Eidgenössische
Technische Hochschule
Zürich

Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

Fast evaluation of nonlinear functionals of tensor product wavelet expansions

C. Schwab and R. Stevenson*

Research Report No. 2010-05
February 2010

Seminar für Angewandte Mathematik
Eidgenössische Technische Hochschule
CH-8092 Zürich
Switzerland

*Korteweg-de Vries Institute for Mathematics, University of Amsterdam, P.O. Box 94248, 1090 GE Amsterdam, The Netherlands

FAST EVALUATION OF NONLINEAR FUNCTIONALS OF TENSOR PRODUCT WAVELET EXPANSIONS

CHRISTOPH SCHWAB AND ROB STEVENSON

Abstract. For a nonlinear functional f, and a function u from the span of a set of tensor product interpolets, it is shown how to compute the interpolant of $f(u)$ from the span of this set of tensor product interpolets in linear complexity, assuming that the index set has a certain multiple tree structure. Applications are found in the field of (adaptive) tensor product solution methods for semilinear operator equations by collocation methods, or after transformations between the interpolet and (bi-) orthogonal wavelet bases, by Galerkin methods.

1. Introduction

1.1. **Background.** Functions from Sobolev spaces on n-fold product domains can be approximated very efficiently by best N-term approximation from a tensor product wavelet basis ([Nit06, SU09]). With this type of non-linear approximation the advantages of the linear hyperbolic cross or sparse-grid approximation are extended to a much larger class of functions. For large classes of functions, the approximation rates can be shown to be (nearly) independent of n.

Adaptive wavelet schemes for solving operator equations were shown to converge with the best possible convergence rate, in linear complexity ([CDD01, CDD02, GHS07, CDD03]). For tensor product wavelet bases, in [SS08, DSS09, SS09] such schemes were applied to linear operator equations, with which the optimal, (nearly) n-independent approximation rates could be realized.

When these schemes are applied to *semilinear* equations, the problem arises how to evaluate the nonlinear term without loosing the linear complexity. To describe this problem, we need to introduce some notation. Let $\Psi := \{\psi_\lambda : \lambda \in \nabla\}$, $\tilde{\Psi} := \{\tilde{\psi}_\lambda : \lambda \in \nabla\}$ denote biorthogonal wavelet bases. The resulting biorthogonal n-fold tensor product wavelet bases are then given by $\boldsymbol{\Psi} := \{\boldsymbol{\psi}_{\boldsymbol{\lambda}} := \otimes_{i=}^{n} \psi_{\lambda_i} : \boldsymbol{\lambda} \in \boldsymbol{\nabla} := \nabla^n\}$ and $\tilde{\boldsymbol{\Psi}} := \{\tilde{\boldsymbol{\psi}}_{\boldsymbol{\lambda}} := \otimes_{i=}^{n} \tilde{\psi}_{\lambda_i} : \boldsymbol{\lambda} \in \boldsymbol{\nabla}\}$.

Given a nonlinear function f, a subset $\bar{\boldsymbol{\Lambda}} \subset \boldsymbol{\nabla}$, and a current approximation $u \in \operatorname{span}\{\boldsymbol{\psi}_{\boldsymbol{\lambda}} : \boldsymbol{\lambda} \in \bar{\boldsymbol{\Lambda}}\}$, main tasks inside an adaptive wavelet scheme are

- to *predict* a $\bar{\boldsymbol{\Lambda}} \subset \boldsymbol{\Lambda} \subset \boldsymbol{\nabla}$ such that the distance between $f(u)$ and the biorthogonal projection $\sum_{\boldsymbol{\lambda} \in \boldsymbol{\Lambda}} \boldsymbol{\psi}_{\boldsymbol{\lambda}}(f(u)) \tilde{\boldsymbol{\psi}}_{\boldsymbol{\lambda}}$ is less than some prescribed tolerance ε in the dual norm, and subsequently
- to *compute* this biorthogonal projection in $\mathcal{O}(\#\boldsymbol{\Lambda})$ operations, possibly up to a quadrature error of order ε.

Date: December 15, 2009.

2000 *Mathematics Subject Classification.* 05C05, 15A69, 41A05, 41A63, 42C40, 65Y20, 68Q25.

Key words and phrases. Interpolets, wavelets, tensor product approximation, tree approximation, linear complexity, semilinear equations.

In this paper, we assume that $\boldsymbol{\Lambda}$ is determined, i.e., we postpone the design of a prediction step to future work, and focus on the second task being the evaluation of the biorthogonal projection in linear complexity. By possibly enlarging $\bar{\boldsymbol{\Lambda}}$, i.e., by adding zero coefficients to the expansion for u, without loss of generality we can take $\bar{\boldsymbol{\Lambda}} = \boldsymbol{\Lambda}$.

The obvious approach to evaluate $\sum_{\boldsymbol{\lambda}\in\boldsymbol{\Lambda}} \boldsymbol{\psi}_{\boldsymbol{\lambda}}(f(u))\tilde{\boldsymbol{\psi}}_{\boldsymbol{\lambda}}$ is to approximate each term $\boldsymbol{\psi}_{\boldsymbol{\lambda}}(f(u))$ using quadrature. For each $\boldsymbol{\lambda}$, this requires the evaluation of $f(u)$ in some quadrature points. Due to the multilevel structure of a wavelet basis, the number of wavelets in the expansion $u = \sum_{\boldsymbol{\lambda}\in\boldsymbol{\Lambda}} \tilde{\boldsymbol{\psi}}_{\boldsymbol{\lambda}}(u)\boldsymbol{\psi}_{\boldsymbol{\lambda}}$ that do not vanish in each of these points is not of order 1 uniformly in $\boldsymbol{\Lambda} \subset \boldsymbol{\nabla}$, and, as a result, the overall complexity of this naive implementation is not of order $\#\boldsymbol{\Lambda}$.

In [DSX00, BDS07], for the non-tensor product case, the following alternative approach was developed: Restricting to $\Lambda \subset \nabla$ that have a *tree* structure, and using the local supports of the primal wavelets, in a bottom-to-top sweep the function $u = \sum_{\lambda\in\Lambda} \tilde{\psi}_\lambda(u)\psi_\lambda$ is re-expressed in $\mathcal{O}(\#\Lambda)$ operations as $u = \sum_{\lambda\in\hat{\Lambda}} c_\lambda\phi_\lambda$, where $\{\phi_\lambda : \lambda \in \hat{\Lambda}\}$ is a collection of scaling functions with $\operatorname{span}\{\phi_\lambda : \lambda \in \hat{\Lambda}\} \supset \operatorname{span}\{\psi_\lambda : \lambda \in \Lambda\}$ and $\#\hat{\Lambda} = \mathcal{O}(\#\Lambda)$. Using the local supports of the dual wavelets, in a top-to-bottom sweep the required coefficients $\{\psi_\lambda(f(u)) : \lambda \in \Lambda\}$ can be computed in $\mathcal{O}(\#\Lambda)$ operations from a set of scaling function coefficients $\{\phi_\lambda(f(u)) : \lambda \in \check{\Lambda}\}$ for some $\check{\Lambda}$ with $\operatorname{span}\{\tilde{\phi}_\lambda : \lambda \in \check{\Lambda}\} \supset \operatorname{span}\{\tilde{\psi}_\lambda : \lambda \in \Lambda\}$ and $\#\check{\Lambda} = \mathcal{O}(\#\Lambda)$. Since the representation $u = \sum_{\lambda\in\hat{\Lambda}} c_\lambda\phi_\lambda$ is *locally finite*, these scaling coefficients can be computed at high accuracy with standard quadrature from the locally finite representation $u = \sum_{\lambda\in\hat{\Lambda}} c_\lambda\phi_\lambda$ such that the overall complexity is $\mathcal{O}(\#\Lambda)$.

The restriction to index sets that are trees which is needed for the above scheme to work is rather harmless. Indeed, in [CDDD01] it has been shown that the corresponding approximation classes are only slightly smaller than with fully unconstrained best N-term approximation.

1.2. **The approximation of the nonlinear term in the tensor product case using interpolets.**

The above approach does not apply to tensor product wavelets, the reason being that generally $\sum_{\boldsymbol{\lambda}\in\boldsymbol{\Lambda}} \tilde{\boldsymbol{\psi}}_{\boldsymbol{\lambda}}(u)\boldsymbol{\psi}_{\boldsymbol{\lambda}}$ has no locally finite scaling function representation with a number of terms that is of the order of $\#\boldsymbol{\Lambda}$. Actually, for "full-grid" collections $\boldsymbol{\Lambda} = \{\boldsymbol{\lambda} \in \boldsymbol{\nabla} : \||\boldsymbol{\lambda}|\|_\infty \leq J\}$ such a scaling function representation does exist, but for sparse-grid collections $\boldsymbol{\Lambda} = \{\boldsymbol{\lambda} \in \boldsymbol{\nabla} : \||\boldsymbol{\lambda}|\|_1 \leq J\}$ it does not.

As in the non-tensor product case, we will impose some structural condition on the sets $\boldsymbol{\Lambda}$. Yet in order to retain the advantages of tensor product approximation, most prominently being the n-independent rates, this condition should allow for sparse-grid index sets, as well as for generalizations of that involving local refinements.

To circumvent the problem that a transformation to a locally finite scaling function representation is prohibitive, our approach to compute a biorthogonal projection of $f(u)$ – with u being given as a linear combination of tensor product wavelets – will be based on the use of *interpolets*, or on tensor products of those. Note that the duals of interpolets are finite linear combinations of Dirac functionals. Furthermore, instead of $\sum_{\boldsymbol{\lambda}\in\boldsymbol{\Lambda}} \boldsymbol{\psi}_{\boldsymbol{\lambda}}(f(u))\tilde{\boldsymbol{\psi}}_{\boldsymbol{\lambda}}$, we will evaluate $\sum_{\boldsymbol{\lambda}\in\boldsymbol{\Lambda}} \tilde{\boldsymbol{\psi}}_{\boldsymbol{\lambda}}(f(u))\boldsymbol{\psi}_{\boldsymbol{\lambda}}$, i.e., we will compute a biorthogonal projection onto a span of *primal* tensor product wavelets. In addition, we will assume that f is such that $(f(u))(x)$ depends only on $u(x)$. Note that in this setting there is no quadrature issue.

Considering first the non-tensor product case, the key property of interpolets is that $\sum_{|\lambda'|>|\lambda|}\tilde{\psi}_{\lambda'}(u)\psi_{\lambda'}$, and so $f(u)-f(\sum_{|\lambda'|\leq|\lambda|}\tilde{\psi}_{\lambda'}(u)\psi_{\lambda'})$, vanishes on $\operatorname{supp}\tilde{\psi}_\lambda$. Based on this property, to evaluate $\{\tilde{\psi}_\lambda(f(u)):\lambda\in\Lambda\}$, we will develop a one way, bottom-to-top scheme with a recurrent increment of the coarsest scale by an application of the refinement equation to ensure that the evaluation of each $\tilde{\psi}_\lambda(f(u))$, i.e., of $f(\sum_{|\lambda'|\leq|\lambda|}\tilde{\psi}_{\lambda'}(u)\psi_{\lambda'})$, takes $\mathcal{O}(1)$ operations. Scaling functions in the expansion for u – that arise in the process by the recurrent applications of the refinement equation – will be dropped whenever their supports have empty intersection with $\operatorname{supp}\tilde{\psi}_\lambda$ for all $\lambda\in\Lambda$ for which $\tilde{\psi}_\lambda(f(u))$ still has to be evaluated. Assuming that Λ is a tree, the overall scheme will be shown to take $\mathcal{O}(\#\Lambda)$ operations.

The above bottom-to-top scheme will be the basis of our scheme to evaluate $\{\tilde{\boldsymbol{\psi}}_{\boldsymbol{\lambda}}(f(u)):\boldsymbol{\lambda}\in\boldsymbol{\Lambda}\}$ in the n-fold tensor product case. Assuming that $\boldsymbol{\Lambda}$ is a multiple tree, this scheme will be shown to take $\mathcal{O}(\#\boldsymbol{\Lambda})$ operations. Here, with $\boldsymbol{\Lambda}$ being a multiple tree, here we mean that for any $\mathbf{m}\in\boldsymbol{\Lambda}$ and $1\leq i\leq n$, $\{k:(m_1,\ldots,m_{i-1},k,m_{i+1},\ldots,m_n)\in\boldsymbol{\Lambda}\}$ is a tree.

The use of interpolets to evaluate nonlinear terms was already advocated in [GK00], mainly because of the absence of quadrature errors. The scheme used in [GK00] in the non-tensor product case is essentially equal to that from [DSX00], which we discussed before. The suggestion that was made that this scheme can be the basis of a scheme in the n-fold tensor product case is however unclear to us, at least when one aims at linear computational complexity.

1.3. **Extension to tensor products of "true" biorthogonal wavelets.**

In order to compute in the tensor product wavelet setting a biorthogonal projection of $f(u)$ in linear complexity, we made some compromises. First of all, we used interpolets which do not form stable bases in both L_2 and, in more than one dimension, in H^1. Secondly, we interchanged the role of primal and dual side. For certain types of "true" wavelets, this can be remedied as we discuss next.

