

ADAPTIVE WAVELET METHODS ON UNBOUNDED DOMAINS

SEBASTIAN KESTLER AND KARSTEN URBAN

ABSTRACT. In this paper, we introduce an adaptive wavelet method for operator equations on unbounded domains. We use wavelet bases on \mathbb{R}^n to equivalently express the operator equation in terms of a well-conditioned discrete problem on sequence spaces. By realizing an approximate adaptive operator application also for unbounded domains, we obtain a scheme that is convergent at an asymptotically optimal rate. As an alternative, we introduce a simplified version of this algorithm. In both cases, we use anisotropic wavelet bases in the multivariate case. We show the quantitative performance of the scheme by various numerical experiments.

1. INTRODUCTION

Operator equations on unbounded domains are relevant in various fields where no boundary conditions, but only the asymptotic behavior of the solution is known. Examples include radiation or wave propagation processes as well as valuation problems in finance. In many cases, the asymptotic nature of the solution allows to truncate the computational domain to a bounded one and to perform all computations by standard methods on that bounded domain. Obviously, this requires a careful compromise of accuracy (sufficiently large truncation domain) and computational complexity (possibly small truncation domain). However, in more complex situations (like for complex structured financial products), such an a priori truncation is not straightforward.

There are several known methods to numerically treat problems on unbounded domains such as infinite elements, inverted finite elements, FEM-BEM coupling and others. In this paper, we introduce an adaptive wavelet method for operator equations on unbounded domains. The idea is as follows. Wavelet bases on Sobolev spaces $H^s(\mathbb{R}^n)$ can easily be constructed in terms of dilations and integer translates of some mother wavelets ψ_e with $e \in E := \{1, \dots, 2^n - 1\}$,

$$(1.1) \quad \Psi^{\mathbb{R}^n} := \{\psi_{j,e,\mathbf{k}} : j \in \mathbb{Z}, e \in E, \mathbf{k} \in \mathbb{Z}^n\}, \quad \psi_{j,e,\mathbf{k}}(x) := 2^{nj/2} \psi_e(2^j x - \mathbf{k}),$$

for $x \in \mathbb{R}^n$. Thus, one can follow the idea from [6] to transform the original operator equation $\mathcal{A}u = f$ into an equivalent well-posed problem $\mathbf{A}\mathbf{u} = \mathbf{f}$ on sequence spaces ℓ_2 for the wavelet coefficients. This idea has been used e.g. in [2, 6, 7, 15, 16, 24] (see also [28]) with wavelet bases on bounded domains. This approach results in adaptive wavelet methods that have been proven to converge at an optimal rate as

Date: June 8, 2010.

2000 Mathematics Subject Classification. 65T60, 35Q68.

Key words and phrases. Wavelets, adaptive numerical methods, unbounded domains.

This work has been supported by the Deutsche Forschungsgemeinschaft within the Research Training Group (Graduiertenkolleg) GrK1100 *Modellierung, Analyse und Simulation in der Wirtschaftsmathematik* at the University of Ulm and within the priority program DFG-SPP 1324 *Mathematical methods for extracting quantifiable information from complex systems*.

compared with the best N -term approximation w.r.t. the same basis. In order to highlight the differences to problems on unbounded domains, let us mention that a wavelet basis on a bounded domain $\Omega \subset \mathbb{R}^n$ typically takes the form

$$(1.2) \quad \Psi^\Omega := \{\psi_{j,\mathbf{k}} : j \geq j_0, \mathbf{k} \in \mathbf{I}_j\},$$

where $|\text{supp } \psi_{j,\mathbf{k}}| \sim 2^{-j}$, but $\psi_{j,\mathbf{k}}$ may not result by scaling and translating mother wavelets, e.g. [4, 10, 12, 15]. Both $\Psi^{\mathbb{R}^n}$ and Ψ^Ω consist of infinitely many basis functions. Whereas $\Psi^{\mathbb{R}^n}$ consists of all dilates and translates, Ψ^Ω has a fixed minimal level j_0 (depending on Ω as well as the type of wavelets) and the location index $\mathbf{k} \in \mathbf{I}_j$ ranges over a finite index set \mathbf{I}_j with $\#\mathbf{I}_j \sim 2^{jn}$.

If we can manage to design an adaptive wavelet method that is able to select appropriate subsets out of $\mathbb{Z} \times E \times \mathbb{Z}^n$, then we can –in principle– use the same adaptive schemes as on Ω . This is precisely the path we follow in this paper. We introduce an adaptive selection procedure on unbounded domains and derive an asymptotically optimal adaptive wavelet method. Let us mention that this approach offers some interesting features:

- Though possible, the construction of wavelet bases on general domains Ω is technically challenging. Here, we completely circumvent the need of constructing a basis on a possibly complicated domain and use the most simple situation that is possible for wavelets, namely, the shift-invariant case.
- Adaptive methods are particularly favorable if the solution has local effects like a singularity of the derivative at a single point. Such effects can result from three different sources, namely the domain, the operator or the right-hand side. The first source does not appear for problems on \mathbb{R}^n . For the remaining two, certain a priori information is available. In fact, for example in the case $\mathcal{A} = -\Delta + I$, the wavelet decomposition of the right-hand side f is already a good prediction for the relevant coefficients of the solution. Thus, this can be used as initial index set in order to improve the efficiency of the method.
- We do not need to truncate the domain, the scheme automatically detects the significant wavelets and determines a ‘computational domain’ automatically. Thus, our method allows to solve a PDE problem on an unbounded domain by a compactly supported and locally refinable basis.
- This approach concerning the treatment of unbounded domains can easily be generalized to higher space dimensions, nonlocal operators or nonlinear problems. In fact, one can easily use our framework in methods to treat high-dimensional [24] or nonlinear problems [8].

Nevertheless, it is a priori not clear how actually the resolution of the asymptotic boundary conditions realized by adaptive wavelet schemes look like. Moreover, as we have to take into account an infinite number of translation indices on each level (recall that in a bounded setting, this number is finite), the question arises how fast the asymptotic behavior of the best N -term approximation is reached by the algorithm.

The remainder of this paper is organized as follows. In Section 2, we review the main ingredients of adaptive wavelet methods. Section 3 contains the modification and extension to unbounded domains. We describe intensive numerical experiments in Section 4 and finish with some conclusions and an outlook in Section 5.

2. ADAPTIVE WAVELET METHODS

2.1. Elliptic operator equations. Let H be a Hilbert space (e.g. $H^1(\mathbb{R}^n)$) and H' its dual w.r.t. $L_2(\mathbb{R}^n)$ (e.g. $H^{-1}(\mathbb{R}^n)$) where we denote by $\langle \cdot, \cdot \rangle$ the duality pairing in $H' \times H$. For a linear operator $\mathcal{A} : H \rightarrow H'$ and a right-hand side $f \in H'$, we are concerned with the numerical solution of the operator equation for $u \in H$

$$(2.1) \quad \mathcal{A}u = f \text{ in } H'.$$

We assume that the bilinear form $a(\cdot, \cdot) : H \times H \rightarrow \mathbb{R}$ defined by $a(\cdot, \cdot) := \langle \mathcal{A} \cdot, \cdot \rangle$ is continuous and coercive, i.e., there exist constants $0 < c_{\mathcal{A}} \leq C_{\mathcal{A}} < \infty$ such that

$$(2.2) \quad c_{\mathcal{A}} \|v\|_H^2 \leq a(v, v), \quad \forall v \in H,$$

$$(2.3) \quad |a(w, v)| \leq C_{\mathcal{A}} \|w\|_H \|v\|_H, \quad \forall v, w \in H.$$

In this article, we focus on PDE problems defined on $H^1(\mathbb{R}^n)$.

2.2. Wavelets. For the discretization of (2.1), we use a wavelet basis Ψ . We start by the univariate case $n = 1$ since we use this as a building block also for higher space dimensions.

2.2.1. Wavelet bases for $H^1(\mathbb{R})$. It is well-known that for $n = 1$, the wavelet bases $\Psi = \{\psi_{j,k} : j, k \in \mathbb{Z}\}$, $\psi_{j,k}(x) := 2^{j/2} \psi(2^j x - k)$, $x \in \mathbb{R}$, constructed e.g. in [9, 13] form Riesz bases of $L_2(\mathbb{R})$. Moreover, it can be shown that (considering Ψ such as $\mathbf{v} \in \ell_2(\mathbb{Z}^2)$ as column vectors) the properly scaled functions $\mathbf{D}\Psi := \{2^{-j} \psi_{j,k} : j, k \in \mathbb{Z}\}$ form a Riesz basis of $H^1(\mathbb{R})$, i.e., there exist constants $0 < c_{\Psi} \leq C_{\Psi} < \infty$ such that

$$(2.4) \quad c_{\Psi} \|\mathbf{v}\|_{\ell_2} \leq \|\mathbf{d}^T \mathbf{D}\Psi\|_{H^1(\mathbb{R})} \leq C_{\Psi} \|\mathbf{v}\|_{\ell_2}, \quad \mathbf{v} \in \ell_2(\mathbb{Z}^2).$$

To shorten notation, as long as it cannot be misunderstood, we skip the index set for norms and scalar products. In the sequel we shall only consider biorthogonal wavelet bases from [9].

In order to avoid arbitrarily coarse levels $j \rightarrow -\infty$, one can also consider a minimal level $-\infty < j_0 < \infty$ and $\Psi = \{\psi_{j,k} : j \geq j_0 - 1, k \in \mathbb{Z}\}$ with $\psi_{j,k}$ $j \geq j_0$ as above and so called scaling functions $\psi_{j_0-1,k} = 2^{j_0/2} \phi(2^{j_0} \cdot - k)$, $k \in \mathbb{Z}$, [20]. Here, ϕ is a refinable function such as a cardinal B-spline. A basic assumption is that ϕ and ψ are compactly supported and therefore we have in particular that $|\text{supp } \psi_{j,k}| \sim 2^{-j}$. Another important feature of the wavelet basis Ψ is its polynomial exactness of order d and the number $\tilde{d} \geq d$ of vanishing moments. Both parameters depend on the particular choice of ϕ and ψ . Often, we abbreviate the double index (j, k) by λ and denote the level j by $|\lambda|$. Moreover, if a minimal level $j_0 > -\infty$ is used, we set $\mathcal{J} := \{(j, k) : j \geq j_0 - 1, k \in \mathbb{Z}\}$.

2.2.2. Tensor product wavelet bases for $n > 1$. In higher dimensions, one can use the wavelet basis $\Psi^{\mathbb{R}^n}$ defined in (1.1) which is an *isotropic* basis and is based upon a multiresolution analysis (see e.g. [28]). In this article, we will concentrate on *anisotropic* wavelets which are generalizations of isotropic ones and have shown some advantages in applications, in particular in higher space dimensions, see e.g. [15, 22, 24]. A (Riesz) tensor product wavelet basis Ψ of $L_2(\mathbb{R}^n)$ can be simply constructed by the tensor product of n wavelet bases $\Psi^{(1)}, \dots, \Psi^{(n)}$ of $L_2(\mathbb{R})$, i.e.,

$\Psi := \Psi^{(1)} \otimes \cdots \otimes \Psi^{(n)}$. Since $H^1(\mathbb{R}^n)$ is isomorphic to $\bigcap_{k=1}^n \bigotimes_{m=1}^n H^{\delta_{mk}}(\mathbb{R})$ with δ_{mk} the Kronecker delta and $H^0(\mathbb{R}) := L_2(\mathbb{R})$, it also holds that (cf. e.g. [17])

$$(2.5) \quad \mathbf{D}\Psi := \left\{ 2^{-|\boldsymbol{\lambda}|_\infty} \bigotimes_{m=1}^n \psi_{\lambda_m}^{(m)} : \boldsymbol{\lambda} = (\lambda^{(1)}, \dots, \lambda^{(n)}) \in \mathbf{J} \right\},$$

where $|\boldsymbol{\lambda}|_\infty := |\mathbf{j}|_\infty := \max_{m=1, \dots, n} |\lambda^{(m)}|$ and $\mathbf{J} := \mathbb{Z}^{2n}$, is a Riesz basis of $H^1(\mathbb{R}^n)$. Hence, the norm equivalence (2.4) also holds for tensor product wavelet bases where the constants c_Ψ and C_Ψ have to be adopted. We emphasize, that as in the univariate case, we can bound the minimal level $\mathbf{j}_0 = (j_0^{(1)}, \dots, j_0^{(n)})$ in the basis Ψ by using scaling functions on levels $j_0^{(1)}, \dots, j_0^{(n)}$ in the different space directions. In this case, we set $\mathbf{J} := \mathcal{J}^{(1)} \times \cdots \times \mathcal{J}^{(n)}$. In the sequel, we only use anisotropic tensor product wavelet bases as they are best suited for problems on \mathbb{R}^n , $n > 1$. As univariate wavelet bases are a special case of a tensor wavelet basis, we only need the notations for tensor bases.

