Adaptive Wavelet Methods using Semiorthogonal Spline Wavelets: Sparse Evaluation of Nonlinear Functions

Kai Bittner and Karsten Urban

Preprint Series: 2004-02

Fakultät für Mathematik und Wirtschaftswissenschaften
UNIVERSITÄT ULM

Adaptive Wavelet Methods using Semiorthogonal Spline Wavelets: Sparse Evaluation of Nonlinear Functions*

Kai Bittner and Karsten Urban

Universität Ulm
Abteilung Numerik, 89069 Ulm, Germany
{kai.bittner,karsten.urban}@mathematik.uni-ulm.de

Abstract

Enormous progress has been made in the construction and analysis of adaptive wavelet methods in the recent years. Cohen, Dahmen, and DeVore showed that such methods converge for a wide class of operator equations, both linear and nonlinear. Moreover, they showed that the rate of convergence is asymptotically optimal and that the methods are asymptotically optimally efficient.

The quantitative behaviour of such methods of course depends on the choice of the wavelet bases used, in particular on the condition number of these bases. It has been observed that compactly supported biorthogonal spline wavelets (for which these adaptive methods are designed for) give rise to condition numbers that cause serious problems in practical applications. An alternative would be the use of semiorthogonal wavelets which are known to have good condition numbers. However, the above mentioned methods do require compactly supported dual wavelets, which in general is not the case for semiorthogonal spline wavelets.

In this paper, we focus on a core ingredient of adaptive wavelet methods for nonlinear problems, namely the adaptive evaluation of nonlinear functions. We present an efficient adaptive method for approximately evaluating nonlinear functions of wavelet expansions using semiorthogonal spline wavelets. This is achieved by modifying and extending a method for compactly supported biorthogonal wavelets by Dahmen, Schneider and Xu.

Using the semiorthogonality, we only need compact support of the primal basis functions. Starting with an adaptive quasi-interpolant in terms of the primal scaling functions, we perform then a fast change of basis into a linear combination of dual scaling functions. Finally, a fast decomposition algorithm is performed, which uses only the finitely supported primal refinement coefficients, to obtain the desired representation in terms of the dual wavelets.

In particular, this paper shows that semiorthogonal spline wavelets can be used in the above mentioned framework of adaptive wavelet schemes for operator equations.

*The work of this paper was partly supported by the EC within the IHP-network *Nonlinear approximation and adaptivity: Breaking complexity in numerical modeling and data representation*, RTN2-2001-00574.

1 Introduction

Enormous progress has been made in the construction and analysis of adaptive wavelet methods for the numerical solution of a wide class of linear and nonlinear operator equations. These operators include partial differential operators (also systems such as in the Stokes equation) and integral operators. It has been shown in [9, 13, 15] that appropriate adaptive wavelet methods converge and in [9, 10, 14] that the rate of convergence is optimal as compared with the best N-term approximation.

Even though the numerical experiments in [1, 18] showed impressive results and confirmed the optimality in qualitative terms, the quantitative behavior of the algorithm did offer some problems. Let us illustrate this by a simple example. The adaptive method in [9, 10] was proven to yield a fixed error reduction per iteration step, i.e.

$$\|u^{(i+1)} - u\| \leq \delta \|u^{(i)} - u\|, \quad 0 < \delta < 1 \ , \tag{1.1}$$

where u is the desired solution. Obviously, (1.1) implies convergence of the sequence $\{u^{(i)}\}_{i\in\mathbb{N}}$ towards u. The constant δ also enters the analysis of the rate of convergence, not qualitatively for $i \to \infty$ but it appears in the size of involved constants.

It has been observed in numerical experiments that this error reduction factor may be poor, e.g. $\delta \approx 0.98$, which would make the whole method practically useless. One core ingredient which influences the size of δ is the *condition number* of the used wavelet basis $\Psi = \{\psi_\lambda : \lambda \in \nabla\}$. These bases form Riesz basis for the respective energy space X (e.g. $H_0^1(\Omega)$ for second order elliptic pde's).

A family $\Psi = \{\psi_\lambda : \lambda \in \nabla\}$ is a called *Riesz basis* for a Hilbert space X if it is complete in X and if the inequalities

$$c_\Psi \|\boldsymbol{d}\|_{\ell_2(\nabla)} \leq \left\| \sum_{\lambda\in\nabla} d_\lambda \psi_\lambda \right\|_X \leq C_\Psi \|\boldsymbol{d}\|_{\ell_2(\nabla)}, \quad \boldsymbol{d} = (d_\lambda)_{\lambda\in\nabla}, \tag{1.2}$$

are satisfied, with constants $0 < c_\Psi \leq C_\Psi < \infty$. The constants c_ψ and C_Ψ are called *lower* and *upper Riesz bound*, respectively. The fraction

$$\rho_\Psi := \frac{C_\Psi}{c_\Psi} \tag{1.3}$$

of the smallest upper and the largest lower Riesz bound is called the *condition* of the Riesz basis. This number gives a quantitative description of the stability of Ψ with respect to X. The optimal case obviously occurs when Ψ is an orthonormal basis, i.e. $\rho_\Psi = 1$.

For numerical purposes we require in addition that the basis function ψ_λ are compactly supported. With respect to these requirements Daubechies' orthonormal compactly support wavelets [19] on $L_2(\mathbb{R})$ would be the optimal choice. However, the use of these functions has some obstructions. Firstly, orthonormality might be a too severe restriction, e.g. when $\Omega \subseteq \mathbb{R}^n$ is complicated or when subspaces of $H_0^1(\Omega)$ have to be considered such as $H(\mathrm{div};\Omega)$ or $H(\mathrm{curl};\Omega)$ (where no compactly supported orthonormal wavelet basis exists). Moreover, these functions are not given by a closed formula but only in terms of a refinement equation. This makes issues like quadrature or point evaluation (though possible exactly) possibly costly in an adaptive framework.

Hence, in particular for the second issue, piecewise polynomial functions would be convenient. However, the above mentioned non-optimal reduction factors δ have been observed numerically for biorthogonal B-spline wavelets from [12, 16]. Several attempts have been made in order to use a preconditioning of these bases, e.g. [2], which was quite successful in 1D but not optimal in higher dimensions.

As an alternative, we consider semiorthogonal spline wavelets here. These wavelets are spline functions, which are levelwise orthogonal and stable on each level. Therefore, Ψ has the Riesz bounds $c_\Psi = \min_j c_{\Psi_j}$ and $C_\Psi = \max_j C_{\Psi_j}$, where Ψ_j consists of all wavelets of the same level (or scale) j. Since stability is much easier to control on a single level, the commonly used examples of semiorthogonal wavelets posses good quantitative stability properties. The main reason why they have not been used so far in adaptive Wavelet-Galerkin methods lies in the fact that the dual system $\tilde{\Psi}$ is global, i.e., the dual wavelets $\tilde{\psi}_\lambda$ are globally supported. This means that the decomposition scheme relies on an infinite mask. It comes out somewhat surprising that this does not pose an obstacle as we show in this paper.

This paper is organized as follows. In Section 2 we review preliminaries on semiorthogonal wavelets and adaptive Wavelet-Galerkin methods. For nonlinear problems, a core ingredient of such methods is the evaluation of nonlinear functions of a wavelet expansion. An efficient approximation method for this task is introduced in [17] for the case of biorthogonal *compactly* supported wavelets. The basic ideas of this approach are also reviewed in Section 2.

The remainder of this paper is devoted to the modification of the DSX-method in [17] for semiorthogonal wavelets. In Section 3, we introduce uniform quasi-interpolation schemes, with emphasis on splines. These schemes are used in Section 4 to construct an adaptive quasi-interpolant. The computational realization of this quasi-interpolant is studied in Section 5, where also the complete algorithm is described.

In particular, we show that semiorthogonal spline wavelets can be used efficiently in the recently developed framework of adaptive wavelet method. In a forthcoming paper, we will report on numerical experiments.

2 Preliminaries

In this section, we recall some preliminary facts that will be needed throughout the paper.

2.1 Semiorthogonal Wavelets

Most examples of semiorthogonal wavelets (also known as pre-wavelets) on $L_2(\mathbb{R})$ are induced by a multiresolution analysis (MRA) which is generated by a refinable function ϕ. We do not want to restrict ourselves to this particular 1-D cardinal case only and consider MRA in $L_2(\Omega)$, where $\Omega \subseteq \mathbb{R}^n$. Let $\mathcal{S} = \{S_j\}_{j\geq 0}$ denote a MRA, i.e. a sequence of nested spaces $S_j \subset S_{j+1}$ with trivial intersection such that the union of all these spaces is dense in $L_2(\Omega)$. However, for notational convenience we restrict ourselves first to the simple case $\Omega = \square := [0,1]^n$ (and tensor products of Wavelets for $L_2([0,1])$).

It should be noted that this is *not* a limitation of generality of the presented method. In fact, all constructions of wavelets on general domains we are aware of include a de-

composition of the domain Ω and parametric mappings of the induced subdomains to $\square$. Thus, it is not just convenient to work on $\square$, but follows also the spirit of wavelet constructions on general domains.

In what follows, the abbreviation $A \lesssim B$ will indicate the existence of an absolute constant $c > 0$ such that $A \leq cB$, and by $A \sim B$ we always mean that $A \lesssim B \lesssim A$. Furthermore, scaling functions and wavelets will be labeled by pairs $\lambda = (j,k)$ and triplets $\lambda = (j,k,e)$ of indices, respectively, where $|\lambda| = j \in \mathbb{N}_0$ denotes the scale or level of a function. The index $k \in \mathbb{Z}^n$ usually refers to the location in space and $e \in \{0,1\}^n$ to the type of wavelet. For any index set Λ we collect in

$$\Lambda_j := \{\lambda \in \Lambda : \ |\lambda| = j\}$$

all indices on a fixed level $j \in \mathbb{N}_0$.

Throughout the remainder of this paper, we pose the following assumptions.

Assumption 2.1 *Let $\Phi := \{\phi_\lambda : \lambda \in \Delta\}$ with a suitable index set $\Delta \subset \mathbb{N}_0 \times \mathbb{Z}^n$, be a family of $L_2(\Omega)$-functions satisfying the following properties:*

1. *The functions are* locally supported, *i.e.,*

$$\sigma_{(j,k)} := \operatorname{supp} \phi_{(j,k)} \subset 2^{-j}[k, k+d]^n, \tag{2.1}$$

   for some $d \in \mathbb{N}$, which in particular implies $|\sigma_\lambda| \lesssim 2^{-n|\lambda|}$, where $|U|$ denotes the Lebesgue measure of $U \subset \mathbb{R}^n$.

