$

This is the hyper-cubic lattice. Translate this lattice by $\mathbf{t}_0 = (1, 1, 1, 1)$ and generate the union. This give us a four-dimensional analog to the three-dimensional BCC lattice with lattice generator,

$$\Gamma_1 = \Gamma_0 \cup (\Gamma_0 + \mathbf{t}_0), \quad \mathbf{L}_1 = \begin{bmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \tag{81}$$

which is the same as subdividing Γ_1 into the two disjoint sublattices,

$$\begin{aligned} \Omega_s &= \mathbf{D}_1\Gamma_1 = \Gamma_0 \\ \Omega_w &= \mathbf{D}_1\Gamma_1 + \mathbf{t}_0, \end{aligned} \tag{82}$$

where

$$\mathbf{D}_1 = \begin{bmatrix} -1 & 0 & 0 & 1 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix}. \tag{83}$$

Next, translate Γ_1 by $\mathbf{t}_1 = (0, 1, 0, 1)$ and generate the union. This gives a four-dimensional, rotated hyper-cubic lattice with lattice generator,

$$\Gamma_2 = \Gamma_1 \cup (\Gamma_1 + \mathbf{t}_1), \quad \mathbf{L}_2 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}, \tag{84}$$

which is the same as subdividing Γ_2 into the two disjoint sublattices,

$$\begin{aligned} \Omega_s &= \mathbf{D}_2\Gamma_2 = \Gamma_1 \\ \Omega_w &= \mathbf{D}_2\Gamma_2 + \mathbf{t}_1, \end{aligned} \tag{85}$$

where

$$\mathbf{D}_2 = \begin{bmatrix} 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}, \tag{86}$$

Now, translate Γ_2 by $\mathbf{t}_2 = (0, 0, 1, 1)$ and generate the union. This gives a four-dimensional, rotated hyper-BCC lattice with lattice generator,

$$\Gamma_3 = \Gamma_2 \cup (\Gamma_2 + \mathbf{t}_2), \quad \mathbf{L}_3 = \begin{bmatrix} 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}, \tag{87}$$

which is the same as subdividing Γ_3 into the two disjoint sublattices,

$$\begin{aligned} \Omega_s &= \mathbf{D}_3\Gamma_3 = \Gamma_2 \\ \Omega_w &= \mathbf{D}_3\Gamma_3 + \mathbf{t}_2, \end{aligned} \tag{88}$$

where

$$\mathbf{D}_3 = \begin{bmatrix} -1 & 0 & 0 & 1 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{bmatrix}. \tag{89}$$

Finally, let us translate Γ_3 by $\mathbf{t}_3 = (0,\ 0,\ 0,\ 1)$ and generate the union. This gives us back a hyper-cubic lattice with lattice generator,

$$\Gamma_4 = \Gamma_3 \cup (\Gamma_3 + \mathbf{t}_3), \quad \mathbf{L}_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{90}$$

which is the same as subdividing Γ_4 into the two disjoint sublattices,

$$\begin{aligned} \Omega_s &= \mathbf{D}_4\Gamma_4 = \Gamma_3 \\ \Omega_w &= \mathbf{D}_4\Gamma_4 + \mathbf{t}_3, \end{aligned} \tag{91}$$

where

$$\mathbf{D}_4 = \begin{bmatrix} 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}. \tag{92}$$

Once again $\mathbf{D}_1\mathbf{D}_2\mathbf{D}_3\mathbf{D}_4 = 2\mathbf{I}$. Tables IV–VII give the PIC's for each lattice refinement in four dimensions. The number before the braces is how many points there are with these coordinates. The number in braces is the placement of the points, where we would apply all cyclic permutations to this point, i.e.,

$$(1,\ 0,\ 0,\ 0) \rightarrow \{(\pm 1,\ 0,\ 0,\ 0),\ (0, \pm 1,\ 0,\ 0),\ (0,\ 0, \pm 1,\ 0),\ (0,\ 0,\ 0, \pm 1)\}. \tag{93}$$

TABLE IV
The PIC's for the Four-Dimensional Filters: Hyper-Cubic ⟶ Hyper-BCC

Order	No. points	Points	$P_{\Gamma_0+t_1}$
1	8	{8(1, 0, 0, 0)}	$\left\{\frac{1}{8}\right\}$
3	40	{8(1, 0, 0, 0), 32(1, 1, 1, 0)}	$\left\{\frac{12}{64}, -\frac{1}{64}\right\}$
5	88	{8(1, 0, 0, 0), 32(1, 1, 1, 0), 48(2, 1, 0, 0)}	$\left\{\frac{114}{640}, -\frac{7}{640}, -\frac{1}{640}\right\}$

TABLE V
The PIC's for the Four-Dimensional Filters: Hyper-BCC ⟶ Rotated Hyper-Cubic

Order	No. points	Points	$P_{\Gamma_1+t_2}$
1	16	{16(1, 1, 0, 0)}	$\left\{\frac{1}{16}\right\}$
3	80	{16(1, 1, 0, 0), 64(2, 1, 1, 0)}	$\left\{\frac{12}{128}, -\frac{1}{128}\right\}$
5	176	{16(1, 1, 0, 0), 64(2, 1, 1, 0), 72(2, 2, 1, 1), 24(3, 1, 0, 0)}	$\left\{\frac{102}{1024}, -\frac{11}{1024}, \frac{1}{1024}, \frac{1}{1024}\right\}$

TABLE VI
The PIC's for the Four-Dimensional Filters: Rotated Hyper-Cubic ⟶ Rotated Hyper-BCC

Order	No. points	Points	$P_{\Gamma_2+t_3}$
1	8	{8(1, 0, 1, 0)}	$\left\{\frac{1}{8}\right\}$
3	40	{8(1, 0, 1, 0), 32(2, 1, 1, 0)}	$\left\{\frac{12}{64}, -\frac{1}{64}\right\}$
5	88	{8(1, 0, 1, 0), 32(2, 1, 1, 0), 36(2, 2, 1, 1), 12(3, 1, 0, 0)}	$\left\{\frac{114}{640}, -\frac{7}{640}, -\frac{1}{640}, -\frac{1}{640}\right\}$

TABLE VII
The PIC's for the Four-Dimensional Filters: Rotated Hyper-BCC ⟶ Hyper-Cubic

Order	No. points	Points	$P_{\Gamma_3+t_4}$
1	16	{16(1, 1, 1, 1)}	$\left\{\frac{1}{8}\right\}$
3	80	{16(1, 1, 1, 1), 64(3, 1, 1, 1)}	$\left\{\frac{12}{128}, -\frac{1}{128}\right\}$
5	176	{16(1, 1, 1, 1), 64(3, 1, 1, 1), 96(3, 3, 1, 1)}	$\left\{\frac{102}{1024}, -\frac{11}{1024}, \frac{1}{1024}\right\}$

4. PROPERTIES OF MULTIDIMENSIONAL MULTIWAVELETS

In this section, we compare the separable and nonseparable multiwavelets with the tensor product wavelets regarding their performance at compressing [7] several discontinuous test functions, such as tetrahedrons, cubes, or spheres. These types of test functions are the "worst" case that could happen, for example, in electronic structure problems, as well as in many other phenomena in physics. We may thus judge how well a wavelet-based variational scheme will perform based upon how well the multiwavelets represent these discontinuous test functions.