2.3. Wavelet discretization. Now we use a tensor wavelet basis $\Psi := \{\psi_\lambda : \lambda \in \mathbf{J}\}$ to transform (2.1) into a well-conditioned discrete operator equation. From the Riesz basis property of $\mathbf{D}\Psi$, we infer that for the solution u to (2.1) there exists a unique $\mathbf{u} \in \ell_2(\mathbf{J})$ with $u = \mathbf{u}^T \mathbf{D}\Psi$. This means that \mathbf{u} is the (unknown) sequence of wavelet coefficients of u . Thus, (2.1) is equivalent to the infinite linear system

$$(2.6) \quad \mathbf{A}\mathbf{u} = \mathbf{f},$$

where $\mathbf{A} := (\langle \mathcal{A}\mathbf{D}\Psi, \mathbf{D}\Psi \rangle)^T$ and $\mathbf{f} := \langle f, \mathbf{D}\Psi \rangle$. Note, that (2.6) is well-posed on $\ell_2(\mathbf{J})$ since by (2.2), (2.3) and (2.4), the bilinear form $\mathbf{a}(\mathbf{v}, \mathbf{v}) := \langle \mathbf{A}\mathbf{v}, \mathbf{v} \rangle_{\ell_2} = a(\mathbf{v}^T \mathbf{D}\Psi, \mathbf{v}^T \mathbf{D}\Psi)$ satisfies

$$(2.7) \quad c_1 \|\mathbf{v}\|_{\ell_2}^2 \leq \mathbf{a}(\mathbf{v}, \mathbf{v}) \leq c_2 \|\mathbf{v}\|_{\ell_2}^2, \quad \mathbf{v} \in \ell_2(\mathbf{J}),$$

with $c_1 := c_\Psi^2 c_{\mathcal{A}}$ and $c_2 := C_\Psi^2 C_{\mathcal{A}}$. Therefore, $\mathbf{a}(\cdot, \cdot)$ is coercive and, by an analogous argumentation using (2.3), also continuous. For this reason, the operator $\mathbf{A} : \ell_2(\mathbf{J}) \rightarrow \ell_2(\mathbf{J})$ is continuous (with continuity constant c_2) and coercive (with coercivity constant c_1). Moreover, from (2.7) follows that $\mathbf{A} : \ell_2(\mathbf{J}) \rightarrow \ell_2(\mathbf{J})$ is boundedly invertible with operator norms on $\ell_2(\mathbf{J})$

$$\|\mathbf{A}\| := \sup_{\mathbf{v} \in \ell_2(\mathbf{J})} \frac{\|\mathbf{A}\mathbf{v}\|_{\ell_2}}{\|\mathbf{v}\|_{\ell_2}} \leq c_2, \quad \|\mathbf{A}^{-1}\| := \sup_{\mathbf{v} \in \ell_2(\mathbf{J})} \frac{\|\mathbf{A}^{-1}\mathbf{v}\|_{\ell_2}}{\|\mathbf{v}\|_{\ell_2}} \leq c_1^{-1}.$$

The condition of \mathbf{A} is defined by $\kappa(\mathbf{A}) := \|\mathbf{A}\| \|\mathbf{A}^{-1}\|$ and is bounded which is in fact a crucial property for the numerical treatment.

Setting $\|\mathbf{v}\|_{\mathbf{a}} := \mathbf{a}(\mathbf{v}, \mathbf{v})$ for $\mathbf{v} \in \ell_2(\mathbf{J})$, we see that the energy norm $\|\cdot\|_{\mathbf{a}}$ is equivalent to the norm $\|\cdot\|_{\ell_2}$ on $\ell_2(\mathbf{J})$, i.e., $\|\mathbf{v}\|_{\mathbf{a}} \sim \|\mathbf{v}\|_{\ell_2}$, $\mathbf{v} \in \ell_2(\mathbf{J})$. We can also define another norm for $\mathbf{v} \in \ell_2(\mathbf{J})$ by $\|\mathbf{v}\|_{\mathbf{A}} := \|\mathbf{A}\mathbf{v}\|_{\ell_2}$ for which we have

$$(2.8) \quad \|\mathbf{v}\|_{\mathbf{A}} \sim \|\mathbf{v}\|_{\ell_2} \sim \|\mathbf{v}^T \mathbf{D}\Psi\|_H, \quad \forall \mathbf{v} \in \ell_2(\mathbf{J}).$$

To avoid the use of various constants, we write $C \lesssim D$ if there exists a constant $c > 0$ such that $C \leq cD$. Analogously, we define \gtrsim . We use $C \sim D$ if $C \lesssim D$ and $C \gtrsim D$. In the sequel, we shall need the restriction of the infinite matrix \mathbf{A} and infinite vectors $\mathbf{v} \in \ell_2(\mathbf{J})$ to finite index sets $\Lambda \subset \mathbf{J}$. For this purpose, we introduce for $\mathbf{v} \in \ell_2(\mathbf{J})$ the projection $\mathbf{P}_\Lambda \mathbf{v} := \mathbf{v}|_{\ell_2(\Lambda)}$. Analogously, we set $\mathbf{A}_\Lambda := (\mathbf{P}_\Lambda \mathbf{A})|_{\ell_2(\Lambda)}$, $\mathbf{f}_\Lambda := \mathbf{P}_\Lambda \mathbf{f}$ and $\mathbf{v}_\Lambda := \mathbf{P}_\Lambda \mathbf{v}$. Thus, we obtain the finite Galerkin system

$$(2.9) \quad \mathbf{A}_\Lambda \mathbf{u}_\Lambda = \mathbf{f}_\Lambda.$$

One possible interpretation of many adaptive schemes is to find a sequence of indices $\mathbf{\Lambda}^{(0)}, \mathbf{\Lambda}^{(1)}, \mathbf{\Lambda}^{(2)}, \dots$ so that the corresponding Galerkin solutions $\mathbf{u}^{(k)}$ of (2.9) for $\mathbf{\Lambda} = \mathbf{\Lambda}^{(k)}$ converge possibly fast against \mathbf{u} with as few active wavelet coefficients as possible.

2.4. Nonlinear approximation theory. The analysis of adaptive schemes automatically leads to nonlinear approximation theory. The reason is that we wish to approximate the unknown solution \mathbf{u} with as few wavelet coefficients as possible. Thus, the optimum would be a *quasi best N -term approximation* \mathbf{u}_N of \mathbf{u} with $\#\text{supp } \mathbf{u}_N = N$, i.e., with some constant $0 < C < \infty$

$$(2.10) \quad \|\mathbf{u} - \mathbf{u}_N\|_{\ell_2} \leq C\sigma_N(\mathbf{u}), \quad \sigma_N(\mathbf{u}) := \inf_{\mathbf{w} \in \Sigma_N} \|\mathbf{u} - \mathbf{w}\|_{\ell_2},$$

where $\Sigma_N := \{\mathbf{v} \in \ell_2(\mathbf{J}) : \#\text{supp } \mathbf{v} \leq N\}$ is a nonlinear manifold in $\ell_2(\mathbf{J})$. A scheme realizing this is called *asymptotically optimal* provided that the number of required operations is linear in the number of active unknowns.

It is a well-known fact from nonlinear approximation theory that the rate of decay of the best N -term approximation is related to decay rates of the wavelet coefficients which in turn is related to the Besov regularity of the function to be approximated. In order to formulate this property, we need weak $\ell_\tau(\mathbf{J})$ -spaces defined as follows (cf. [14]). For each $0 < \tau < 2$ and $\mathbf{v} \in \ell_2(\mathbf{J})$, we define $|\mathbf{v}|_{\ell_\tau^w} := \sup_{n \geq 1} n^{1/\tau} v_n^*$, where v_n^* is the n -th largest entry in modulus of \mathbf{v} and $\mathbf{v}^* := (v_n^*)_{n \in \mathbb{N}}$. Then we set $\ell_\tau^w(\mathbf{J}) := \{\mathbf{v} \in \ell_2(\mathbf{J}) : |\mathbf{v}|_{\ell_\tau^w} < \infty\}$ with the corresponding norm $\|\mathbf{v}\|_{\ell_\tau^w} := |\mathbf{v}|_{\ell_\tau^w} + \|\mathbf{v}\|_{\ell_2}$, $\mathbf{v} \in \ell_\tau^w$. With this notation at hand, $\sigma_N(\mathbf{u})$ decays with a fixed rate $s > 0$ if $\mathbf{u} \in \ell_\tau^w(\mathbf{J})$ for

$$(2.11) \quad \frac{1}{\tau} = s + \frac{1}{2}.$$

In this case, there exists a constant $C_\tau > 0$ depending on τ such that (see [6, Proposition 3.2])

$$(2.12) \quad \sigma_N(\mathbf{u}) \leq C_\tau \|\mathbf{u}\|_{\ell_\tau^w} N^{-s}.$$

Hence, the best possible value s^* for which (2.12) holds, is the optimal rate. It turns out that this is related to the Besov regularity of the underlying function.

For $n = 1$, it is known that for $0 < s < d - 1$ (with d being the polynomial exactness), we have $\|u\|_{B_\tau^{s+1}(L_\tau(\mathbb{R}))} \sim \|\mathbf{u}\|_{\ell_\tau}$, provided that $u = \mathbf{u}^T \mathbf{D} \Psi \in B_\tau^{s+1}(L_\tau(\mathbb{R}))$. In particular, this means that independent of the Besov regularity of u , s is bounded from above by $d - 1$, which depends only on the wavelet basis Ψ . For larger values of s , \mathbf{u} is in general no longer contained in $\ell_\tau(\mathbf{J})$. Concluding, for operator equations on $H^1(\mathbb{R})$, we can set $s^* = d - 1$ if $u \in B_\tau^{s+1}(L_\tau(\mathbb{R}))$.

For $n > 1$, there is a significant difference between isotropic and anisotropic wavelets. In the isotropic case, one has $s^* = \frac{d-1}{n}$, which means that the best rate deteriorates with increasing space dimension n , [5, Theorem 30.7]. For anisotropic wavelets, it is known from [18, 22, 24, 26] that if $u \in \bigcap_{k=1}^n \otimes_{\tau, m=1}^n B_\tau^{s+\delta_{mk}}(L_\tau(\mathbb{R}))$, then $\mathbf{u} \in \ell_\tau^w(\mathbf{J})$ for $0 < s < d - 1$. Here, \otimes_τ is a so called τ -tensor product introduced in [22]. Thus, also for $n > 1$, we can set $s^* = d - 1$ provided that we use anisotropic wavelets and the solution of (2.1) is sufficiently smooth in the above Besov sense.

2.5. Optimality and locality. Before we come to the formulation of adaptive wavelet methods, we need one more ingredient. In order to obtain a best possible method, it is not enough to generate a scheme which converges as fast as a best N -term approximation. In fact, we also need to be able to actually compute such an approximation with at most linear complexity. One key ingredient is that wavelets allow for a compression of a large class of operators due to their locality and their vanishing moments. An operator $\mathbf{A} : \ell_2(\mathbf{J}) \rightarrow \ell_2(\mathbf{J})$ is said to be in the class \mathcal{B}_s if there are two positive, summable sequences $(\alpha_j)_{j \geq 0}$ and $(\beta_j)_{j \geq 0}$ such that for every $j \in \mathbb{N}_0$ there exists a matrix \mathbf{A}_j with at most $2^j \alpha_j$ nonzero entries per row and column such that

$$(2.13) \quad \|\mathbf{A} - \mathbf{A}_j\| \leq \beta_j 2^{-js}.$$

Compression estimates which fit into the setting of (2.13) have been discussed in detail for different type of operators for example in [11, 23, 24]. This property can be used for the design of efficient algorithms as we shall review now. If we define $\mathbf{v}_{[j]}$ as a best 2^j term approximation to $\mathbf{v} \in \ell_\tau^w(\mathbf{J})$ (e.g. the first 2^j entries of \mathbf{v}^*), then it holds

$$(2.14) \quad \|\mathbf{v} - \mathbf{v}_{[j]}\|_{\ell_2} \leq \|\mathbf{v}\|_{\mathcal{A}^s} 2^{-js}, \quad \|\mathbf{v}\|_{\mathcal{A}^s} := \sup_{N \geq 0} (N+1)^s \sigma_N(\mathbf{v}),$$

if τ is chosen as in (2.11). One can use this observation to show that if $\mathbf{A} \in \mathcal{B}_s$, then it is a bounded operator on $\ell_\tau^w(\mathbf{J})$ and also derive a method for approximating an infinite matrix-vector product $\mathbf{A}\mathbf{v}$.

2.6. An optimal adaptive wavelet algorithm. Now, we describe the adaptive wavelet solver **ORIG-ADWAV** from [16] which we used as a basis for our extension to unbounded domains. Here, we assume furthermore that \mathcal{A} is symmetric. The core scheme is shown in Algorithm 1. We start with a real number ν_{-1} to be explained below and a desired tolerance $\varepsilon > 0$. Finally, we need to choose constants $\alpha, \gamma, \theta, \omega$ (see, e.g. [16] for possible choices):

- $0 < \omega < \alpha < 1$ such that $\frac{\alpha+\omega}{1-\omega} < \kappa(\mathbf{A})^{-\frac{1}{2}}$,
- $0 < \gamma < \frac{1}{6} \kappa(\mathbf{A})^{-1/2} \frac{\alpha-\omega}{1+\omega}$ and $\theta > 0$.

Algorithm 1 $[\mathbf{u}(\varepsilon), \mathbf{\Lambda}(\varepsilon)] = \text{ORIG-ADWAV}[\nu_{-1}, \varepsilon]$

- 1: $\mathbf{\Lambda}^{(0)} = \emptyset, k := 0, \mathbf{w}^{(0)} := \mathbf{0}$
 - 2: **repeat**
 - 3: $[\mathbf{\Lambda}^{(k+1)}, \nu_k] = \text{GROW}[\mathbf{w}^{(k)}, \theta \nu_{k-1}, \varepsilon]$
 - 4: $\mathbf{g}^{(k+1)} = \mathbf{P}_{\mathbf{\Lambda}^{(k+1)}}(\text{RHS}[\gamma \nu_k])$
 - 5: $\mathbf{w}^{(k+1)} = \text{GALSOLVE}[\mathbf{\Lambda}^{(k+1)}, \mathbf{g}^{(k+1)}, \mathbf{w}^{(k)}, (1 + \gamma) \nu_k, \gamma \nu_k]$
 - 6: $k = k + 1$
 - 7: **until** $\nu_k \leq \varepsilon$
 - 8: $\mathbf{u}(\varepsilon) = \mathbf{w}^{(k)}, \mathbf{\Lambda}(\varepsilon) = \mathbf{\Lambda}^{(k)}$
-

Before we detail the subroutines called within **ORIG-ADWAV**, let us recall the properties of this adaptive wavelet scheme.