2. *The system $\Phi_j := \{\phi_\lambda : \ \lambda \in \Delta_j\}$ is a Riesz basis of S_j with Riesz bounds c_Φ and C_Φ independent of j, i.e.,*

$$c_\Phi \|\boldsymbol{c}_j\|_{\ell_2(\Delta_j)} \leq \Big\| \sum_{\lambda \in \Delta_j} c_\lambda \phi_\lambda \Big\|_{L_2} \leq C_\Phi \|\boldsymbol{c}_j\|_{\ell_2(\Delta_j)}, \quad \boldsymbol{c}_j = (c_\lambda)_{\lambda \in \Delta_j}.$$

3. *The system Φ_j is* locally linear independent, *i.e., if for any open set $U \subset \Omega$*

$$\sum_{\lambda \in \Delta_j} c_\lambda \, \phi_\lambda \Big|_U = 0,$$

   then $c_\lambda = 0$ for all $\lambda \in \Delta_j$ with $|\sigma_\lambda \cap U| \neq 0$.

4. *The system is* refinable, *i.e., there exist coefficients $a_{\lambda,\mu} \in \mathbb{R}$ such that*

$$\phi_\lambda = \sum_{\mu \in \Delta_{|\lambda|+1}} a_{\lambda,\mu} \, \phi_\mu. \tag{2.2}$$

5. *The system Φ_j is* exact of order $m \in \mathbb{N}$, *i.e., polynomials up to order m (degree $m-1$) can be represented exactly by Φ_j, i.e. $\Pi_{m-1} \subset S_j$. In particular, we assume that $2^{-nj/2}\Phi_j$ forms a* partition of unity, *i.e.,*

$$2^{-nj/2} \sum_{\lambda \in \Delta_j} \phi_\lambda = 1. \tag{2.3}$$

The local linear independence implies that if $\sigma_{(j+1,\ell)} \not\subset \sigma_{(j,k)}$ then $a_{(j,k),(j+1,\ell)} = 0$, i.e., for any λ only $(d+1)^n$ summands in the refinement equation (2.2) can be non-zero.

Since Φ_j is a Riesz basis of S_j there exists a dual basis $\{\tilde{\phi}_\lambda,\ \lambda \in \Delta_j\} \subset S_j$, which is *uniquely determined* by the conditions

$$\tilde{\phi}_{(j,k)} \in S_j \quad \text{and} \quad (\phi_\lambda, \tilde{\phi}_\mu)_{L_2} = \delta_{\lambda,\mu},\ \lambda, \mu \in \Delta_j. \tag{2.4}$$

In general, the dual scaling functions $\tilde{\phi}_\lambda$ do *not* have local support, but are supported in the entire domain Ω.

The wavelet space W_j is the *orthogonal complement* of S_j in S_{j+1}, i.e.,

$$S_{j+1} = S_j \oplus W_j \quad \text{and} \quad S_j \perp W_j.$$

Obviously, the spaces W_j are mutually orthogonal. By choosing orthonormal bases for the W_j one can construct an orthonormal wavelet basis for $L^2(\Omega)$. However, the elements of such a basis will have global support for most choices of the MRA $\mathcal{S}$. This problem can often be circumvented by demanding only Riesz bases for the wavelet spaces W_j. In the sequel, we will assume that we have wavelets ψ_λ available which satisfy the following assumptions

Assumption 2.2 *Let $\Psi := \{\psi_\lambda : \lambda \in \nabla\}$ with a suitable index set $\nabla \subset (-1 \cup \mathbb{N}_0) \times \mathbb{Z}^n \times \{0,1\}^n$ ($\#\nabla_j = \#\Delta_{j+1} - \#\Delta_j \sim 2^{nj}$) satisfy the following properties:*

1. *$W_{-1} := S_0$, $\Psi_{-1} := \{\psi_\lambda :\ \lambda \in \nabla_{-1}\} = \Phi_0$.*
2. *The system $\Psi_j := \{\psi_\lambda :\ \lambda \in \nabla_j\}$, $j \in -1 \cup \mathbb{N}_0$ is a Riesz basis of W_j with Riesz bounds c_Ψ and C_Ψ independent of j, i.e.,*

$$c_\Psi \|\boldsymbol{c}_j\|_{\ell_2(\nabla_j)} \le \Big\| \sum_{\lambda \in \nabla_j} c_\lambda \psi_\lambda \Big\|_{L_2} \le C_\Psi \|\boldsymbol{c}_j\|_{\ell_2(\nabla)}, \quad \boldsymbol{c}_j = (c_\nabla)_{\nabla \in \Delta_j}.$$

3. *The functions are* locally supported, *i.e.,*

$$\omega_{(j,k,e)} := \operatorname{supp} \psi_{(j,k,e)} \subset 2^{-j}(k + \omega) \tag{2.5}$$

for some $\omega \in \mathbb{R}^n$, which in particular implies $|\omega_\lambda| \lesssim 2^{-n|\lambda|}$.

Since $W_j \subset S_{j+1}$ we know that there exist coefficients $b_{\lambda,\mu}$ so that

$$\psi_\lambda = \sum_{\mu \in \Delta_{|\lambda|+1}:\ \sigma_\mu \subset \omega_\lambda} b_{\lambda,\mu}\, \phi_\mu. \tag{2.6}$$

From the local support of ψ_λ we deduce that the number of terms in the above sum is bounded by a constant independent of λ. Again, there is a dual basis $\{\tilde{\psi}_\lambda,\ \lambda \in \Delta_j\} \subset W_j$, which is uniquely determined by $\tilde{\psi}_\lambda \in W_j$ and $(\psi_\lambda, \tilde{\psi}_\mu)_{L_2} = \delta_{\lambda,\mu}$, $\lambda, \mu \in \Delta_j$. Usually, also $\tilde{\psi}_\lambda$ does not have local support.

Since $\bigcup_j S_j$ is dense in $L^2(\Omega)$ we know that Ψ is complete. Using the orthogonality between the levels it is easy to show that Ψ is a Riesz basis of $L^2(\Omega)$ with Riesz bounds

c_Ψ and C_Ψ. Usually it is easy to determine Riesz bounds for a single level, but without semiorthogonality it is often impossible to determine exact Riesz bounds for Ψ. The dual basis of Ψ is obviously given by $\widetilde{\Psi} = \{\tilde{\psi}_\lambda : \lambda \in \nabla\}$.

Frequently used examples of semiorthogonal wavelets are the spline wavelets for $L_2(\mathbb{R})$ of Chui and Wang [5, 7, 8] and their modification to $L_2([0,1])$ by Chui and Quak [6]. The construction of semiorthogonal spline wavelets is motivated by the fact that there are no compactly supported orthogonal spline wavelets (except for spline order 1) and for the biorthogonal spline wavelets from [11, 16] the dual basis functions are not splines. Further studies of semiorthogonal wavelets can be found in [3, 4, 20, 21, 23, 24, 22]

In Table 1, condition numbers for semiorthogonal spline wavelets and scaling functions are listed. For the interval, we determined these numbers only up to level 10, but the behavior of the values for increasing j gives strong evidence that these values are already close to the condition numbers $\rho_\Phi^{[0,1]}$ and $\rho_\Psi^{[0,1]}$. Furthermore, we want to mention that we have L_2-normalized these wavelets and scaling functions (in contrast to [6]).

m	$\rho_\Phi^{\mathbb{R}}$	$\rho_\Psi^{\mathbb{R}}$	$\rho_{\Phi_j}^{[0,1]}$	$\rho_{\Psi_j}^{[0,1]}$
2	3	2.25	3.	2.24998
3	7.5	3.52972	7.57288	3.52968
4	18.5294	5.9035	19.2725	6.03137
5	45.7258	10.4028	49.8107	11.6751
6	112.826	18.6978	130.416	21.1974
7	278.387	33.8866	345.046	—
8	686.891	61.6394	920.713	—

Table 1: Condition numbers for scaling functions and wavelets of Chui and Wang ($\rho_\Phi^{\mathbb{R}}$ and $\rho_\Psi^{\mathbb{R}}$) for $L_2(\mathbb{R})$, and of Chui and Quak ($\rho_{\Phi_j}^{[0,1]}$ and $\rho_{\Psi_j}^{[0,1]}$) for $L_2([0,1])$ and $j \le 10$.

2.2 Adaptive Wavelet-Galerkin Methods

We briefly review the main facts of adaptive Wavelet-Galerkin methods that we will need in the sequel. For simplicity, we focus on a particular example, namely the operator equation

$$-\Delta u + f(u) = g \quad \text{on } \Omega$$

with a given (nonlinear) function $f : \mathbb{R} \to \mathbb{R}$. Its variational formulation amounts finding $u \in H_0^1(\Omega)$ such that

$$a(u,v) + (f(u),v)_{L_2} = (g,v)_{L_2} \quad \text{for all } v \in H_0^1(\Omega),$$

where the bilinear form is given by $a(u,v) := (\nabla u, \nabla v)_{L_2}$. Finally, in operator form, the problem reads

$$Au + F(u) = G,$$

where the operators $A, F : H_0^1(\Omega) \to H^{-1}(\Omega)$ are defined by

$$\langle Au, v\rangle = a(u,v), \quad \langle F(u), v\rangle = (f(u),v)_{L_2}, \qquad v \in H_0^1(\Omega),$$

where $\langle \cdot, \cdot \rangle$ denotes the standard duality pairing.

Using an adaptive wavelet method based upon a wavelet basis Ψ of $H_0^1(\Omega)$ then requires the computation (or at least, the approximation) of terms like $(Au_\Lambda, \psi_\mu)_{L_2}$ or $(F(u_\Lambda), \psi_\mu)_{L_2}$ for a given (finite) linear approximation $u_\Lambda = \sum_{\lambda \in \Lambda} d_\lambda \, \psi_\lambda$.

It is known, that a sufficiently good finite approximation of Au_Λ can be computed with cost $\mathcal{O}(\#\Lambda)$, if the entries $a(\psi_\lambda, \psi_\mu)$ of the stiffness matrix can be computed at unit cost. This is the case if the wavelets ψ_λ are spline functions of compact support (cf. [9]). Since the dual wavelets are not needed here, the global support of these functions is no obstruction at this point.

For the nonlinear function $f(u)$ the problem is solved if we have a suitable finite approximation in terms of the dual system, i.e.,

$$f(u_\Lambda) \approx \sum_{\lambda \in \nabla} d_\lambda \, \tilde{\psi}_\lambda.$$

From the Riesz stability we know that the coefficients d_λ, $\lambda \in \nabla$, are a good approximation of $(f(u_\Lambda), \psi_\lambda)_{L_2}$, $\lambda \in \nabla$. Thus, we seek an approximation of $f(u_\Lambda)$ w.r.t. the dual system.

2.3 The DSX-Algorithm

Now we briefly describe the main ingredients of the algorithm introduced by Dahmen, Schneider and Xu in [17] for evaluating nonlinear functions of wavelet expansions which we will briefly refer to as DSX-Algorithm. This algorithm consists of four main steps which we will detail in the sequel. We always start with a known approximation u_Λ^ε of a function $u \in L_2(\Omega)$, i.e.,

$$u_\Lambda^\varepsilon = \sum_{\lambda \in \Lambda^{u,\varepsilon}} c_\lambda \, \psi_\lambda, \qquad \|u - u_\Lambda^\varepsilon\|_{L_2} \leq \varepsilon.$$

The goal is to determine an approximation g of $f(u)$ of similar accuracy, i.e.,

$$g = \sum_{\lambda \in \hat{\Lambda}} d_\lambda \, \tilde{\psi}_\lambda, \qquad \|g - f(u)\|_{L_2} \lesssim \varepsilon, \tag{2.7}$$

where the index set Λ is as small as possible. The core algorithm reads as follows.