4.1. Compression of Functions Using Multidimensional Multiwavelets

First, let us present some nomenclature:

- $\{P_1, P_2, \ldots P_d\}$ describes the order of the polynomial interpolation for the nonseparable multiwavelets, where P_i is the polynomial order of the i'th filter.
- $\{\{P_1\} \oplus \{P_2\} \oplus \cdots \oplus \{P_d\}\}$ describes the order of the polynomial interpolation for the separable multiwavelets, where P_i is the polynomial order of the i'th one-dimensional filter.
- $\{\{P_1\} \otimes \{P_2\} \otimes \cdots \otimes \{P_d\}\}$ describes the order of the polynomial interpolation for the tensor product wavelets, where P_i is the polynomial order of the i'th one-dimensional filter.
- Compression ratio (CR),

$$CR = \left[1 - \frac{N_w}{2^{J_{max}}}\right], \tag{94}$$

where N_w is the number of wavelet coefficients above the tolerance ϵ, and J_{max} is the maximum resolution scale.

- Reconstruction error (RE)

$$RE = \frac{\int_V d\mathbf{x}\, |f(\mathbf{x}) - f_R(\mathbf{x})|^2}{\int_V d\mathbf{x}\, |f(\mathbf{x})|^2}, \tag{95}$$

where the reconstructed function is

$$f_R(\mathbf{x}) = \sum_{\mathbf{k}\in\Gamma_{J_0}} s_{J_0\mathbf{k}} \mathbf{\Phi}_{J_0\mathbf{k}}(\mathbf{x}) + \sum_{j=J_0,\mathbf{k}\in\Gamma_j}^{j=J_{max}} \eta_{j\mathbf{k}} \Psi_{j\mathbf{k}}(\mathbf{x}), \tag{96}$$

and

$$\{\eta_{j\mathbf{k}}\} \equiv \{|d_{j\mathbf{k}}| > \epsilon\} \tag{97}$$

is the restricted set of coefficients which are above the tolerance ϵ.

4.1.1. *Compression of Functions in Two Dimensions*

Figure 6 shows an analysis of the reconstruction error vs compression ratio for two-dimensional multiwavelets of order {7, 7} and {{7} ⊕ {7}} as compared to a tensor product wavelet of order {{7} ⊗ {7}} for the discontinuous test functions,

$$f^{(1)}_{test}(\mathbf{x}) = \begin{cases} 1 & if\ \mathbf{x} \in circle \\ 0 & otherwise \end{cases} \quad \text{and} \quad f^{(2)}_{test}(\mathbf{x}) = \begin{cases} 1 & if\ \mathbf{x} \in triangle \\ 0 & otherwise \end{cases}. \tag{98}$$

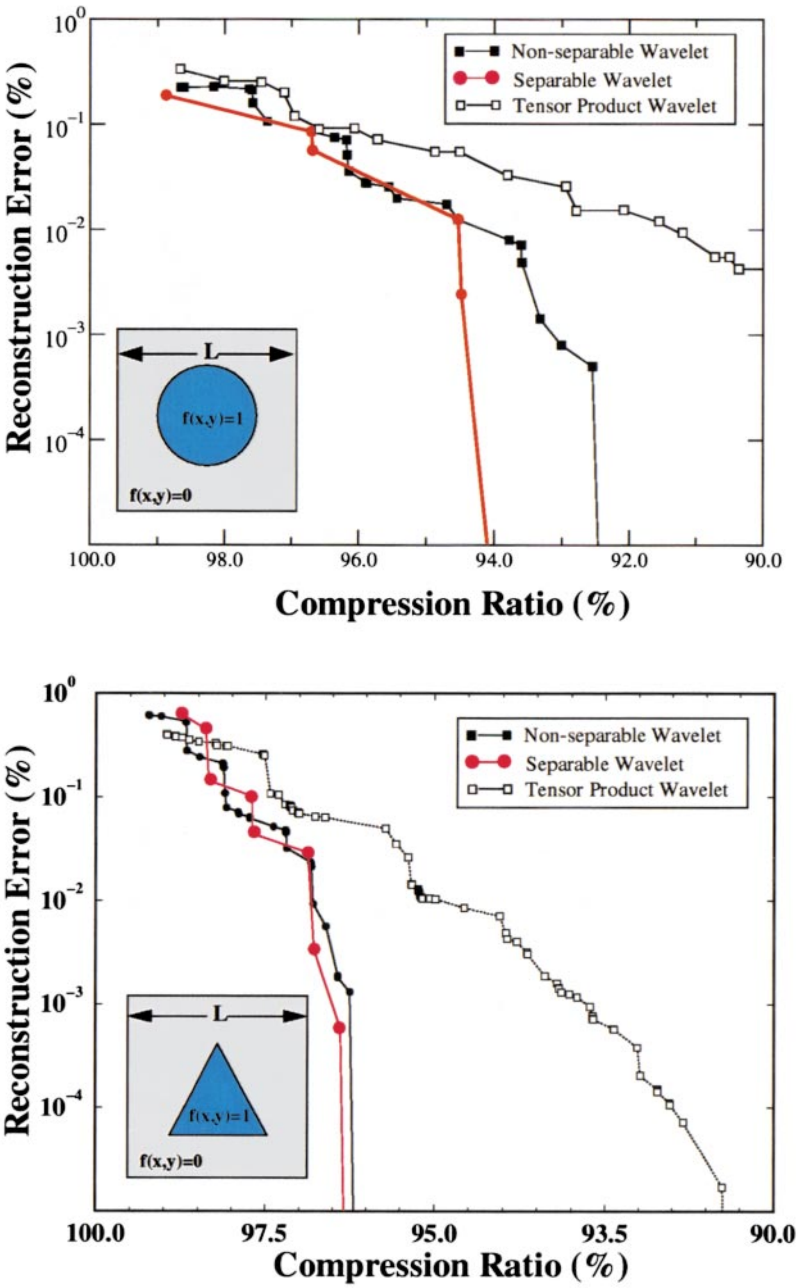

FIG. 6. The reconstruction error compression ratio for the nonseparable and separable, two-dimensional multiwavelet of order {7, 7} and {{7} ⊕ {7}} for the discontinuous test functions $f^1_{test}(\mathbf{x})$ and $f^2_{test}(\mathbf{x})$ as compared to the tensor product wavelet of order {{7} ⊗ {7}}.

For almost the entire range of compression, the nonseparable and separable two-dimensional multiwavelets are an order of magnitude smaller in the reconstruction error than the tensor product wavelets. Since both the separable and nonseparable wavelets have a similar performance at compressing discontinuous functions, we will focus on the nonseparable wavelets in the following.