Theorem 2.1 ([16, Theorem 2.7]). *The output $\mathbf{w} = \mathbf{u}(\varepsilon)$ of the routine **ORIG-ADWAV** $[\nu_{-1}, \varepsilon]$ satisfies $\|\mathbf{A}\mathbf{w} - \mathbf{f}\|_{\ell_2} \leq \varepsilon$. If $\nu_{-1} \sim \|\mathbf{f}\|_{\ell_2} \gtrsim \varepsilon$, and $\mathbf{u} \in \ell_\tau^w(\mathbf{J})$ for some $s < s^*$, $\frac{1}{\tau} = s + \frac{1}{2}$, then $\#\text{supp } \mathbf{w} \lesssim \varepsilon^{-1/s} |\mathbf{u}|_{\ell_\tau^w}^{1/s}$ and the number of arithmetic*

operations and storage locations is bounded by some absolute multiple of the same expression. \square

Now, we are going to detail all subroutines involved in **ORIG-ADWAV**. Within **ORIG-ADWAV** routine **GROW** shown in Algorithm 2 enlarges the current index set $\Lambda^{(k)}$ in such a way that the new index set $\Lambda^{(k+1)}$ guarantees a fixed error reduction in terms of the so called *saturation property*

$$(2.15) \quad \|\mathbf{P}_{\Lambda^{(k+1)}}(\mathbf{A}\mathbf{u}_{\Lambda^{(k)}} - \mathbf{f})\|_{\ell_2} \geq \beta \|\mathbf{A}\mathbf{u}_{\Lambda^{(k)}} - \mathbf{f}\|_{\ell_2}$$

for some $0 < \beta < 1$. Then, due to Galerkin orthogonality (cf. [6, Lemma 4.1]), one has the following error reduction $\|\mathbf{u} - \mathbf{u}_{\Lambda^{(k+1)}}\|_{\mathbf{a}} \leq (1 - \frac{c_2}{c_1}\beta^2)^{1/2} \|\mathbf{u} - \mathbf{u}_{\Lambda^{(k)}}\|_{\mathbf{a}}$.

Algorithm 2 GROW $[\mathbf{w}, \bar{\nu}, \varepsilon] \rightarrow [\Lambda, \nu]$

- 1: Define $\zeta := 2\frac{\omega\bar{\nu}}{1-\omega}$.
 - 2: **repeat**
 - 3: $\zeta := \zeta/2$, $\mathbf{r} := \mathbf{RHS}[\zeta/2] - \mathbf{APPLY}[\mathbf{w}, \zeta/2]$,
 - 4: **until** $\nu := \|\mathbf{r}\|_{\ell_2} + \zeta \leq \varepsilon$ or $\zeta \leq \omega\|\mathbf{r}\|_{\ell_2}$
 - 5: **if** $\nu > \varepsilon$ **then**
 - 6: determine a minimal set $\Lambda \supset \text{supp } \mathbf{w}$ such that $\|\mathbf{P}_{\Lambda}\mathbf{r}\| \geq \alpha\|\mathbf{r}\|_{\ell_2}$.
 - 7: **else**
 - 8: set $\Lambda := \emptyset$.
 - 9: **end if**
-

Under the same assumptions as in Theorem 2.1 and if $\mathbf{w} \in \ell_{\tau}^{\mathbf{w}}(\mathbf{J})$, then the number of operations and storage locations required by $[\Lambda, \nu] = \mathbf{GROW}[\mathbf{w}, \bar{\nu}, \varepsilon]$ is bounded by some absolute multiple of $\min\{\bar{\nu}, \nu\}^{-1/s} [|\mathbf{w}|_{\ell_{\tau}^{\mathbf{w}}}^{1/s} + |\mathbf{u}|_{\ell_{\tau}^{\mathbf{w}}}^{1/s} + \bar{\nu}^{1/s}(\#\text{supp } \mathbf{w} + 1)]$. Moreover, we have $\nu \geq \|\mathbf{A}\mathbf{w} - \mathbf{f}\|_{\ell_2}$ and, in case of $\nu > \varepsilon$, the saturation property

$$(2.16) \quad \frac{\alpha - \omega}{1 + \omega} \nu \leq \|\mathbf{P}_{\Lambda}(\mathbf{A}\mathbf{w} - \mathbf{f})\|_{\ell_2}, \quad \#(\Lambda \setminus \text{supp } \mathbf{w}) \lesssim \nu^{-1/s} |\mathbf{u}|_{\ell_{\tau}^{\mathbf{w}}}$$

holds with the constants α and ω described above.

The routine **RHS** $[\delta]$ produces an approximation $\bar{\mathbf{f}}$ to \mathbf{f} such that $\|\mathbf{f} - \bar{\mathbf{f}}\|_{\ell_2} \leq \delta$. Finally, **GALSOLVE** (short for Galerkin solver) produces an approximate solution $\tilde{\mathbf{w}}_{\Lambda}$ with $\|\mathbf{A}_{\Lambda}\tilde{\mathbf{w}}_{\Lambda} - \mathbf{f}_{\Lambda}\| \leq \varepsilon$ starting with an initial guess \mathbf{w}_{Λ} satisfying $\|\mathbf{A}_{\Lambda}\mathbf{w}_{\Lambda} - \mathbf{f}_{\Lambda}\| \leq \delta$.

Algorithm 3 GALSOLVE $[\Lambda, \mathbf{f}_{\Lambda}, \mathbf{w}_{\Lambda}, \delta, \varepsilon] \rightarrow [\tilde{\mathbf{w}}_{\Lambda}]$

- 1: Determine \mathbf{A}_J in the sense of (2.13) with $J = J(\varepsilon)$ as small as possible and $\|\mathbf{A} - \mathbf{A}_J\| \leq \frac{\varepsilon}{3}$.
 - 2: Assemble $\mathbf{B} := \mathbf{P}_{\Lambda}[\frac{1}{2}(\mathbf{A}_J + \mathbf{A}_J^*)]_{\ell_2(\Lambda)}$ with \mathbf{A}_J^* being the adjoint of \mathbf{A}_J .
 - 3: Compute $\mathbf{r}_0 := \mathbf{f}_{\Lambda} - \mathbf{P}_{\Lambda}(\mathbf{APPLY}[\mathbf{w}_{\Lambda}, \frac{\varepsilon}{3}])$.
 - 4: Determine \mathbf{x} as the solution of $\mathbf{B}\mathbf{x} = \mathbf{r}_0$ and set $\tilde{\mathbf{w}}_{\Lambda} = \mathbf{w}_{\Lambda} + \mathbf{x}$.
-

One key ingredient both in **GROW** and **GALSOLVE** is the routine **APPLY** shown in Algorithm 4 which is an adaptive approximate application of the biinfinite operator \mathbf{A} to a given compactly supported input \mathbf{v} with the following properties. The output $\mathbf{w} = \mathbf{APPLY}[\mathbf{v}, \eta]$ satisfies $\|\mathbf{A}\mathbf{v} - \mathbf{w}\|_{\ell_2} \leq \eta$ as well as $\text{supp } \mathbf{w} \lesssim$

$\|\mathbf{v}\|_{\ell_\tau^{\mathbf{w}}}^{1/s} \eta^{-\frac{1}{s}}$ provided that $\mathbf{v} \in \ell_\tau^{\mathbf{w}}(\mathbf{J})$, $\frac{1}{\tau} = s + \frac{1}{2}$, see [6, Properties 6.4]. We remark that necessary sorting operations in **GROW** and **APPLY** which are not of linear complexity can be replaced by approximative sorting procedures introduced in [1, 15] without destroying the approximation properties.

Algorithm 4 **APPLY** $[\mathbf{v}, \eta] \rightarrow \mathbf{w}$

- 1: Set $N := \#\text{supp } \mathbf{v}$
 - 2: Set $k(\eta)$ as the smallest integer such that $2^{k(\eta)} \geq \eta^{-\frac{1}{s}} \|\mathbf{v}\|_{\ell_\tau^{\mathbf{w}}(\mathbf{J})}^{\frac{1}{s}}$.
 - 3: Compute $\mathbf{v}_{[0]}, \mathbf{v}_{[i]} - \mathbf{v}_{[i-1]}$ for $i = 1, \dots, \lfloor \log N \rfloor$ and set $\mathbf{v}_{[i]} := \mathbf{v}$ for $i > \log N$.
 - 4: Determine $k \leq k(\eta)$ minimal such that $2^{-ks} \|\mathbf{v}\|_{\ell_\tau^{\mathbf{w}}} \leq \eta$.
 - 5: Compute $\mathbf{w} := \mathbf{w}_k := \mathbf{A}_k \mathbf{v}_{[0]} + \mathbf{A}_{k-1}(\mathbf{v}_{[1]} - \mathbf{v}_{[0]}) + \dots + \mathbf{A}_0(\mathbf{v}_{[k]} - \mathbf{v}_{[k-1]})$.
-

3. ADAPTIVE WAVELET ALGORITHMS ON UNBOUNDED DOMAINS

3.1. Choice of a minimal level. As already mentioned above, we can bound the minimal level \mathbf{j}_0 for the anisotropic wavelet basis Ψ on \mathbb{R}^n by using scaling functions. These have a significant impact for the approximation. As an example, a function that is piecewise constant can be represented by very few scaling functions (often B-splines), whereas one would need many wavelets. This shows that for efficiency reasons, we need to introduce a minimal level. We propose to use the right-hand side $f \in H^{-1}(\mathbb{R}^n)$ for this. Since $f \in H^{-1}(\mathbb{R}^n)$, the norm equivalence (2.5) ensures

$$(3.1) \quad 0 < \sum_{\mathbf{j} \in \mathbb{Z}^n} \sum_{\mathbf{k} \in \mathbb{Z}^n} 2^{-|\mathbf{j}|_\infty} |\langle f, \psi_{\mathbf{j}, \mathbf{k}} \rangle| < \infty.$$

This means that we have (at least theoretically) access to the largest values of $2^{-|\mathbf{j}|_\infty} |\langle f, \psi_{\mathbf{j}, \mathbf{k}} \rangle|$ by looking only at some finite index set $\mathbf{\Lambda}$. Thus, we introduce the following routine to determine an estimate for \mathbf{j}_0 . In Section 4 below, we investigate the effect of \mathbf{j}_0 to the efficiency of the adaptive algorithm.

Algorithm 5 **INITIALIZE** $\rightarrow [\mathbf{\Lambda}_0, \mathbf{j}_0]$

- 1: Compute a sufficiently large index set $\mathbf{\Lambda}$ such that $\bigcup_{\lambda \in \mathbf{\Lambda}} \text{supp } \psi_\lambda$ contains all singularities and the “bulk” of f .
 - 2: For all $\lambda \in \mathbf{\Lambda}$, compute $\mathbf{f}_{0, \lambda} := 2^{-|\lambda|_\infty} |\langle f, \psi_\lambda \rangle|$ for $\mathbf{f}_0 = (\mathbf{f}_{0, \lambda})_{\lambda \in \mathbf{\Lambda}}$.
 - 3: Compute $\mathbf{f}_0^* := (\mathbf{f}_{0, \lambda_1}^*, \mathbf{f}_{0, \lambda_2}^*, \dots)$ and set $\mathbf{j}_0 := |\lambda_1|$.
 - 4: Insert the scaling indices $(\mathbf{j}_0 - \mathbf{1}, \mathbf{k}_i)$ into $\mathbf{\Lambda}_0$ for $i = 1, \dots, K$, where K is the smallest integer such that $\bigcup_{i=1}^K \text{supp } \psi_{(\mathbf{j}_0 - \mathbf{1}, \mathbf{k}_i)}$ contains all singularities of \mathbf{f} . If there are no singularities, set $K = 1$.
-

We insert the initialization into the original method in Algorithm 1 and obtain Algorithm 6 below. The unchanged parts are shown in gray.

3.2. Some computational aspects of the unbounded domain setting.

Determination of the constants. It is known that the quantitative efficiency of **AD-WAV** is quite sensitive to the choice of its parameters. In particular, a good estimate of $\kappa(\mathbf{A})$ and the constants in (2.13) should be available.

Algorithm 6 $[\mathbf{u}(\varepsilon), \Lambda(\varepsilon)] = \mathbf{ADWAV}[\nu_{-1}, \varepsilon]$

-
- 1: $\mathbf{j}_0 = \mathbf{INITIALIZE}$
 - 2: $\Lambda^{(0)} = \emptyset, k := 0, \mathbf{w}^{(0)} := \mathbf{0}$
 - 3: **repeat**
 - 4: $[\Lambda^{(k+1)}, \nu_k] = \mathbf{GROW}[\mathbf{w}^{(k)}, \theta\nu_{k-1}, \varepsilon]$
 - 5: $\mathbf{g}^{(k+1)} = \mathbf{P}_{\Lambda^{(k+1)}}(\mathbf{RHS}[\gamma\nu_k])$
 - 6: $\mathbf{w}^{(k+1)} = \mathbf{GALSOLVE}[\Lambda^{(k+1)}, \mathbf{g}^{(k+1)}, \mathbf{w}^{(k)}, (1 + \gamma)\nu_k, \gamma\nu_k]$
 - 7: $k = k + 1$
 - 8: **until** $\nu_k \leq \varepsilon$
 - 9: $\mathbf{u}(\varepsilon) = \mathbf{w}^{(k)}, \Lambda(\varepsilon) = \Lambda^{(k)}$
-

Computation of the right-hand side. We assume the availability of a right-hand side $f \in H'$ in wavelet coordinates. This is usually achieved by the assumption that the analytic form of f is known. In this case, one knows exactly which entries in \mathbf{f} have to be calculated. If this is not the case, however, a very careful scan of the right-hand side data is necessary since otherwise the adaptive wavelet method might not be able to catch parts of the solution.