Algorithm 2.3 *Input:* u_Λ^ε, $\Lambda^{u,\varepsilon}$

1. Prediction*: Given $\Lambda^{u,\varepsilon}$, predict the set $\hat{\Lambda}$ of significant indices in* (2.7)*;*
2. Reconstruction*: Determine a local scaling function representation of u_Λ^ε, which permits a fast computation of function values.*
3. Quasi-Interpolation*: Compute a quasi-interpolant $g = P_\Lambda f(u_{\Lambda^{u,\varepsilon}})$ in local scaling function representation, using function values of $f(u_{\Lambda^{u,\varepsilon}})$;*
4. Decomposition*: Compute the wavelet coefficients d_λ, $\lambda \in \hat{\Lambda}$ of g.*

Let us now briefly give some background information on these steps.

2.3.1 Prediction

The prediction is based upon a local error function E which is defined on subsets of Ω. Let E be such an error function satisfying

$$E(\square') \le E(\square'') \quad \text{for } \square' \subseteq \square'', \qquad E(\square') \to 0 \text{ as } \operatorname{diam}(\square') \to 0 \tag{2.8}$$

for $\square', \square'' \subset \Omega$. We are particularly interested in the dyadic cubes

$$\square_{(j,k)} := 2^{-j}(k + \square).$$

The goal is to find for a given $\varepsilon > 0$ a disjoint partition $\{\square_\lambda : \lambda \in \Lambda_\varepsilon \subset \Delta\}$ of the unit cube $\square$ in the sense that

$$\square = \bigcup_{\lambda \in \Lambda_\varepsilon} \square_\lambda \qquad \text{and} \qquad \left|\square_\lambda \cap \square_\mu\right| = 0, \ \lambda, \mu \in \Lambda_\varepsilon, \ \lambda \ne \mu, \tag{2.9}$$

so that

$$\varepsilon_p(\Lambda_\varepsilon) := \Big(\sum_{\lambda \in \Lambda_\varepsilon} E(\square_\lambda)^p \Big)^{\frac{1}{p}} \lesssim \varepsilon. \tag{2.10}$$

We will call an index set $\Lambda_\varepsilon \in \Delta$, which defines a partition by (2.9), a *partition set*.

The analysis in [17] is mainly concerned with two particular cases, namely

$$E(\square_\lambda) = E(\square_\lambda, g) := \inf_{P \in \Pi_{m-1}} \|g - P\|_{L_p(\square_\lambda^*)}, \tag{2.11}$$

and

$$E^*(\square_\lambda, g) := 2^{|\lambda|(\frac{n}{\tau} - \frac{n}{p} - s)} |g|_{B_q^s(L_\tau(\square_\lambda^*))}, \qquad \tau > (\tfrac{s}{n} + \tfrac{1}{p})^{-1}. \tag{2.12}$$

where for any λ the cube $\square_\lambda^*$ satisfies

$$\operatorname{diam} \square_\lambda^* \lesssim 2^{-|\lambda|}, \qquad \square_\lambda \subset \square_\lambda^*. \tag{2.13}$$

In [17], it is shown that for sufficiently smooth functions g a partition can be found so that (2.10) is satisfied for E from (2.11) or (2.12). In particular, if g is contained in the Besov space $B_q^s(L_\tau(\square))$, $\tau > (\frac{s}{n} + \frac{1}{p})^{-1}$, $s \le m$, then for each $N \in \mathbb{N}$ there is a partition set Λ with $\#\Lambda = N$ such that for E

$$\varepsilon_p(\Lambda) \le C \, N^{-\frac{s}{n}} |g|_{B_q^s(L^\tau(\square))}$$

with C independent of g and N (cf. [17, Lemmata 3.6 and 3.13]). Having determined the partition set Λ the next ingredient is to find a suitable mapping $P_\Lambda : L_p \to \widetilde{S}_{\hat{\Lambda}} := \operatorname{span}\{\tilde{\psi}_\lambda : \lambda \in \hat{\Lambda}\}$ with the same order of approximation.

Definition 2.4 *For any partition set Λ, let P_Λ denote a mapping from L_p into $\widetilde{S}_{\hat{\Lambda}}$. P_Λ is called* E-admissible, *if*

$$\|P_\Lambda g - g\|_{L_p(\square_\lambda)} \lesssim E(\square_\lambda), \quad \lambda \in \Lambda,$$

and $\#\hat{\Lambda} \lesssim \#\Lambda$.

Remark 2.5 *It is easy to verify that the index set $\hat{\Lambda}$ will have tree structure. The approach in [17] is completely based on (M-graded) tree-like index sets. For our purpose this restriction is not explicitly needed, but may be contained implicitly.*

In conclusion, if we have a suitable method for choosing Λ_ε satisfying (2.10) and have an E-admissible operator P_Λ we know that

$$\|P_\Lambda g - g\|_{L_p(\square_\lambda)} \lesssim \#\hat{\Lambda}^{-\frac{s}{n}} |g|_{B^s_q(L^\tau(\square))},$$

which is almost the best possible N-term approximation order (where $\tau = q = (\frac{s}{n} + \frac{1}{p})^{-1}$). For $g = f(u)$, $f \in C^m(\mathbb{R})$, the prediction of Λ is based on the local regularity of g, which in turn can be estimated by the local regularity of u. The local regularity of u can be predicted from the index set $\Lambda^{u,\varepsilon}$.

Finally, one obtains the following result ([17, Theorem 4.2])

Theorem 2.6 *Assume that $f \in C^m(\mathbb{R})$, that P_Λ is E-admissible for E defined by* (2.11) *or by* (2.12)*, and that $u \in B^s_q(L^\tau(\square))$, $\tau > (\frac{s}{n} + \frac{1}{p})^{-1}$, $s \le m$. For any $\varepsilon > 0$ and $\Lambda^{u,\varepsilon}$ described above there exists a set Λ so that $\Lambda^{u,\varepsilon} \subset \hat{\Lambda}$ such that*

$$\|f(u) - P_\Lambda f(u)\|_{L_p} \lesssim \varepsilon, \qquad \#\Lambda \lesssim \varepsilon^{-n/s},$$

with constants that are independent of ε and u.

Roughly speaking, we may assume that such a prediction scheme is available that determines Λ. A simple example of such a scheme is presented in [17]. It remains to find an E-admissible operator P_Λ, which will be applied in the third step of Algorithm 2.3. For the case of semiorthogonal wavelets, we will present such an operator in Section 4.

2.3.2 Reconstruction

The input u^ε_Λ is given in terms of its wavelet expansion. For the next steps in the algorithm, however, a representation with respect to scaling functions on several levels is more convenient. This is often called *local scaling function representation*. One starts with

$$u^\varepsilon_\Lambda = \sum_{\lambda \in \Lambda^{u,\varepsilon}} d_\lambda \, \psi_\lambda$$

and we aim at the alternative representation

$$u^\varepsilon_\Lambda = \sum_{\lambda \in \mathcal{I}} c_\lambda \, \phi_\lambda, \tag{2.14}$$

with some subset $\mathcal{I} \subset \Delta$. Note that such a representation is not unique. For computational reasons one wants to find a representation which satisfies the following two conditions.

1. The number of nonzero coefficients should be 'small', i.e., $\#\mathcal{I} \lesssim \#\Lambda^{u,\varepsilon}$.

2. To compute the function value $u_\Lambda^\varepsilon(x)$, $x \in \Omega$, only a fixed number of terms in (2.14) shall be needed, i.e.,
$$\#\big\{\lambda \in \mathcal{I};\ x \in \sigma_\lambda\big\} \lesssim 1$$
with a constant independent of x, Λ, and u_Λ^ε.

The first condition makes sure that the representation can be computed with $\mathcal{O}(\#\Lambda)$ operations using the refinement equations (2.2) and (2.6) (if primal scaling functions and wavelets have local support). The second condition permits the computation of function values at unit cost.

An example of such a representation is given in [17].

2.3.3 Quasi-Interpolation

Given the predicted set Λ and the local scaling function representation of $u = u_\Lambda$, a computable scheme is needed in order to obtain an E-admissible approximation in terms of dual scaling functions according to Theorem 2.6. This is done in [17] by a particular approximate evaluation of wavelet coefficients of compositions. At this point, it makes a difference if one works with biorthogonal or semiorthogonal wavelets. Hence, we concentrate on this issue in Section 3 below where we construct an adaptive quasi-interpolation scheme for semiorthogonal spline wavelets.

2.3.4 Decomposition

The quasi-interpolation results in an approximation of $f(u)$ again with respect to a local scaling function representation in terms of the *dual* scaling functions. In order to obtain the desired approximation of $f(u)$ in terms of the dual wavelet basis, we need a decomposition algorithm. Since the decomposition algorithm for *dual* wavelets needs the refinement coefficients of the *primal* basis functions, local support of the *primal* scaling functions is sufficient in order to have a fast decomposition algorithm (cf. [17]).

3 Quasi-interpolation schemes

In this section, we consider quasi-interpolation schemes in general (i.e., non-adaptive) and give examples in terms of splines. These schemes will be used in Section 4 to construct a corresponding adaptive quasi-interpolant. Let us start by defining such schemes.

Definition 3.1 (a) *A linear operator $L : L_p(\Omega) \to L_p(\Omega)$ is called* local *if there exists a compact set M so that $f\big|_{x+M} = 0$ implies $Lf(x) = 0$ for all $x \in \Omega$.*

(b) *A linear functional $c : L_p(\Omega) \to \mathbb{R}$ is called* local *if there exists a compact set M so that $f\big|_M = 0$ implies $c(f) = 0$. In this case we call M the* support *of c.*

Definition 3.2 *We say the linear operators $Q_j : L_p(\Omega) \to S_j$, $j \in \mathbb{N}_0$, form a* quasi-interpolation scheme*, if the following conditions are satisfied:*

(a) *Q_j preservers polynomials of degree less then $m \in \mathbb{N}$, i.e., $Q_j P = P$ for $P \in \Pi_{m-1}$.*

(b) *The operators Q_j are local in the sense of Definition 3.1 and uniformly bounded, i.e., there is a constant C_Q and a compact set K so that for each $U \subseteq \Omega$*

$$\|Q_j f\|_{L_p(U)} \leq C_Q \, \|f\|_{L_p(U+2^{-j}K)}. \tag{3.1}$$

The locality in Definition 3.2 (b) in particular implies that for any fixed $x \in \Omega$ the value $Q_j f(x)$ depends only on values $f(y)$ with $|x-y| \lesssim 2^{-j}$. In particular, the operators Q_j satisfy (3.1) if and only if they are uniformly bounded, local linear operators (with the compact set M in Definition 3.1 being $M = 2^{-j}K$).