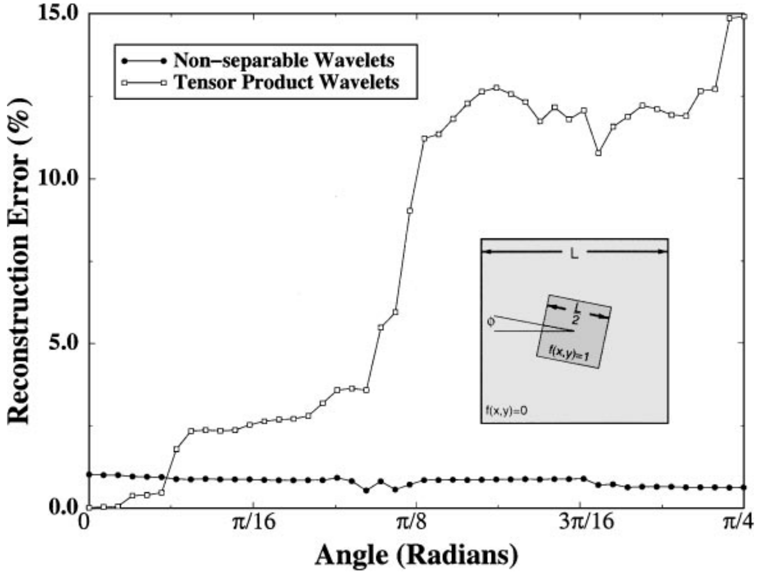

FIG. 7. The reconstruction error vs angle for the nonseparable two-dimensional multiwavelet of order {7, 7} and the tensor product wavelet of polynomial order {{7} ⊗ {7}} for the test function $f_{test}(\mathbf{x}, \phi)$.

The reduction of performance of the tensor product wavelets is due to the strong mixing of scales which biases these wavelets along coordinate directions. To demonstrate this let us consider how the tensor wavelets and nonseparable multiwavelets compress the test function

$$f_{test}(\mathbf{x}, \phi) = \begin{cases} 1 & \textit{if } \mathbf{x} \in \textit{square rotated by } \phi \\ 0 & \textit{otherwise} \end{cases}, \tag{99}$$

where the number of wavelet components N_w is fixed by adjusting the tolerance ϵ. Figure 7 shows the reconstruction error vs angle for our two-dimensional multiwavelets of order {7, 7} as compared to a tensor product wavelet of order {{7} ⊗ {7}}. The number of wavelet components was fixed to $N_w = 2000 \pm 5$. What can be seen from the figure is that the tensor product wavelet is very sensitive to the squares orientation, whereas the nonseparable multiwavelets show very little dependence on the squares orientation. This illustrates the isotropic behavior of the multiwavelets.

4.1.2. *Compression of Functions in Three Dimensions*

Figure 8 shows an analysis of the reconstruction error vs compression ratio for our three-dimensional multiwavelets of order {7, 7, 7} as compared to a tensor product wavelet of order {{7} ⊗ {7} ⊗ {7}} for the discontinuous test functions

$$f^{(3)}_{test}(\mathbf{x}) = \begin{cases} 1 & \textit{if } \mathbf{x} \in \textit{tetrahedron} \\ 0 & \textit{otherwise} \end{cases} \quad \text{and} \quad f^{(4)}_{test}(\mathbf{x}) = \begin{cases} 1 & \textit{if } \mathbf{x} \in \textit{sphere} \\ 0 & \textit{otherwise} \end{cases}. \tag{100}$$

In both cases, our nonseparable multiwavelets compress the test functions significantly better then the tensor product wavelet of comparable order. The gain in the degree at which the nonseparable multiwavelets outperform the tensor wavelet is even higher than that in two dimensions. This trend should continue in higher dimensions and is due to the increased scale mixing of the tensor product wavelets.

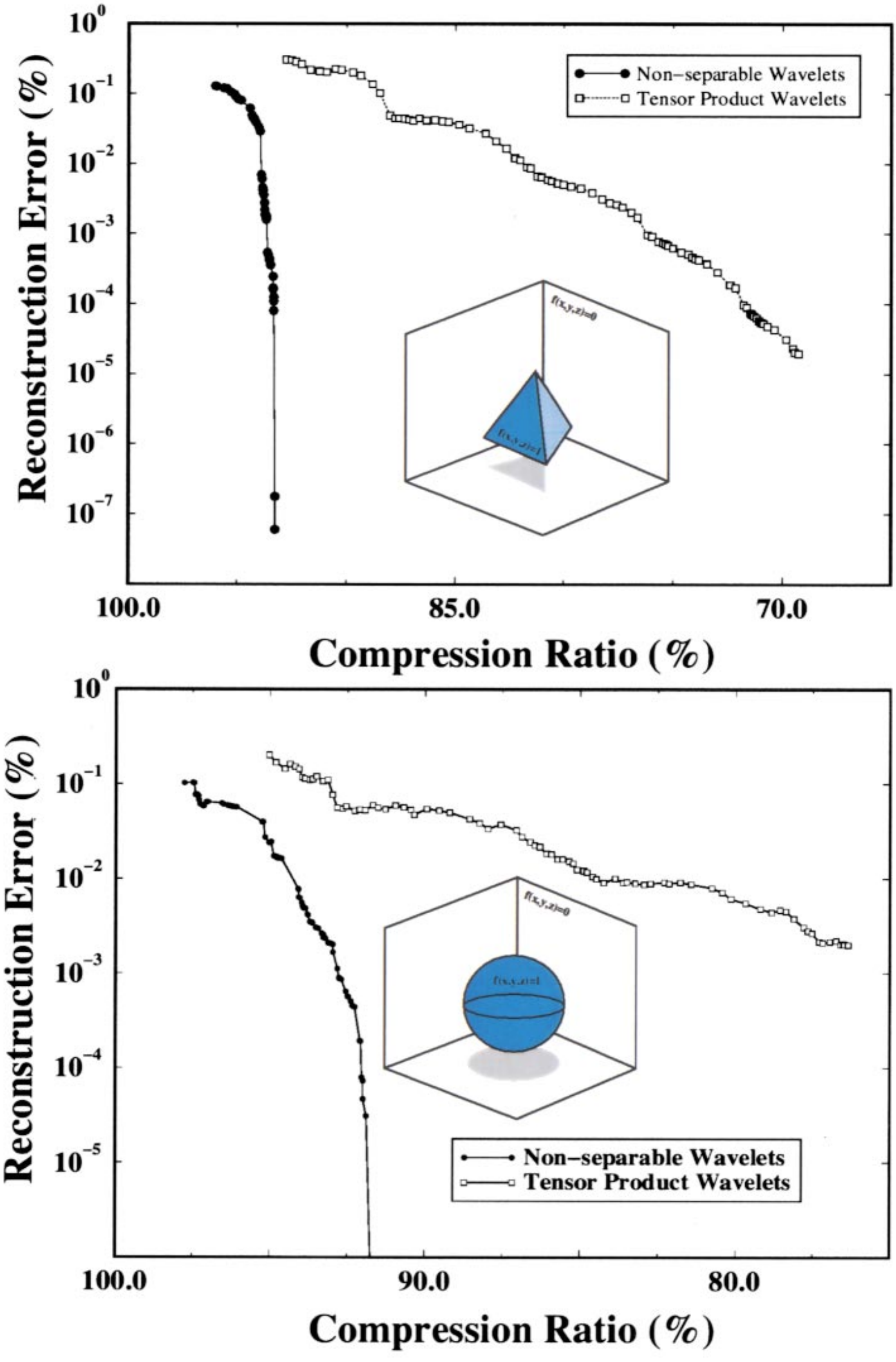

FIG. 8. The reconstruction error vs compression ratio for the nonseparable three-dimensional multiwavelet of order $\{7, 7, 7\}$ and the tensor product wavelet of order $\{\{7\} \otimes \{7\} \otimes \{7\}\}$ for the test functions $f^3_{test}(\mathbf{x})$ and $f^4_{test}(\mathbf{x})$.