*Realization of **GROW**.* The new index set $\Lambda^{(k+1)}$ is obtained by **GROW** which adds wavelet indices to $\Lambda^{(k)}$. We do not have any control on this selection process which might be problem in the unbounded domain case. Moreover, as already stated above, it might be more efficient to add certain scaling function indices instead of wavelet indices.

Convergence and optimality. As it can be seen from the proofs of the optimality result in Theorem 2.1 in [16], the properties of the routines **ORIG-ADWAV** do depend neither on the number of translation indices in the underlying wavelet basis nor on the minimal level, but only on the adherence of the solution \mathbf{u} to $\ell_\tau^w(\mathbf{J})$. Moreover, as the size of the index set Λ in **INITIALIZE** can be kept small in comparison to $\text{supp } \mathbf{u}(\varepsilon)$, the complexity of this initial routine does not influence the overall complexity of **ADWAV** in comparison to **ORIG-ADWAV** which leads to one of the main results of this paper.

Theorem 3.1. *Under the same assumptions as in Theorem 2.1 and independent of the minimal level \mathbf{j}_0 in Ψ , it holds that $\|\mathbf{A}\mathbf{w} - \mathbf{f}\|_{\ell_2} \leq \varepsilon$ where $\mathbf{w} = \mathbf{u}(\varepsilon)$ is the output of $\mathbf{ADWAV}[\nu_{-1}, \varepsilon]$. Moreover, the overall complexity is bounded by some multiple of $\text{supp } \mathbf{w} \lesssim \varepsilon^{-1/s} |\mathbf{u}|_{\ell_\tau^w}^{1/s}$. \square*

Remark 3.2. The quantitative efficiency and the proven convergence of **ADWAV** depends strongly on the quality of the mentioned estimates for the required constants, e.g. $\kappa(\mathbf{A})$ and the choice of the parameters $\alpha, \gamma, \omega, \theta$. Though these estimates are accessible, the actual estimate for a given PDE might not be easy at all, in particular when convection or reaction terms may be varying. This is e.g. the case for the valuation of financial products. Moreover, the symmetry assumption in Subsection 2.6 made for **ADWAV** is a limitation and can only be lifted by adopting the more complicated algorithms from [7, ?] in the same way as described above. For this reason, we propose a simplified adaptive wavelet algorithm referred to as **S-ADWAV** that requires much less requirements and also applies to nonsymmetric problems at the price of non-proven convergence.

3.3. A simplified adaptive wavelet algorithm. The simplified adaptive wavelet algorithm we present in this paragraph is a slight modification and adaption of the algorithm proposed in [2, 28]. To our knowledge, there is no proof of convergence or optimality. Nevertheless, numerical experiments have shown that this adaptive wavelet algorithm performs very well in practice. The simplified algorithm passes on the usage of the routines **RHS** and **APPLY**, but explicitly determines $\mathbf{\Lambda}^{(k+1)}$ from $\mathbf{\Lambda}^{(k)}$. This is done by a heuristically motivated approach. We still have to solve Galerkin systems $\mathbf{A}_{\mathbf{\Lambda}^{(k)}} \mathbf{u}_{\mathbf{\Lambda}^{(k)}} = \mathbf{f}_{\mathbf{\Lambda}^{(k)}}$ for $k = 1, 2, \dots$ in each iteration.

3.3.1. Numerical solution of the Galerkin system. In each iteration of the algorithm, we have to solve a Galerkin system (2.9) for an index set $\mathbf{\Lambda}$. But in general one only solves a perturbed linear system

$$(3.2) \quad \tilde{\mathbf{A}}_{\mathbf{\Lambda}} \tilde{\mathbf{u}}_{\mathbf{\Lambda}} = \tilde{\mathbf{f}}_{\mathbf{\Lambda}} + \tilde{\mathbf{s}}_{\mathbf{\Lambda}}$$

with $\|\tilde{\mathbf{s}}_{\mathbf{\Lambda}}\|_{\ell_2} < \text{tol}_{\text{iter}}$, $\text{tol}_{\text{iter}} > 0$ a given tolerance, instead of solving (2.9) exactly. These perturbations may arise from numerical integration or, for the stiffness matrix, also from matrix compression techniques. We estimate the error as follows.

Proposition 3.3. *Let $\mathbf{\Lambda}$ be a finite subset of \mathbf{J} and assume that $\|\mathbf{A}_{\mathbf{\Lambda}} - \tilde{\mathbf{A}}_{\mathbf{\Lambda}}\| < \eta_{\mathbf{A}}$ with $\eta_{\mathbf{A}} < c_1$ and $\|\mathbf{f}_{\mathbf{\Lambda}} - \tilde{\mathbf{f}}_{\mathbf{\Lambda}}\|_{\ell_2} < \eta_{\mathbf{f}}$. Then, it holds that*

$$(3.3) \quad \|\mathbf{u}_{\mathbf{\Lambda}} - \tilde{\mathbf{u}}_{\mathbf{\Lambda}}\|_{\ell_2} \leq c_1^{-1} \left(\frac{\eta_{\mathbf{A}}}{c_1 - \eta_{\mathbf{A}}} (\eta_{\mathbf{f}} + \|\mathbf{f}_{\mathbf{\Lambda}}\|_{\ell_2} + \text{tol}_{\text{iter}}) + \eta_{\mathbf{f}} + \text{tol}_{\text{iter}} \right),$$

where $\mathbf{u}_{\mathbf{\Lambda}}$ is the solution of (2.9) and $\tilde{\mathbf{u}}_{\mathbf{\Lambda}}$ is a solution of (3.2).

Proof. First, we prove continuity and coercivity of $\tilde{\mathbf{A}}_{\mathbf{\Lambda}}$ on $\ell_2(\mathbf{\Lambda})$. From the assumptions, we infer from (2.7) that for all $\mathbf{w}_{\mathbf{\Lambda}}, \mathbf{v}_{\mathbf{\Lambda}} \in \ell_2(\mathbf{\Lambda})$

$$|\langle \tilde{\mathbf{A}}_{\mathbf{\Lambda}} \mathbf{w}_{\mathbf{\Lambda}}, \mathbf{v}_{\mathbf{\Lambda}} \rangle_{\ell_2}| \leq (\eta_{\mathbf{A}} + c_2) \|\mathbf{w}_{\mathbf{\Lambda}}\|_{\ell_2} \|\mathbf{v}_{\mathbf{\Lambda}}\|_{\ell_2},$$

such as for all $\mathbf{v}_{\mathbf{\Lambda}} \in \ell_2(\mathbf{\Lambda})$ we have

$$(3.4) \quad (c_1 - \eta_{\mathbf{A}}) \|\mathbf{v}_{\mathbf{\Lambda}}\|_{\ell_2} \leq \langle \tilde{\mathbf{A}}_{\mathbf{\Lambda}} \mathbf{v}_{\mathbf{\Lambda}}, \mathbf{v}_{\mathbf{\Lambda}} \rangle_{\ell_2}.$$

Now, the following estimate is straightforward:

$$\begin{aligned} \|\mathbf{u}_{\mathbf{\Lambda}} - \tilde{\mathbf{u}}_{\mathbf{\Lambda}}\|_{\ell_2}^2 &\leq c_1^{-1} \langle \mathbf{A}_{\mathbf{\Lambda}} (\mathbf{u}_{\mathbf{\Lambda}} - \tilde{\mathbf{u}}_{\mathbf{\Lambda}}), \mathbf{u}_{\mathbf{\Lambda}} - \tilde{\mathbf{u}}_{\mathbf{\Lambda}} \rangle_{\ell_2} \\ &\leq c_1^{-1} \langle \mathbf{A}_{\mathbf{\Lambda}} \mathbf{u}_{\mathbf{\Lambda}} - \tilde{\mathbf{A}}_{\mathbf{\Lambda}} \tilde{\mathbf{u}}_{\mathbf{\Lambda}} + \tilde{\mathbf{A}}_{\mathbf{\Lambda}} \tilde{\mathbf{u}}_{\mathbf{\Lambda}} - \mathbf{A}_{\mathbf{\Lambda}} \tilde{\mathbf{u}}_{\mathbf{\Lambda}}, \mathbf{u}_{\mathbf{\Lambda}} - \tilde{\mathbf{u}}_{\mathbf{\Lambda}} \rangle_{\ell_2} \\ &\leq c_1^{-1} (\|(\mathbf{A}_{\mathbf{\Lambda}} - \tilde{\mathbf{A}}_{\mathbf{\Lambda}}) \tilde{\mathbf{u}}_{\mathbf{\Lambda}}\|_{\ell_2} \|\mathbf{u}_{\mathbf{\Lambda}} - \tilde{\mathbf{u}}_{\mathbf{\Lambda}}\|_{\ell_2} + \langle \mathbf{f}_{\mathbf{\Lambda}} - \tilde{\mathbf{f}}_{\mathbf{\Lambda}} - \tilde{\mathbf{s}}_{\mathbf{\Lambda}}, \mathbf{u}_{\mathbf{\Lambda}} - \tilde{\mathbf{u}}_{\mathbf{\Lambda}} \rangle_{\ell_2}) \\ &\leq c_1^{-1} \|\mathbf{u}_{\mathbf{\Lambda}} - \tilde{\mathbf{u}}_{\mathbf{\Lambda}}\|_{\ell_2} (\eta_{\mathbf{A}} \|\tilde{\mathbf{u}}_{\mathbf{\Lambda}}\|_{\ell_2} + \eta_{\mathbf{f}} + \text{tol}_{\text{iter}}). \end{aligned}$$

From estimate (3.4) we finally get

$$(3.5) \quad \|\tilde{\mathbf{u}}_{\mathbf{\Lambda}}\|_{\ell_2} \leq (c_1 - \eta_{\mathbf{A}})^{-1} \|\tilde{\mathbf{f}}_{\mathbf{\Lambda}} + \tilde{\mathbf{s}}_{\mathbf{\Lambda}}\|_{\ell_2} \leq (c_1 - \eta_{\mathbf{A}})^{-1} (\eta_{\mathbf{f}} + \|\mathbf{f}_{\mathbf{\Lambda}}\|_{\ell_2} + \text{tol}_{\text{iter}}),$$

which proves the claim. \square

From Proposition 3.3, we infer that there is no gain if one of the tolerances $\eta_{\mathbf{f}}$, $\eta_{\mathbf{A}}$ or tol_{iter} is much smaller than the others. For this reason, we assume from now on that

$$\eta_{\mathbf{f}} \sim \text{tol}_{\text{iter}}, \text{ and } \eta_{\mathbf{A}} < \min\{c_1, \text{tol}_{\text{iter}}\},$$

so that (3.3) can be replaced by $\|\mathbf{u}_{\mathbf{\Lambda}} - \tilde{\mathbf{u}}_{\mathbf{\Lambda}}\|_{\ell_2} \lesssim \text{tol}_{\text{iter}} \|\mathbf{f}_{\mathbf{\Lambda}}\|_{\ell_2}$. In particular, we have in mind to use a sparse, compressed matrix $\mathbf{A}_{J,\mathbf{\Lambda}} = \mathbf{P}_{\mathbf{\Lambda}} \mathbf{A}_J|_{\ell_2(\mathbf{\Lambda})}$ for $\mathbf{A} \in \mathcal{B}_s$ to solve the Galerkin system (2.9) approximately. But in order to use the results

from Proposition 3.3, we should have (at least rough) estimates of the constants α_j , β_j and s in (2.13). Otherwise, the above error estimate for the perturbed linear system are no longer valid. Under the assumption that this is possible, we propose the following routine to solve appearing linear systems.

Algorithm 7 $\mathbf{LINSOLVE}[\mathbf{\Lambda}, \mathbf{w}_{\mathbf{\Lambda}}, \text{tol}_{\text{iter}}] \rightarrow \tilde{\mathbf{u}}_{\mathbf{\Lambda}}$

- 1: Estimate $J \in \mathbb{N}$ such that $\|\mathbf{A}_{\mathbf{\Lambda}} - \mathbf{A}_{J,\mathbf{\Lambda}}\|_{\ell_2} < \min\{c_1, \text{tol}_{\text{iter}}\}$.
 - 2: Compute $\tilde{\mathbf{f}}_{\mathbf{\Lambda}}$ such that $\|\mathbf{f}_{\mathbf{\Lambda}} - \tilde{\mathbf{f}}_{\mathbf{\Lambda}}\|_{\ell_2} < \text{tol}_{\text{iter}}$.
 - 3: Use a linear system solver like CG or GMRES with initial guess $\mathbf{w}_{\mathbf{\Lambda}}$ to compute $\tilde{\mathbf{u}}_{\mathbf{\Lambda}}$ such that $\|\mathbf{A}_{J,\mathbf{\Lambda}}\tilde{\mathbf{u}}_{\mathbf{\Lambda}} - \tilde{\mathbf{f}}_{\mathbf{\Lambda}}\|_{\ell_2} \leq \text{tol}_{\text{iter}}$.
-

As the assembling of $\tilde{\mathbf{f}}_{\mathbf{\Lambda}}$ and of the sparse matrix $\mathbf{A}_{J,\mathbf{\Lambda}}$ can be obtained in $\mathcal{O}(\#\mathbf{\Lambda})$ operations, the complexity of $\mathbf{LINSOLVE}$ is of the same order. Moreover, it holds $\|\mathbf{u}_{\mathbf{\Lambda}} - \tilde{\mathbf{u}}_{\mathbf{\Lambda}}\|_{\ell_2} \lesssim \text{tol}_{\text{iter}}\|\mathbf{f}_{\mathbf{\Lambda}}\|_{\ell_2}$.

3.3.2. Residual computation. The main part of the computation time of the **AD-WAV** algorithm is used by **APPLY** for the computation of index sets satisfying the saturation property (2.15). Alternatively, coming from an index set $\mathbf{\Lambda}$ and the corresponding Galerkin system (2.9), one can compute a finite so called *security zone* $\hat{\mathbf{\Lambda}} \supset \mathbf{\Lambda}$ using Algorithm 8 (see [28, p.235]) for a constant $c > 0$ using the following notation for $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_n) \in \mathbf{\Lambda}$, namely $|\boldsymbol{\lambda}|_1 := |\lambda_1| + \dots + |\lambda_n|$.