To distinguish them from "true" biorthogonal wavelets, let us here denote with $\psi_\lambda^{(\mathrm{I})}$ and $\tilde{\psi}_\lambda^{(\mathrm{I})}$ an interpolet and its dual, and let $\boldsymbol{\psi}_{\boldsymbol{\lambda}}^{(\mathrm{I})}=\otimes_{i=1}^n\psi_{\lambda_i}^{(\mathrm{I})}$ and similarly $\tilde{\boldsymbol{\psi}}_{\boldsymbol{\lambda}}^{(\mathrm{I})}$. Let $V_0^{(\mathrm{I})}\subset V_1^{(\mathrm{I})}\subset\cdots$ denote the multiresolution analysis corresponding to the interpolets, and let $V_0\subset V_1\subset\cdots$ and $\tilde{V}_0\subset\tilde{V}_1\subset\cdots$ denote the primal and dual biorthogonal multiresolution analyses corresponding some "true" locally supported biorthogonal primal and dual wavelet systems Ψ and $\tilde{\Psi}$.

Let us consider the situation that the interpolets and biorthogonal wavelets are such that for some constants L and $\tilde{L}$, it holds that

$$V_j\subset V_{j+L}^{(\mathrm{I})},\quad V_j^{(\mathrm{I})}\subset\tilde{V}_{j+\tilde{L}}\quad(j\in\mathbb{N}_0).\tag{1.1}$$

Then for $u=\sum_{\boldsymbol{\lambda}\in\hat{\boldsymbol{\Lambda}}}\tilde{\boldsymbol{\psi}}_{\boldsymbol{\lambda}}(u)\boldsymbol{\psi}_{\boldsymbol{\lambda}}$, with $\hat{\boldsymbol{\Lambda}}$ being a multiple tree and $\boldsymbol{\psi}_{\boldsymbol{\lambda}}=\otimes_{i=1}^n\psi_{\lambda_i}$ and similarly $\tilde{\boldsymbol{\psi}}_{\boldsymbol{\lambda}}$, $f(u)$ can be approximated in the dual basis by the following 3 steps:

- Re-express u in terms of the interpolets, i.e., as $u=\sum_{\boldsymbol{\lambda}\in\bar{\boldsymbol{\Lambda}}}c_{\boldsymbol{\lambda}}\boldsymbol{\psi}_{\boldsymbol{\lambda}}^{(\mathrm{I})}$. As we will show, such a representation exists with $\bar{\boldsymbol{\Lambda}}$ being a multiple tree with $\#\bar{\boldsymbol{\Lambda}}=\mathcal{O}(\#\hat{\boldsymbol{\Lambda}})$, and it can be computed in linear complexity.
- Approximate $f(u)$ by $\sum_{\boldsymbol{\lambda}\in\boldsymbol{\Lambda}}\tilde{\boldsymbol{\psi}}_{\boldsymbol{\lambda}}^{(\mathrm{I})}(f(u))\boldsymbol{\psi}_{\boldsymbol{\lambda}}^{(\mathrm{I})}$ where $\boldsymbol{\Lambda}\supset\bar{\boldsymbol{\Lambda}}$ is a multiple tree that is sufficiently large to meet a prescribed tolerance. The fact that the

computation of this approximation requires only $\mathcal{O}(\#\mathbf{\Lambda})$ operations is the main topic of this work, and was discussed before.

- Re-express the obtained approximation for $f(u)$ in terms of the dual wavelets, i.e., in the form $\sum_{\boldsymbol{\lambda}\in\hat{\mathbf{\Lambda}}} d_{\boldsymbol{\lambda}}\tilde{\psi}_{\boldsymbol{\lambda}}$. Analogous arguments that are used in the first step show that such a representation exists with $\hat{\mathbf{\Lambda}}$ being a multiple tree with $\#\hat{\mathbf{\Lambda}} = \mathcal{O}(\#\mathbf{\Lambda})$, and that it can be computed in linear complexity.

Note that the first and last steps are exact and so do not introduce additional errors.

As an example of a setting in which (1.1) holds , for $d \in \{2,3,\dots\}$, we consider $V_j^{(\mathrm{I})} = \sum_{k=0}^{2^j-1} P_{d-1}(k2^{-j},(k+1)2^{-j}) \cap C(0,1)$, possibly intersected with $H_0^1(0,1)$. Clearly these spaces can be equipped with interpolating basis functions, and so the corresponding multiresolution analysis with interpolets. As shown in [DGH99], there exists a multiresolution analysis $V_0 \subset V_1 \subset \cdots$ that satisfies (1.1) with $\tilde{V}_j = V_j$, and for which $V_{j+1} \cap V_j^{\perp_{L_2(0,1)}}$ can be equipped with locally supported orthogonal bases.

Apart from their application to semilinear equations discussed here, these piecewise polynomial orthogonal wavelets are very well suited for application in tensor product wavelet algorithms. Firstly, with respect to a range of Sobolev norms, including L_2, thanks to their orthogonality, the condition numbers of the n-fold tensor product basis are bounded uniformly in n. Secondly, thanks to their piecewise smoothness, the representation of differential operators that have smooth coeefficients with respect to the tensor product basis can be very well approximated by sparse matrices.

The condition (1.1) can also be satisfied by biorthogonal, i.e., non-orthogonal wavelets. In a forthcoming work, we will construct such wavelets that have their application in the solution of time-dependent problems.

Finally, we note that the whole setting of this paper can be generalized to the use of *two* systems of interpolets, one for the representation of u and the other for the representation of the approximation for $f(u)$. In this case, condition (1.1) should read as $V_j \subset V_{j+L}^{(\mathrm{I}_1)}$ and $V_j^{(\mathrm{I}_2)} \subset \tilde{V}_{j+\tilde{L}}$, where $(V_j^{(\mathrm{I}_1)})_j$ and $(V_j^{(\mathrm{I}_2)})_j$ are the multiresolution analyses corresponding to both systems of interpolets. In order that the computation of the biorthogonal projection can be performed in linear complexity, the *interpolation points*, however, should be the same for both systems.

For the case that $(f(u))(x)$ does not only depend on $u(x)$ but also on first order partial derivatives of u in x, a promising option is to take the first system of interpolets as being based on *Hermite* interpolation.

1.4. **Organization and notation.**

This paper is organized as follows: In Sect. 2, we introduce interpolets, and define parent-child relations on the index set of these functions. Given a nonlinear functional f, and a u from the span of a set of interpolets, where the index set forms a tree, we give an algorithm to compute the interpolant of $f(u)$ from the span of this set of interpolets in linear complexity. In Sect. 3, we extent this algorithm to tensor products of interpolets, with index sets that forms multiple trees. Finally, in Sect. 4, we show that for certain types of (bi)-orthogonal wavelet bases, basis transformations in linear complexity can be made between (primal) wavelets and interpolets, and interpolets and (dual) wavelets, with which the application of the algorithm for evaluating the nonlinear functional is extended to "true" wavelet systems.

In this paper, with $C \lesssim D$ we will mean that C can be bounded by a multiple of D, independently of parameters on which C and D may depend, possibly with the exception n being the number of factors in a tensor product. Obviously, $C \gtrsim D$ is defined as $D \lesssim C$, and $C \eqsim D$ as $C \lesssim D$ and $C \gtrsim D$.

2. Evaluation of a nonlinear functional in the non-tensor product case

2.1. **Interpolating scaling functions and wavelets.** Let

$$V_0 \subset V_1 \subset \cdots$$

be a nested sequence of finite dimensional spaces of real-valued functions on some domain Ω, where we have in mind a sequence constructed using dyadic refinements. For simplicity, we will assume that Ω is convex, but the results we are going to derive can be extended straightforwardly to non-convex Ω by re-defining the distance between points in Ω as the length of the shortest path in Ω connecting them.

We assume to have bases

$$\Phi_j = \{\phi_{j,k} : k \in I_j\}$$

for V_j available, where the $\phi_{j,k}$ are known as *scaling functions*, such that

$$\operatorname{diam}\operatorname{supp}\phi_{j,k} \lesssim 2^{-j} \quad (\textit{localness}), \tag{2.1}$$

such that any subset of Ω with diameter 2^{-j} has non-empty intersection with the supports of a uniformly bounded number of $\phi_{j,k}$ (the bases are *locally finite*), and such that the scaling functions are *interpolating* with respect to a nested sequence of points in Ω. That is, identifying I_j with a set of points in Ω, we assume that $I_j \subset I_{j+1}$ and

$$\phi_{j,k}(m) = \delta_{k,m} \quad (m \in I_j).$$

Setting

$$J_{j+1} := I_{j+1} \setminus I_j,$$

a further natural assumption is that there exists a constant $c_1 > 0$ such that for $j \geq 1$ and all $x \in \Omega$,

$$B(x; c_1 2^{-j}) \cap J_j \neq \emptyset. \tag{2.2}$$

Because of the nestedness of the V_j and since Φ_{j+1} is interpolating, each $\phi_{j,k}$ can be expressed as

$$\phi_{j,k} = \sum_{m \in I_{j+1}} \phi_{j,k}(m)\phi_{j+1,m} = \phi_{j+1,k} + \sum_{m \in J_{j+1}} \phi_{j,k}(m)\phi_{j+1,m}, \tag{2.3}$$

where the second equality is a consequence of $I_j \subset I_{j+1}$ and the fact that Φ_j is interpolating. The localness and locally finiteness of the scaling functions shows that the numbers of non-zero coefficients $\phi_{j,k}(m)$ in these expansions are bounded, uniformly in j and k.

From (2.3), one infers that

$$\Phi_j \cup \{\phi_{j+1,m} : m \in J_{j+1}\}$$

is a basis for V_{j+1}. The functions $\phi_{j+1,m}$ for $m \in J_{j+1}$ are also known as the *interpolatory wavelets* or *interpolets* on level $j+1$, and we will denote them as $\psi_{j+1,m}$. A repetition of the argument shows that for any $\mathbb{N}_0 \ni \ell \leq j+1$,

$$\Phi_\ell \cup \{\psi_{\ell+1,m} : m \in J_\ell\} \cup \cdots \cup \{\psi_{j+1,m} : m \in J_{j+1}\}$$

is a basis for V_{j+1}. Throughout this work, the integer ℓ will indicate the (current) "coarsest level". For convenience, we set $\psi_{0,m} := \phi_{0,m}$ for $m \in J_0 := I_0$.

Proposition 2.1. *With $\delta_m \in C(\Omega)'$ being defined by $\delta_m(f) = f(m)$, and $I_{-1} := \emptyset$, the collections*

$$\bigcup_{j=0}^{\infty} \{\psi_{j,m} : m \in J_j\}$$

and

$$\bigcup_{j=0}^{\infty} \Big\{\tilde{\psi}_{j,m} := \delta_m - \sum_{k \in I_{j-1}} \phi_{j-1,k}(m)\delta_k : m \in J_j\Big\}$$

are biorthogonal.

Proof. For $j < j'$ and $m \in J_j$, $m' \in J_{j'}$, it holds that $\tilde{\psi}_{j,m}(\psi_{j',m'}) = 0$.

For $m, m' \in J_j$, we have $\tilde{\psi}_{j,m}(\psi_{j,m'}) = \delta_{m,m'}$.

For $m \in J_j$ and $m' \in I_{j-1}$, using the refinement equation (2.3) we obtain that

$$\begin{aligned}\tilde{\psi}_{j,m}(\phi_{j-1,m'}) &= (\delta_m - \sum_{k \in I_{j-1}} \phi_{j-1,k}(m)\delta_k)(\phi_{j,m'} + \sum_{q \in J_j} \phi_{j-1,m'}(q)\phi_{j,q}) \\ &= \phi_{j-1,m'}(m) - \phi_{j-1,m'}(m) = 0,\end{aligned}$$

and so $\tilde{\psi}_{j,m}(\psi_{j',m'}) = 0$ for any $j' < j$ and $m' \in J_{j'}$. □

2.2. **Graded trees.** The localness assumption (2.1) implies that there exists a constant $c_2 > 0$ such that

$$\operatorname{supp} \tilde{\psi}_{j+1,m} \subset B(m, c_2 2^{-j}). \tag{2.4}$$

Definition 2.2. Let $c_3 \geq 0$ be some constant such that $c_1 \leq c_2 + c_3$, where c_1 is from (2.2), and let $\ell \in \mathbb{N}_0$. Then for $j \geq \ell$ and $m \in J_{j+1}$, all points k in $B(m;(c_2+c_3)2^{-j}) \cap \begin{Bmatrix} I_\ell & \text{when } j = \ell \\ J_j & \text{when } j > \ell \end{Bmatrix}$ are called *ℓ-parents* of m, and m is called an *ℓ-child* of any of these parents k. This will be denoted as $k \prec_\ell m$.

A point k is called an *ℓ-ancestor* of m, and m an *ℓ-descendant* of k, denoted as $k \prec\!\!\prec_\ell m$, when m is an ℓ-child of either k or of an ℓ-descendant of k.

A set $\Lambda \subset I := \cup_{j \in \mathbb{N}_0} I_j$ is called a *graded ℓ-tree* if whenever for some $j \geq \ell$, $m \in \Lambda \cap J_{j+1}$, then so are all its ℓ-parents.

Note that the actual grading of an ℓ-graded tree is determined by the value of the constant $c_2 + c_3$.