4.2. Isotropy of the Multidimensional Multiwavelets

Let us define a measure to determine the isotropy of these wavelets. Consider

$$ISO[\psi] = 1 - \sqrt{\frac{\int_V dR\, d\Omega\, \left|\psi^{(n)}(R,\Omega) - \psi^{(n)}_{ave}(R)\right|^2}{\int_V dR\, d\Omega\, \left|\psi^{(n)}(R,\Omega)\right|^2}}, \tag{101}$$

where

$$\psi^{(n)}_{ave}(R) = \frac{1}{\Omega}\int_\Omega d\Omega\, \psi^{(n)}(R,\Omega). \tag{102}$$

If the function is radial symmetric, this measure is one, and if the function becomes nonradial symmetric, then this measure approaches zero. In the next sections we use this measure to show that the multiwavelets are more isotropic than the tensor product wavelets.

4.2.1. *Three-Dimensional Polynomial Interpolating Multiwavelet*

To illustrate the isotropy and smoothness of a nonseparable multiwavelet in three dimensions, we present Figs. 9 and 10. Figure 9a shows an iso-surface plot for the three-dimensional multiwavelet $\psi^{(1)}(\mathbf{x})$ of order $\{7, 7, 7\}$. Figures 9b and 9c show a surface and contour plot of this wavelet for the $z = 0$ plane. This wavelet is fairly isotropic and smooth; the smoothness is due to the high degree of the polynomial interpolation, and the isotropy is due to the symmetry of the lattice decomposition. Figure 10 shows the radial plot of the same wavelet function of order $\{7, 7, 7\}$ in the three different symmetry directions $\{x,\ y = 0,\ z = 0\}$, $\{x = y,\ z = 0\}$, and $\{x = y = z\}$. As can be seen from Fig. 10, as the radius approaches zero the three-dimensional multiwavelet becomes more isotropic. This should be contrasted with the tensor product wavelet in which this does not happen. Generally the tensor product wavelet will remain anisotropic as the radius approaches zero.

Table VIII shows the isotropy measure of our three-dimensional nonseparable and separable multiwavelet of order $\{7, 7, 7\}$ and $\{\{7\} \oplus \{7\} \oplus \{7\}\}$ compared to tensor product wavelets of order $\{\{7\} \otimes \{7\} \otimes \{7\}\}$. The tensor product wavelets are reasonably isotropic as long as the difference in the j-scales are not too large but become very anisotropic ($ISO[\psi] < 0.50$) once the difference in the j-scales exceeds about 2. However, the multiwavelets are isotropic ($ISO[\psi] > 0.50$) in all cases, especially the nonseparable multiwavelets ($ISO[\psi] > 0.75$).

4.2.2. *Four-Dimensional, Polynomial Interpolating Multiwavelets*

We present Figs. 11 and 12 to further illustrate the isotropy and smoothness of the four-dimensional nonseparable multiwavelet. Figure 11 is a surface plot for the four-dimensional multiwavelet $\psi^{(1)}(\mathbf{x})$ of order $\{5, 5, 5, 5\}$ for the $z = 0$, $t = 0$ plane. Figure 12 shows the radial plot of the same wavelet function of order $\{5, 5, 5, 5\}$ in the four symmetry directions $\{x,\ y = 0,\ z = 0,\ t = 0\}$, $\{x = y,\ z = 0,\ t = 0\}$, $\{x = y = z,\ t = 0\}$, and $\{x = y = z = t\}$. Because the order of the polynomial interpolation is not as large as it was in the three-dimensional case, this wavelet is not as smooth as its three-dimensional counterpart. However, it is more isotropic. This can be clearly seen in Fig. 12, where the

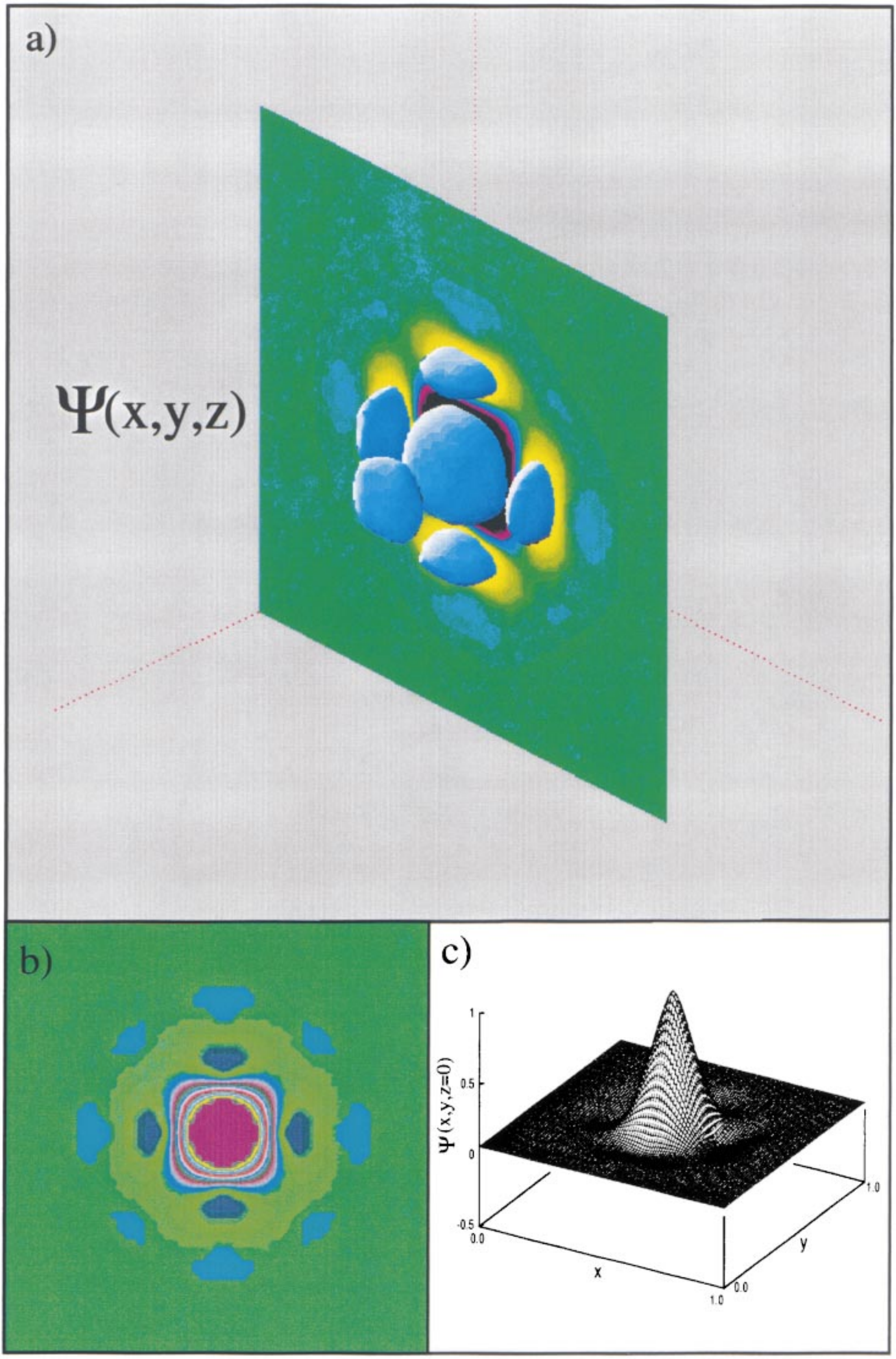

FIG. 9. (a) An iso-surface plot of the three-dimensional nonseparable multiwavelet $\psi^{(1)}(\mathbf{x})$ for order {7, 7, 7}. Also shown, (b) and (c), a density and surface plot of the $z = 0$ plane for this wavelet.