Since Q_j is a linear operator onto $S_j = \operatorname{span}\{\phi_\lambda : \lambda \in \Delta_j\}$, there exist (uniquely determined) functionals q_λ so that

$$Q_j f = \sum_{\lambda \in \Delta_j} q_\lambda(f)\phi_\lambda. \tag{3.2}$$

For computational purposes it is usually important that the functionals are local. In particular, if the $q_\lambda = q_{(j,k)}$ are uniformly bounded, are supported on $2^{-j}k + K'$ and if $\operatorname{supp}\phi_{(j,k)} \subset 2^{-j}[k, k+d]^n$, then condition (3.1) is satisfied with $K = K' + [0,d]$.

It is well-known that quasi-interpolation schemes are well-suited for the approximation of functions. For later use, we formulate the following result.

Lemma 3.3 *If Q_j is a quasi-interpolation scheme, we have for $p \geq 2$*

$$\|f - Q_{|\lambda|} f\|_{L_2(\square_\lambda)} \lesssim 2^{-|\lambda|(\frac{n}{2}-\frac{n}{p})} \inf_{P \in \Pi_{m-1}} \|f - P\|_{L_p(\square_\lambda^*)},$$

where $\square_\lambda^* := \square_\lambda + 2^{-|\lambda|}K$. In particular, we have $\square_\lambda \subset \square_\lambda^*$ and $\operatorname{diam}\square_\lambda^* = (1 + \operatorname{diam} K)2^{-|\lambda|}$.

Proof. By definition, we have for any $P \in \Pi_{m-1}$ and $U \subset \mathbb{R}^n$

$$\begin{aligned} \|f - Q_{|\lambda|} f\|_{L_p(U)} &\leq \|f - P\|_{L_p(U)} + \|Q_{|\lambda|}(f-P)\|_{L_p(U)} \\ &\leq (1 + C_Q)\|f - P\|_{L_p(U+2^{-|\lambda|}K)}. \end{aligned}$$

Hence,

$$\|f - Q_{|\lambda|} f\|_{L_p(U)} \leq (1 + C_Q) \inf_{P \in \Pi_{m-1}} \|f - P\|_{L_p(U+2^{-|\lambda|}K)}$$

In particular, for $U = \square_\lambda$ and $p \geq 2$ we obtain by Hölder's inequality

$$\begin{aligned} \|f - Q_{|\lambda|} f\|_{L_2(\square_\lambda)} &\leq 2^{-n|\lambda|(\frac{1}{2}-\frac{1}{p})} \, \|f - Q_{|\lambda|} f\|_{L_p(\square_\lambda)} \\ &\leq 2^{-n|\lambda|(\frac{1}{2}-\frac{1}{p})} \, (1 + C_Q) \inf_{p \in \Pi_{m-1}} \|f - p\|_{L_p(\square_\lambda^*)}, \end{aligned}$$

where $\square_\lambda^* := \square_\lambda + 2^{-|\lambda|}K$, which proves the claim. $\square$

Let us give some examples that will be important in the sequel.

Example 3.4 (The cardinal case.) Let $\Omega = \mathbb{R}$ and $\phi_{(j,k)} = 2^{\frac{j}{2}}\phi(2^j \cdot -k)$, $j,k \in \mathbb{Z}$, for a given refinable function ϕ. Usually, one considers functionals $q_{(j,k)}$ defined by $q_{(j,k)}(f) = 2^{-\frac{j}{2}} q\big(f(2^{-j} \cdot +k)\big)$ for some $q : L_p \to \mathbb{R}$. Hence, we obtain $Q_j f(x) = Q\big(f(2^{-j}\cdot)\big)(2^j x)$, where

$$Qf(x) = \sum_{k\in\mathbb{Z}} q\big(f(\cdot + k)\big)\phi(x-k). \tag{3.3}$$

If Q is a bounded, linear local operator which preserves polynomials, then it follows immediately that Q_j is a quasi-interpolation scheme in the sense of Definition 3.2 with $C_Q = \|Q\|_{L_p(\mathbb{R})\to L_p(\mathbb{R})}$.

One often is interested in the case $p = \infty$ ($f \in C(\mathbb{R})$), which is of particular interest because one can define q as a weighted sum of point values $q(f) = \sum_{\ell=-L}^{L} w_\ell \; f(\ell + \tau)$ with some suitable $\tau \in \mathbb{R}$ and suitable weights $w_\ell \in \mathbb{R}$.

Example 3.5 (Cardinal splines.) Let $\phi = N_d$ be the cardinal B-spline of order d. Following a result of Zheludev [25] one can choose the specific functional

$$q(f) := \sum_{\ell=0}^{\lceil \frac{d-1}{2} \rceil} (-1)^\ell \beta_{d,\ell} \; \tilde{\Delta}_1^{2\ell} f(\tfrac{d}{2}) \tag{3.4}$$

to obtain quasi-interpolation operators, which are exact for polynomials of degree $d-1$. Here, $\tilde{\Delta}_h^\ell$ denotes the symmetric difference defined by $\tilde{\Delta}_h^1 f(x) := f(x+\frac{h}{2}) - f(x-\frac{h}{2})$ and $\tilde{\Delta}_h^{\ell+1} f(x) = \tilde{\Delta}_h^1 \tilde{\Delta}_h^\ell f(x)$. The coefficients $\beta_{d,\ell}$ are defined via the generating function

$$\left(\frac{2\arcsin\frac{t}{2}}{t}\right)^d = \sum_{\ell=0}^{\infty} \beta_{d,\ell} t^{2\ell}. \tag{3.5}$$

For completeness we give a proof of the polynomial exactness of the quasi-interpolant obtained from (3.4).

Lemma 3.6 *The cardinal spline operator Q defined in (3.3) with the functional q from (3.4) is exact for polynomials of degree less than d.*

Proof. Using the centered B-spline $M_d = N_d(\cdot + \frac{d}{2})$ we have to show that

$$Q^* f(x) := \sum_{k\in\mathbb{Z}} \Big(\sum_{\ell=0^d}^{\lceil \frac{m-1}{2} \rceil} (-1)^\ell \beta_{d,\ell} \; \tilde{\Delta}_1^{2\ell} f(k) \Big) M_d(x-k)$$

preserves polynomials of degree less than d. Since $Qf(x) = Q^*\big(f(\cdot + \frac{d}{2})\big)(x - \frac{d}{2})$ it then follows that Q is exact of the same order.

Note that $\tilde{\Delta}_1^2 f(x) = f(x-1) - 2f(x) + f(x+1)$, and hence

$$\sum_{k\in\mathbb{Z}} \tilde{\Delta}_1^{2\ell} f(k) \; M_d(x-k) = \sum_{k\in\mathbb{Z}} \tilde{\Delta}_1^{2(\ell-1)} f(k) \; \tilde{\Delta}_1^2 M_d(x-k) = \sum_{k\in\mathbb{Z}} f(k) \; \tilde{\Delta}_1^{2\ell} M_d(x-k).$$

Changing the order of summation, we obtain

$$\begin{aligned} Q^*f(x) &= \sum_{\ell=0}^{\lceil\frac{d-1}{2}\rceil} (-1)^\ell \beta_{d,\ell} \sum_{k\in\mathbb{Z}} \tilde{\Delta}_1^{2\ell} f(k)\; M_d(x-k) \\ &= \sum_{\ell=0}^{\lceil\frac{d-1}{2}\rceil} (-1)^\ell \beta_{d,\ell} \sum_{k\in\mathbb{Z}} f(k)\; \tilde{\Delta}_1^{2\ell} M_d(x-k). \end{aligned}$$

For B-splines we have the well-known relation $\tilde{\Delta}_1^\ell M_d(x) = M_{d+\ell}^{(\ell)}(x)$ between differences and derivatives (see, e.g., [5, Theorem 4.3 (vii)]). Hence, we obtain the representation

$$Q^*f = \sum_{k\in\mathbb{Z}} f(k)\vartheta(x-k),$$

where

$$\vartheta(x) := \sum_{\ell=0}^{\lceil\frac{d-1}{2}\rceil} (-1)^\ell \beta_{d,\ell} M_{d+2\ell}^{(2\ell)}(x).$$

Now, we consider the monomials $p_\nu(x) = (-2\pi i x)^\nu$. For any compactly supported function $g \in L^1(\mathbb{R})$ it is well-know that

$$\int_{\mathbb{R}} p_\nu(x)\, g(x)\, dx = \hat{g}^{(\nu)}(0),$$

where we denote the Fourier transform of a function $f : \mathbb{R}^n \to \mathbb{R}$ by

$$\hat{f}(\xi) := \int_{\mathbb{R}^n} f(x) e^{2\pi i x\cdot\xi}\, dx, \qquad \xi \in \mathbb{R}^n.$$

On the other hand

$$\begin{aligned} \int_{\mathbb{R}} Q^*p_\nu(x)\, g(x)\, dx &= \sum_{k\in\mathbb{Z}} (-2\pi i k)^\nu \int_{\mathbb{R}} e^{-2\pi i k\xi}\; \hat{\vartheta}(\tau)\, \hat{g}(\tau)\, d\tau \\ &= \sum_{k\in\mathbb{Z}} (-2\pi i k)^\nu \int_0^1 e^{-2\pi i k\tau} \sum_{n\in\mathbb{Z}} \hat{\vartheta}(\tau+k)\, \hat{g}(\tau+k)\, d\tau \\ &= \sum_{k\in\mathbb{Z}} (-2\pi i k)^\nu \int_0^1 e^{-2\pi i k\tau} \sum_{n\in\mathbb{Z}} \hat{\vartheta}(\tau+k)\, \hat{g}(\tau+k)\, d\tau\; e^{2\pi i k\xi}\Big|_{\xi=0} \\ &= \frac{d^\nu}{d\xi^\nu} \sum_{n\in\mathbb{Z}} \hat{\vartheta}(\xi+k)\, \hat{g}(\xi+k)\Big|_{\xi=0}. \end{aligned}$$

From $\hat{M}_d(\xi) = \mathrm{sinc}^d(\xi)$ and we conclude

$$\begin{aligned} \hat{\vartheta}(\xi) &= \sum_{\ell=0}^{\lceil\frac{d-1}{2}\rceil} \beta_{d,\ell}\, (2\pi\xi)^{2\ell}\, \mathrm{sinc}^{d+2\ell}(\xi) \\ &= \mathrm{sinc}^d(\xi) \sum_{\ell=0}^{\lceil\frac{d-1}{2}\rceil} \beta_{d,\ell}\, (2)^{2\ell}\, \sin^{2\ell}(\pi\xi). \end{aligned}$$