Remark 2.3. In above definition, generally a parent has more than one child and a child has more than one parent. Actually, the condition $c_2 + c_3 \geq c_1$ guarantees that for $j \geq \ell$ any $m \in J_{j+1}$ has an ℓ-parent. In an *implementation* of the algorithm that we are going to present it might be more convenient to use a datastructure where each node in a tree has at most one parent. The graded ℓ-tree condition on a collection Λ can then be enforced by requiring that if $m \in J_{j+1}$ is in Λ, then so is its parent as well as all elements in I_ℓ or J_j in a sufficiently large neighbourhood this parent.

Since the sets $J_0, J_1, J_2, \ldots$ are disjoint, for any $m \in I$ there exists a unique $j = j(m)$ such that $m \in J_j$. We set

$$\tilde{\psi}_m := \tilde{\psi}_{j(m),m}.$$

At the primal side, besides $\Phi_0 + \bigcup_{j=1}^{\infty}\{\psi_{j,m} : m \in J_j\}$, for varying $\ell \in \mathbb{N}_0$ we also will consider the collections $\Phi_\ell + \bigcup_{j=\ell+1}^{\infty}\{\psi_{j,m} : m \in J_j\}$ which have the same spans. For $\ell \in \mathbb{N}_0$ and $m \in I$, we set

$$\psi_m^{(\ell)} := \begin{cases} \psi_{j(m),m} & \text{when } j(m) > \ell, \\ \phi_{\ell,m} & \text{otherwise.} \end{cases}$$

With this definition, we have

$$\Phi_\ell + \bigcup_{j=\ell+1}^{\infty} \{\psi_{j,m} : m \in J_j\} = \{\psi_m^{(\ell)} : m \in I\}.$$

Remark 2.4. The most important concepts in Definition 2.2 are that of graded ℓ-trees, ℓ-children, parents or descendants for $\ell = 0$. For $\ell > 0$, these concepts become relevant only after the coarsest scale has been changed from 0 into ℓ by a repeated application of the refinement equation (2.3).

Obviously, any I_j is a graded 0-tree. For constructing efficient approximations of (locally) non-smooth functions, it is relevant to consider the spans of collections $\{\phi_m^{(0)} : m \in \Lambda\}$ for general $\Lambda \subset I$, i.e., not necessarily equal to some I_j. It has been shown, cf. [CDDD01], that the class of functions that can be approximated with a certain rate from the spans of a sequence of such collections becomes only slightly smaller when instead fully general sets only index sets are considered that are graded 0-trees.

Proposition 2.5. *Let $\Lambda \subset I$ be a graded 0-tree. Then for all $m \in \Lambda$,* $\operatorname{supp} \tilde{\psi}_m \subset \Lambda$.

Proof. In view of $c_3 \geq 0$ and the definition of c_2 in (2.4), it suffices to show that for $j \in \mathbb{N}_0$ and $m \in \Lambda \cap J_{j+1}$, $B(m; (c_2 + c_3)2^{-j}) \cap I_j \subset \Lambda$ which we will do using induction.

For $j = 0$, this property follows from the definition of a graded 0-tree. Suppose the property is valid for some $j - 1 \in \mathbb{N}_0$. Let $m \in \Lambda \cap J_{j+1}$ and $p \in B(m; (c_2 + c_3)2^{-j}) \cap I_j$. If $p \in J_j$, then $p \in \Lambda$ follows from the definition of a graded 0-tree. If $p \in I_{j-1}$, then by (2.2) there exists a $q \in J_j$ with $|q - m| \leq c_1 2^{-j} \leq (c_2 + c_3)2^{-j}$, so that $q \in \Lambda$ by definition of a graded 0-tree. Since $|p - q| \leq (c_2 + c_3)2^{-(j-1)}$, we conclude that $p \in \Lambda$ by the induction hypothesis. □

Remark 2.6. The above proof shows that a graded 0-tree is a graded ℓ-tree for any $\ell \in \mathbb{N}_0$. Similarly, a graded ℓ-tree is a graded ℓ'-tree for any $\mathbb{N}_0 \ni \ell' \geq \ell$.

Proposition 2.7. *Let $\Lambda \subset I$ be a graded 0-tree and $g \in C(\Omega)$. Then the biorthogonal projection $\mathcal{I}_\Lambda g := \sum_{k\in\Lambda} \tilde{\psi}_k(g)\psi_k^{(0)}$ is the unique function from* $\operatorname{span}\{\psi_k^{(0)} : k \in \Lambda\}$ *that is equal to g in all points of Λ, i.e., $\mathcal{I}_\Lambda g$ is the* interpolant.

Proof. For $m \in \Lambda \cap I_0$, $g(m) = (\mathcal{I}_\Lambda g)(m)$ follows from $\tilde{\psi}_k(g) = g(k)$ when $k \in I_0$ and the fact that the primal functions are interpolating with respect to nested sequences of points. Suppose that $g(m) = (\mathcal{I}_\Lambda g)(m)$ for all $m \in \Lambda \cap I_{j-1}$. Then for $m \in \Lambda \cap J_j$, from Proposition 2.5, Proposition 2.1, the induction hypothesis, and again Proposition 2.5, we have

$$\begin{aligned}\delta_m\Big(\sum_{k\in\Lambda}\tilde\psi_k(g)\phi_k^{(0)}\Big) &= \tilde\psi_m\Big(\sum_{k\in\Lambda}\tilde\psi_k(g)\psi_k^{(0)}\Big) + \sum_{p\in I_{j-1}\cap\Lambda}\phi_{j-1,p}(m)\delta_p\Big(\sum_{k\in\Lambda}\tilde\psi_k(g)\phi_k^{(0)}\Big)\\ &= \tilde\psi_m(g) + \sum_{p\in I_{j-1}\cap\Lambda}\phi_{j-1,p}(m)\delta_p\Big(\sum_{k\in\Lambda}\tilde\psi_k(g)\phi_k^{(0)}\Big)\\ &= \tilde\psi_m(g) + \sum_{p\in I_{j-1}\cap\Lambda}\phi_{j-1,p}(m)\delta_p(g) = \delta_m(g).\end{aligned}$$

The uniqueness of $\mathcal{I}_\Lambda g$ follows easily from the basis functions being interpolating. □

Proposition 2.8. *Let the constant c_3 in Definition 2.2 be such that*

$$\operatorname{supp}\phi_{j,k'} \subset B(k';c_3 2^{-j}) \quad (k'\in I_j) \quad \textit{and} \quad c_1 \le c_2 + \tfrac{1}{2}c_3. \tag{2.5}$$

Then for $k \neq m \in I$, $\operatorname{supp}\psi_k^{(\ell)} \cap \operatorname{supp}\tilde\psi_m = \emptyset$ whenever $k \not\prec\!\!\prec_\ell m$.

Proof. For $j > j'$ and $k \in I_j$, $m \in I_{j'}$, or $j = j'$ and $k \neq m \in I_j$, $\operatorname{supp}\phi_{j,k} \cap \operatorname{supp}\tilde\psi_{j',m} = \emptyset$. So in view of (2.4) and our first assumption on c_3, it suffices to prove that for $k, m \in I$ with $j' := j(m) > \max(j(k),\ell) := j$ and $k \not\prec\!\!\prec_\ell m$, $|k-m| > c_3 2^{-j} + c_2 2^{-(j'-1)}$.

By definition of an ℓ-child, this statement is valid when $j' = j+1$. Assuming that it is valid when $j' = j+n$, we consider the case that $j' = j+n+1$. Because of (2.2), there exists a $p \in J_{j'-1}$ in a ball with radius $c_1 2^{-(j'-1)}$ around the point on the line connecting m and k on distance $\frac{1}{2}c_3 2^{-(j'-1)}$ to m, see Figure 1. By the triangle inequality and $\frac{1}{2}c_3 + c_1 \le c_2 + c_3$ by the second assumption on c_3, we have $|m-p| \le (c_2+c_3)2^{-(j'-1)}$ and so $p \prec_\ell m$.

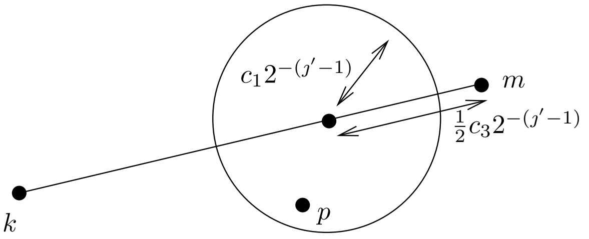

Figure 1. Illustration with the proof of Proposition 2.8.

Necessarily $k \not\prec\!\!\prec_\ell p$, and so $|k-p| > c_3 2^{-j} + c_2 2^{-(j'-2)}$ by the induction hypothesis. From $|k-p| \le (|k-m| - \frac{1}{2}c_3 2^{-(j'-1)}) + c_1 2^{-(j'-1)}$ and $\frac{1}{2}c_3 - c_1 \ge -c_2$, it follows that $|k-m| > c_3 2^{-j} + c_2 2^{-(j'-1)}$. □

In the following, silently we will always assume that the constant c_3 satisfies the conditions formulated in (2.5).

2.3. **The evaluation algorithm.**

Let $f : C(\Omega) \to C(\Omega)$ be a function of type

$$(f(u))(x) = \bar{f}(u(x)),$$

where $\bar{f} \in C(\mathbb{R})$, i.e., the value of $f(u)$ in x depends only on that of u in x. Given a graded 0-tree $\Lambda \subset I$ and $u = \sum_{k\in\Lambda} c_k^{(0)} \psi_k^{(0)}$, our goal is to compute $\mathcal{I}_\Lambda(f(u))$ in $\mathcal{O}(\#\Lambda)$ operations, where we assume that each $\bar{f}$ evaluation takes $\mathcal{O}(1)$ operations. In view of Proposition 2.7, computing $\mathcal{I}_\Lambda(f(u))$ amounts to computing $\{\tilde{\psi}_m(f(u)) : m \in \Lambda\}$.

The advantage of using interpolets instead of general biorthogonal wavelets is that in view of Proposition 2.8 only for $k = m$ or $k \prec\!\!\prec_0 m$ a term $c_k^{(0)}\psi_k^{(0)}$ might contribute to $\tilde{\psi}_m(f(u))$. Despite of this, since for $m \in \Lambda \cap J_\ell$ the number of such terms is of order ℓ, a direct evaluation of all $\tilde{\psi}_m(f(u))$ cannot be performed in linear complexity.

The idea behind the algorithm presented below is that for $m \in \Lambda \cap J_0$, $\tilde{\psi}_m(f(u))$ can be evaluated in $\mathcal{O}(1)$ operations, and that for $\ell = 1, 2, \ldots$, after u is re-expressed in terms of $\{\psi_k^{(\ell)} : k \in I\}$, the same holds true for $m \in \Lambda \cap J_\ell$. These recurrent re-expressions of u will be performed by applying the refinement equation (2.3). To ensure that the total cost of all applications of (2.3) is $\mathcal{O}(\#\Lambda)$, it is needed that at the ℓth stage of this process, the number of terms in the expansion with respect to $\{\psi_k^{(\ell)} : k \in I\}$ corresponding to $k \in I_\ell$ is $\mathcal{O}(\#(\Lambda \cap I_\ell))$. This will be realized by dropping all terms corresponding to $k \in I_\ell$ for which $\psi_k^{(\ell)}$ has no ℓ-children in Λ. In view of Proposition 2.8, this is allowed since it will not change u on the supports of $\tilde{\psi}_m$ for $m \in \Lambda \setminus I_\ell$.

Algorithm 2.9.

```
eval($\Lambda, (c_k^{(0)})_{k\in\Lambda}$)
% $\Lambda$ has to be a graded 0-tree.
$\Lambda^{(0)} := \Lambda,\ u := \sum_{k\in\Lambda} c_k^{(0)}\psi_k^{(0)}$.
for $\ell = 0, 1, \ldots$ do
   % the current $u$ is of the form $\sum_{k\in\Lambda^{(\ell)}} c_k^{(\ell)}\psi_k^{(\ell)}$ with $\Lambda \cap J_\ell \subset \Lambda^{(\ell)} \cap J_\ell$.
   forall $m \in \Lambda \cap J_\ell$ do compute $\delta_m(f(u)) = \bar{f}(u(m)) = \bar{f}(c_m^{(\ell)})$ and
      $\tilde{\psi}_m(f(u)) = \delta_m(f(u)) - \sum_{p\in I_{\ell-1}} \phi_{\ell-1,p}(m)\delta_p(f(u))$
      % all $\delta_p(f(u))$ with $p \in I_{\ell-1}$ that are used here have been computed previously.
   enddo
   forall $k \in \Lambda^{(\ell)} \cap I_\ell$ that have no $\ell$-child in $\Lambda$ do
      remove $k$ from $\Lambda^{(\ell)}$, and with that $c_k^{(\ell)}\psi_k^{(\ell)}$ from the expansion for $u$
   enddo
   if $\Lambda^{(\ell)} = \emptyset$ then goto return-statement endif
   forall $m \in (\Lambda^{(\ell)} \cap I_\ell)\backslash(\Lambda \cap J_\ell)$ do compute $\delta_m(f(u)) = \bar{f}(u(m)) = \bar{f}(c_m^{(\ell)})$
   enddo
   by applying (2.3), write the current $u$ in the form $\sum_{k\in\Lambda^{(\ell+1)}} c_k^{(\ell+1)}\psi_k^{(\ell+1)}$.
enddo
return $\{\tilde{\psi}_m(f(u)) : m \in \Lambda\}$
```

Theorem 2.10. *Algorithm 2.9 produces* $\{\tilde{\psi}_m(f(u)) : m \in \Lambda\}$ *in* $\mathcal{O}(\#\Lambda)$ *operations.*

Proof. Let $\ell > 0$ be given. Assume that in the previous iterations all elements of $\{\tilde{\psi}_m(f(u)) : m \in \Lambda \cap I_{\ell-1}\}$ have been computed, as well as all those $\delta_p(f(u))$ with

$p \in I_{\ell-1}$ that are needed for the computation of $\{\tilde{\psi}_m(f(u)) : m \in \Lambda \cap J_\ell\}$, and that the current u, being of the form $\sum_{k \in \Lambda^{(\ell)}} c_k^{(\ell)} \psi_k^{(\ell)}$, is equal to the original u on $\operatorname{supp} \tilde{\psi}_m$ for $m \in \Lambda \setminus I_{\ell-1}$. The arguments that will be given below show that these three assumptions, in particular the last one, are valid for $\ell = 1$.