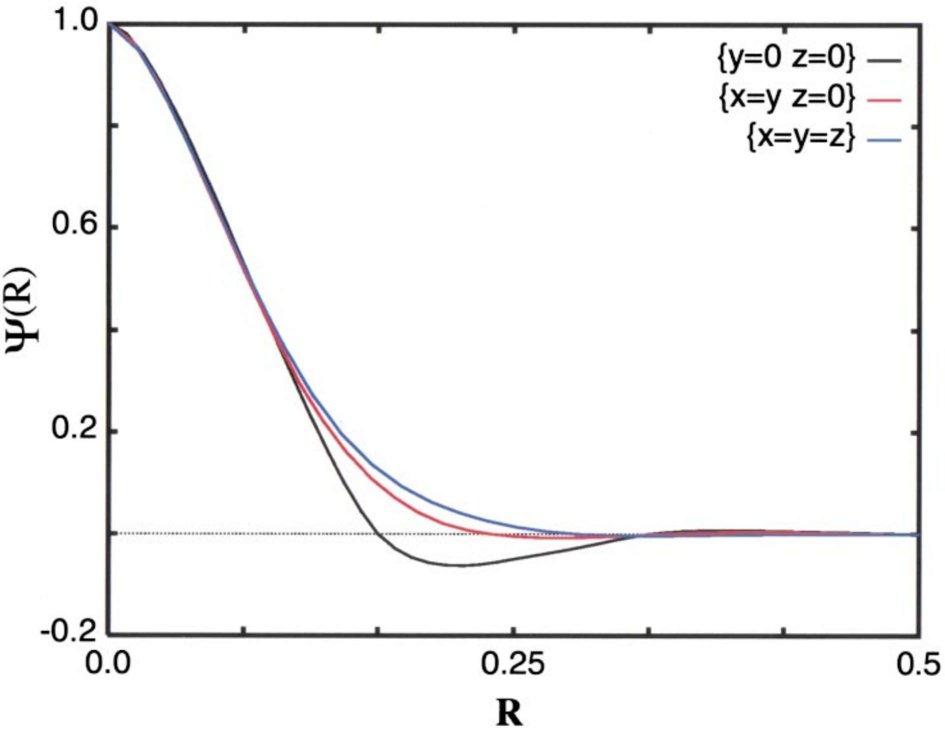

FIG. 10. A radial plot of the three-dimensional nonseparable multiwavelet $\psi^{(1)}(\mathbf{x})$ of order {7, 7, 7} in the three symmetry directions.

relative degree of isotropy around the core region is greater than in the three-dimensional case.

In Table IX we show the isotropy measure of the four-dimensional nonseparable multiwavelet of order {5, 5, 5, 5}. Again, as the isotropy measure indicates, this multiwavelet is even more isotropic than its three-dimensional counterpart.

TABLE VIII
The Isotropy Measure for Comparison of the Multiwavelets with the Tensor Product Wavelets

Tensor-product wavelet	Order	$ISO[\psi]$
$\psi_3(x)\psi_3(y)\psi_3(z)$	$\{\{7\}\otimes\{7\}\otimes\{7\}\}$	0.8116
$\psi_3(x)\psi_4(y)\psi_4(z)$	$\{\{7\}\otimes\{7\}\otimes\{7\}\}$	0.5094
$\psi_3(x)\psi_4(y)\psi_5(z)$	$\{\{7\}\otimes\{7\}\otimes\{7\}\}$	0.3452
Separable multiwavelet	Order	$ISO[\psi]$
$\psi_3^{(1)}(x, y, z)$	$\{\{7\}\oplus\{7\}\oplus\{7\}\}$	0.8116
$\psi_3^{(2)}(x, y, z)$	$\{\{7\}\oplus\{7\}\oplus\{7\}\}$	0.5196
$\psi_3^{(3)}(x, y, z)$	$\{\{7\}\oplus\{7\}\oplus\{7\}\}$	0.5094
Nonseparable multiwavelet	Order	$ISO[\psi]$
$\psi_3^{(1)}(x, y, z)$	{7, 7, 7}	0.9433
$\psi_3^{(2)}(x, y, z)$	{7, 7, 7}	0.8585
$\psi_3^{(3)}(x, y, z)$	{7, 7, 7}	0.7531

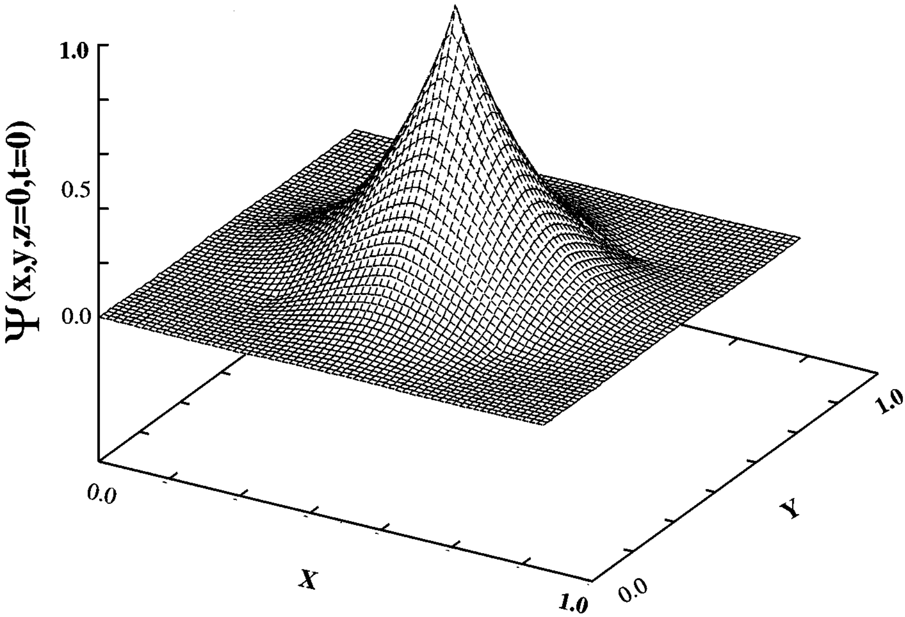

FIG. 11. A surface plot of the four-dimensional nonseparable multiwavelet $\psi^{(1)}(\mathbf{x})$ of order {5, 5, 5, 5} for the $z = 0$ and $t = 0$ plane.