Applying (3.5), we obtain for $|\xi| < \frac{1}{2}$

$$\hat{\vartheta}(\xi) = \operatorname{sinc}^d(\xi)\left(\operatorname{sinc}^{-d}(\xi) - \sum_{\ell=\lceil\frac{d+1}{2}\rceil}^{\infty} \beta_{d,\ell}(2)^{2\ell}\sin^{2\ell}(\pi\xi)\right) = 1 + \mathcal{O}(\xi^d), \quad \xi \to 0.$$

For $\xi \to 0$ and $k \in \mathbb{Z} \setminus \{0\}$ we have $\operatorname{sinc}^d(k+\xi) = \mathcal{O}(\xi^d)$ and hence $\hat{\vartheta}(\xi + k) = \mathcal{O}(\xi^d)$. Thus,

$$\int_{\mathbb{R}} Q^* p_\nu(x)\, g(x)\, dx = \hat{g}^{(\nu)}(0) = \int_{\mathbb{R}} p_\nu(x)\, g(x)\, dx, \quad \nu < d,$$

for all $g \in L_2(\mathbb{R})$, i.e., $Q^* p_\nu = p_\nu$. Finally, note that $\{p_\nu : \ \nu = 0, \ldots, d-1\}$ is a basis for Π_{d-1}, which completes the proof. $\square$

Let us conclude this example by noting that the functional q defined in (3.4) can be expanded in terms of point values of f in the following form

$$q(f) = \begin{cases} \sum_{\ell=1}^{d-1} \gamma_{d,\ell}\, f(\ell), & \text{if } d \text{ is even,} \\ \sum_{\ell=1}^{d} \gamma_{d,\ell}\, f(\ell - \frac{1}{2}), & \text{if } d \text{ is odd,} \end{cases}$$

where the weights $\gamma_{d,\ell}$ can be determined explicitly from (3.4). The values for $d \le 6$ are given in Table 2.

d	$\gamma_{d,\ell}$	C_Q
1	1	1
2	1	1
3	$-\frac{1}{8}, \frac{5}{4}, -\frac{1}{8}$	$\frac{3}{2}$
4	$-\frac{1}{6}, \frac{4}{3}, -\frac{1}{6}$	$\frac{5}{3}$
5	$\frac{47}{1152}, -\frac{107}{288}, \frac{319}{192}, -\frac{107}{288}, \frac{47}{1152}$	$\frac{179}{72}$
6	$\frac{13}{240}, -\frac{7}{15}, \frac{73}{40}, -\frac{7}{15}, \frac{13}{240}$	$\frac{43}{15}$

Table 2: Coefficients $\gamma_{d,\ell}$.

One easily checks that in this case the compact set K in Definition 3.2 can be chosen as $K = [1-d, d-1]$ for even d and $K = [\frac{1}{2} - d, d - \frac{1}{2}]$ for odd d.

Example 3.7 (Splines on the unit interval.) Here, we we consider splines with uniformly distributed knots as they are used for the construction of wavelets on the interval [6, 16]. Let the level j be fixed. We assume that the basis $\{\phi_{(j,k)} : k = 1-d, \ldots, 2^j - 1\}$ of S_j consists of the 'interior' B-splines $\phi_{(j,k)} = N_d(2^j \cdot - k)$, $k \in I_j^{\text{int}} := \{0, \ldots, 2^j - d\}$, as well as left and right boundary splines $\phi_{j,k}$, $k \in I_j^{\text{l}} := \{1-d, \ldots, -1\}$ and $k \in I_j^{\text{r}} := \{2^j + 1 - d, \ldots, 2^j - 1\}$. The boundary functions are restrictions of linear combinations of functions in $\operatorname{span}\{N_d(2^j \cdot - k) : k \in I_j^{\text{l}}\}$ and $\operatorname{span}\{N_d(2^j \cdot - k) : k \in I_j^{\text{r}}\}$, respectively.

Since $\Pi_{d-1}([0,1]) \subset \operatorname{span}\{\phi_{j,k} : k \in I_j\}$, $I_j := I_j^{\mathrm{l}} \cup I_j^{\mathrm{int}} \cup I_j^{\mathrm{r}}$, we conclude that for each polynomial $p \in \Pi_{d-1}([0,1])$ there exist uniquely determined coefficients $(c_k)_{k\in I_j}$ such that $p = \sum_{k\in I_j} c_k(p)\phi_{(j,k)}$. Obviously, for the coefficients of the interior B-splines we can use the functional q from (3.4) explained above, i.e., we set $q_{(j,k)}(f) := q\big(f\big(2^{-j}(\cdot + k)\big)\big)$, $k \in I_j^{\mathrm{int}}$.

Now, consider a fixed $k \in I_j^{\mathrm{l}} \cup I_j^{\mathrm{r}}$. For the monomials $p_\nu(x) = x^\nu$ we can determine the coefficients $c_k(p_\nu)$ as follows. We choose knots $x_{k,\ell} \in \sigma_{(j,k)}$, $\ell = 0, \dots, d-1$, which are mutually different. The linear system of equations

$$\sum_{\ell=0}^{d-1} w_{k,\ell}\, x_{k,\ell}^\nu = c_k(p_\nu), \quad \ell = 0, \dots, d-1,$$

has a unique solution, since the corresponding system matrix is a transposed Vandermonde matrix. If we set

$$q_{(j,k)}^\nu(f) := \sum_{\ell=0}^{d-1} w_{k,\ell}\, f(x_{k,\ell}^\nu), \quad k \in I_j^{\mathrm{l}} \cup I_j^{\mathrm{r}},$$

then $q_{j,k}(p) = c_k(p)$ and thus polynomials of order at most $d-1$ are reproduced by the quasi-interpolation operator Q_j defined by these functionals in terms of (3.2). Obviously, this operator is also local.

However, it is not clear at this point how one should choose the knots $x_{k,\ell}$ in order to obtain a particular 'good' quasi-interpolation operator (whatever this may mean in the particular application at hand).

4 Adaptive quasi-interpolation

We now extend the above constructed spline quasi-interpolation schemes to an adaptive version. Even though the technical details differ from [17], the idea behind is quite similar.

Let us assume that we have predicted a partition set Λ satisfying (2.9). We set

$$j_0 = \min_{\lambda\in\Lambda} |\lambda| \quad \text{and} \quad J = \max_{\lambda\in\Lambda} |\lambda|.$$

Denote now by

$$\Omega_j = \bigcup_{\lambda\in\Lambda_j} \square_\lambda$$

the area covered by the cubes from level j. Note that Ω is the union of the disjoint sets Ω_j. Now we define

$$P_j f := \sum_{\lambda\in\mathcal{I}_j} q_\lambda(f)\ \phi_\lambda$$

with

$$\mathcal{I}_j := \{\lambda \in \Delta_j :\ |\sigma_\lambda \cap \Omega_j| \neq 0\}.$$

Note that $\mathcal{I}_j$ is the minimal index set for which $P_j f\big|_{\Omega_j} = Q_j f\big|_{\Omega_j}$.

Starting with $P_{\Lambda,j_0} := P_{j_0}$ we want to define successive updates $P_{\Lambda,j}$ so that finally the operator $P_\Lambda := P_{\Lambda,J}$ is E-admissible. We set

$$P_{\Lambda,j} f := P_{\Lambda,j-1} f + P_j f - \sum_{\lambda \in \mathcal{I}_j} (P_{\Lambda,j-1} f, \tilde{\phi}_\lambda)_{L_2} \, \phi_\lambda. \tag{4.1}$$

Note, that $\tilde{\phi}_\lambda$ in general does not have local support. However, that does not pose a problem here, since $P_{\Lambda,j-1} f \in S_{j-1}$ so that each coefficient $(P_{\Lambda,j-1} f, \tilde{\phi}_\lambda)_{L_2}$ can be determined by a finite number of operations, using the refinement relation (2.2) and duality. For the numerical realization of P_Λ we refer to the next section.

Theorem 4.1 *The operator P_Λ defined in* (4.1) *is an* adaptive quasi-interpolation operator *in the sense that*

(a) *it preserves all polynomials $p \in \Pi_{m-1}$, i.e., $P_\Lambda p = p$,*

(b) *it is local with respect to Λ, i.e., for any open set $U \subset \Omega_j$ we have*

$$P_\Lambda f\Big|_U = P_\Lambda\big(\chi_{U+2^{-j}K} f\big)\Big|_U,$$

where K is the compact set from Definition 3.2.

Proof. (a) We will show by induction that

$$P_{\Lambda,j} p(x) = p(x), \qquad p \in \Pi_{m-1}, \quad x \in \Omega_r, \ r = j_0, \dots, j. \tag{4.2}$$

From the definition of $P_{\Lambda,j_0} := P_{j_0}$ we know that the statement is true for $j = j_0$.

Assume now that (4.2) holds for $j-1$. Since $Q_j p = p$ we have $q_\lambda(p) = (p, \tilde{\phi}_\lambda)_{L_2}$. If $x \in \Omega_r$, $r = j_0, \dots, j-1$, we know by induction hypothesis that $P_{\Lambda,j-1} p(x) = p(x)$. Hence,

$$P_{\Lambda,j} p(x) = p(x) + \sum_{\lambda \in \mathcal{I}_j} q_\lambda(p) \phi_\lambda(x) - \sum_{\lambda \in \mathcal{I}_j} (p, \tilde{\phi}_\lambda)_{L_2} \, \phi_\lambda = p(x), \quad x \in \Omega_r, \ r < j.$$

On the other hand, if $x \in \Omega_j$ we have $P_j p(x) = Q_j p(x) = p(x)$ and since $P_{\Lambda,j-1} f \in S_j$ we have

$$\sum_{\lambda \in \mathcal{I}_j} (P_{\Lambda,j-1} f, \tilde{\phi}_\lambda)_{L_2} \, \phi_\lambda(x) = \sum_{\lambda \in \Delta_j} (P_{\Lambda,j-1} f, \tilde{\phi}_\lambda)_{L_2} \, \phi_\lambda(x) = P_{\Lambda,j-1} f(x), \quad x \in \Omega_j, \tag{4.3}$$

which immediately yields $P_{\Lambda,j} p(x) = p(x)$, $x \in \Omega_j$, and (4.2) is shown. In particular, for $j = J$ we obtain (a).

(b) Let $U \subset \Omega_j$ be open. By (4.3) we know, that $P_{\Lambda,j} f(x) = P_j f(x) = Q_j f(x)$. From the locality of Q_j (see Definition 3.2 (b)) we know for $g := \chi_{U+2^{-j}K} f$ that $Q_j g\big|_U = Q_j f\big|_U$. Hence, $P_{\Lambda,j} g\big|_U = P_{\Lambda,j} f\big|_U$ and

$$\sum_{\lambda \in \Delta_{j+1}} \Big(P_{\Lambda,j} g - P_{\Lambda,j} f, \tilde{\phi}_\lambda\Big)_{L_2} \phi_\lambda\Big|_U = 0.$$

Now, local linear independence implies that $(P_{\Lambda,j}g, \tilde{\phi}_\lambda)_{L_2} = (P_{\Lambda,j}f, \tilde{\phi}_\lambda)_{L_2}$ if $|\sigma_\lambda \cap U| \neq 0$, i.e., the relevant coefficients in (4.1) are equal.