Then what is left to compute is $\{\delta_m(f(u)) : m \in \Lambda \cap J_\ell\}$ – which together with previously computed $\delta_p(f(u))$ with $p \in I_{\ell-1}$ yields $\{\tilde{\psi}_m(f(u)) : m \in \Lambda \cap J_\ell\}$ –, as well as $\{\tilde{\psi}_m(f(u)) : m \in \Lambda \backslash I_\ell\}$.

In the current iteration, after performing the first task, it is allowed to drop all terms $c_k^{(\ell)} \psi_k^{(\ell)}$ for those $k \in \Lambda^{(\ell)} \cap I_\ell$ that have no ℓ-childs in Λ. Indeed, since Λ is a graded 0-tree, such k have no ℓ-descendants in Λ, and so $\psi_k^{(\ell)}$ vanishes on $\operatorname{supp} \tilde{\psi}_m$ for any $m \in \Lambda \backslash I_\ell$ by Proposition 2.8.

Since, besides those that were already computed earlier in this iteration, in the remainder of this iteration the $\delta_p(f(u))$ for all remaining $p \in \Lambda^{(\ell)} \cap I_\ell$ are computed, and the current u is expressed in the form $\sum_{k \in \Lambda^{(\ell+1)}} c_k^{(\ell+1)} \psi_k^{(\ell+1)}$, an induction argument shows that the algorithm produces $\{\tilde{\psi}_m(f(u)) : m \in \Lambda\}$.

The statement about the cost is a consequence of the fact that by the dropping of indices from $\Lambda^{(\ell)}$ before the application of (2.3), $\#(\Lambda^{(\ell+1)} \cap I_{\ell+1}) \lesssim \#(\Lambda \cap J_{\ell+1})$. □

3. Evaluation of a nonlinear functional in the tensor product case

3.1. Tensor product bases and multiple trees.

Let $n \in \mathbb{N}$. For $\mathbf{m} \in \mathbf{I} := I^n$ and $\boldsymbol{\ell} \in \mathbb{N}_0^n$, we set

$$\tilde{\boldsymbol{\psi}}_{\mathbf{m}} := \tilde{\psi}_{m_1} \otimes \cdots \otimes \tilde{\psi}_{m_n}, \quad \boldsymbol{\psi}_{\mathbf{m}}^{(\boldsymbol{\ell})} := \psi_{m_1}^{(\ell_1)} \otimes \cdots \otimes \psi_{m_n}^{(\ell_n)}.$$

Remark 3.1. For ease of presentation, we assume that the collections of primal and (thus) dual functions are the same in all n coordinate directions. The general case that the collections in the various coordinate directions are possibly different causes no additional difficulties, apart from a more complicated notation.

Clearly, the collections $\{\boldsymbol{\psi}_{\mathbf{m}}^{(\mathbf{0})} : \mathbf{m} \in \mathbf{I}\}$ ($\mathbf{0} := (0, \ldots, 0)$) and $\{\tilde{\boldsymbol{\psi}}_{\mathbf{m}} : \mathbf{m} \in \mathbf{I}\}$ are biorthogonal.

Our substitute for the concept of graded ℓ-trees in the non-tensor product case is given by the following.

Definition 3.2. For $\boldsymbol{\ell} \in \mathbb{N}_0^n$, we call $\boldsymbol{\Lambda} \subset \mathbf{I}$ a *graded $\boldsymbol{\ell}$-tree* when for all $1 \leq i \leq n$ and all $\mathbf{m} \in \boldsymbol{\Lambda}$, the collection of all $k \in I$ with $(m_1, \ldots, m_{i-1}, k, m_{i+1}, \ldots, m_n) \in \boldsymbol{\Lambda}$ is a graded ℓ_i-tree.

If $\boldsymbol{\Lambda}$ is a graded $\boldsymbol{\ell}$-tree, then $\Lambda_1 := \{m_1 : (m_1, \ldots, m_n) \in \boldsymbol{\Lambda}\}$ is a graded ℓ_1-tree. For $k \in \Lambda_1$, the collection of $\mathbf{p} \in I^{n-1}$ with $(k, \mathbf{p}) \in \boldsymbol{\Lambda}$ is a graded $(\ell_2, \ldots, \ell_n)$-tree, that will be denoted as $\boldsymbol{\Lambda}(k)$. Obviously, $\boldsymbol{\Lambda} = \cup_{k \in \Lambda_1} \{k\} \times \boldsymbol{\Lambda}(k)$.

Remark 3.3. Conversely, if Λ is a graded ℓ_1-tree, and for each $k \in \Lambda$, $\bar{\boldsymbol{\Lambda}}(k)$ is some graded $\bar{\boldsymbol{\ell}} := (\ell_2, \ldots, \ell_n)$-tree, then $\cup_{k \in \Lambda} \{k\} \times \bar{\boldsymbol{\Lambda}}(k)$ is not necessarily a graded $\boldsymbol{\ell}$-tree. Indeed, if $k \prec_{\ell_1} \check{k} \in \Lambda$, and $\mathbf{p} \in \bar{\boldsymbol{\Lambda}}(\check{k})$, then not necessarily $\mathbf{p} \in \bar{\boldsymbol{\Lambda}}(k)$. So, in other words, a graded ℓ_1-tree of graded $\bar{\boldsymbol{\ell}}$-trees is not necessarily a graded $\boldsymbol{\ell}$-tree.

As we will see, by a combination of Proposition 3.8 and Corollary 3.10, if $\boldsymbol{\Lambda}$ is a graded $\boldsymbol{\ell}$-tree, and $\mathbf{k} = (k_1, \bar{\mathbf{k}}) \in I_{\ell_1} \times I^{n-1}$ such that k_1 has no ℓ_1-child p with $(p, \bar{\mathbf{k}}) \in \boldsymbol{\Lambda}$, then $\operatorname{supp} \boldsymbol{\psi}_{\mathbf{k}}^{(\boldsymbol{\ell})} \cap \operatorname{supp} \tilde{\boldsymbol{\psi}}_{\mathbf{m}} = \emptyset$ for all $\mathbf{m} \in \boldsymbol{\Lambda}$ with $m_1 \in I \backslash I_{\ell_1}$. This property will turn out to be essential, and cannot be guaranteed when $\boldsymbol{\Lambda}$ is only a graded ℓ_1-tree of graded $\bar{\boldsymbol{\ell}}$-trees. The application of this property will be in the evaluation of $\tilde{\boldsymbol{\psi}}_{\mathbf{m}}(f(u))$ for an f such that $f(u)(x)$ depends only on $u(x)$. If $\operatorname{supp} \boldsymbol{\psi}_{\mathbf{k}}^{(\boldsymbol{\ell})} \cap \operatorname{supp} \tilde{\boldsymbol{\psi}}_{\mathbf{m}} = \emptyset$ for all $\mathbf{m}$ for which $\tilde{\boldsymbol{\psi}}_{\mathbf{m}}(f(u))$ still has to be evaluated, then obviously any multiple of $\boldsymbol{\psi}_{\mathbf{k}}^{(\boldsymbol{\ell})}$ can be removed from an expansion for u. The key is that the verification whether $(k_1, \bar{\mathbf{k}}) \in I_{\ell_1} \times I^{n-1}$ is such that k_1 has no ℓ_1-child p with $(p, \bar{\mathbf{k}}) \in \boldsymbol{\Lambda}$ requires only local information from the graded $\boldsymbol{\ell}$-tree $\boldsymbol{\Lambda}$ that, in all but the first coordinate, is “frozen” at $\bar{\mathbf{k}}$.

Remark 3.4. The question which class of functions can be approximated with a certain rate from the spans of $\{\boldsymbol{\psi}_{\mathbf{m}}^{(\mathbf{0})}\}$ with $\mathbf{m}$ running over some graded $\mathbf{0}$-trees is outside the scope of this paper. Yet to indicate that this is a relevant class we note the following: For $1 \leq i \leq n$, let $\Lambda_0^{(i)} \subset \Lambda_1^{(i)} \subset \cdots \subset I$ be a nested sequence of graded 0-trees. Then with $\Delta_j^{(i)} := \Lambda_j^{(i)} \setminus \Lambda_{j-1}^{(i)}$ ($\Lambda_{-1}^{(i)} := \emptyset$), and $\mathbf{S} \subset \mathbb{N}_0^n$ such that if $\mathbf{k} \in \mathbf{S}$ and $k_i > 0$, then $\mathbf{k} - \mathbf{e}_i \in \mathbf{S}$, the set $\cup_{\mathbf{k} \in \mathbf{S}} \Delta_{k_1}^{(1)} \times \cdots \times \Delta_{k_1}^{(n)}$ is a graded $\mathbf{0}$-tree. Sparse grid index sets are of this type, as well as certain generalizations that involve local refinements, cf. [DS09].

Remark 3.5. In the discussion in the introduction, the coarsest scale $\boldsymbol{\ell}$ was always $\mathbf{0}$, and the issue of gradedness was ignored. Apart from the latter, what was called a multiple tree there corresponds to a graded $\mathbf{0}$-tree here.

Proposition 3.6. *Let $\boldsymbol{\Lambda} \subset \mathbf{I}$ be a graded $\mathbf{0}$-tree. Then for all $\mathbf{m} \in \boldsymbol{\Lambda}$, $\operatorname{supp} \tilde{\boldsymbol{\psi}}_{\mathbf{m}} \subset \boldsymbol{\Lambda}$.*

Proof. For $n = 1$, the statement is that from Proposition 2.5. Suppose that the statement is valid for $n - 1 \geq 1$. Let $\mathbf{m} \in \boldsymbol{\Lambda}$. Proposition 2.5 gives that for all $k \in \operatorname{supp} \tilde{\psi}_{m_1}$, $(k, m_2, \ldots, m_n) \in \boldsymbol{\Lambda}$. From the induction hypothesis we conclude that for each of these k, $\{k\} \times \operatorname{supp} \tilde{\psi}_{m_2} \times \cdots \times \operatorname{supp} \tilde{\psi}_{m_n} \subset \boldsymbol{\Lambda}$ or $\operatorname{supp} \tilde{\boldsymbol{\psi}}_{\mathbf{m}} \subset \boldsymbol{\Lambda}$. □

Proposition 3.7. *Let $\boldsymbol{\Lambda} \subset \mathbf{I}$ be a graded $\mathbf{0}$-tree and $g \in C(\Omega^n)$. Then the biorthogonal projection $\boldsymbol{\mathcal{I}}_{\boldsymbol{\Lambda}} g := \sum_{\mathbf{k} \in \boldsymbol{\Lambda}} \tilde{\boldsymbol{\psi}}_{\mathbf{k}}(g) \boldsymbol{\psi}_{\mathbf{k}}^{(\mathbf{0})}$ is the unique function from* $\operatorname{span}\{\boldsymbol{\psi}_{\mathbf{k}}^{(\mathbf{0})} : \mathbf{k} \in \boldsymbol{\Lambda}\}$ *that is equal to g in all points of $\boldsymbol{\Lambda}$, known as the* interpolant.

Proof. For $n = 1$, the statement is that from Proposition 2.7. As a first induction hypothesis, let us assume that the statement is valid for $n - 1 \geq 1$. Let $\mathbf{m} \in \Lambda \subset \mathbf{I} = I^n$. If $(j(m_1), \ldots, j(m_n)) = \mathbf{0}$, then $g(\mathbf{m}) = (\boldsymbol{\mathcal{I}}_{\boldsymbol{\Lambda}} g)(\mathbf{m})$. As a second induction hypothesis, let us assume that for some $1 \leq i \leq n$ and all $\mathbf{m}' \in \boldsymbol{\Lambda}$ with $j(m_k') = j(m_k)$ when $k \neq i$ and $j(m_i') = j(m_i) - 1 \geq 0$ it holds that $g(\mathbf{m}') = (\boldsymbol{\mathcal{I}}_{\boldsymbol{\Lambda}} g)(\mathbf{m}')$. We show that this implies that $g(\mathbf{m}) = (\boldsymbol{\mathcal{I}}_{\boldsymbol{\Lambda}} g)(\mathbf{m})$. Without loss of generality we can take $i = 1$ and $n = 2$.