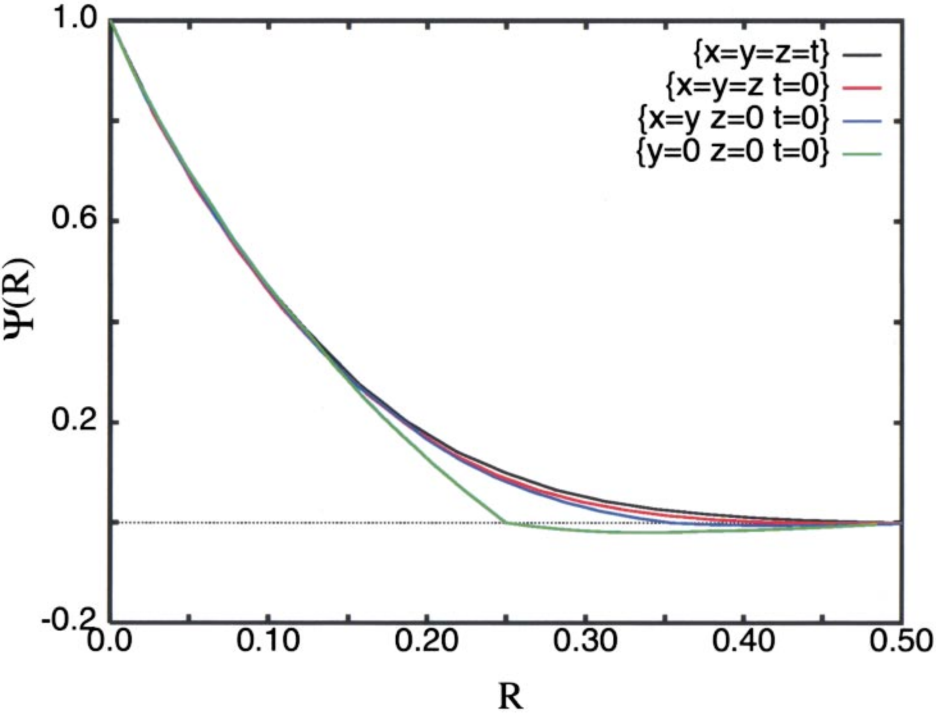

FIG. 12. A radial plot of the four-dimensional nonseparable multiwavelet $\psi^{(1)}(\mathbf{x})$ of order {5, 5, 5, 5} in the four symmetry directions.

TABLE IX

The Isotropy Measure for the Four-Dimensional Nonseparable Multiwavelets of Order {5, 5, 5, 5}

Nonseparable multiwavelet	$ISO[\psi]$
$\psi_3^{(1)}(x, y, z, t)$	0.9612
$\psi_3^{(2)}(x, y, z, t)$	0.8765
$\psi_3^{(3)}(x, y, z, t)$	0.9615
$\psi_3^{(4)}(x, y, z, t)$	0.8759

5. CONCLUSIONS

Using the lifting scheme devised by Sweldens *et al.* [21, 33, 34] and polynomial interpolation in $\mathbf{R}^d$ [17], we have constructed nonseparable and separable polynomial interpolating multiwavelets in multiple dimensions. This was done by devising a scheme for periodically subdividing multi-dimensional lattices such that in $N = d$ number of steps we return to the original lattice scaled by a factor of 2 in the coordinate directions. The scheme was demonstrated by constructing three-dimensional separable and nonseparable multiwavelets as well as four-dimensional nonseparable multiwavelets. Finally, the compressibility was investigated and analyzed in terms of the isotropy of the wavelets. It was found that the multiwavelets are more isotropic and efficient at compressing functions as compared to the tensor product wavelets of similar order.

In the appendices, we present two practical algorithms which are a natural outcome of the presented work (i) the in-place inverse wavelet transform which allows the calculation of the value of a function at a point, where the function is represented in the wavelet basis and (ii) the top–down algorithm which allows for the adaptive wavelet analysis of a function, where as an example of the efficiency of the top–down algorithm, we present the wavelet transform of the potential generated by 32 arbitrarily placed point charges.

It is possible to extend the presented methodology toward developing new classes of wavelets. One possible extension of the presented research is in constructing orthogonal multiwavelets. Another area of interest is constructing biorthogonal multiwavelets with specific properties, such as ultra-sparse operator representations [12, 18]. We are working toward this end by constructing a biorthogonal multiwavelet basis which diagonalizes the Poisson operator [35]. In the immediate future, we are implementing the separable multiwavelets into a generalized object-oriented code for use in solving integral–differential equations. This library should be partially suited for systems in which multiscale phenomena are important, i.e., the electronic structure of condensed matter systems.

APPENDIX A

In-Place Inverse Wavelet Transform

Given a multidimensional wavelet representation of a function, Eq. (96), we would like to devise an algorithm for calculating the function value at some point $\mathbf{x}$. The standard approach is to sum over all the expansion coefficients multiplied by the basis functions at point **x**. However, this can be extremely time consuming for large basis sets, and for the wavelet basis this approach does not take advantage of the wavelets' locality. Instead, we have devised an algorithm for calculating the value of functions, represented in the wavelet basis, at point **x** which takes advantage of the wavelets' locality. This algorithm, instead of scaling $O(N_w)$, where N_w is the number of wavelet coefficients, scales $O(J_{max})$, where J_{max} is the finest scale of the wavelet expansion. The algorithm takes advantage of the wavelet locality by calculating the inverse wavelet transform only in a region necessary to continue onto the next scale; this is coded as follows:

ALGORITHM 2 (Inverse WTR (**x**, WTR(f(**x**)))).

```
begin
j = 0
  while $j \leq j_{max}$ do
    calculate $\mathbf{x}_{Range} \approx 2\sup\{\tilde{\psi}_{j,2^j\mathbf{x}}\}$
    loop over k
      if $\|\mathbf{k} - 2^j\mathbf{x}\| < \mathbf{x}_{Range}$
        calculate $s_{j,\mathbf{k}} \leftarrow s_{j-1,\mathbf{k}} - \sum_{\mathbf{l}\in\Gamma_\mathbf{j}+\mathbf{t_j}} U_{j,\mathbf{1}} d_{j-1,\mathbf{k}-\mathbf{l}}$    $\mathbf{k} \in \Gamma_j$
        calculate $s_{j,\mathbf{k}} \rightarrow d_{j-1,\mathbf{k}} + \sum_{\mathbf{l}\in\Gamma_\mathbf{j}} P_{j,\mathbf{l}} s_{j-1,\mathbf{k}-\mathbf{l}}$    $\mathbf{k} \in \Gamma_J + \mathbf{t}_J$
      end if
    end loop
end while
end
```

APPENDIX B

Top–Down Algorithm

We have implemented a practical algorithm which calculates only the significant wavelet coefficients, above some predetermined threshold, ϵ, of a function. The traditional approach is to use the pyramid algorithm [10], which calculates all the wavelet coefficients from a chosen finest scale of resolution up to the coarsest. This can be highly inefficient because a large percentage of the coefficients are usually discarded. Here we have devised a method for eliminating this problem, via a top–down algorithm [38].