By an analogous argument we conclude $P_{j+1}g\big|_U = P_{j+1}f\big|_U$ from $Q_{j+1}g\big|_U = Q_{j+1}f\big|_U$. Thus, $P_{\Lambda,j+1}g\big|_U = P_{\Lambda,j+1}f\big|_U$. Now, by induction one shows immediately that $P_\Lambda g\big|_U = P_\Lambda f\big|_U$. $\square$

The operator P_Λ is a finite sum of bounded operators and hence it is bounded itself. Unfortunately, $\|P_\Lambda\|$ may depend on Λ, because the number of terms in this sum grows with $J - j_0$. One possibility to obtain a fixed upper bound for $\|P_\Lambda\|$ is to consider only graded (or M-graded) partition sets Λ. Then, one needs locally only a fixed number of upgrades and can therefore prove uniform boundedness. Moreover, for certain classes of scaling functions and quasi-interpolants, one can show uniform boundedness independent of Λ. We restrict ourselves to the case $p = \infty$. This includes the particular relevant case when the functionals $q_{j,k}$ are based on a finite number of function values as in the Examples 3.5 and 3.7.

Theorem 4.2 *Let the quasi-interpolation operators $Q_j : C(\Omega) \to S_j$ satisfy the following conditions:*

(a) *The functionals $q_\lambda : C(\Omega) \to \mathbb{R}$ are bounded by $\|q_\lambda\| \leq C_q 2^{-\frac{n|\lambda|}{2}}$, $\lambda \in \Delta$.*

(b) *The scaling functions and the refinement coefficients from* (2.2) *are non-negative, i.e., $\phi_\lambda(x) \geq 0$ and $a_{\lambda,\mu} \geq 0$.*

Then,

$$\|P_\Lambda\|_{C(\Omega)\to C(\Omega)} \leq C_q.$$

Proof. First, we will show by induction that for any $f \in C(\Omega)$ and $j \geq j_0$ the operator $P_{\Lambda,j}$ can be written in the form

$$P_{\Lambda,j}f = \sum_{\lambda\in\Delta_j} p_\lambda(f)\ \phi_\lambda, \tag{4.4}$$

where $|p_\lambda(f)| \leq C_q 2^{-\frac{n|\lambda|}{2}} \|f\|_\infty$. For $j = 0$ the statement is obviously true with

$$p_{(j_0,k)}(f) = \begin{cases} q_{(j_0,k)}(f), & \text{if } k \in \mathcal{I}_j, \\ 0 & \text{otherwise.} \end{cases}$$

We assume now that the statement is already shown for $j-1$. The refinement equations (2.2) yield

$$P_{\Lambda,j-1}f = \sum_{\lambda\in\Delta_j} c_\lambda\ \phi_\lambda$$

with

$$c_\lambda = \sum_{\mu\in\Delta_{j-1}} p_\mu(f)\ a_{\mu,\lambda}.$$

In particular, from $a_{\mu,\lambda} \geq 0$ we conclude

$$|c_\lambda| \leq C_q 2^{-\frac{n(|\lambda|-1)}{2}} \|f\|_\infty \sum_{\mu \in \Delta_{j-1}} a_{\mu,\lambda}.$$

Since the ϕ_λ form a partition of unity (2.3), we know that

$$\sum_{\mu \in \Delta_{j-1}} a_{\mu,\lambda} = 2^{-\frac{n}{2}}, \quad \lambda \in \Delta_j,$$

i.e., $|c_\lambda| \leq C_q 2^{-\frac{n|\lambda|}{2}} \|f\|_\infty$. Applying this to the definition of $P_{\Lambda,j}$ in (4.1) we obtain the representation (4.4) with

$$p_\lambda(f) = \begin{cases} q_\lambda(f), & \text{if } \lambda \in \mathcal{I}_j, \\ c_\lambda & \text{otherwise.} \end{cases}$$

Now, we obtain with $\phi_\lambda(x) \geq 0$ and (2.3) that

$$|P_{\Lambda,j} f(x)| \leq C_q \|f\|_\infty 2^{-\frac{jn}{2}} \sum_{\lambda \in \Delta_j} \phi_\lambda(x) = C_q.$$

Taking the supremum over x for $j = J$ proves the theorem. $\square$

In particular, the assumptions of Theorem 4.2 are satisfied for B-splines.

Theorem 4.3 *Let the assumptions of Theorem 4.2 be satisfied. Then, one has for $\lambda \in \Lambda$*

$$\|P_\Lambda f - f\|_{L_2(\square_\lambda)} \lesssim 2^{-|\lambda|\frac{n}{2}} \inf_{p \in \Pi_{m-1}} \|f - p\|_{C(\square_\lambda^*)}.$$

Proof. For $\lambda \in \Lambda$ we have $\square_\lambda \subset \Omega_{|\lambda|}$ and hence by Theorem 4.1 (b) and Theorem 4.2

$$\|P_\Lambda f\|_{C(\square_\lambda)} \leq \big\|P_\Lambda\big(\chi_{\square_\lambda^*} f\big)\big\|_{L_p(\Omega)} \leq C_q \|f\|_{C(\square_\lambda^*)},$$

where $\square_\lambda^* = \square_\lambda + 2^{-|\lambda|} K$. Then for any $p \in \Pi_{m-1}$ we have by Theorem 4.1 (a)

$$\begin{aligned} \|f - P_\Lambda f\|_{C(\square_\lambda)} &\leq \|f - p\|_{C(\square_\lambda)} + \|P_\Lambda(f-p)\|_{C(\square_\lambda)} \\ &\leq \big(1 + C_q\big)\|f - p\|_{C(\square_\lambda^*)}. \end{aligned}$$

Hence,

$$\begin{aligned} \|f - P_\Lambda f\|_{L_2(\square_\lambda)} &\leq 2^{-\frac{n}{2}|\lambda|} \|f - P_\Lambda f\|_{L_p(\square_\lambda)} \\ &\leq 2^{-n|\lambda|(\frac{1}{2}-\frac{1}{p})}\big(1 + C_q\big) \inf_{p \in \Pi_{m-1}} \|f - p\|_{L_p(\square_\lambda^*)}. \end{aligned}$$ $\square$

Remark 4.4 *Under the assumption that the partition set Λ is M-graded (for some $M \in \mathbb{N}$) one can show a similar statement for $p \leq \infty$ where the restrictions from Theorem 4.2 are not necessary. The estimate then reads*

$$\|P_\Lambda f - f\|_{L_2(\square_\lambda)} \lesssim 2^{|\lambda|(\frac{n}{p}-\frac{n}{2})} \inf_{p \in \Pi_{m-1}} \|f - Pf\|_{L_p(\square_\lambda^*)}.$$

The proof is similar to [17, Theorem 5.3].

We are now able to show that P_Λ is an E-admissible mapping.

Theorem 4.5 *Let the assumptions of Theorem 4.2 be satisfied and assume that $s \leq m$ and $\tau \geq \frac{n}{s}$ (so that $B^s_q(L_\tau(\Omega))$ is embedded in $C(\Omega)$). Then the mapping P_Λ is E^*-admissible with E^* defined in* (2.12)*, i.e.,*

$$\|P_\Lambda f - f\|_{L_2(\square_\lambda)} \lesssim 2^{|\lambda|(\frac{n}{\tau}-\frac{n}{2}-s)} |f|_{B^s_q(L_\tau(\square^*_\lambda))}, \qquad \lambda \in \Lambda, \tag{4.5}$$

and $\#\hat{\Lambda} \lesssim \#\Lambda$.

Proof. It was proven in [17, Lemma 3.10] that for $\tau > (\frac{s}{n} + \frac{1}{p})^{-1}$ one has

$$\inf_{p\in\Pi_{m-1}} \|f-p\|_{L_p(\square_\lambda)} \lesssim 2^{|\lambda|(\frac{n}{\tau}-\frac{n}{p}-s)} \, |f|_{B^s_q(L_\tau(\square_\lambda))}.$$

Combining this with Theorem 4.3 yields (4.5).

Now, we define

$$\widetilde{\Omega}_j := \bigcup_{\lambda\in\mathcal{I}_j} \sigma_\lambda. \tag{4.6}$$

By the local support of ϕ_λ we conclude that $|\widetilde{\Omega}_j| \lesssim 2^{-nj}\#\Lambda_j$. Furthermore, we choose

$$\hat{\Lambda} := \{\lambda \in \nabla : \ |\omega_\lambda \cap \widetilde{\Omega}_r| = 0, \ r > j\}. \tag{4.7}$$

From (4.6) it follows that $(P_\lambda f, \psi_\lambda)_{L_2} = 0$ for $\lambda \notin \hat{\Lambda}$, i.e., for any f we have $P_\Lambda f \in \widetilde{S}_{\hat{\Lambda}}$. By (2.5) we conclude that

$$\#\hat{\Lambda}_j \lesssim 2^{nj} \sum_{r=j+1}^{J} |\widetilde{\Omega}_r|.$$

Hence,

$$\begin{aligned} \#\hat{\Lambda} \ &\lesssim \ \sum_{j=-1}^{J-1} 2^{nj} \sum_{r=j+1}^{J} 2^{-nr}\#\Lambda_r \ \lesssim \ \sum_{r=0}^{J} \#\Lambda_r \sum_{j=-1}^{r-1} 2^{n(j-r)} \\ &\lesssim \ \sum_{r=0}^{J} \#\Lambda_r = \#\Lambda. \qquad \square \end{aligned} \tag{4.8}$$

5 Computational aspects and realization

We conclude this paper with some remarks concerning the realization of the presented method. Let us assume that we can compute the functionals $q_\lambda(f)$ with a fixed number of operations. This was shown to be the case in the presented examples in Section 3.