We write

$$\delta_{m_1} \otimes \delta_{m_2} = \tilde{\psi}_{m_1} \otimes \delta_{m_2} + \sum_{p \in I_{j(m_1)-1}} \phi_{j(m_1)-1,p}(m_1)\, \delta_p \otimes \delta_{m_2}.$$

Since $\boldsymbol{\Lambda}$ is a graded $\mathbf{0}$-tree, Proposition 2.5 shows that for all $p \in I_{j(m_1)-1}$ for which $\phi_{j(m_1)-1,p}(m_1) \neq 0$, we have $(p, m_2) \in \boldsymbol{\Lambda}$, and so the second induction hypothesis shows that for these p, $(\delta_p \otimes \delta_{m_2})(\boldsymbol{\mathcal{I}}_{\boldsymbol{\Lambda}} g) = (\delta_p \otimes \delta_{m_2})(g)$.

With $\bar{g}(y) := \tilde{\psi}_{m_1}(x \mapsto g(x,y))$, Proposition 2.1 and the first induction hypothesis show that

$$\begin{aligned}(\tilde{\psi}_{m_1} \otimes \delta_{m_2})(\boldsymbol{\mathcal{I}}_{\mathbf{\Lambda}} g) &= (\tilde{\psi}_{m_1} \otimes \delta_{m_2})(\sum_{\mathbf{k}\in\mathbf{\Lambda}} (\tilde{\psi}_{k_1} \otimes \tilde{\psi}_{k_2})(g)\, \psi^{(0)}_{k_1} \otimes \psi^{(0)}_{k_2}) \\ &= \sum_{k_2 \in \mathbf{\Lambda}(m_1)} (\tilde{\psi}_{m_1} \otimes \tilde{\psi}_{k_2})(g) \delta_{m_2}(\psi^{(0)}_{k_2}) = \delta_{m_2}(\sum_{k_2 \in \mathbf{\Lambda}(m_1)} \tilde{\psi}_{k_2}(\bar{g}) \psi^{(0)}_{k_2}) \\ &= \delta_{m_2}(\bar{g}) = (\tilde{\psi}_{m_1} \otimes \delta_{m_2})(g).\end{aligned}$$

By a combination of both results, we conclude $g(\mathbf{m}) = (\boldsymbol{\mathcal{I}}_{\mathbf{\Lambda}} g)(\mathbf{m})$.

The uniqueness follows from the basis functions being interpolating. □

Proposition 3.8. *Let $\mathbf{\Lambda} \subset \mathbf{I}$ be a graded $\boldsymbol{\ell}$-tree. Let $\bar{\mathbf{\Lambda}}$ be constructed from $\mathbf{\Lambda}$ by removing all $\mathbf{k} = (k_1, \bar{\mathbf{k}}) \in \mathbf{\Lambda}$ with $k_1 \in I_{\ell_1}$ such that k_1 has no ℓ_1-child p with $(p, \bar{\mathbf{k}}) \in \mathbf{\Lambda}$. Then also $\bar{\mathbf{\Lambda}}$ is a graded $\boldsymbol{\ell}$-tree.*

Proof. Suppose the statement is false. Then there exists a $\mathbf{k}$ that is removed and an $i \in \{2, \ldots, n\}$ and a $q \in I$ such that $k_i \prec_{\ell_i} q$ and $(k_1, \ldots, q, \ldots, k_n) \in \bar{\mathbf{\Lambda}}$. But that means that there exists a $p \succ_{\ell_1} k_1$ with $(p, \ldots, q, \ldots, k_n) \in \mathbf{\Lambda}$ (otherwise $(k_1, \ldots, q, \ldots, k_n)$ would have been removed). But then also $(p, \ldots, k_i, \ldots, k_n) \in \mathbf{\Lambda}$ but this contradicts the removement of $\mathbf{k}$ from $\mathbf{\Lambda}$. □

A direct consequence of Proposition 2.8 is the following result:

Proposition 3.9. *Let $\boldsymbol{\ell} \in \mathbb{N}_0^n$ and $\mathbf{k}, \mathbf{m} \in \mathbf{I}$. If for some $1 \leq i \leq n$, $k_i \neq m_i$ and $k_i \not\prec\!\!\prec_{\ell_i} m_i$, then* $\operatorname{supp} \boldsymbol{\psi}^{(\boldsymbol{\ell})}_{\mathbf{k}} \cap \operatorname{supp} \tilde{\boldsymbol{\psi}}_{\mathbf{m}} = \emptyset$.

Corollary 3.10. *Let $\mathbf{\Lambda} \subset \mathbf{I}$ be a graded $\boldsymbol{\ell}$-tree. Then for $\mathbf{m} \in \mathbf{\Lambda}$ and $\mathbf{k} \in \mathbf{I} \backslash \mathbf{\Lambda}$,* $\operatorname{supp} \boldsymbol{\psi}^{(\boldsymbol{\ell})}_{\mathbf{k}} \cap \operatorname{supp} \tilde{\boldsymbol{\psi}}_{\mathbf{m}} = \emptyset$.

Proof. If, for some $\mathbf{m} \in \mathbf{\Lambda}$ and $\mathbf{k} \in \mathbf{I}$, $\operatorname{supp} \boldsymbol{\psi}^{(\boldsymbol{\ell})}_{\mathbf{k}} \cap \operatorname{supp} \tilde{\boldsymbol{\psi}}_{\mathbf{m}} \neq \emptyset$, then for all $1 \leq i \leq n$, either $k_i = m_i$ or $k_i \prec\!\!\prec_{\ell_i} m_i$, but then $\mathbf{k} \in \mathbf{\Lambda}$. □

3.2. **The evaluation algorithm.** Let $f : C(\Omega^n) \to C(\Omega^n)$ be given of the form $(f(u))(x) = \bar{f}(u(x))$ where $\bar{f} \in C(\mathbb{R})$, i.e, the value of $f(u)$ in x depends only on that of u in x.

Remark 3.11. Conversely, such an $\bar{f}$ defines an $f : C(\Omega^n) \to C(\Omega^n)$ for any n. In the following, we write "f" for any of these functions.

Given $\mathbf{\Lambda} \subset \mathbf{I}$ and $u = \sum_{\mathbf{k}\in\mathbf{\Lambda}} c^{(0)}_{\mathbf{k}} \boldsymbol{\psi}^{(0)}_{\mathbf{k}}$, our goal is to compute $\boldsymbol{\mathcal{I}}_{\mathbf{\Lambda}}(f(u))$. Assuming that $\mathbf{\Lambda}$ is a graded $\mathbf{0}$-tree, this amounts to computing $\{\tilde{\boldsymbol{\psi}}_{\mathbf{m}}(f(u)) : \mathbf{m} \in \mathbf{\Lambda}\}$. Since the number of $\mathbf{k}$ with $k_i \prec\!\!\prec_0 m_i$ or $k_i = m_i$ $(1 \leq i \leq m)$ is of order $\prod_{i=1}^n j(m_i)$, and for all these $\mathbf{k}$, $c^{(0)}_{\mathbf{k}} \boldsymbol{\psi}^{(0)}_{\mathbf{k}}$ might contribute to $\boldsymbol{\mathcal{I}}_{\mathbf{\Lambda}}(f(u))$, a direct evaluation of $\{\tilde{\boldsymbol{\psi}}_{\mathbf{m}}(f(u)) : \mathbf{m} \in \mathbf{\Lambda}\}$ is prohibitive. Instead we apply the following recursive algorithm.

Algorithm 3.12.

```
tensoreval($n, \mathbf{\Lambda}, (c^{(0)}_{\mathbf{k}})_{\mathbf{k}\in\mathbf{\Lambda}}$)
% $\mathbf{\Lambda}$ has to be a graded $\mathbf{0}$-tree.
if $n = 1$ then eval($\mathbf{\Lambda}, (c^{(0)}_{\mathbf{k}})_{\mathbf{k}\in\mathbf{\Lambda}}$) else
    $\mathbf{\Lambda}^{(0)} := \mathbf{\Lambda}$, $u := \sum_{\mathbf{k}\in\mathbf{\Lambda}^{(0)}} c^{(0)}_{\mathbf{k}} \boldsymbol{\psi}^{(\mathbf{0})}_{\mathbf{k}}$.
    for $\ell = 0, 1, \ldots$ do
    % the current $u$ is of the form $\sum_{\mathbf{k}\in\mathbf{\Lambda}^{(\ell)}} c^{(\ell)}_{\mathbf{k}} \boldsymbol{\psi}^{(\ell,0,\ldots,0)}_{\mathbf{k}}$ with
    % $\mathbf{\Lambda} \cap (J_\ell \times I^{n-1}) \subset \mathbf{\Lambda}^{(\ell)} \cap (J_\ell \times I^{n-1})$.
    forall $m \in \Lambda_1 \cap J_\ell$ do call tensoreval$(n-1, \mathbf{\Lambda}(m), (c^{(\ell)}_{(m,\mathbf{p})})_{\mathbf{p}\in\mathbf{\Lambda}(m)})$,
      % this yields $\{(\delta_m \otimes \tilde{\boldsymbol{\psi}}_{\mathbf{p}})(f(u)) : \mathbf{p} \in \mathbf{\Lambda}(m)\}$
      forall $\mathbf{p} \in \mathbf{\Lambda}(m)$ do compute
        $(\tilde{\psi}_m \otimes \tilde{\boldsymbol{\psi}}_{\mathbf{p}})(f(u)) = (\delta_m \otimes \tilde{\boldsymbol{\psi}}_{\mathbf{p}})(f(u)) - \sum_{q\in I_{\ell-1}} \phi_{\ell-1,q}(m)(\delta_q \otimes \tilde{\boldsymbol{\psi}}_{\mathbf{p}})(f(u))$
        % all $(\delta_q \otimes \tilde{\boldsymbol{\psi}}_{\mathbf{p}})(f(u))$ with $q \in I_{\ell-1}$ that are used here have been computed
        % previously.
      enddo
    enddo
    forall $\mathbf{k} = (k_1, \bar{\mathbf{k}}) \in \mathbf{\Lambda}^{(\ell)}$ with $k_1 \in I_\ell$ and such that $k_1$ has no $\ell$-child $p$
      with $(p, \bar{\mathbf{k}}) \in \mathbf{\Lambda}$ do remove $\mathbf{k}$ from $\mathbf{\Lambda}^{(\ell)}$, and with that the
      term $c^{(\ell)}_{\mathbf{k}} \boldsymbol{\psi}^{(\ell,0,\ldots,0)}_{\mathbf{k}}$ from the expansion for $u$
    enddo
    if $\mathbf{\Lambda}^{(\ell)} = \emptyset$ then goto return-statement endif
    forall $m \in (\Lambda^{(\ell)}_1 \cap I_\ell) \backslash (\Lambda_1 \cap J_\ell)$
      do call tensoreval$(n-1, \mathbf{\Lambda}^{(\ell)}(m), (c^{(\ell)}_{(m,\mathbf{p})})_{\mathbf{p}\in\mathbf{\Lambda}^{(\ell)}(m)})$
      % this yields $\{(\delta_m \otimes \tilde{\boldsymbol{\psi}}_{\mathbf{p}})(f(u)) : \mathbf{p} \in \mathbf{\Lambda}^{(\ell)}(m)\}$
    enddo
    by applying (2.3) in the first coordinate direction, write the current $u$ in the
    form $\sum_{\mathbf{k}\in\mathbf{\Lambda}^{(\ell+1)}} c^{(\ell+1)}_{\mathbf{k}} \boldsymbol{\psi}^{(\ell+1,0,\ldots,0)}_{\mathbf{k}}$.
  enddo
endif
return $\{\tilde{\boldsymbol{\psi}}_{\mathbf{m}}(f(u)) : \boldsymbol{m} \in \mathbf{\Lambda}\}$
```

Theorem 3.13. *Algorithm 3.12 produces $\{\tilde{\boldsymbol{\psi}}_{\mathbf{m}}(f(u)) : \mathbf{m} \in \mathbf{\Lambda}\}$ in $\mathcal{O}(\#\mathbf{\Lambda})$ operations.*

Proof. Let $\ell > 0$ be given. Assume that in the previous iterations all elements of $\{\tilde{\boldsymbol{\psi}}_{\mathbf{m}}(f(u)) : \mathbf{m} \in \mathbf{\Lambda} \cap (I_{\ell-1} \times I^{n-1})\}$ have been computed, as well as all those $(\delta_q \otimes \tilde{\boldsymbol{\psi}}_{\mathbf{p}})(f(u))$ with $(q, \mathbf{p}) \in I_{\ell-1} \times I^{n-1}$ that are needed for the computation of $\{\tilde{\boldsymbol{\psi}}_{\mathbf{m}}(f(u)) : \mathbf{m} \in \mathbf{\Lambda} \cap (J_\ell \times I^{n-1})\}$; and that the current u – being of the form $\sum_{\mathbf{k}\in\mathbf{\Lambda}^{(\ell)}} c^{(\ell)}_{\mathbf{k}} \boldsymbol{\psi}^{(\ell,0,\ldots,0)}_{\mathbf{k}}$ for some graded $(\ell, 0, \ldots, 0)$-tree $\mathbf{\Lambda}^{(\ell)}$ with $\mathbf{\Lambda}^{(\ell)} \backslash (I_\ell \times I^{n-1}) = \mathbf{\Lambda} \setminus (I_\ell \times I^{n-1})$ – is equal to the original u on $\operatorname{supp} \tilde{\boldsymbol{\psi}}_{\mathbf{m}}$ for $\mathbf{m} \in \mathbf{\Lambda} \setminus (I_{\ell-1} \times I^{n-1})$. The arguments that will be given below show that these assumptions are valid for $\ell = 1$.