Algorithm 8 $\mathbf{C}[\mathbf{\Lambda}, c] \rightarrow \hat{\mathbf{\Lambda}}$

- 1: For each $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_n) \in \mathbf{\Lambda}$, compute for $i = 1, \dots, n$

$$\mathcal{C}(\lambda_i, c) := \{\mu \in \mathcal{J}^{(i)} : |\text{supp } \psi_{\mu} \cap c \cdot \text{supp } \psi_{\lambda_i}| > 0, |\lambda_i| \leq |\mu| \leq |\lambda_i| + 1\}$$
 where $c \cdot \text{supp } \psi_{\lambda}$ is the contracted support of ψ_{λ} by the factor c around its barycenter.
 - 2: Define $\mathcal{C}(\boldsymbol{\lambda}, c) := \{\boldsymbol{\mu} \in \mathcal{C}(\lambda_1, c) \times \dots \times \mathcal{C}(\lambda_n, c) : |\boldsymbol{\mu}|_1 \leq |\boldsymbol{\lambda}|_1 + 1\}$.
 - 3: Set $\hat{\mathbf{\Lambda}} := \bigcup_{\boldsymbol{\lambda} \in \mathbf{\Lambda}} \mathcal{C}(\boldsymbol{\lambda}, c)$.
-

Depending on the underlying wavelet basis, there exists a constant C such that $\#\mathcal{C}(\boldsymbol{\lambda}, c) \leq (n+1)C^n$ where C is chosen such that

$$\begin{aligned} \max_{i \in \{1, \dots, n\}} \max_{\lambda \in \mathcal{J}^{(i)}} \#\{\mu \in \mathcal{J}^{(i)} : |\text{supp } \psi_{\mu} \cap c \cdot \text{supp } \psi_{\lambda}| > 0, |\lambda| = |\mu|\} &\leq C, \text{ and} \\ \max_{i \in \{1, \dots, n\}} \max_{\lambda \in \mathcal{J}^{(i)}} \#\{\mu \in \mathcal{J}^{(i)} : |\text{supp } \psi_{\mu} \cap c \cdot \text{supp } \psi_{\lambda}| > 0, |\lambda| = |\mu| + 1\} &\leq C. \end{aligned}$$

Thus, the overall complexity of \mathbf{C} is bounded by some multiple of $(n+1)C^n \cdot \#\mathbf{\Lambda}$. We stress that the particular construction of a security zone $\hat{\mathbf{\Lambda}}$ is crucial to solve an operator equation on an unbounded domain. As it can be seen from Algorithm 8, $\mathbf{\Lambda}$ is not only expanded by wavelets on higher levels, but also by additional scaling functions. This will later on permit an adaptive truncation of the formerly unbounded computational domain.

As an estimate of the (full) residual $\mathbf{A}\mathbf{u}_{\mathbf{\Lambda}} - \mathbf{f}$ of the Galerkin solution $\mathbf{u}_{\mathbf{\Lambda}}$ from (2.9), one now takes

$$(3.6) \quad \mathbf{r}_{\hat{\mathbf{\Lambda}}} := \mathbf{P}_{\hat{\mathbf{\Lambda}}}(\mathbf{A}\mathbf{u}_{\mathbf{\Lambda}} - \mathbf{f}),$$

where $\widehat{\Lambda}$ is the output of $\mathbf{C}[\Lambda, c]$. To reduce the complexity of the residual computation, we alternatively compute

$$(3.7) \quad \widetilde{\mathbf{r}}_{\widehat{\Lambda}} := \mathbf{P}_{\widehat{\Lambda}}(\mathbf{A}_{J,\Lambda}\widetilde{\mathbf{u}}_{\Lambda} - \widetilde{\mathbf{f}}),$$

using the compressed matrix $\mathbf{A}_{J,\Lambda}$ and an approximate solution $\widetilde{\mathbf{u}}_{\Lambda}$ from (3.2). For the next result we need the following notation: If \mathbf{v}_{Λ} is a vector in $\ell_2(\Lambda)$ with finite support Λ , then $\mathbf{v}_{\widehat{\Lambda}}$ denotes its extension by zeros to $\widehat{\Lambda}$.

Proposition 3.4. *Let $\Lambda, \widehat{\Lambda}$ be finite subsets of \mathbf{J} with $\widehat{\Lambda} \supset \Lambda$ and assume that the assumptions from Proposition 3.3 hold. Then, by setting $\widetilde{\mathbf{A}}_{\widehat{\Lambda}} := \mathbf{A}_{J,\widehat{\Lambda}}$ for sufficiently large J , we have*

$$(3.8) \quad \|\widetilde{\mathbf{r}}_{\widehat{\Lambda}} - \mathbf{r}_{\widehat{\Lambda}}\|_{\ell_2} \lesssim \text{tol}_{\text{iter}} \|\mathbf{f}_{\widehat{\Lambda}}\|_{\ell_2}.$$

Proof. By definition, we have for $\mathbf{v}_{\Lambda} \in \ell_2(\Lambda)$ that $\|\mathbf{v}_{\widehat{\Lambda}}\|_{\ell_2} = \|\mathbf{v}_{\Lambda}\|_{\ell_2}$ and, moreover, $\mathbf{A}_{\widehat{\Lambda}}\mathbf{v}_{\widehat{\Lambda}} = \mathbf{P}_{\widehat{\Lambda}}\mathbf{A}\mathbf{v}_{\Lambda}$. Using this, we get the following estimate:

$$\begin{aligned} \|(\mathbf{A}_{J,\widehat{\Lambda}}\widetilde{\mathbf{u}}_{\widehat{\Lambda}} - \widetilde{\mathbf{f}}_{\widehat{\Lambda}}) - (\mathbf{A}_{\widehat{\Lambda}}\mathbf{u}_{\widehat{\Lambda}} - \mathbf{f}_{\widehat{\Lambda}})\|_{\ell_2} &\leq \|\mathbf{u}_{\Lambda}\|_{\ell_2} \|\mathbf{A}_{\widehat{\Lambda}} - \mathbf{A}_{J,\widehat{\Lambda}}\| \\ &\quad + \|\mathbf{A}_{J,\widehat{\Lambda}}\| \|\widetilde{\mathbf{u}}_{\Lambda} - \mathbf{u}_{\Lambda}\|_{\ell_2} + \|\mathbf{f}_{\widehat{\Lambda}} - \widetilde{\mathbf{f}}_{\widehat{\Lambda}}\|_{\ell_2}. \end{aligned}$$

From Proposition 3.3 we get that $\|\widetilde{\mathbf{u}}_{\Lambda} - \mathbf{u}_{\Lambda}\|_{\ell_2} \lesssim \text{tol}_{\text{iter}} \|\mathbf{f}_{\Lambda}\|_{\ell_2}$. Moreover, similar as in (3.5), we see that $\|\mathbf{u}_{\Lambda}\|_{\ell_2} \|\mathbf{A}_{\widehat{\Lambda}} - \mathbf{A}_{J,\widehat{\Lambda}}\| \lesssim \text{tol}_{\text{iter}} \|\mathbf{f}_{\Lambda}\|_{\ell_2}$. \square

At this point we remember that replacing the exact, infinite residual $\mathbf{A}\mathbf{u}_{\Lambda} - \mathbf{f}$ by $\widetilde{\mathbf{r}}_{\widehat{\Lambda}}$ is an heuristic approach. On one hand, there is no proof of the existence of $0 < \beta < 1$ such that $\|\mathbf{P}_{\widehat{\Lambda}}(\mathbf{A}\mathbf{u}_{\Lambda} - \mathbf{f})\|_{\ell_2} \geq \beta \|\mathbf{A}\mathbf{u}_{\Lambda} - \mathbf{f}\|_{\ell_2}$. On the other hand, the advantage is that we do not need **APPLY** or **RHS**. If we use $\widehat{\Lambda} = \mathbf{C}[\Lambda, c]$, then due the sparsity of $\mathbf{A}_{J,\widehat{\Lambda}}$ and the fact that $\#\widehat{\Lambda} \lesssim \#\Lambda$, the routine **RESIDUAL** is of complexity $\mathcal{O}(\#\Lambda)$.

Algorithm 9 RESIDUAL $[\widehat{\Lambda}, \widetilde{\mathbf{u}}_{\Lambda}, \text{tol}_{\text{iter}}] \rightarrow \widetilde{\mathbf{r}}_{\widehat{\Lambda}}$

- 1: Estimate $J \in \mathbb{N}$ such that $\|\mathbf{A}_{\widehat{\Lambda}} - \mathbf{A}_{J,\widehat{\Lambda}}\|_{\ell_2} < \min\{c_1, \text{tol}_{\text{iter}}\}$.
 - 2: Compute $\widetilde{\mathbf{r}}_{\widehat{\Lambda}}$ according to (3.7).
-

3.3.3. Coefficient thresholding. Obviously, if we call iteratively $\Lambda^{(k+1)} = \mathbf{C}(\Lambda^{(k)}, c)$ starting with some initial set $\Lambda^{(0)}$, then the sizes of the index sets $(\Lambda^{(k)})_{k \in \mathbb{N}}$ grow exponentially fast. For this reason, we have to keep the index sets for which we call **C** small. This is realized by thresholding the wavelet coefficients in $\widetilde{\mathbf{u}}_{\Lambda^{(k)}}$ and in the estimated residual $\widetilde{\mathbf{r}}_{\Lambda^{(k)}}$. For this purpose, Algorithm 10 realizes a threshold on finitely supported vectors \mathbf{v} and returns a vector $\bar{\mathbf{v}}$ such that $\|\mathbf{v} - \bar{\mathbf{v}}\|_{\ell_2} \leq \delta$ for a given tolerance $0 < \delta$. Here, also approximate sorting procedures from [1, 15] can be used, so that **THRESH** is also of linear complexity.

For convenience, let us now see what the effect of **THRESH** is when it is used to threshold the output of **LINSOLVE**. Therefore, let \mathbf{u}_{Λ} be the solution of (2.9) and $\widetilde{\mathbf{u}}_{\Lambda}$ be the output of **LINSOLVE** $[\Lambda, \mathbf{w}_{\Lambda}, \text{tol}_{\text{iter}}]$ for some starting point \mathbf{w}_{Λ} . Then, as $\|\mathbf{u}_{\Lambda} - \widetilde{\mathbf{u}}_{\Lambda}\|_{\ell_2} \lesssim \text{tol}_{\text{iter}} \|\mathbf{f}_{\Lambda}\|_{\ell_2}$, it holds for $\bar{\mathbf{u}} = \mathbf{THRESH}[\widetilde{\mathbf{u}}_{\Lambda}, \text{tol}_{\text{iter}}]$ that

$$\|\mathbf{u}_{\Lambda} - \bar{\mathbf{u}}\|_{\ell_2} \lesssim \text{tol}_{\text{iter}}(1 + \|\mathbf{f}_{\Lambda}\|_{\ell_2}),$$

Algorithm 10 THRESH $[\mathbf{v}, \delta] \rightarrow \bar{\mathbf{v}}$

-
- 1: Sort the vector $\mathbf{v} = (\mathbf{v}_\lambda)_{\lambda \in \text{supp } \mathbf{v}}$ by decreasing order which yields the sorted vector $\mathbf{v}^* = (\mathbf{v}_{(i, \lambda_i)}^*)_{i=1, \dots, M}$ where $M := \#(\text{supp } \mathbf{v})$ and (i, λ_i) for $i = 1, \dots, M$ indicate the ordering in \mathbf{v}^* such as the corresponding index in $\text{supp } \mathbf{v}$.
 - 2: Compute $\|\mathbf{v}\|_{\ell_2}$. The vector $\bar{\mathbf{v}}$ and its support Λ are given by $\bar{\mathbf{v}} := (\mathbf{v}_{\lambda_i})_{i=1, \dots, K}$ and $\Lambda := \{\lambda_1, \dots, \lambda_K\}$ where K is the smallest integer such that $\sum_{i=1}^K |\mathbf{v}_{(i, \lambda_i)}^*|^2 \geq \|\mathbf{v}\|_{\ell_2}^2 - \delta^2$.
-

where $\text{supp } \bar{\mathbf{u}} \subseteq \Lambda$. That is, we get an approximation of order tol_{iter} to the Galerkin solution \mathbf{u}_Λ but in general with a much smaller support. This observation can also be made for the residual computation. Let \mathbf{r}_Λ be the residual defined in (3.6) and $\tilde{\mathbf{r}}_{\hat{\Lambda}} = \mathbf{RESIDUAL}[\hat{\Lambda}, \tilde{\mathbf{u}}_\Lambda, \text{tol}_{\text{iter}}]$. Then, by Proposition 3.4, $\|\mathbf{r}_{\hat{\Lambda}} - \tilde{\mathbf{r}}_{\hat{\Lambda}}\|_{\ell_2} \lesssim \text{tol}_{\text{iter}} \|\mathbf{f}_{\hat{\Lambda}}\|_{\ell_2}$. Thus, for $\bar{\mathbf{r}} = \mathbf{THRESH}[\tilde{\mathbf{r}}_{\hat{\Lambda}}, \text{tol}_{\text{iter}}]$, it holds $\|\mathbf{r}_{\hat{\Lambda}} - \bar{\mathbf{r}}\|_{\ell_2} \lesssim \text{tol}_{\text{iter}} (1 + \|\mathbf{f}_{\hat{\Lambda}}\|_{\ell_2})$, where, as above, the support $\bar{\mathbf{r}}$ is in general much smaller than $\hat{\Lambda}$.