5.1 Computation of the Quasi-Interpolant

To determine $P_\Lambda f$ we will compute the coefficients γ_λ in the expansion

$$P_\Lambda f = \sum_{\lambda\in\mathcal{I}} \gamma_\lambda \, \phi_\lambda.$$

To compute the update, we furtherly need some of the coefficients β_λ from the level-wise representations

$$P_{\Lambda,j} f = \sum_{\lambda\in\Delta_j} \beta_\lambda \, \phi_\lambda.$$

By local linear independence we know that $\beta_\lambda \neq 0$ only if $\sigma_\lambda \subset \widetilde{\Omega}_s$, $s \leq j$, with $\widetilde{\Omega}_r$ defined in (4.6). On the other hand, we only need coefficients β_λ for the remaining updates for those λ which satisfy $\sigma_\mu \subset \sigma_\lambda$ for some $\mu \in \mathcal{I}_r$, $r > j$. Therefore, we compute for every level only the coefficients

$$\beta_\lambda := \left(P_{\Lambda,j} f, \tilde{\phi}_\lambda\right)_{L_2}, \quad \lambda \in \mathcal{I}_j^+,$$

where

$$\mathcal{I}_j^+ := \{\lambda \in \Delta_j : \ \sigma_\mu \subset \sigma_\lambda \subset \widetilde{\Omega}_s, \ \text{for some } \mu \in \mathcal{I}_r, \ s \leq j < r\}.$$

We start by setting $\gamma_\lambda = q_\lambda(f)$, $\lambda \in \mathcal{I}_{j_0}$, i.e., we consider

$$P_{\Lambda,j_0} f = \sum_{\lambda\in\mathcal{I}_{j_0}} \gamma_\lambda \ \phi_\lambda.$$

Furthermore, we set $\beta_\lambda = \gamma_\lambda$, $\lambda \in \mathcal{I}_{j_0}^+$. Now, assume that we have already computed the coefficients γ_λ in the representation

$$P_{\Lambda,j-1} f = \sum_{r=j_0}^{j-1} \sum_{\lambda\in\mathcal{I}_r} \gamma_\lambda \ \phi_\lambda,$$

as well as the coefficients β_λ, $\lambda \in \mathcal{I}_{j-1}^+$. For $\lambda \in \mathcal{I}_j^+$ we deduce from (4.1) that

$$\beta_\lambda = \begin{cases} q_\lambda(f), & \text{if } \lambda \in \mathcal{I}_j, \\ (P_{\Lambda,j-1} f, \tilde{\phi}_\lambda)_{L_2} & \text{otherwise.} \end{cases}$$

From the refinement equation (2.2) we know that for $\lambda \in \mathcal{I}_j^+$

$$\left(P_{\Lambda,j-1} f, \tilde{\phi}_\lambda\right)_{L_2} = \sum_{\mu\in\mathcal{I}_{j-1}^+} \beta_\mu a_{\mu,\lambda}.$$

Moreover, for $\lambda \in \mathcal{I}_j$ we have

$$\gamma_\lambda = \left(P_{\Lambda,j} f, \tilde{\phi}_\lambda\right)_{L_2} - \left(P_{\Lambda,j-1} f, \tilde{\phi}_\lambda\right)_{L_2} = q_\lambda(f) - \sum_{\mu\in\Delta_{j-1}:\ \sigma_\lambda\subset\sigma_\mu} \beta_\mu a_{\mu,\lambda},$$

i.e.

$$\gamma_\lambda = \begin{cases} \beta_\lambda - \sum_{\mu\in\mathcal{I}_{j-1}^+} \beta_\mu a_{\mu,\lambda}, & \text{if } \lambda \in \mathcal{I}_j^+, \\ q_\lambda(f) & \text{otherwise.} \end{cases}$$

Repeating this scheme for each level, leads to the following efficient algorithm.

Algorithm 5.1 QI_ADAPT(f, Λ)

1. *Initialization:*
   Determine $\mathcal{I}$, $\mathcal{I}_{j_0}^+$,
   $\gamma_\lambda := q_\lambda(f)$, $\lambda \in \mathcal{I}_{j_0}$,
   $\beta_\lambda := \gamma_\lambda$, $\lambda \in \mathcal{I}_{j_0}^+$.

2. *For* $j = j_0 + 1, \ldots, J$

   (a) *Determine* $\mathcal{I}_j^+$,

   (b) $\beta_\lambda := q_\lambda(f), \qquad \lambda \in \mathcal{I}_j^+ \cap \mathcal{I}_j,$

   (c) $\beta_\lambda := \sum_{\mu \in \mathcal{I}_{j-1}^+} \beta_\mu a_{\mu,\lambda}, \qquad \lambda \in \mathcal{I}_j^+ \setminus \mathcal{I}_j,$

   (d) $\gamma_\lambda = \beta_\lambda - \sum_{\mu \in \mathcal{I}_{j-1}^+} \beta_\mu a_{\mu,\lambda}, \qquad \lambda \in \mathcal{I}_j^+ \cap \mathcal{I}_j,$

   (e) $\gamma_\lambda = q_\lambda(f), \qquad \lambda \in \mathcal{I}_j \setminus \mathcal{I}_j^+.$

Output: γ, $\mathcal{I}$

We will assume that we can compute function values $f(x)$ at unit cost independent of x. Then the number of function values needed to determine $q_\lambda(f)$ is independent of λ. This requirements are usually fulfilled, and we can estimate the complexity of the algorithm as follows.

By definition of $\mathcal{I}$ we conclude immediately that $\#\mathcal{I} \lesssim \#\Lambda$. Furthermore $\lambda \in \mathcal{I}_j^+$ implies $|\sigma_\lambda \cap \widetilde{\Omega}_r| \neq 0$ and by the local support of ϕ_λ (2.1) we conclude analogously to (4.8) that

$$\sum_{j=j_0}^{J} \#\mathcal{I}_j^+ \lesssim \#\Lambda.$$

As we know for any $\lambda \in \Delta$ the number of non-vanishing refinement coefficients $a_{\mu,\lambda}$ is bounded by a constant independent of λ, i.e., for each $\lambda \in \mathcal{I}$ and each $\lambda \in \mathcal{I}^+$ we need only a fixed number of operations independent of λ. Hence, the number of operations as well as the size of the output can be estimated by $\mathcal{O}(\Lambda)$.

Remark 5.2 *If* $|\widetilde{\Omega}_r \cap \widetilde{\Omega}_s| = 0$, $|r - s| > 1$, *one has always* $\mathcal{I}_j^+ \subset \mathcal{I}_j$ *and* $\beta_\lambda = q_\lambda(f) = \gamma_\lambda$, $\lambda \in \mathcal{I}_j^+$, *i.e., the coefficients* β_λ *are not needed in the algorithm. Such a partition can always be determined (cf.* [17]*). However, the size of* Λ *may be increased by a factor depending on the support size of* ϕ_λ, *while the above algorithm works for any partition set* Λ *and keeps the size of* $\mathcal{I}$ *minimal at no extra cost.*

5.2 Change of Basis

The final ingredient in the DSX-algorithm is the decomposition of the approximation of $f(u)$ from local scaling function representation to wavelet representation (in terms of the dual wavelets). Algorithm 5.1 is based on the finite support of the *primal* scaling functions. The decomposition algorithm in terms of the primal functions is expensive

(or not even possible exactly), because it is based on the infinitely supported mask of the dual scaling functions and wavelets. The way-out is rather simple. Since primal and dual functions span the *same* space (as opposed to general biorthogonal systems), we can perform a change of basis, which can be done fast in this direction and gives the desired representation of $f(u)$.

This means that we want to represent a scaling function expansion of the form

$$P_\Lambda f = \sum_{\lambda \in \mathcal{I}} \gamma_\lambda \phi_\lambda$$

in terms of the *dual* scaling functions $\tilde{\phi}_\lambda$. Since the dual scaling functions form a basis for the *same* space S_j, we obtain the following expansion for any $v \in S_j$

$$v = \sum_{\mu \in \Delta_j} (v, \phi_\mu)_{L_2} \, \tilde{\phi}_\mu.$$

Thus,

$$v_j := \sum_{\lambda \in \mathcal{I}_j} \gamma_\lambda \phi_\lambda = \sum_{\mu \in \tilde{\mathcal{I}}_j} \sum_{\lambda \in \mathcal{B}_\mu} \gamma_\lambda (\phi_\lambda, \phi_\mu)_{L_2} \, \tilde{\phi}_\mu,$$

with

$$\begin{aligned} \mathcal{B}_\mu &= \{\lambda \in \Delta_{|\mu|} : \ |\sigma_\lambda \cap \sigma_\mu| \neq 0\}, \\ \tilde{\mathcal{I}}_j &= \big\{\lambda \in \Delta_j : \ |\sigma_\lambda \cap \sigma_\mu| \neq 0 \text{ for some } \mu \in \mathcal{I}_j\big\}. \end{aligned}$$

From the local support of ϕ_λ (2.1) we know that $\#\mathcal{B}_\mu = (2d-1)^n$ and $\#\tilde{\mathcal{I}}_j \lesssim \#\mathcal{I}_j$, i.e., for $\tilde{\mathcal{I}} := \bigcup_{j=j_0}^{J} \tilde{\mathcal{I}}_j$ we have $\#\tilde{\mathcal{I}} \lesssim \#\mathcal{I} \lesssim \#\Lambda$. Now

$$P_\Lambda f = \sum_{\lambda \in \tilde{\mathcal{I}}} \tilde{\gamma}_\lambda \, \tilde{\phi}_\lambda.$$

The quantities $g_{(j,k),(j,\ell)} = (\phi_{(j,k)}, \phi_{(j,\ell)})_{L_2}$ are usually known (these are just integrals of splines on the *same* level), so that we can compute the coefficients

$$\tilde{\gamma}_\lambda := \sum_{\mu \in \mathcal{B}_\mu} \gamma_\mu \, g_{\lambda,\mu}. \tag{5.1}$$

with at most $(2d-1)^n$ operations, i.e., The change of basis can be performed with $\mathcal{O}(\#\Lambda)$ operations.

This shows that the change of basis can be performed efficiently. This can also be seen from the fact that this change of basis is nothing else then the multiplication of the given vector with the Gramian matrix $\big((\phi_\lambda, \phi_\mu)_{L_2}\big)_{\lambda,\mu \in \Delta_j}$ which is a sparse matrix. It should be noted that the number of operations per coefficient is *independent* of the level.

5.3 Decomposition

After performing the above described change of basis we now have a vector of expansion coefficients with respect to the dual scaling functions. For a function

$$v = \sum_{\mu \in \mathcal{I}_v} \gamma_\mu \, \tilde{\phi}_\mu, \qquad \mathcal{I} \subset \Delta_{j+1}$$

we know by biorthogonality that

$$v = \sum_{\lambda \in \mathcal{J}_v} (v, \phi_\lambda)_{L_2} \, \tilde{\phi}_\lambda + \sum_{\lambda \in \mathcal{K}_v} (v, \psi_\lambda)_{L_2} \, \tilde{\psi}_\lambda$$

with

$$\mathcal{J}_v = \{\lambda \in \Delta_j : \ |\sigma_\lambda \cap \operatorname{supp} v| \neq 0\} \quad \text{and} \quad \mathcal{K}_v = \{\lambda \in \nabla_j : \ |\omega_\lambda \cap \operatorname{supp} v| \neq 0\}.$$

With the refinement equations (2.2) of the *primal* scaling functions, we have

$$c_\lambda \ := \ (v, \phi_\lambda)_{L_2} \ = \ \sum_{\mu \in \mathcal{I}_v} \sum_{\nu \in \Delta_{j+1}} \gamma_\mu \, a_{\lambda,\nu} \left(\phi_\mu, \tilde{\phi}_\nu \right)_{L_2} \ = \ \sum_{\mu \in \mathcal{I}_v} \gamma_\mu \, a_{\lambda,\mu}.$$

Analogously, one obtains

$$d_\lambda := (v, \psi_\lambda)_{L_2} = \sum_{\mu \in \mathcal{I}_v} \gamma_\mu \, b_{\lambda,\mu}.$$

Recall that for any λ the number of non-zero coefficients $a_{\lambda,\mu}$ and $b_{\lambda,\mu}$ is bounded by a constant independent of λ, i.e., the coefficients c_λ and d_λ can be computed at fixed cost.