Then what is left to compute is $\{(\delta_m \otimes \tilde{\boldsymbol{\psi}}_{\mathbf{p}})(f(u)) : m \in \Lambda_1 \cap J_\ell, \mathbf{p} \in \mathbf{\Lambda}(m)\}$ – which together with previously computed $(\delta_q \otimes \tilde{\boldsymbol{\psi}}_{\mathbf{p}})(f(u))$ with $(q, \mathbf{p}) \in I_{\ell-1} \times I^{n-1}$ yields $\{\tilde{\boldsymbol{\psi}}_{\mathbf{m}}(f(u)) : \mathbf{m} \in \mathbf{\Lambda} \cap (J_\ell \times I^{n-1})\}$ –, as well as $\{\tilde{\boldsymbol{\psi}}_{\mathbf{m}}(f(u)) : \mathbf{m} \in \mathbf{\Lambda} \setminus (I_\ell \times I^{n-1})\}$.

Concerning the first task, recalling that $u = \sum_{\mathbf{k}\in\mathbf{\Lambda}^{(\ell)}} c^{(\ell)}_{\mathbf{k}} \boldsymbol{\psi}^{(\ell,0,\ldots,0)}_{\mathbf{k}}$, given $m \in \Lambda_1 \cap J_\ell$ let us denote $v^{(m)} = \sum_{\mathbf{p}\in\mathbf{\Lambda}^{(\ell)}(m)} c^{(\ell)}_{m,\mathbf{p}} \psi^{(0)}_{p_1} \otimes \cdots \otimes \psi^{(0)}_{p_{n-1}}$. Then for any y, we have $(f(u))(m, y) = \bar{f}(u(m, y)) = \bar{f}(v^{(m)}(y)) = (f(v^{(m)}))(y)$, where we used the convention introduced in Remark 3.11, and so in particular,

$$(\delta_m \otimes \tilde{\boldsymbol{\psi}}_{\mathbf{p}})(f(u)) = \tilde{\boldsymbol{\psi}}_{\mathbf{p}}(f(v^{(m)})).$$

Since $\mathbf{\Lambda}^{(\ell)}(m)$ is a graded $\mathbf{0}$-tree, where $\mathbf{0} \in \mathbb{N}_0^{n-1}$, using induction to the number of factors n and Theorem 2.10, we conclude that the first task is performed by means of the recursive call of **tensoreval**. After the subsequent loop over $\mathbf{p}$, $\{\tilde{\boldsymbol{\psi}}_{\mathbf{m}}(f(u)) : \mathbf{m} \in \mathbf{\Lambda} \cap (J_\ell \times I^{n-1})\}$ has been evaluated.

Next we come to the part in the algorithm in which elements are dropped. Since $\mathbf{\Lambda}^{(\ell)} \backslash (I_\ell \times I^{n-1}) = \mathbf{\Lambda} \backslash (I_\ell \times I^{n-1})$, the condition whether $k_1 \in I_\ell$ has an ℓ-child p with $(p, \bar{\mathbf{k}}) \in \mathbf{\Lambda}$ is the same as whether it has such a child with $(p, \bar{\mathbf{k}}) \in \mathbf{\Lambda}^{(\ell)}$. As a consequence, Proposition 3.8 shows that the resulting $\mathbf{\Lambda}^{(\ell)}$ after the dropping of terms is again a graded $(\ell, 0, \ldots, 0)$-tree. Corollary 3.10 shows that for any $\mathbf{k}$ outside this tree, so in particular for those that have been dropped, and for any $\mathbf{m}$ inside this tree, $\operatorname{supp} \boldsymbol{\psi}_{\mathbf{k}}^{(\boldsymbol{\ell})} \cap \operatorname{supp} \tilde{\boldsymbol{\psi}}_{\mathbf{m}} = \emptyset$. Since the tree contains $\mathbf{\Lambda} \backslash (I_\ell \times I^{n-1})$, we conclude that the terms that have been dropped are irrelevant for the forthcoming computation of $\{\tilde{\boldsymbol{\psi}}_{\mathbf{m}}(f(u)) : \mathbf{m} \in \mathbf{\Lambda} \setminus (I_\ell \times I^{n-1})\}$.

Besides those that were already computed earlier in this iteration, in the next loop, the $\{(\delta_m \otimes \tilde{\boldsymbol{\psi}}_{\mathbf{p}})(f(u)) : \mathbf{p} \in \mathbf{\Lambda}^{(\ell)}(m)\}$ for all remaining $m \in \Lambda_1^{(\ell)} \cap I_\ell$ are computed, again using a recursive call of **tensoreval**.

The application of (2.3) in the first coordinate direction means that for any $k \in \Lambda_1^{(\ell)} \setminus J_{\ell+1}$ and $\mathbf{p} \in \mathbf{\Lambda}^{(\ell)}(k)$, $c_{k,\mathbf{p}}^{(\ell+1)} = c_{k,\mathbf{p}}^{(\ell)}$, and that for $k \in J_{\ell+1}$ and $\mathbf{p} \in I^{n-1}$,

$$c_{k,\mathbf{p}}^{(\ell+1)} = c_{k,\mathbf{p}}^{(\ell)} + \sum_{q \in I_\ell \cap \Lambda_1^{(\ell)}} \phi_{\ell,q}(k) c_{q,\mathbf{p}}^{(\ell)},$$

and so

$$\mathbf{\Lambda}^{(\ell+1)} = \mathbf{\Lambda}^{(\ell)} + \bigcup_{k \in J_{\ell+1}} \bigcup_{\{q \in I_\ell \cap \Lambda_1^{(\ell)} : \phi_{\ell,q}(k) \neq 0\}} \{k\} \times \mathbf{\Lambda}^{(\ell)}(q).$$

Since $\mathbf{\Lambda}^{(\ell)}$ is a graded $(\ell, 0, \ldots, 0)$-tree, it is a graded $(\ell + 1, 0, \ldots, 0)$-tree (cf. Remark 3.11). Since furthermore, for any $k \in I_{\ell+1}$ and a graded $\mathbf{0}$-tree $\bar{\mathbf{\Lambda}}$, where $\mathbf{0} \in \mathbb{N}_0^{n-1}$, $\{k\} \times \bar{\mathbf{\Lambda}}$ is a graded $(\ell + 1, 0, \ldots, 0)$-tree, and the union of graded $(\ell + 1, 0, \ldots, 0)$-trees is a graded $(\ell + 1, 0, \ldots, 0)$-tree, we conclude that $\mathbf{\Lambda}^{(\ell+1)}$ is a graded $(\ell + 1, 0, \ldots, 0)$-tree.

By combining above statements, we conclude that Algorithm 3.12 produces $\{\tilde{\boldsymbol{\psi}}_{\mathbf{m}}(f(u)) : \mathbf{m} \in \mathbf{\Lambda}\}$.

The statement about the cost is a consequence of the fact that, by the dropping of indices from $\mathbf{\Lambda}^{(\ell)}$, before the application of (2.3),

$$\#(\mathbf{\Lambda}^{(\ell+1)} \cap (I_{\ell+1} \times I^{n-1})) \lesssim \#(\mathbf{\Lambda} \cap (J_{\ell+1} \times I^{n-1})),$$

whereas an induction argument shows that the work for each value of the counter ℓ is $\mathcal{O}(\#(\mathbf{\Lambda}^{(\ell)} \cap (I_\ell \times I^{n-1}))$. □

4. Transformation from one wavelet basis to another

4.1. **The non-tensor product case.** Let $V_0 \subset V_1 \subset \cdots$ be a multiresolution analysis on some domain Ω with V_j spanned by both $\Phi_j := \{\phi_\lambda : \lambda \in \Delta_j\}$ and, for $j > 0$, by the two-level collection $\Psi_j := \{\psi_\lambda : \lambda \in \nabla_j\} \cup \Phi_{j-1}$; and let $\tilde{\Phi}_j$ and, for $j > 0$, $\tilde{\Psi}_j$ be dual collections, i.e., $\tilde{\Phi}_j(\Phi_j) = I$, $\tilde{\Psi}_j(\Psi_j) = I$ and $\tilde{\Psi}_j(\Phi_{j-1}) = 0$. For convenience, we set $\nabla_0 := \Delta_0$ and $\Psi_0 := \Phi_0$. W.l.o.g. assuming that both the Δ_j's and ∇_j's are mutually disjoint, we set $\Delta := \cup_{j\geq 0}\Delta_j$, $\nabla := \cup_{j\geq 0}\nabla_j$, $\Psi := \cup_{j\geq 0}\Psi_j = \{\psi_\lambda : \lambda \in \nabla\}$, and $|\lambda| := j$ when $\lambda \in \Delta$ is in Δ_j or when $\lambda \in \nabla$ is in ∇_j.

Besides $(V_j)_j$, we consider another multiresolution analysis $\breve{V}_0 \subset \breve{V}_1 \subset \cdots$ on Ω, and corresponding wavelet collection $\breve{\Psi} = \cup_{j\geq 0}\breve{\Psi}_j = \cup_{j\geq 0}\{\breve{\psi}_\lambda : \lambda \in \breve{\nabla}_j\}$, i.e., $V_j = \operatorname{span} \cup_{p=0}^{j} \breve{\Psi}_p$. Assuming that the $\breve{\nabla}_j$'s are mutually disjoint, we set $\breve{\nabla} := \cup_{j\geq 0}\breve{\nabla}_j$, and $|\lambda| := j$ when $\lambda \in \breve{\nabla}$ is in $\breve{\nabla}_j$.

We assume that all above collections are both *locally finite* – i.e., any subset of Ω with diameter 2^{-j} has non-empty intersection with the supports of a uniformly bounded number of functions of Φ_j, $\tilde{\Phi}_j$, Ψ_j, $\tilde{\Psi}_j$ or $\breve{\Psi}_j$ – and *local*, i.e.,

$$\begin{aligned} d_\Phi &:= \sup_{\lambda\in\Delta} 2^{|\lambda|}\operatorname{diam}\operatorname{supp}\phi_\lambda, & d_{\tilde{\Phi}} &:= \sup_{\lambda\in\Delta} 2^{|\lambda|}\operatorname{diam}\operatorname{supp}\tilde{\phi}_\lambda, \\ d_\Psi &:= \sup_{\lambda\in\nabla} 2^{|\lambda|}\operatorname{diam}\operatorname{supp}\psi_\lambda, & d_{\tilde{\Psi}} &:= \sup_{\lambda\in\nabla} 2^{|\lambda|}\operatorname{diam}\operatorname{supp}\tilde{\psi}_\lambda, \\ d_{\breve{\Psi}} &:= \sup_{\lambda\in\breve{\nabla}} 2^{|\lambda|}\operatorname{diam}\operatorname{supp}\breve{\psi}_\lambda \end{aligned}$$

are all finite. Finally, we assume that for any $j \in \mathbb{N}_0$

$$\cup_{\lambda\in\breve{\nabla}_j}\operatorname{supp}\breve{\psi}_\lambda = \overline{\Omega}. \tag{4.1}$$

Definition 4.1. Fixing some constant $t \geq 0$, we call $\mu \in \nabla$ $(\breve{\nabla})$ a parent of $\lambda \in \nabla$ $(\breve{\nabla})$ or, equivalently, λ a child of μ, when $|\lambda| = |\mu| + 1$ and

$$\operatorname{dist}(\operatorname{supp}\psi_\lambda, \operatorname{supp}\psi_\mu) \leq t2^{-|\mu|}.$$

$(\operatorname{dist}(\operatorname{supp}\breve{\psi}_\lambda, \operatorname{supp}\breve{\psi}_\mu) \leq t2^{-|\mu|})$. A subset $\Lambda \subset \nabla$ $(\breve{\nabla})$ is now called a graded tree or, to make the dependance on t explicit, a *t-graded tree*, if whenever $\lambda \in \nabla$ $(\breve{\nabla})$ with $|\lambda| > 0$ is in Λ, then so are all its parents.

Remark 4.2. For the special case that Ψ (or $\breve{\Psi}$) is a collection of *interpolets*, in the preceding sections we gave already a different definition of a graded tree. Whereas in Definition 4.1, distances between indices in ∇ $(\breve{\nabla})$ are measured in terms of the distances between the supports of the corresponding wavelets, in Definition 2.2, ∇ is a collection of interpolation points, so that the distance between indices could be simply defined as the distance between these points. Condition (4.1) can be viewed as a substitute for condition (2.2) in the interpolet case. Since an interpolation point is inside the support of the corresponding interpolet, and the interpolets were always assumed to be local, actually both definitions of graded trees are equivalent. More precisely, with ∇ being a collection of interpolation points, given a gradedness parameter c_2+c_3 as in Definition 2.2, a set $\Lambda \subset \nabla$ that, for t being sufficiently large, is a t-graded tree, is also a graded 0-tree according to Definition 2.2. Conversely, given a t, a set $\Lambda \subset \nabla$ that, for $c_2 + c_3$ being sufficiently large, is a graded 0-tree, is also a t-graded tree.