3.3.4. The simplified algorithm S-ADWAV. Now that we have all necessary routines together, we can now describe the complete algorithm. The **S-ADWAV** algorithm described in detail in Algorithm 11 below computes in each iteration an approximate solution $\mathbf{w}^{(k)}$ to the Galerkin system $\mathbf{A}_{\Lambda_k^{\text{cand.}}} \mathbf{w}^{(k)} = \mathbf{f}_{\Lambda_k^{\text{cand.}}}$ where $\Lambda_k^{\text{cand.}}$ is referred to as the set of *candidate* indices, i.e., indices that *can be activated* in the current iteration. The target precision for solving the linear system is tol_{iter} . As already stated above, we have to keep index sets for which we call the **C** routine small. Therefore, with the (approximate) Galerkin solution $\mathbf{w}^{(k)}$ at hand, we threshold this vector in order to obtain the *active* wavelet coefficients $\mathbf{u}^{(k)}$ that satisfy $\|\mathbf{u}^{(k)} - \mathbf{w}^{(k)}\|_{\ell_2} \leq \text{tol}_{\text{iter}}$. Its support $\Lambda^{(k)} := \text{supp } \mathbf{u}^{(k)}$ is referred to as the set of *active* indices. Around the support of $\mathbf{u}^{(k)}$, the security zone $\hat{\Lambda}_k$ is constructed using the routine **C** and the residual $\mathbf{r}_{\hat{\Lambda}_k} = \mathbf{P}_{\hat{\Lambda}_k}(\mathbf{A}\mathbf{u}^{(k)} - \mathbf{f})$ is computed. If $\|\mathbf{r}_{\hat{\Lambda}_k}\|_{\ell_2}$ is smaller than the given target tolerance, then we accept $\mathbf{u}^{(k)}$ as solution. Otherwise, a new candidate set of activable indices $\Lambda_{k+1}^{\text{cand.}}$ is constructed by thresholding the residual $\mathbf{r}_{\hat{\Lambda}}$. As we always use the same tolerance tol_{iter} for thresholding and the numerical solution of the Galerkin system, the approximation errors we generate are all of order tol_{iter} (see Propositions 3.3 and 3.4 such as Paragraph 3.3.3). But if we fix this tolerance, it may happen that the algorithm stagnates before the target accuracy ε is reached. Namely, by thresholding the approximate Galerkin solution $\mathbf{w}^{(k)}$, it may occur that no higher levels or translation indices on the coarsest level are added in the course of the algorithm and we end up with $\Lambda^{(k)} = \Lambda^{(k+1)}$. Therefore, in addition to the algorithm described in [2] and [28], we decrease the threshold tolerance tol_{iter} by factor $\frac{1}{2}$ if the difference of the relative residuals of two iterations is too close to zero. Thus, in case of $\Lambda^{(k+1)} = \Lambda^{(k)}$, the threshold tolerance is decreased and therefore, we obtain $\mathbf{w}^{(k)} = \mathbf{w}^{(k+1)}$ but $\text{supp } \mathbf{u}^{(k+1)} \supseteq \text{supp } \mathbf{u}^{(k)}$ which means that also finer information on high levels or further translations on the coarsest level remain in the set of active indices. Moreover, as cycles of type $\Lambda_k = \Lambda_{k+m}$ for some $k \geq 2$ cannot not be theoretically excluded, we add an inner loop with a maximal number M of iterations which prevents such loops and ensures that the tolerance tol_{iter} decreases.

Algorithm 11 $[\mathbf{u}(\varepsilon), \Lambda(\varepsilon)] = \mathbf{S-ADWAV}[\varepsilon]$

Let $h > 0$ be a control width, M a fixed number of inner loops, $c > 0$ and $\text{tol}_{\text{iter}} > 0$ an initial tolerance.

- 1: $[\mathbf{j}_0, \Lambda_{1,1}^{\text{cand.}}] = \mathbf{INITIALIZE}$
- 2: **for** $k = 1, 2, 3, \dots$ **do**
- 3: **for** $m = 1, 2, \dots, M$ **do**
- 4: $\mathbf{w}^{(k,m)} = \mathbf{LINSOLVE}[\Lambda_{k,m}^{\text{cand.}}, \mathbf{u}^{(k-1,m)}, \text{tol}_{\text{iter}}]$
- 5: $\mathbf{u}^{(k,m)} = \mathbf{THRESH}[\mathbf{w}^{(k,m)}, \text{tol}_{\text{iter}}]$
- 6: $\Lambda^{(k,m)} = \text{supp } \mathbf{u}^{(k,m)}$; $\widehat{\Lambda}_{k,m} = \mathbf{C}[\Lambda^{(k,m)}, c]$
- 7: $\mathbf{r}^{(k,m)} = \mathbf{RESIDUAL}[\widehat{\Lambda}_{k,m}, \mathbf{u}^{(k,m)}, \text{tol}_{\text{iter}}]$
- 8: **if** $\|\mathbf{r}^{(k,m)}\|_{\ell_2} \leq \varepsilon \|\mathbf{f}_{\widehat{\Lambda}_{k,m}}\|_{\ell_2}$ **then**
- 9: $\mathbf{u}(\varepsilon) := \mathbf{u}^{(k,m)}$, $\Lambda(\varepsilon) := \Lambda^{(k,m)}$; **EXIT**
- 10: **end if**
- 11: $\bar{\mathbf{r}}^{(k)} = \mathbf{THRESH}[\mathbf{r}^{(k,m)}, \text{tol}_{\text{iter}}]$; $\Lambda_{k+1,m}^{\text{cand.}} = \text{supp } \mathbf{u}^{(k)} \cup \text{supp } \bar{\mathbf{r}}^{(k,m)}$
- 12: **if** $\left| \frac{\|\mathbf{r}^{(k,m)}\|_{\ell_2}}{\|\mathbf{P}_{\widehat{\Lambda}_{k,m}} \mathbf{f}\|_{\ell_2}} - \frac{\|\mathbf{r}^{(k-1,m)}\|_{\ell_2}}{\|\mathbf{P}_{\widehat{\Lambda}_{k-1,m}} \mathbf{f}\|_{\ell_2}} \right| < h$ **then**
- 13: **BREAK**
- 14: **end if**
- 15: **end for**
- 16: $\text{tol}_{\text{iter}} = \frac{1}{2} \text{tol}_{\text{iter}}$
- 17: **end for**

At this point, it is important to note that the adaptive truncation of a computational domain, i.e., the support of the computed solutions $\text{supp } (\mathbf{u}^{(m,k)})^T \mathbf{D}\Psi$, is done implicitly. Every time $\mathbf{C}[\Lambda^{(k,m)}, c]$ is called, additional scaling function indices on the coarsest level are added to the security zone $\widehat{\Lambda}^{(k,m)}$. If now these added scaling function indices are relevant for a more precise approximation of the solution, their corresponding value in $\mathbf{r}^{(k,m)}$ is relatively large and they will be added in the new candidate $\Lambda_{k+1,m}^{\text{cand.}}$ set after the call of $\mathbf{THRESH}[\mathbf{r}^{(k,m)}, \text{tol}_{\text{iter}}]$. This proceeding provides the possibility that in each iteration, the computational domain can be extended, but also truncated as we have another call of \mathbf{THRESH} after the solution of the Galerkin system. We emphasize that this is in difference to \mathbf{ADWAV} where we do not have this control. Therefore, the call of \mathbf{THRESH} in line 5 of Algorithm 11 is not only necessary to reduce the complexity but also to estimate if further scaling function translation indices are relevant. Otherwise, the computational domain would grow too fast.

3.3.5. Complexity. Although there is no proof for the convergence of $\mathbf{S-ADWAV}$, we emphasize that the complexity in each iteration is bounded by some constant multiple of $\mathcal{O}(\Lambda^{(k,m)})$ which can be derived directly from the complexity of the subroutines.

4. NUMERICAL EXPERIMENTS

In this section, we demonstrate the performance of the presented adaptive wavelet algorithms for some examples in 1D and 2D where we used the biorthogonal wavelets from [9]. We compare approximation rates with a best N-term approximation and

demonstrate the adaptive truncation of the computational domain. All examples are realized in C++ using the software libraries FLENS and LAWA, [19, 27].

4.1. **Examples in one dimension.** We start with some examples in 1D, namely instances of a Helmholtz problem as well as a convection-diffusion problem.

4.1.1. *Helmholtz problems.* We start with the following Helmholtz problem on $H^1(\mathbb{R})$,

$$(4.1) \quad -u''(x) + u(x) = f(x), x \in \mathbb{R}, \quad \lim_{|x| \rightarrow \infty} u(x) = 0, u \in H^1(\mathbb{R}),$$

for $f \in H^{-1}(\mathbb{R})$. Note that the operator fulfills all required assumptions. We consider six different choices of the right-hand side f which permit a reference solution in closed form. The solutions are shown in Figure 4.1. We have chosen these particular examples due to the following reasons:

- (P1) Global, smooth solution.
- (P2) Global solution with peak, large significant domain.
- (P3) Global solution with 2 peaks.
- (P4) Compactly supported, smooth solution.
- (P5) Compactly supported function with strong gradient.
- (P6) Global, piecewise defined solution with strong peak.

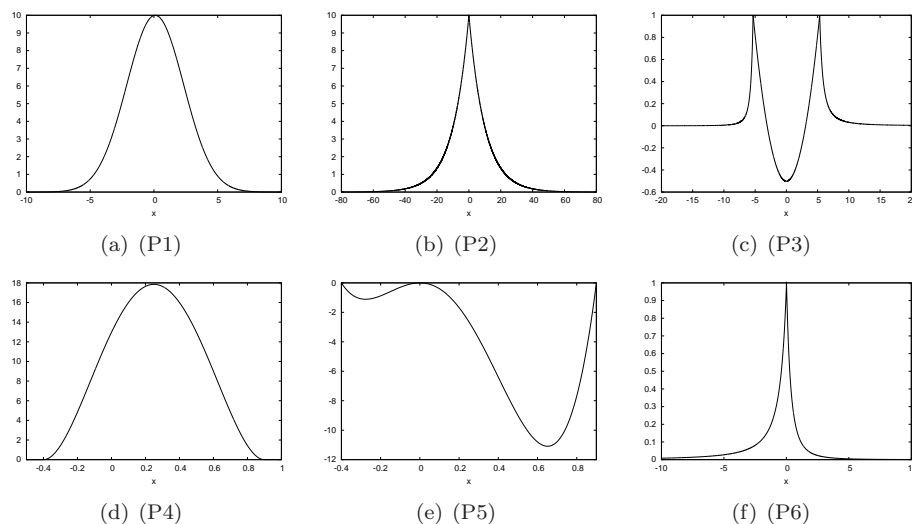
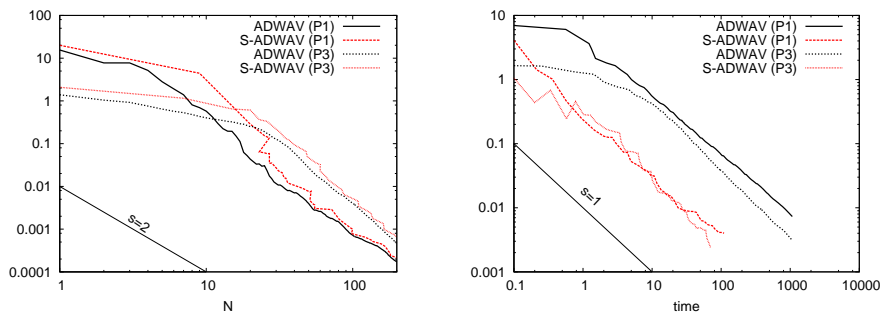


FIGURE 4.1. Solutions u_i for the Helmholtz problems (P1)-(P6).

4.1.2. *Comparison of the adaptive algorithms.* We start by comparing **S-ADWAV** and **ADWAV** with respect to approximation rates and CPU times. For **ADWAV** we used the parameters $\theta = 0.5$, such as $\alpha = 0.307$, $\omega = 0.01$, $\gamma = 0.016$ and $\kappa(\mathbf{A}) \approx 9.3$ for $d = \tilde{d} = 2$ and $\alpha = 0.185$, $\omega = 0.01$, $\gamma = 0.005$ and $\kappa(\mathbf{A}) \approx 26.5$. The parameters were chosen in a way that **GROW** terminates within one iteration and α sufficiently large such that the constant for the saturation property (2.15) $\frac{\alpha - \omega}{1 + \omega}$ in (2.16) takes a large value. At the same time ω should not be too small as then **APPLY** is called with a very small target accuracy which deteriorates the

quantitative efficiency. Concerning the parameters of **S-ADWAV**, here we used $\text{tol}_{\text{iter}} = 0.1$, $c = 0.25$, $h = 0.0001$ and $M = 10$ for all examples.

Representative for all examples, we consider (P1) and (P3) in Figure 4.2. The graphs on the left-hand side of Figure 4.2 show the error versus the number of used wavelets, i.e., the slope indicates the rate of convergence. In all cases that we have tested, both schemes show the same asymptotic behavior, but **ADWAV** is quantitatively slightly better and the sizes of the estimated index set $\Lambda^{(k)}$ do not decrease as in Figure 4.2(a) **S-ADWAV** for (P1). In the graphs on the right-hand side of Figure 4.2, we have plotted the corresponding error estimator, i.e., the output ν_k of **GROW** and the output of **RESIDUAL**, against the CPU time (wall clock time in seconds). In all our test cases, **S-ADWAV** performs slightly better, again having the same asymptotic performance. Although we have no proven convergence or optimality result for **S-ADWAV**, we observe the same asymptotic behaviour as for **ADWAV**, also for the other examples. However, comparing the required CPU times we also have to take into account that the algorithms require different data structures. Optimized data structures for adaptive algorithms are a current field of research and are not our subject here.