Given the coefficients $\tilde{\gamma}_{j,\ell}$ from (5.1) we are now able to compute the desired representation of $P_\Lambda f$ in terms of the dual wavelets. Set

$$\begin{aligned}
\mathcal{J}_J &:= \emptyset, \\
\mathcal{J}_j &:= \left\{ \lambda \in \Delta_j : \ |\sigma_\lambda \cap \sigma_\mu| \neq 0 \text{ for some } \mu \in \tilde{\mathcal{I}}_{j+1} \cup \mathcal{J}_{j+1} \right\}, \quad j < J, \\
\mathcal{K}_j &:= \left\{ \lambda \in \nabla_j : \ |\omega_\lambda \cap \sigma_\mu| \neq 0 \text{ for some } \mu \in \tilde{\mathcal{I}}_{j+1} \cup \mathcal{J}_{j+1} \right\}, \quad j < J.
\end{aligned}$$

Algorithm 5.3 LOC_DECOMP($\tilde{\gamma}$, $\tilde{\mathcal{I}}$)

1. *For j=J-1 downto 0*

   (a) $c_\lambda := \sum_{\mu \in \mathcal{I}} \gamma_\mu \, a_{\lambda,\mu}$, $\lambda \in \mathcal{J}_j$,
   $d_\lambda := \sum_{\mu \in \mathcal{I}} \gamma_\mu \, b_{\lambda,\mu}$, $\lambda \in \mathcal{K}_j$.

   (b) $\tilde{\gamma}_\lambda := c_\lambda + \tilde{\gamma}_\lambda$, $\lambda \in \tilde{\mathcal{I}}_j \cup \mathcal{J}_j$.

2. $d_{(-1,k)} := c_{(0,k)}$, $(-1,k) \in \mathcal{K}_{-1} = \mathcal{J}_0$

3. $\hat{\Lambda} := \bigcup_{j=-1}^{J-1} \mathcal{K}_j$

Output: $\boldsymbol{d}$, $\hat{\Lambda}$.

Note that this adaptive algorithm is highly efficient, in particular the number of operations per coefficient is *independent* of the level. The reason is that the decomposition of an expansion with respect to the *dual* functions uses the refinement coefficients of the *primal* functions (which is a finite set). Hence, this is like a Fast Wavelet Transform. In particular, for a given $P_\Lambda f$

This means that we now have an approximation $P_\Lambda f$ of $f(u)$ with respect to the dual wavelets at hand. That is, with the coefficients $\tilde{d}_\lambda$ from the above algorithm we have

$$P_\Lambda f = \sum_{\lambda \in \hat{\Lambda}} d_\lambda \tilde{\psi}_\lambda,$$

where

$$d_\lambda = (P_\Lambda f, \psi_\lambda)_{L_2} \approx (f, \psi_\lambda)_{L_2}.$$

Since this representation is unique, we conclude immediately that set $\hat{\Lambda}$ is exactly the one we have found in (4.7). Furthermore, $\#\mathcal{J}_j \lesssim \#\mathcal{K}_j$, i.e., the complexity of the decomposition step is $\mathcal{O}(\#\Lambda)$.

5.4 Approximate evaluation of wavelet coefficient of $f(u)$

Combining the methods described above yields the following algorithm.

Algorithm 5.4 EVAL_f(u)($\boldsymbol{u}_\Lambda^\varepsilon$, f, ε, $\Lambda^{u,\varepsilon}$)

1. *Prediction of Λ corresponding to Section 2.3.1.*
2. *Local reconstruction of u_Λ^ε corresponding to Section 2.3.2.*
3. $(\gamma, \mathcal{I})$ =**QI_ADAPT**$(f(u_\Lambda^\varepsilon), \Lambda)$.
4. *Change of basis:* $\tilde{\gamma}_\lambda := \sum_{\mu \in \mathcal{B}_\mu} \gamma_\mu \, g_{\lambda,\mu}, \quad \lambda \in \tilde{\mathcal{I}}$.
5. $(\boldsymbol{d}, \hat{\Lambda})$ =**LOC_DECOMP**$(\tilde{\gamma}, \tilde{\mathcal{I}})$.

Output: $\boldsymbol{d}$, $\hat{\Lambda}$.

The prediction and reconstruction step can be done in complete analogy to biorthogonal wavelets (see [17]), since this steps are only based on the compact support of the primal scaling functions and wavelets. Furthermore, we have already shown that the complexity of step 3, 4, and 5 is $\mathcal{O}(\#\Lambda)$. Hence, we have the following result.

Theorem 5.5 *Let $f \in C^m(\mathbb{R})$ and $u \in B_q^s(L^\tau(\square))$, $\tau > (\frac{s}{n} + \frac{1}{p})^{-1}$, $s \le m$. Assume that for any $\varepsilon > 0$ the input satisfies $\|u - u_\Lambda^\varepsilon\|_{L_2} \le \varepsilon$, $\#\Lambda^{u,\varepsilon} \lesssim \varepsilon^{-n/s}$. Then Algorithm 5.4 computes the coefficients d_λ, $\lambda \in \Lambda$, with*

$$\left(\sum_{\lambda \in \nabla} \left| d_\lambda - (f(u), \psi_\lambda)_{L_2} \right|^2 \right)^{\frac{1}{2}} \lesssim \varepsilon \tag{5.2}$$

($d_\lambda = 0$, $\lambda \notin \Lambda$) at cost $\mathcal{O}(\#\Lambda) = \mathcal{O}(\varepsilon^{-n/s})$.

Proof. With a suitable prediction of Λ we have $\#\Lambda \lesssim \varepsilon^{-n/s}$. Furthermore, the algorithm will yield an approximation $P_\Lambda f(u_\Lambda^\varepsilon)$ with

$$\|f(u) - P_\Lambda f(u_\Lambda^\varepsilon)\|_{L_2} \le \|f(u) - P_\Lambda f(u)\|_{L_2} + C_q \|f'\|_{L_\infty} \|u - u_\Lambda^\varepsilon\|_{L_2} \lesssim \varepsilon.$$

Now, (5.2) follows from the Riesz basis property of Ψ. $\square$

References

[1] A. Barinka, T. Barsch, P. Charton, A. Cohen, S. Dahlke, W. Dahmen, and K. Urban. Adaptive Wavelet Schemes for Elliptic Problems — Implementation and Numerical Experiments. *SIAM J. Scient. Comput.*, 23(3):910–39, 2001.

[2] T. Barsch. *Adaptive Multiskalenverfahren für elliptische partielle Differentialgleichungen — Realisierung, Umsetzung und numerische Ergebnisse (in german).* PhD thesis, RWTH Aachen, Shaker-Verlag, Aachen, 2001.

[3] T. Berger and O. Strömberg. Exact reconstruction algorithms for the discrete wavelet transform using spline wavelets. *Appl. Comput. Harmon. Anal.*, 2:392–397, 1995.

[4] K. Bittner. Fast algorithms for spline wavelets on sparse grids. *SIAM J. Scient. Computing*, 20:1192–1213, 1999.

[5] C. K. Chui. *An Introduction to Wavelets.* Academic Press, Boston, 1992.

[6] C. K. Chui and E. Quak. Wavelets on a bounded interval. In D. Braess and L. Schumaker, editors, *Numerical methods in approximation theory*, volume 9, pages 53–75. Birkh"auser, Basel, 1992.

[7] C. K. Chui and J. Wang. A general framework of compactly supported splines and wavelets. *JAT*, 71:54–68, 1992.

[8] C. K. Chui and J. Wang. On compactly supported spline wavelets and a duality principle. *TAMS*, 330:903–915, 1992.

[9] A. Cohen, W. Dahmen, and R.A. DeVore. Adaptive wavelet schemes for elliptic operator equations — convergence rates. *Math. Comput.*, 70(233):27–75, 2001.

[10] A. Cohen, W. Dahmen, and R.A. DeVore. Adaptive wavelet methods II — beyond the elliptic case. *Found. Comp. Math.*, 2(3):203–245, 2002.

[11] A. Cohen, I. Daubechies, and J.-C. Feauveau. Biorthogonal bases of compactly supported wavelets. *Comm. Pure and Appl. Math.*, 45:485–560, 1992.

[12] A. Cohen, I. Daubechies, and P. Vial. Wavelets on the interval and fast wavelet transform. *Appl. Comp. Harm. Anal.*, 1:54–81, 1993.

[13] S. Dahlke, W. Dahmen, R. Hochmuth, and R. Schneider. Stable multiscale bases and local error estimation for elliptic problems. *Appl. Numer. Math*, 23(1):21–48, 1997.

[14] S. Dahlke, W. Dahmen, and K. Urban. Adaptive Wavelet Methods for Saddle Point Problems — Optimal Convergence Rates. *SIAM J. Numer. Anal.*, 40(4):1230–1262, 2002.

[15] S. Dahlke, R. Hochmuth, and K. Urban. Adaptive wavelet methods for saddle point problems. *Math. Model. Numer. Anal.*, 34(5):1003–1022, 2000.

[16] W. Dahmen, A. Kunoth, and K. Urban. Biorthogonal Spline-wavelets on the Interval — Stability and Moment Condition. *Appl. Comp. Harm. Anal.*, 6:132–196, 1999.

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

[18] W. Dahmen, K. Urban, and J. Vorloeper. Adaptive Wavelet Methods — Basic Concepts and Applications to the Stokes Problem. In D.-X. Zhou, editor, *Wavelet Analysis*, pages 39–80. orld Scientific, New Jersey, 2002.

[19] I. Daubechies. *Ten lectures on wavelets*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1992.

[20] R.-Q. Jia and C. A. Micchelli. Using the refinement equation for the construction of pre-wavelets II: Powers of two. In P. J. Laurent, A. Le Méhauté, and L. L. Schumaker, editors, *Curves and Surfaces*, pages 209–246. Academic Press, New York, 1991.

[21] C. A. Micchelli. Using the refinement equation for the construction of pre-wavelets. *Numer. Algorithms*, 1(1):75–116, 1991.

[22] G. Plonka and M.Tasche. On the computation of periodic spline wavelets. *Appl. Comput. Harmon. Anal.*, 2:1–14, 1995.

[23] E. G. Quak and N. Weyrich. Decomposition and reconstruction algorithms for spline wavelets on a bounded interval. *Appl. Comp. Harm. Anal.*, 1:217–23, 1994.

[24] Sherman D. Riemenschneider and Zuowei Shen. Wavelets and pre-wavelets in low dimensions. *J. Approx. Theory*, 71(1):18–38, 1992.

[25] V. A. Zheludev. Local quasi-interpolatory splines and Fourier transforms. *Dokl. Akad. Nauk SSSR*, 282:1293–1298, 1985. English transl. in *Soviet Math. Dokl.*, 31:573–577, (1985).