Assuming that for some constant L,

$$\breve{V}_j \subset V_{j+L} \quad (j \in \mathbb{N}_0), \tag{4.2}$$

in this subsection we will show that an expansion $u = \sum_{\lambda\in\breve{\Lambda}} \breve{c}_\lambda\breve{\psi}_\lambda$ where $\breve{\Lambda} \subset \breve{\nabla}$ is a graded tree, can be rewritten in $\mathcal{O}(\#\breve{\Lambda})$ operations as $u = \sum_{\lambda\in\Lambda} c_\lambda\psi_\lambda$ for some graded tree $\Lambda \subset \nabla$ with $\#\Lambda \lesssim \#\breve{\Lambda}$. Subsequently, in the next subsection, we prove a corresponding statement for multiple graded trees. By applying this result with either $V_j = V_j^{(\mathrm{I})}$ or $\breve{V}_j = V_j^{(\mathrm{I})}$, we have proven the claims made in §1.3 about the possibility to make transformations in linear complexity between representations with respect to tensor products of interpolets and tensor products of certain types of "true" biorthogonal wavelet bases.

For notational convenience only, we will assume that (4.2) is valid with

$$L = 0.$$

In the following proposition, it is shown that we may always assume that trees are sufficiently "fat" or strongly graded.

Proposition 4.3. *Let $\breve{\Lambda} \subset \breve{\nabla}$ be a $\breve{t}$-graded tree. Then for $\bar{t} > \breve{t}$, $\breve{\Lambda}$ can be enlarged to a $\bar{t}$-graded tree $\bar{\Lambda} \subset \breve{\nabla}$ with $\#\bar{\Lambda} \lesssim \#\breve{\Lambda}$, only dependent on $\breve{t}$, $\bar{t}$, and the local finiteness and localness of $\breve{\Psi}$.*

Proof. Enlarge $\breve{\Lambda}$ to $\bar{\Lambda}$ by adding to it any $\lambda \in \breve{\nabla}$ for which there exists a $\lambda' \in \breve{\Lambda}$ with $|\lambda'| = |\lambda|$ and $\operatorname{dist}(\operatorname{supp}\breve{\psi}_\lambda, \operatorname{supp}\breve{\psi}_{\lambda'}) \leq (2(\bar{t}-\breve{t}) + d_{\breve{\Psi}})2^{-|\lambda|}$. One easily verifies the claim about $\#\bar{\Lambda}$.

To show that $\bar{\Lambda}$ is a $\bar{t}$-graded tree, let $\lambda \in \bar{\Lambda}$ with $|\lambda| > 0$, and let $\mu \in \breve{\nabla}$ with $|\lambda| = |\mu| + 1$ and $\operatorname{dist}(\operatorname{supp}\breve{\psi}_\lambda, \operatorname{supp}\breve{\psi}_\mu) \leq \bar{t}2^{-|\mu|}$. Let $\lambda' \in \breve{\Lambda}$ be such that $|\lambda'| = |\lambda|$ and $\operatorname{dist}(\operatorname{supp}\breve{\psi}_\lambda, \operatorname{supp}\breve{\psi}_{\lambda'}) \leq (2(\bar{t}-\breve{t}) + d_{\breve{\Psi}})2^{-|\lambda|}$, so that $\operatorname{dist}(\operatorname{supp}\breve{\psi}_{\lambda'}, \operatorname{supp}\breve{\psi}_\mu) \leq \bar{t}2^{-|\mu|} + d_{\breve{\Psi}}2^{-(|\mu|+1)} + (2(\bar{t}-\breve{t}) + d_{\breve{\Psi}})2^{-(|\mu|+1)}$.

Let us assume that $\operatorname{dist}(\operatorname{supp}\breve{\psi}_{\lambda'}, \operatorname{supp}\breve{\psi}_\mu) > \breve{t}2^{-|\mu|}$ since otherwise we already know that $\mu \in \breve{\Lambda} \subset \bar{\Lambda}$. Let x on the shortest path between $\operatorname{supp}\breve{\psi}_{\lambda'}$ and $\operatorname{supp}\breve{\psi}_\mu$ with $\operatorname{dist}(x, \operatorname{supp}\breve{\psi}_{\lambda'}) = \breve{t}2^{-|\mu|}$. Thanks to (4.1), $x \in \operatorname{supp}\breve{\psi}_{\mu'}$ for some $\mu' \in \breve{\nabla}$ with $|\mu'| = |\mu|$, and so $\mu' \in \breve{\Lambda}$. We infer that

$$\begin{aligned}\operatorname{dist}(\operatorname{supp}\breve{\psi}_{\mu'}, \operatorname{supp}\breve{\psi}_\mu) &\leq \bar{t}2^{-|\mu|} + d_{\breve{\Psi}}2^{-(|\mu|+1)} + (2(\bar{t}-\breve{t}) + d_{\breve{\Psi}})2^{-(|\mu|+1)} - \breve{t}2^{-|\mu|} \\ &= (2(\bar{t}-\breve{t}) + d_{\breve{\Psi}})2^{-|\mu|},\end{aligned}$$

and so $\mu \in \bar{\Lambda}$ which completes the proof. □

Although the proof of the next lemma is very similar to that of Proposition 4.3, since it is short we include it for convenience.

Lemma 4.4. *Let $\breve{\Lambda} \subset \breve{\nabla}$ be a $\breve{t}$-graded tree. For some constant $d \geq 0$ such that $t := \frac{1}{2}d + \breve{t} - \frac{1}{2}c_\Psi \geq 0$, let Λ be the set of $\lambda \in \nabla$ for which there exists a $\breve{\lambda} \in \breve{\Lambda}$ with $|\breve{\lambda}| = |\lambda|$ and $\operatorname{dist}(\operatorname{supp}\breve{\psi}_{\breve{\lambda}}, \operatorname{supp}\psi_\lambda) \leq d2^{-|\lambda|}$. Then $\Lambda \subset \nabla$ is a t-graded tree with $\#\Lambda \lesssim \#\breve{\Lambda}$ dependent only on $\breve{t}$, t, the localness of $\breve{\Psi}$ and the local finiteness and localness of Ψ.*

Proof. The statement about $\#\Lambda$ is obvious.

Let $\lambda \in \Lambda$, $|\lambda| > 0$, $\mu \in \nabla$, with $|\mu| = |\lambda| + 1$ and $\operatorname{dist}(\operatorname{supp}\psi_\lambda, \operatorname{supp}\psi_\mu) \leq t2^{-|\mu|}$. Let $\breve{\lambda} \in \breve{\Lambda}$ be such that $|\breve{\lambda}| = |\lambda|$ and $\operatorname{dist}(\operatorname{supp}\breve{\psi}_{\breve{\lambda}}, \operatorname{supp}\psi_\lambda) \leq d2^{-|\lambda|}$, so that $\operatorname{dist}(\operatorname{supp}\breve{\psi}_{\breve{\lambda}}, \operatorname{supp}\psi_\mu) \leq d2^{-|\lambda|} + c_\Psi 2^{-|\lambda|} + t2^{-|\mu|}$. Let x on the shortest path between $\operatorname{supp}\breve{\psi}_{\breve{\lambda}}$ and $\operatorname{supp}\psi_\mu$ with $\operatorname{dist}(x, \operatorname{supp}\breve{\psi}_{\breve{\lambda}}) = \breve{t}2^{-|\mu|}$. Thanks to (4.1), $x \in \operatorname{supp}\breve{\psi}_{\breve{\mu}}$ for some $\breve{\mu} \in \breve{\nabla}$ with $|\breve{\mu}| = |\mu|$, and so $\breve{\mu} \in \breve{\Lambda}$. We infer that

$$\operatorname{dist}(\operatorname{supp}\breve{\psi}_{\breve{\mu}}, \operatorname{supp}\breve{\psi}_\mu) \leq d2^{-|\lambda|} + c_\Psi 2^{-|\lambda|} + t2^{-|\mu|} - \breve{t}2^{-|\mu|} = d2^{-|\mu|},$$

and so $\mu \in \Lambda$ which completes the proof. □

Algorithm 4.5.

```
**transform**$(\breve\Lambda,(\breve c_{\breve\lambda})_{\breve\lambda\in\breve\Lambda})$
% Let $\breve\Lambda\subset\breve\nabla$ be a $\breve t$-graded tree with $\breve t\geq c_\Phi+c_{\tilde\Phi}$. Let $\breve V_j\subset V_j$ $(j\in\mathbb{N}_0)$.
For a constant $d\geq\max(c_{\tilde\Psi},c_\Psi-2\breve t)$, let $\Lambda$ be the set of $\lambda\in\nabla$ for which
there exists a $\breve\lambda\in\breve\Lambda$ with $|\breve\lambda|=|\lambda|$ and $\operatorname{dist}(\operatorname{supp}\breve\psi_{\breve\lambda},\operatorname{supp}\psi_\lambda)\leq d2^{-|\lambda|}$.
$J:=\max_{\breve\lambda\in\breve\nabla}|\breve\lambda|$, $w_{J+1}:=0$
for $j=J,J-1,\ldots,1$ do
   $\breve v_j:=\sum_{\breve\lambda\in\breve\Lambda\cap\breve\nabla_j}\breve c_{\breve\lambda}\breve\psi_{\breve\lambda}$
   for $\lambda\in\Lambda\cap\nabla_j$ do $c_\lambda:=\tilde\psi_\lambda(w_{j+1}+\breve v_j)$ enddo
   $w_j:=\sum_{\lambda\in\Delta_{j-1}}\tilde\phi_\lambda(w_{j+1}+\breve v_j)\phi_\lambda$
enddo
for $\lambda\in\Lambda\cap\nabla_0$ do $c_\lambda:=\tilde\psi_\lambda(w_1+\breve v_0)$ enddo
return $(\Lambda,(c_\lambda)_{\lambda\in\Lambda})$
```

Theorem 4.6. *Algorithm 4.5 produces* $(c_\lambda)_{\lambda\in\Lambda}$ *with* $\sum_{\lambda\in\Lambda}c_\lambda\psi_\lambda=\sum_{\breve\lambda\in\breve\Lambda}\breve c_{\breve\lambda}\breve\psi_{\breve\lambda}$ *in* $\mathcal{O}(\#\breve\Lambda)$ *operations. Furthermore, with* $t:=\frac{1}{2}d+\breve t-\frac{1}{2}c_\Psi$, $\Lambda\subset\nabla$ *is a* t*-graded tree with* $\#\Lambda\lesssim\#\breve\Lambda$.

Proof. By the condition $d\geq c_\Psi-2\breve t$, the last statement is shown in Lemma 4.4.

For $j=J,J-1,\ldots,0$, we have

$$w_{j+1}+\breve v_j\in V_j \tag{4.3}$$

by $\breve V_j\subset V_j$, as well as

$$\operatorname{supp}(w_{j+1}+\breve v_j)\subset\cup_{\breve\lambda\in\breve\Lambda\cap\breve\nabla_j}\operatorname{supp}\breve\psi_{\breve\lambda}. \tag{4.4}$$

Indeed, the last statement is valid for $j=J$, let us assume that it valid for $j+1$. Now it suffices to show that $\operatorname{supp}w_{j+1}\subset\cup_{\breve\lambda\in\breve\Lambda\cap\breve\nabla_j}\operatorname{supp}\breve\psi_{\breve\lambda}$. For any $x\in\operatorname{supp}w_{j+1}$, there exists a $\lambda\in\Delta_j$ and a $\breve\lambda\in\breve\Lambda\cap\breve\nabla_{j+1}$ such that $x\in\operatorname{supp}\phi_\lambda$ and $\operatorname{supp}\tilde\phi_\lambda\cap\operatorname{supp}\breve\psi_{\breve\lambda}\neq\emptyset$, and so $\operatorname{dist}(x,\operatorname{supp}\breve\psi_{\breve\lambda})\leq(c_\Phi+c_{\tilde\Phi})2^{-j}\leq\breve t2^{-j}$. But then there exists a $\breve\mu\in\breve\Lambda\cap\breve\nabla_j$ with $x\in\operatorname{supp}\breve\psi_{\breve\mu}$.