(a) (P1) and (P3) for $d = \tilde{d} = 3$ and $j_0 = -3$ (b) (P1) and (P3) for $d = \tilde{d} = 2$ and $j_0 = -3$

FIGURE 4.2. Comparison of **S-ADWAV** and **ADWAV**. Left: error in $\|\cdot\|_{\mathbf{A}}$, right: Required CPU time.

By this first experiment, we do not intend to make a suggestion which algorithm to use. We rather aim to show that *both* algorithms can be used to solve PDE problems on \mathbb{R}^n and also to highlight differences between the algorithms without any weighting. In the following, we present results for **S-ADWAV** but keep in mind that **ADWAV** would give comparable numbers if the involved constants are estimated sufficiently well.

4.1.3. Convergence Rates. Now, we use **S-ADWAV** to solve the Helmholtz problems (P2), (P4), (P5) and (P6). In Figure 4.3, we show the rate of convergence and compare it with the rate of a best N -term approximation measured in $\|\cdot\|_{\ell_2}$. In Figure 4.3, we use linear wavelets ($d = \tilde{d} = 2$) and quadratic ones ($d = \tilde{d} = 3$). In all cases, we obtain the optimal rate. The error of the solution by **S-ADWAV** is always measured in $\|\cdot\|_{\mathbf{A}}$.

We also see in Figure 4.3(c) that a best N -term approximation might converge faster than with the best possible rate if N is not sufficiently large. Even in this

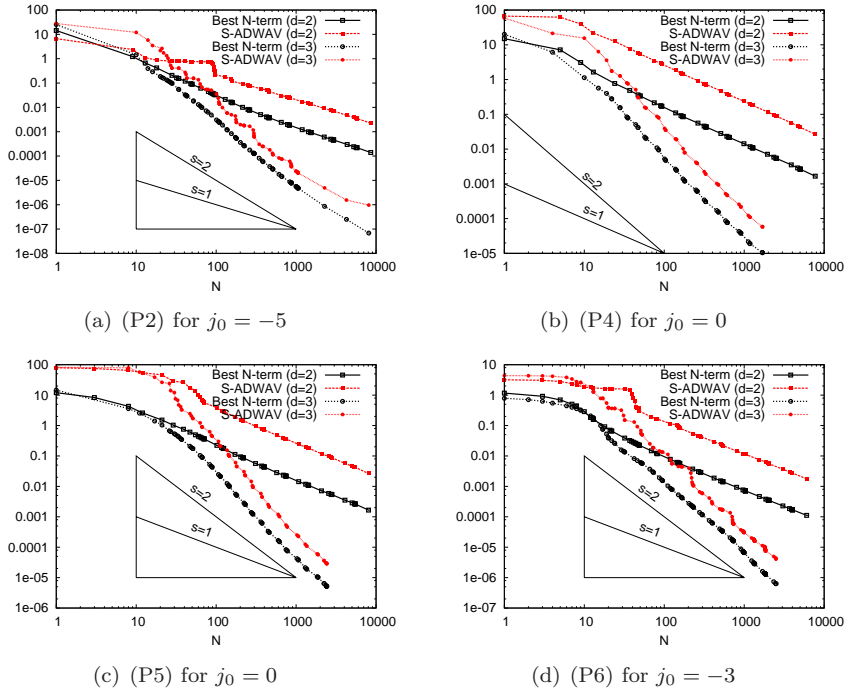


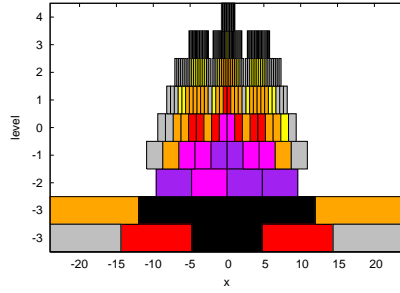
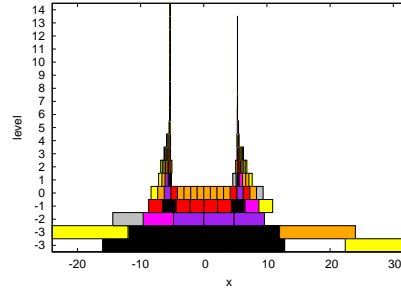
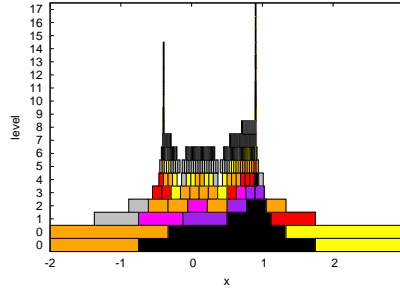
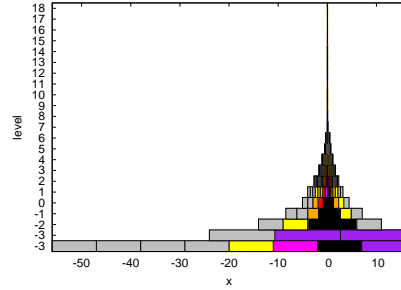
FIGURE 4.3. Convergence rates for (P2), (P4), (P5) and (P6) for $d = 2$ and $d = 3$ in comparison to a best N-term approximation.

case, however, it is remarkable that **S-ADWAV** still converges as fast as a best N-term approximation. The approximation rates in Figure 4.3 are computed for the indicated minimal level determined by **INITIALIZE**.

4.1.4. *Adaptive truncation of the computational domain.* Besides the convergence in $\|\cdot\|_{\mathbf{A}}$, it is particularly interesting how the adaptive wavelet algorithms adaptively truncate the computational domain, i.e., the support of the calculated approximate solution $u_{\mathbf{\Lambda}} = \mathbf{u}_{\mathbf{\Lambda}}^T \Psi$. Results for (P1), (P3), (P5) and (P6) can be found in Table 4.1 and Figure 4.4. For all examples, we started **S-ADWAV** with an initial set $\mathbf{\Lambda}_0$ consisting only of one scaling index such as **ADWAV** which starts with an empty set.

In Figure 4.4, the supports of the approximate solution obtained by **S-ADWAV** are shown. These plots are to be read as follows: On the x-axis, the support $\text{supp } u_{\mathbf{\Lambda}}$ of the corresponding function is shown. Each plotted rectangle corresponds to one index $\lambda \in \mathbf{\Lambda}$. The coloring of a rectangle for the index λ indicates the relative size of the corresponding coefficient \mathbf{u}_{λ} . Here, the more intensive the color, the larger the corresponding coefficient, black indicates a relative large coefficients whereas gray indicates a relatively small coefficients. On the y-axis, the level on the indices is plotted where at the bottom one finds the indices corresponding to scaling functions and above those corresponding to wavelets. The number of rectangles on each level indicates how many scaling functions respectively wavelets are used to cover the indicated interval range.

	S-ADWAV				ADWAV			
	It.	$\#\Lambda$	$[a, b]$	$L_\infty(\mathbb{R})$	It.	$\#\Lambda$	$[a, b]$	$L_\infty(\mathbb{R})$
(P1)	7	54	$[-24, 24]$	$< 3.0 \cdot 10^{-2}$	32	52	$[-24, 24]$	$< 2.0 \cdot 10^{-2}$
	13	90	$[-24, 24]$	$< 7.5 \cdot 10^{-3}$	42	93	$[-24, 24]$	$< 5.5 \cdot 10^{-3}$
	23	233	$[-24, 24]$	$< 1.6 \cdot 10^{-3}$	57	228	$[-24, 24]$	$< 1.7 \cdot 10^{-3}$
(P5)	6	52	$[-2, 3]$	$< 4.5 \cdot 10^{-1}$	51	52	$[-2, 2]$	$< 1.6 \cdot 10^{-1}$
	8	139	$[-2, 3]$	$< 8.5 \cdot 10^{-2}$	91	140	$[-2, 3]$	$< 1.4 \cdot 10^{-1}$
	10	272	$[-2, 3]$	$< 6.5 \cdot 10^{-3}$	123	273	$[-2, 3]$	$< 3.3 \cdot 10^{-2}$
(P6)	21	51	$[-16, 16]$	$< 8.0 \cdot 10^{-3}$	38	56	$[-16, 16]$	$< 3.5 \cdot 10^{-3}$
	30	92	$[-24, 16]$	$< 2.0 \cdot 10^{-3}$	47	94	$[-32, 16]$	$< 1.8 \cdot 10^{-3}$
	50	239	$[-56, 16]$	$< 4.0 \cdot 10^{-4}$	63	229	$[-56, 16]$	$< 6.0 \cdot 10^{-4}$

TABLE 4.1. Adaptive computational domain $[a, b]$ with linear wavelets.(a) (P1) with $\#\Lambda = 233$.(b) (P3) with $\#\Lambda = 127$.(c) (P5) with $\#\Lambda = 299$.(d) (P6) with $\#\Lambda = 239$.FIGURE 4.4. Support of \mathbf{u}_Λ obtained by **S-ADWAV** with $d = \tilde{d} = 2$.

Obviously, **S-ADWAV** is capable to work out singularities of the solution. We see that for (P5), the homogeneous Dirichlet boundary conditions are treated as singularities as the solution u_5 is not smooth on \mathbb{R} . Nevertheless, we observe that the boundary conditions are resolved with increasing level where **S-ADWAV** performs better. For all examples, one can observe also a decreasing $\|\cdot\|_{L_\infty(\mathbb{R})}$ error. Moreover, for an increasing number of iterations, not only singularities are resolved with a higher precision, also the computational domain for non-local examples is enlarged w.r.t. to the current accuracy in the algorithm. It can also be observed

that the asymptotic behaviour of the solution to (P6) is detected by both algorithms. For $x \rightarrow +\infty$, we have $u_5(x) \lesssim \frac{1}{x^3}$, whereas for $x \rightarrow -\infty$, $u_5(x) \lesssim \frac{1}{x^2}$. Thus, we need more scaling functions for $x < 0.01$ to obtain a good approximation than for $x > 0.01$.

4.1.5. *A convection-diffusion problem.* For the Helmholtz examples one might argue that it would also be possible to determine a computational domain a priori and then to use standard methods for PDEs on bounded domains. In order to treat a problem where this is not that obvious, we consider a convection diffusion problem of the form

$$(4.2) \quad -u''(x) + \beta u'(x) + u(x) = f_1(x), \quad x \in \mathbb{R},$$

using the right-hand side from (P1) which also fulfills all required assumptions (cf. e.g. [3]). For increasing values of β , the solution exhibits a strong layer at $x = 0$, see the left part of Figure 4.5. On the right side of Figure 4.5 we see the adaptive truncation of the computational domain. In particular, the layer is detected automatically. We stress the fact that in our setting a stabilization is *not* required, as there is no boundary. However, a stabilization for convection dominated problems for very large values of $|\beta|$ is still necessary.

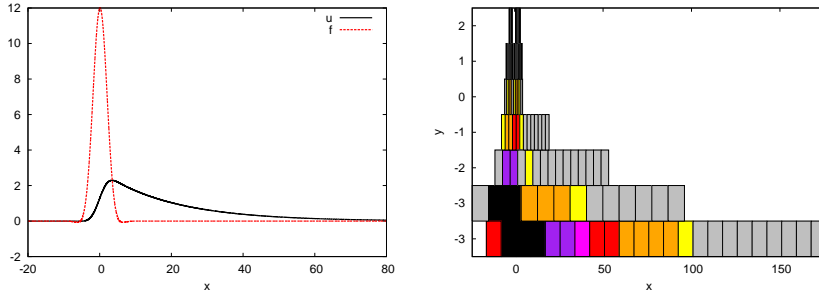


FIGURE 4.5. Solution u , right-hand side f (left), and estimated index set ($d = \tilde{d} = 2$ and $j_0 = -3$) (right) for (4.2) with $\beta = 20$.

4.2. **Examples in two dimensions.** Finally, we consider some bivariate problems. As above, we consider the Helmholtz problem on $H^1(\mathbb{R}^2)$, i.e.,

$$(4.3) \quad -\Delta u + u = f, \quad u \in H^1(\mathbb{R}^2),$$

for $f \in H^{-1}(\mathbb{R}^2)$. Here, we consider the following examples:

$$(P7) \quad u_7(x_1, x_2) := e^{-\frac{(x_1+0.1)^2}{10}} \cdot e^{-\frac{(x_2-0.1)^2}{2}},$$

$$(P8) \quad u_8(x_1, x_2) := e^{-2|x_1-\frac{1}{3}|} \cdot e^{-\frac{(x_2-\frac{1}{3})^2}{10}},$$

$$(P9) \quad u_9(x_1, x_2) := e^{-\sqrt{(x_1-0.1)^2+(x_2-0.1)^2}}.$$

$$(P10) \quad u_{10}(x_1, x_2) = e^{-(2(x_1-0.1)^2+(x_1-0.1)\cdot(x_2-0.1)+(x_2-0.1)^2)}.$$

The tensor product structure of the reference solutions u_7 and u_8 permits us, at last in theory, to obtain the best possible approximation $s^* = d - 1$ for (P7) and (P8). Concerning (P9), to our knowledge there does not exist a theoretical proof as e.g. in [21] that u_9 is sufficiently smooth to obtain the optimal rate. It is known

that $u_9 \in H^1(\mathbb{R}^2)$ but $u_9 \notin H^2(\mathbb{R}^2)$. The function u_{10} is of Schwartz type and thus has the required tensor regularity. We emphasize that the tensor product structure and the symmetry of the solutions was *not* exploited in the numerical solution. We use anisotropic wavelet bases for the simulation.

4.2.1. *Convergence rates.* We shall investigate whether it is possible to obtain the best possible approximation rate $s^* = d - 1$ with **S-ADWAV**. In addition, we compute a best N-term approximation. The error of the computed N-term approximation \mathbf{u}_N is measured in the ℓ_2 -norm, i.e., $\|\mathbf{u} - \mathbf{u}_N\|_{\ell_2}$.