The basis philosophy of the top–down algorithm is to use the coefficients at scale j to estimate the coefficients at the finer scale $j+1$; this is possible because the wavelet coefficients at scale j are usually related to the wavelet coefficients at scale $j+1$ via,

$$d_{j+1,\mathbf{k}} \approx 2^{-(P+1)} d_{j,\mathbf{k}'} \quad \mathbf{k} \in \sup\left[\tilde{\psi}^{(n)}_{j,\mathbf{k}'}\right], \tag{B.1}$$

where P is the number of vanishing moments of the wavelet, and the function $f(\mathbf{x}) \in L_2$. This relation suggests an algorithm for calculating the significant wavelet coefficients.

1. Calculate all the wavelets coefficients on scale j_{min}, where j_{min} is a resolution scale which captures the "relevant" structure of the function $f(\mathbf{x})$.

2. Keep only those wavelet coefficients above some predetermined threshold ϵ. Since the wavelets we use are interpolating, no prefiltering is necessary.

3. For each wavelet coefficient at scale j, calculate all wavelets coefficients at scale $j+1$ which are within the support of the wavelets at scale j.

4. Keep only the wavelet coefficients above the threshold ϵ.

5. Repeat this process using the wavelet coefficients at scale $j+1$ to deduce the wavelet coefficients at scale $j+2$, etc.

In pseudo-code:

ALGORITHM 3 (TopDownWTR (ϵ, j_{min}, j_{max}, $f(\mathbf{x})$)).

```
begin
    $j = j_{min}$: calculate $\{\{d_{0,\mathbf{k}}\}, \ldots \{d_{j_{min},\mathbf{k}}\}\}$
    while $(j \leq j_{max}$ and $N_W^j \neq 0)$ do
        loop over $\mathbf{k} \in \{d_{j,\mathbf{k}}\}$
            loop over $\mathbf{k}' \in \sup[d_{j,\mathbf{k}}]$
                calculate $d_{j+1,\mathbf{k}'} = \sum_{\mathbf{l}} \tilde{g}_{j+1,\mathbf{l}} f(\mathbf{x} - \mathbf{k}' - \mathbf{1})$
                if $|d_{j+1,\mathbf{k}'}| < \epsilon$
                    store $d_{j+1,\mathbf{k}'} \rightarrow \{d_{j+1,\mathbf{k}}\} : N_W^j \rightarrow N_W^j + 1$
                end if
            end loop
        end loop
        $j \rightarrow j+1$
    end while
    return $\{\{d_{0\mathbf{k}}\}, \{d_{1\mathbf{k}}\}, \ldots \{d_{j\mathbf{k}}\}\}$
end
```

This algorithm has several advantages over the traditional approach,

- The time for the transform scales like $O(N_W)$, where N_W is the number of wavelet coefficients above the threshold ϵ.
- It is possible to calculate the wavelet coefficients to very fine scales of resolution, where, in principle, no maximum scale is assumed.
- Function structures are adaptively resolved where they are needed.

However, one disadvantage of this algorithm is that there is no guarantee of obtaining all the wavelet coefficients above the threshold ϵ; i.e., some details could be missed. One remedy for this is to decrease the threshold. Another remedy of this problem is to calculate the L_2 norm of the function and the L_2 norm of the wavelet representation of the function. If the difference between these two norms is greater than the threshold, then some coefficients have been missed, and the transform has to be restarted at a finer scale. However, for most problems in physics we have to approximate knowledge of the function's behavior, and all the "relevant" features are already known. For example, Fig. 13 shows a slice of a wavelet transform of 32 arbitrarily placed point charges; i.e.,

$$f_{test}(\mathbf{x}) = \sum_{i=1}^{32} \frac{1}{|\mathbf{x} - \mathbf{x}_i|} \quad \mathbf{x}_i \in [0,1]^3. \tag{B.2}$$

Eight of the charges lie within the cut plane. Both the size of the circles and the colors encode the scale on which the particular wavelet resides, blue being coarse and red being fine. We also include a contour plot of the potential generated by these 32 point charges to emphasize the adaptive placements of the wavelets. It is easily seen that the above algorithm

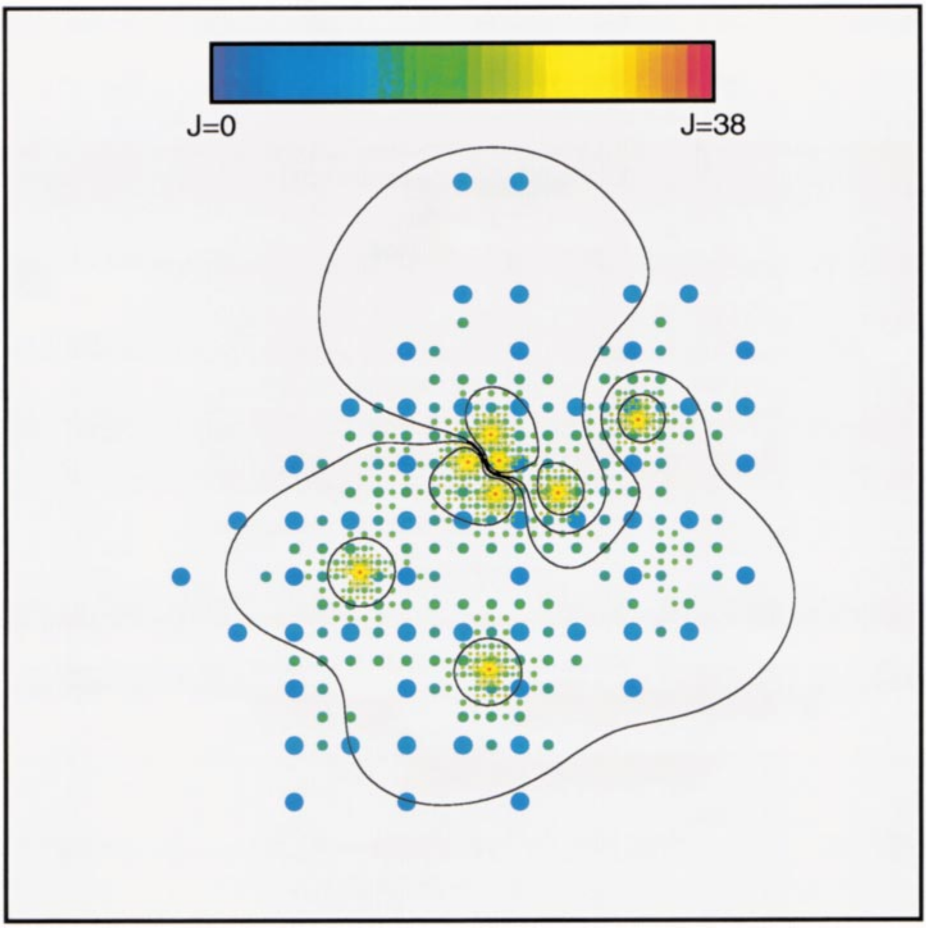

FIG. 13. The placement of the multiwavelets on the $z = 0$ plane for 32 arbitrarily placed point charges in a cubic box. Also shown is a contour plot of the potential.

very nicely resolves all the relevant structures of this function. It should also be noted to what scale we were able to do the analysis, $j_{max} = 38$ or 2^{38} number of points. This would normally take a 500-MHz processor about $2\frac{1}{2}$ CPU years to calculate, whereas the top–down algorithm took about 10 CPU min.