From (4.4) and the localness and local finiteness assumptions, it follows that the number of non-zero terms in the expansion for w_j is $\mathcal{O}(\#(\breve\Lambda\cap\breve\nabla_j))$, and also that the total number of operations needed for the algorithm is $\mathcal{O}(\#\breve\nabla_j)$. Furthermore, thanks to $d\geq c_{\tilde\Psi}$, (4.4) also implies that

$$\tilde\psi_\lambda(w_{j+1}+\breve v_j)=0\quad(\lambda\in\nabla_j\setminus\Lambda). \tag{4.5}$$

With, for $j\in\mathbb{N}_0$, $u_j:=\sum_{\{\lambda\in\Lambda,|\lambda|\geq j\}}c_\lambda\psi_\lambda$, and $w_0:=0$, we have

$$\sum_{\breve\lambda\in\breve\Lambda}\breve c_{\breve\lambda}\breve\psi_{\breve\lambda}=u_j+w_j+\sum_{p=0}^{j-1}\breve v_p\quad(j=J+1,J,\ldots,0). \tag{4.6}$$

Indeed (4.6) is valid for $j=J+1$. Now let it be valid for $j+1$. Then (4.6) is equivalent to $u_j+w_j+\sum_{p=0}^{j-1}\breve v_p=u_{j+1}+w_{j+1}+\sum_{p=0}^{j}\breve v_p$ or

$$\sum_{\lambda\in\Lambda\cap\nabla_j}c_\lambda\psi_\lambda+w_j=w_{j+1}+\breve v_j.$$

This equality is a consequence of (4.3), (4.5), and the fact that $\Psi_j \cup \Phi_{j-1}$ ($\Phi_{-1} := \emptyset$) is a basis for V_j. The equality (4.6) for $j = 0$ is the last statement that was to be shown. □

4.2. **The tensor product case.** Let $n \in \mathbb{N}$. For $\boldsymbol{\lambda} \in \boldsymbol{\nabla} := \nabla^n$ and $\check{\boldsymbol{\lambda}} \in \check{\boldsymbol{\nabla}} := \check{\nabla}^n$, we set

$$\boldsymbol{\psi}_{\boldsymbol{\lambda}} := \psi_{\lambda_1} \otimes \cdots \otimes \psi_{\lambda_n}, \quad \check{\boldsymbol{\psi}}_{\check{\boldsymbol{\lambda}}} := \check{\psi}_{\check{\lambda}_1} \otimes \cdots \otimes \check{\psi}_{\check{\lambda}_n}.$$

In view of Remark 4.2 and Definition 3.2, an appropriate definition of a multiple graded tree is given by the following.

Definition 4.7. Given $\mathbf{t} \in \mathbb{R}^n_{\geq 0}$, $\boldsymbol{\Lambda} \subset \boldsymbol{\nabla}$ ($\check{\boldsymbol{\nabla}}$) is called a $\mathbf{t}$-*graded tree* when for all $1 \leq i \leq n$ and $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}$, the set of $\mu \in \nabla$ ($\check{\nabla}$) with $(\lambda_1, \ldots, \lambda_{i-1}, \mu, \lambda_{i+1}, \ldots, \lambda_n) \in \boldsymbol{\Lambda}$ is a t_i-graded tree.

Given a $(\check{t}, \ldots, \check{t})$-graded tree $\check{\boldsymbol{\Lambda}} \subset \check{\boldsymbol{\nabla}}$ and $u = \sum_{\check{\boldsymbol{\lambda}} \in \check{\boldsymbol{\Lambda}}} \check{c}_{\check{\boldsymbol{\lambda}}} \check{\boldsymbol{\psi}}_{\check{\boldsymbol{\lambda}}}$, our remaining task is to represent u, in $\mathcal{O}(\#\check{\boldsymbol{\Lambda}})$ operations, as $u = \sum_{\boldsymbol{\lambda} \in \boldsymbol{\Lambda}} c_{\boldsymbol{\lambda}} \boldsymbol{\psi}_{\boldsymbol{\lambda}}$, where $\boldsymbol{\Lambda}$ is a $(t, \ldots, t)$-graded tree with $\#\boldsymbol{\Lambda} \lesssim \#\check{\boldsymbol{\Lambda}}$. As we will see, this task can be performed simply by applying Algorithm 4.5 in all coordinate directions.

We set $P : \boldsymbol{\lambda} \mapsto (\lambda_1, \cdots, \lambda_{n-1})$. Then

$$\check{\boldsymbol{\Lambda}} = \cup_{\boldsymbol{\mu} \in P(\check{\boldsymbol{\Lambda}})} P^{-1}(\{\boldsymbol{\mu}\}) = \cup_{\boldsymbol{\mu} \in P(\check{\boldsymbol{\Lambda}})} (\boldsymbol{\mu}, P^{-1}(\{\boldsymbol{\mu}\}) \cdot \mathbf{e}_n),$$

whereas Definition 4.7 shows us that for any of these $\boldsymbol{\mu}$, $\check{\Lambda}(\boldsymbol{\mu}) := P^{-1}(\{\boldsymbol{\mu}\}) \cdot \mathbf{e}_n$ is a $\check{t}$-graded tree. Now for any fixed $\boldsymbol{\mu} \in P(\check{\boldsymbol{\Lambda}})$, if necessary we enlarge $\check{\Lambda}(\boldsymbol{\mu})$ to a $\max(\check{t}, c_\Phi + c_{\tilde{\Phi}})$-graded tree (cf. Proposition 4.3), and then apply

$$(\Lambda(\boldsymbol{\mu}), (c_{\boldsymbol{\mu},\lambda})_{\lambda \in \Lambda(\boldsymbol{\mu})}) := \mathbf{transform}(\check{\Lambda}(\boldsymbol{\mu}), (\check{c}_{\boldsymbol{\mu},\check{\lambda}})_{\check{\lambda} \in \check{\Lambda}(\boldsymbol{\mu})}),$$

giving

$$\sum_{\boldsymbol{\mu} \in P(\check{\boldsymbol{\Lambda}})} \sum_{\lambda \in \Lambda(\boldsymbol{\mu})} c_{\boldsymbol{\mu},\lambda} \check{\psi}_{\mu_1} \otimes \cdots \otimes \check{\psi}_{\mu_{n-1}} \otimes \psi_\lambda = \sum_{\check{\boldsymbol{\lambda}} \in \check{\boldsymbol{\Lambda}}} c_{\check{\boldsymbol{\lambda}}} \check{\boldsymbol{\psi}}_{\check{\boldsymbol{\lambda}}}.$$

As follows from Theorem 4.6, the total number of operations required by all these calls is $\mathcal{O}(\#\check{\boldsymbol{\Lambda}})$, and $\cup_{\boldsymbol{\mu} \in P(\check{\boldsymbol{\Lambda}})} (\boldsymbol{\mu}, \Lambda(\boldsymbol{\mu}))$ is a $(\check{t}, \ldots, \check{t}, t)$-graded tree, where $t = \frac{1}{2}d + \max(\check{t}, c_\Phi + c_{\tilde{\Phi}}) - \frac{1}{2}c_\Psi$. Indeed, for any $\boldsymbol{\mu} \in P(\check{\boldsymbol{\Lambda}})$, $\Lambda(\boldsymbol{\mu})$ is a t-graded tree. To show the analogous statement when $\cup_{\boldsymbol{\mu} \in P(\check{\boldsymbol{\Lambda}})} (\boldsymbol{\mu}, \Lambda(\boldsymbol{\mu}))$ is frozen in any other set of $n-1$ coordinates, it is sufficient to consider the case that $n = 2$. Let $(\mu, \lambda) \in \cup_{\mu \in P(\check{\boldsymbol{\Lambda}})} (\mu, \Lambda(\mu))$ and let γ be a parent of μ. By construction of $\Lambda(\mu)$ in Algorithm 4.5, there exists a $\check{\lambda} \in \check{\Lambda}(\mu)$ with $|\check{\lambda}| = |\lambda|$ and $\operatorname{dist}(\operatorname{supp} \check{\psi}_{\check{\lambda}}, \operatorname{supp} \psi_\lambda) \leq d2^{-|\lambda|}$. Since $\check{\boldsymbol{\Lambda}}$ is a $(\check{t}, \check{t})$-graded tree, it holds that $(\gamma, \check{\lambda}) \in \check{\boldsymbol{\Lambda}}$, but then also $(\gamma, \lambda) \in \cup_{\mu \in P(\check{\boldsymbol{\Lambda}})} (\mu, \Lambda(\mu))$ as required.

By repeating the application of **transform** in each of the other $n-1$ coordinate directions, we have proven our claim about the transformation of $u = \sum_{\check{\boldsymbol{\lambda}} \in \check{\boldsymbol{\Lambda}}} \check{c}_{\check{\boldsymbol{\lambda}}} \check{\boldsymbol{\psi}}_{\check{\boldsymbol{\lambda}}}$ into $u = \sum_{\boldsymbol{\lambda} \in \boldsymbol{\Lambda}} c_{\boldsymbol{\lambda}} \boldsymbol{\psi}_{\boldsymbol{\lambda}}$ in linear complexity.

References

[BDS07] A. Barinka, W. Dahmen, and R. Schneider. Fast computation of adaptive wavelet expansions. *Numer. Math.*, 105(4):549–589, 2007.

[CDD01] A. Cohen, W. Dahmen, and R. DeVore. Adaptive wavelet methods for elliptic operator equations – Convergence rates. *Math. Comp*, 70:27–75, 2001.

[CDD02] A. Cohen, W. Dahmen, and R. DeVore. Adaptive wavelet methods II - Beyond the elliptic case. *Found. Comput. Math.*, 2(3):203–245, 2002.

[CDD03] A. Cohen, W. Dahmen, and R. DeVore. Adaptive wavelet schemes for nonlinear variational problems. *SIAM J. Numer. Anal.*, 41:1785–1823, 2003.

[CDDD01] A. Cohen, W. Dahmen, I. Daubechies, and R. DeVore. Tree approximation and optimal encoding. *Appl. Comput. Harmon. Anal.*, 11(2):192–226, 2001.

[DGH99] G.C. Donovan, J.S. Geronimo, and D.P. Hardin. Orthogonal polynomials and the construction of piecewise polynomial smooth wavelets. *SIAM J. Math. Anal.*, 30(5):1029–1056, 1999.

[DS09] M. Dauge and R.P. Stevenson. Sparse tensor product wavelet approximation of singular functions. Technical report, 2009. Submitted.

[DSS09] T.J. Dijkema, Ch. Schwab, and R.P. Stevenson. An adaptive wavelet method for solving high-dimensional elliptic PDEs. *Constr. Approx.*, 30(3):423–455, 2009.

[DSX00] W. Dahmen, R. Schneider, and Y. Xu. Nonlinear functionals of wavelet expansions—adaptive reconstruction and fast evaluation. *Numer. Math.*, 86(1):49–101, 2000.

[GHS07] T. Gantumur, H. Harbrecht, and R.P. Stevenson. An optimal adaptive wavelet method without coarsening of the iterands. *Math. Comp.*, 76:615–629, 2007.

[GK00] M. Griebel and F. Koster. Adaptive wavelet solvers for the unsteady incompressible Navier-Stokes equations. In *Advances in mathematical fluid mechanics (Paseky, 1999)*, pages 67–118. Springer, Berlin, 2000.

[Nit06] P.-A. Nitsche. Best N-term approximation spaces for tensor product wavelet bases. *Constr. Approx.*, 24(1):49–70, 2006.

[SS08] Ch. Schwab and R.P. Stevenson. Adaptive wavelet algorithms for elliptic PDEs on product domains. *Math. Comp.*, 77:71–92, 2008.

[SS09] Ch. Schwab and R.P. Stevenson. A space-time adaptive wavelet method for parabolic evolution problems. *Math. Comp.*, 78:1293–1318, 2009.

[SU09] W. Sickel and T. Ullrich. Tensor products of Sobolev-Besov spaces and applications to approximation from the hyperbolic cross. *J. Approx. Theory*, 161:748–786, 2009.

Seminar for Applied Mathematics, ETHZ HG G58.1, ETH Zürich, CH 8092 Zürich, Switzerland

E-mail address: schwab@sam.math.ethz.ch

Korteweg-de Vries Institute for Mathematics, University of Amsterdam, P.O. Box 94248, 1090 GE Amsterdam, The Netherlands.

E-mail address: R.P.Stevenson@uva.nl

Research Reports

No.	Authors/Title
10-05	*C. Schwab and R. Stevenson* Fast evaluation of nonlinear functionals of tensor product wavelet expansions
10-04	*B.N. Khoromskij and C. Schwab* Tensor-structured Galerkin approximation of parametric and stochastic elliptic PDEs
10-03	*A. Cohen, R. DeVore and C. Schwab* Analytic regularity and polynomial approximation of parametric and stochastic elliptic PDEs
10-02	*V. Gradinaru, G.A. Hagedorn, A. Joye* Tunneling dynamics and spawning with adaptive semi-classical wavepackets
10-01	*N. Hilber, S. Kehtari, C. Schwab and C. Winter* Wavelet finite element method for option pricing in highdimensional diffusion market models
09-41	*E. Kokiopoulou, D. Kressner, N. Paragios, P. Frossard* Optimal image alignment with random projections of manifolds: algorithm and geometric analysis
09-40	*P. Benner, P. Ezzatti, D. Kressner, E.S. Quintana-Ortí, A. Remón* A mixed-precision algorithm for the solution of Lyapunov equations on hybrid CPU-GPU platforms
09-39	*V. Wheatley, P. Huguenot, H. Kumar* On the role of Riemann solvers in discontinuous Galerkin methods for magnetohydrodynamics
09-38	*E. Kokiopoulou, D. Kressner, N. Paragios, P. Frossard* Globally optimal volume registration using DC programming
09-37	*F.G. Fuchs, A.D. McMurray, S. Mishra, N.H. Risebrom, K. Waagan* Approximate Riemann solvers and stable high-order finite volume schemes for multi-dimensional ideal MHD
09-36	*Ph. LeFloch, S. Mishra* Kinetic functions in magnetohydrodynamics with resistivity and hall effects
09-35	*U.S. Fjordholm, S. Mishra* Vorticity preserving finite volume schemes for the shallow water equations