The results for the tensor examples (P7) and (P8) are shown in Figure 4.6. We observe that the best possible convergence rate is not exactly obtained for (P7) or only asymptotically (P8). In particular for (P7) and wavelets of order $d = 3$ with $\mathbf{j}_0 = (-3, -2)$, we only measured a rate of $s = 1.85$ which is still higher than the best possible for an isotropic wavelet basis which would be $s = 1$. For linear wavelets and $\mathbf{j}_0 = (-2, -2)$, $s = 0.95$ was realized. For (P8), we used $\mathbf{j}_0 = (0, -3)$ for linear wavelets and $\mathbf{j}_0 = (1, -3)$ for the quadratic ones. This result is not really surprising since the number of degrees of freedom in two dimensions is much higher than in only one space dimension (curse of dimension), see also [15] for similar observations. Note that this is a matter of the bases that we used and not of the algorithm that in fact performs in an optimal way.

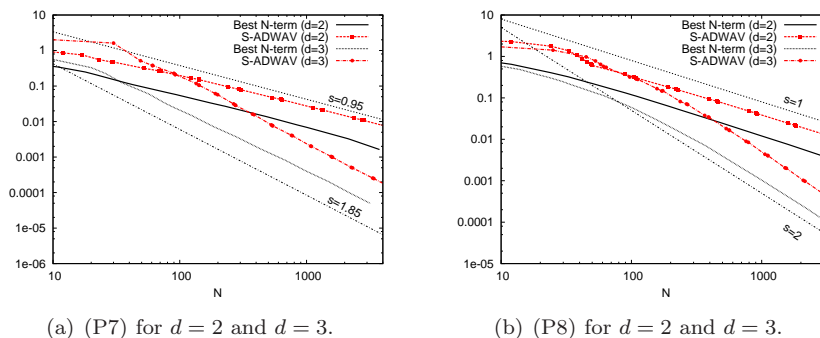
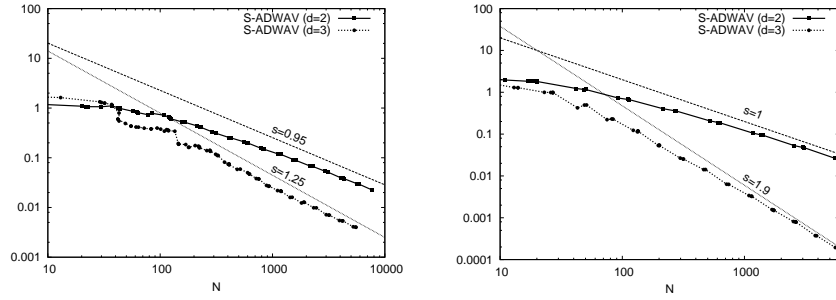


FIGURE 4.6. Convergence rate (measured in $\|\cdot\|_H$) for the 2D examples.

The minimal level \mathbf{j}_0 has been estimated by **INITIALIZE**. We see that the estimates of \mathbf{j}_0 influence the start of the asymptotic regime. Here, the minimal level $\mathbf{j}_0 = (0, -3)$ indicates that the corresponding solution as a larger support in x_2 -direction and therefore, scaling functions in the x_2 -direction should be chosen on a smaller minimal level than in x_1 -direction. This also shows that anisotropic wavelets are useful here.

For (P9) which does not have a tensor structure, we observed that the approximation rate for $d = 2$ is almost the expected optimal value. As it can be seen from Figure 4.7(a), the optimal rate is not reached for $d = 3$ but is still better than in the isotropic case. On the other hand, we do observe the optimal rates for (P10) which is also not of tensor product structure but has the required regularity. Even though we do not have a proof, our results seem to indicate that u_9 is lacking the required tensor regularity.



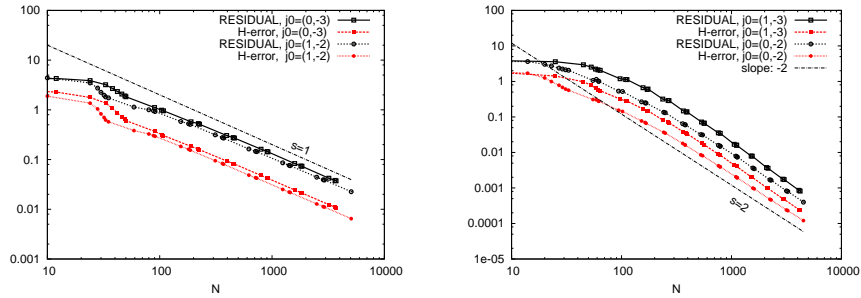
(a) (P9) for $d = 2$ and $d = 3$ with $\mathbf{j}_0 = (0, 0)$. (b) (P10) for $d = 2$ and $d = 3$ with $\mathbf{j}_0 = (0, 0)$.

FIGURE 4.7. Convergence rate (measured in $\|\cdot\|_{\mathbf{A}}$), (P9) left, (P10) right.

Next, we show that it makes sense to invest into a better estimate of the minimal level for higher dimensional computations, see Figure 4.8, where we have tested an improved estimate for \mathbf{j}_0 . We observe that there is a significant quantitative difference. This effect can also be observed in one dimension where it is less significant. To obtain such a better estimate of \mathbf{j}_0 , we computed in **INITIALIZE** the values

$$2^{-\max\{|\lambda_1|, |\lambda_2|\}} \langle f, \psi_{\lambda_1} \otimes \psi_{\lambda_2} \rangle, \quad \lambda_1 \in \Lambda_1, \lambda_2 \in \Lambda_2,$$

where the index sets Λ_1, Λ_2 also contain scaling coefficients on different (!) minimal levels j_0 . Obviously, the computational effort is larger and not all calculated coefficients can later be used in the algorithm. We also see from this experiment that **RESIDUAL** is a reliable estimator for the approximation error which has also been observed for the other examples.



(a) (P8) for $d = 2$ and different \mathbf{j}_0 .

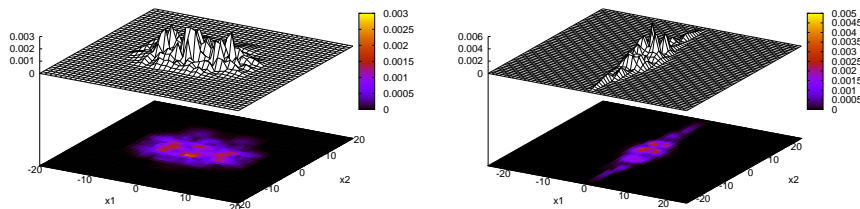
(b) (P8) for $d = 3$ and different \mathbf{j}_0 .

FIGURE 4.8. Convergence rate (measured in $\|\cdot\|_H$) for different \mathbf{j}_0 .

4.2.2. Adaptive truncation of the computational domain. As in the one-dimensional setting, we show that **S-ADWAV** not only produces approximations that converge in the H^1 -norm, but also estimates a convenient computational domain. As we can see from Figure 4.9 and 4.7(a), the algorithm does not need many iterations to find a finite support for the approximation of the solution of (P7).

In case of problem (P8), this estimate takes more iterations due to the singularity in x_1 direction. We see that this singularity is detected and a higher resolution is obtained with an increasing number of iterations.

We remark that the computational domains are not symmetric for none of the two examples as the solutions are not symmetric to a scaling function grid point.



(a) (P7) with $d = 2$ after 15 iterations, $\#\text{supp } \mathbf{u}_\Lambda = 291$.
 (b) (P8) with $d = 2$ after 15 iterations, $\#\text{supp } \mathbf{u}_\Lambda = 226$.

FIGURE 4.9. Absolute errors $|u(x) - u_\Lambda(x)|$.

5. CONCLUSIONS AND OUTLOOK

We have introduced an adaptive wavelet method for operator problems on \mathbb{R}^n . We have shown that the method converges and is asymptotically optimal. We have seen the benefit of using anisotropic wavelet bases in the multivariate case. We have also introduced a simplified adaptive wavelet method without prove of optimality but with very good quantitative numerical results. The performance of the scheme has been demonstrated by a variety of numerical experiments in 1D and 2D. It has been shown that the scheme also performs quantitatively very well.

This opens the door to several questions which will be subject to future research. The extension to different kind of problems (nonlinear, integral equations, obstacle problems, etc.) has already been mentioned in the introduction. Moreover, a coupling with the space-time adaptive method proposed in [25] might be interesting.

REFERENCES

- [1] A. BARINKA, *Fast computation tools for adaptive wavelet schemes*, PhD thesis, RWTH Aachen, 2005.
- [2] S. BERRONE AND T. KOZUBEK, *An adaptive WEM algorithm for solving elliptic boundary value problems in fairly general domains*, SIAM Journal on Scientific Computing, 28 (2006), pp. 2114–2138.
- [3] C. CANUTO AND A. TABACCO, *An anisotropic functional setting for convection-diffusion problems*, East-West Journal of Numerical Mathematics, 9 (2001), pp. 199–231.
- [4] C. CANUTO, A. TABACCO, AND K. URBAN, *The wavelet element method (part I): construction and analysis*, Applied and Computational Harmonic Analysis, 6 (1999), pp. 1–52.
- [5] A. COHEN, *Wavelet methods in numerical analysis*, Handbook of Numerical Analysis, 7 (2000), pp. 417–711.
- [6] A. COHEN, W. DAHMEN, AND R. DEVORE, *Adaptive wavelet methods for elliptic operator equations: convergence rates*, Mathematics of Computation, 70 (2001), pp. 27–75.
- [7] ———, *Adaptive wavelet methods II - beyond the elliptic case*, Foundations of Computational Mathematics, 2 (2002), pp. 203–245.

- [8] ———, *Adaptive wavelet schemes for nonlinear variational problems*, SIAM Journal on Numerical Analysis, 41 (2003), pp. 1785–1823.
- [9] A. COHEN, I. DAUBECHIES, AND J.-C. FEAUVEAU, *Biorthogonal bases of compactly supported wavelets*, Communications on Pure Applied Mathematics, 45 (1992), pp. 485–560.
- [10] A. COHEN AND R. MASSON, *Wavelet adaptive method for second order elliptic problems: boundary condition and domain decomposition*, Numerische Mathematik, 86 (2000), pp. 193–238.
- [11] W. DAHMEN, H. HARBRECHT, AND R. SCHNEIDER, *Compression techniques for boundary integral equations - asymptotically optimal complexity estimates*, SIAM Journal On Numerical Analysis, 43(6) (2006), pp. 2251–2271.
- [12] W. DAHMEN, A. KUNOTH, AND K. URBAN, *Biorthogonal spline wavelets on the interval - stability and moment conditions*, Applied Computational Harmonic Analysis, 6(2) (1999), pp. 132–196.
- [13] I. DAUBECHIES, *Ten lectures on wavelets*, SIAM, Pennsylvania, 1992.
- [14] R. DEVORE, *Nonlinear approximation*, Acta Numerica, 7 (1998), pp. 51–150.
- [15] T. DIJKEMA, *Adaptive tensor product wavelet methods for solving PDEs*, PhD thesis, University of Utrecht, 2009.
- [16] T. GANTUMUR, H. HARBRECHT, AND R. STEVENSON, *An optimal adaptive wavelet method without coarsening of the iterands*, Mathematics of Computation, 76 (2007), pp. 615–629.
- [17] M. GRIEBEL AND P. OSWALD, *Tensor product type subspace splitting and multilevel iterative methods for anisotropic problems*, Advances in Computational Mathematics, 4 (1995), pp. 171–206.
- [18] M. HANSEN AND W. SICKEL, *Best m -term approximation and tensor products of Sobolev and Besov spaces – the case of non-compact embeddings*, DFG-SPP 1324, Preprint 39 (March 2010).
- [19] M. LEHN, *FLENS - A Flexible Library for Efficient Numerical Solutions*, <http://flens.sourceforge.net>, (2008).
- [20] S. MALLAT, *Multiresolution approximations and wavelet orthonormal bases of $L^2(\mathbb{R})$* , Transactions of the American Mathematical Society, 315 (1989), pp. 69–87.
- [21] P.-A. NITSCHKE, *Sparse Approximation of Singularity Functions*, Constructive Approximation, 21 (2005), pp. 63–81.
- [22] ———, *Best N -term approximation spaces for tensor product wavelet bases*, Constructive Approximation, 24 (2006), pp. 49–70.
- [23] R. SCHNEIDER, *Multiskalen- und Wavelet-Matrixkompression: Analysisbasierte Methoden zur effizienten Lösung großer vollbesetzter Gleichungssysteme*, B.G. Teubner, 1998.
- [24] C. SCHWAB AND R. STEVENSON, *Adaptive wavelet algorithms for elliptic PDE's on product domains*, Mathematics of Computation, 77 (2008), pp. 71–92.
- [25] ———, *Space-time adaptive wavelet methods for parabolic evolution problems*, Mathematics of Computation, 78 (2009), pp. 1293–1318.
- [26] W. SICKEL AND T. ULLRICH, *Tensor products of Sobolev-Besov spaces and applications to approximation from the hyperbolic cross*, Journal of Approximation Theory, 161 (2009), pp. 748–786.
- [27] A. STIPLER, *LAWA - Library for Adaptive Wavelet Applications*, <http://lawa.sourceforge.net>, (2009).
- [28] K. URBAN, *Wavelet methods for elliptic partial differential equations*, Oxford University Press, 2009.

SEBASTIAN KESTLER, UNIVERSITY OF ULM, INSTITUTE FOR NUMERICAL MATHEMATICS, HELMHOLTZSTRASSE 18, D-89069 ULM, GERMANY
E-mail address: `sebastian.kestler@uni-ulm.de`

KARSTEN URBAN, UNIVERSITY OF ULM, INSTITUTE FOR NUMERICAL MATHEMATICS, HELMHOLTZSTRASSE 18, D-89069 ULM, GERMANY
E-mail address: `karsten.urban@uni-ulm.de`