ACKNOWLEDGMENTS

We acknowledge Brad Clements, John Wills, Duane Wallace, Pieter Swart, and Galen Straub for help and insightful discussions. A. M. N. Niklasson is grateful for support from the Swedish Foundation for International Cooperation in Research and Higher Education (STINT).

REFERENCES

1. O. K. Anderson, Linear methods in band theory, *Phys. Rev. B* **12**, No. 8, 3060 (1975).
2. Sankar Basu, Multi-dimensional filter banks and wavelets—A system theoretic perspective, *J. Franklin Inst.* **335B**, No. 8, 1367 (1998).

3. G. Beylkin, Fast wavelet transforms and numerical algorithms I. *Comm. Pure Appl. Math.* **44**, No. 2, 141 (1991).
4. S. F. Boys, Electronic wave functions. II. A calculation for the ground state of the beryllium atom, *Proc. Roy. Soc. London Ser. A* **201**, 125 (1950).
5. M. E. Brewster, G. I. Fann, and Z. Y. Yang, Wavelets for electronic structure calculations, *J. Math Chem.* **22**, Nos. 2–4, 117 (1997).
6. M. Challacombe, A simplified density matrix minimization for linear scaling self-consistent field theory, *J. Chem. Phys.* **110**, No. 5, 2332 (1999).
7. K. Charles, Chui Montefusco, Laura Montefusco, and Luigia Puccio, Eds. *Wavelets: Theory, Algorithms, and Applications*, Wavelet Analysis and Its Applications (Academic Press, San Diego, 1994), Vol. 5.
8. Chen and Debao, *Cardinal Spline Wavelets*, Ph.D. thesis (University of Texas at Austin, 1994).
9. K. Cho, T. A. Arias, J. D. Joannopoulos, and P. K. Lam, Wavelets in electronic-structure calculations, *Phys. Rev. Lett.* **71**, No. 12, 1808 (1993).
10. C. K. Chui, *An Introduction to Wavelets* (Academic Press, San Diego, 1992).
11. I. Daubechies, Orthonormal bases of compactly supported wavelets, *Comm. Pure Appl. Math.* **41**, 909 (1988).
12. I. Daubechies, Two recent results on wavelets: Wavelet bases for the interval, and biorthogonal wavelets diagonalizing the derivative operator, in *Recent Advances in Wavelet Analysis*, edited by Larry L. Schumaker and Glenn Webb (Academic Press, San Diego 1994), p. 237.
13. I. Daubechies and J. Lagarias, Two-scale difference equations, I. *SIAM J. Math. Anal.* **22**, 1388 (1991).
14. I. Daubechies and J. Lagarias, Two-scale difference equations, II. *SIAM J. Math. Anal.* **23**, 1031 (1992).
15. Ingrid Daubechies, *Ten Lectures on Wavelets* (Society for Industrial and Applied Math., Philadelphia, 1992).
16. Ingrid Daubechies, Orthonormal bases of compactly supported wavelets: II. Variations on a theme, *SIAM J. Math. Anal.* **24**, 499 (1993).
17. Carl de Boor and Amos Ron, Computational aspects of polynominal interpolation in several variables, *Math Comput.* **58**, 198 (1997).
18. Fredrik Ekstedt, *Wavelet Diagonalization of Convolution Operators*, Ph.D. thesis (Göteborg University, Göteborg, Sweden, 1997).
19. S. Goedecker and O. Ivanov, Algorithms & applications: Solution of multiscale partial differential equations using wavelets, *Comput. Phys.* **12**, No. 6, 548 (1998).
20. D. P. Hardin and J. A. Marasovich, Biorthogonal multiwavelets on $[-1, 1]$, *ACJHA* **7**, No. 1, 34 (1999).
21. Jelena Kovačević and Wim Sweldens, Wavelet families of increasing order in arbitrary dimensions, *IEEE Trans. Image Process.* **9**, No. 3, 480 (2000).
22. J. Kovačević and M. Vetterli, Nonseparable multidimensional perfect reconstruction filter banks and wavelet basis for $\Re r^d$, *IEEE Trans. Inform. Theory* **38**, No. 2, 533 (1992).
23. Stéphane Mallat, *A Wavelet Tour of Signal Processing* (Academic Press, San Diego, 1998).
24. M. C. Payne, M. P. Teter, D. C. Allan, T. A. Arias, and J. D. Joannopoulos, Iterative minimization techniques for abinitio total-energy calculation: Molecular-dynamics and conjugate gradients, *Rev. Mod. Phys.* **64**, No. 4, 1045 (1992).
25. W. E. Pickett, Pseudopotential methods in condensed matter applications, *Comput. Phys. Rep.* **9**, No. 3, 115 (1989).
26. Gerlind Plonka and Vasily Strela, From wavelets to multiwavelets, in *Mathematical Methods for Curves and Surfaces II*, edited by M. Dahlem, T. Lyche, and L. Shumaker (Vanderbilt University Press, 1998).
27. D. A. Richie and K. Hess, Wavelet based electronic structure calculations, *Micro. Eng.* **47**, Nos. 1–4, 333 (1999).
28. S. D. Riemenschneider and Z. Shen, Wavelets and pre-wavelets in low dimensions, *J. Approx. Theory* **71**, No. 1, 18 (1992).
29. J. B. Smith and J. G. Gay, Inherently self-consistent procedure for obtaining electronic structure: Results for a lithium particle, *Phys. Rev. B* **12**, No. 10, 4238 (1975).
30. David Stanhill and Yehoshua Y. Zeevi, Two-dimensional orthognal filter banks and wavelets with linear phase, *IEEE Trans. Signal Process.* **46**, No. 1, 183 (1998).

31. J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis* (Springer-Verlag, New York, 1980).

32. Vasily Strela, *Multiwavelets: Theory and Applications*, Ph.D. thesis (Massachusetts Institute of Technology, 1996).

33. Wim Sweldens, The lifting scheme: A custom-design construction of biorthogonal wavelets, *Appl. Comput. Harmon. Anal.* **3**, 186 (1996).

34. Wim Sweldens, The lifting scheme: A construction of second generation wavelets, *SIAM J. Math. Anal.* **29**, No. 2, 511 (1997).

35. C. J. Tymczak, Anders M. N. Niklasson, and Heinrich Röder, Diagonalization of the poisson operator using bi-orthogonal multi-wavelets, unpublished manuscript.

36. C. J. Tymczak and X. Q. Wang, Orthonormal wavelet basis for quantum molecular-dynamics, *Phy. Rev. Lett.* **78**, 3654 (1997).

37. S. O. Wei and M. Y. Chou, Wavelets in self-consistent electronic structure calculations, *Phys. Rev. Lett.* **76**, No. 15, 2650 (1996).

38. D. Zorin, P. Schröder, and W. Sweldens, Interpolating subdivision for meshes with arbitrary topology, in *Computer Graphics Proc. (SIGGRAPH'96)*, p. 189.