Adaptive Wavelet Methods on Unbounded Domains

Sebastian Kestler und Karsten Urban

Preprint Series: 2010-10

Fakultät für Mathematik und Wirtschaftswissenschaften
UNIVERSITÄT ULM

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 $H^1(\mathbb{R}^n)$ can easily be constructed as tensor products of dilations and integer translates of some mother wavelet ψ ($\mathbf{j} = (j_1, \dots, j_n)^T$, $\mathbf{k} = (k_1, \dots, k_n)^T$)

$$\Psi^{\mathbb{R}^n} := \{\psi_{\mathbf{j},\mathbf{k}} : \mathbf{j}, \mathbf{k} \in \mathbb{Z}^n\},\ \psi_{\mathbf{j},\mathbf{k}}(x) := \left(1 + \sum_{i=1}^{n} 2^{2j_i}\right)^{-\frac{1}{2}} \prod_{i=1}^{n} 2^{\frac{j_i}{2}} \psi(2^{j_i} x_i - k_i), \tag{1.1}$$

for $x \in \mathbb{R}^n$. Thus, one can follow the idea from [5] to transform the original operator equation $\mathcal{A}u = f$ on $H^{-1}(\mathbb{R}^n)$ into an equivalent well-posed problem $\mathbf{Au} = \mathbf{f}$ on sequence spaces ℓ_2 for the wavelet coefficients. This idea has been used e.g. in

Date: March 16, 2011.

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.*

[2, 5, 6, 13, 17, 25] (see also [30]) with wavelet bases on bounded domains. This approach results in adaptive wavelet methods that have been proven to converge at an optimal rate as compared with the best N-term approximation w.r.t. the same basis. To highlight the differences to bounded settings, let us mention that a wavelet basis on a bounded domain $\Omega \subset \mathbb{R}^n$ typically takes the form

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

where $|\operatorname{supp} \psi_{j,\mathbf{k}}| \sim 2^{-j}$, but $\psi_{j,\mathbf{k}}$ may not result by scaling and translating mother wavelets, e.g. [3, 9, 11, 13]. 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 $\mathbb{Z}^n \times \mathbb{Z}^n$, 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 $\mathcal{A} = -\Delta + \mathrm{I}$, 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 be generalized to higher space dimensions, nonlocal operators or nonlinear problems. In fact, our framework can be used in methods to treat high-dimensional [25] or nonlinear problems [7].

Nevertheless, it is a priori not clear how actually the resolution of the asymptotic boundary conditions realized by adaptive wavelet schemes look like. 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 of the adaptive scheme from [17]. We shall describe a second heuristic adaptive scheme and a comparison of the two algorithms 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, self-adjoint operator $\mathcal{A}: H\to H'$ and a right-hand side $f\in H'$, we are concerned with the numerical solution of the operator equation for $u\in H$

$$\mathcal{A}u = f \text{ in } H'. \tag{2.1}$$

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

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

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

In this article, we focus on the case $H=H^1(\mathbb{R}^n)$ and $\mathcal{A}$ a differential operator.

2.2. **Wavelets.** For the discretization of (2.1), we use the tensor product wavelet basis $\Psi^{\mathbb{R}^n}$ on $\mathbb{R}^n$ which is also known as *anisotropic* wavelet basis. Such systems have shown some advantages in applications, in particular in higher space dimensions, see e.g. [13, 14, 24, 25]. Since $H^1(\mathbb{R}^n)$ is isomorphic to $\bigcap_{k=1}^n\bigotimes_{i=1}^n H^{\delta_{i,k}}(\mathbb{R})$ with $\delta_{i,k}$ the Kronecker delta and $H^0(\mathbb{R}) := L_2(\mathbb{R})$, it also holds that (cf. e.g. [20])

$$\Psi := \{\psi_{\boldsymbol{\lambda}} := \psi_{\mathbf{j},\mathbf{k}} : \boldsymbol{\lambda}=(\lambda_1,\dots,\lambda_n)\in\mathbf{J}\}, \tag{2.4}$$

where $\lambda_i=(j_i,k_i)$ for $i=1,\dots,n$ and $\mathbf{J} := \mathcal{J}\times\cdots\times\mathcal{J}$, $\mathcal{J}:=\mathbb{Z}^2$, is a Riesz basis of $H^1(\mathbb{R}^n)$, i.e., there exist constants $0<c_\Psi\le C_\Psi<\infty$ such that

$$c_\Psi\|\mathbf{d}\|_{\ell_2(\mathbf{J})} \le \|\mathbf{d}^T\Psi\|_{H^1(\mathbb{R}^n)} \le C_\Psi\|\mathbf{d}\|_{\ell_2(\mathbf{J})}, \quad \mathbf{d}\in\ell_2(\mathbf{J}). \tag{2.5}$$

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 B-spline wavelet bases from [8].

In order to avoid arbitrarily coarse levels $j_i\to-\infty$ (keeping in mind that the support size of the wavelets grows exponentially with decreasing levels), one can also consider a minimal level $\mathbf{j}_0=(j_0^{(1)},\dots,j_0^{(n)})$, $-\infty<j_0^{(i)}<\infty$, and the system

$$\bar\Psi = \Phi_{\mathbf{j}_0}\cup\{\psi_{\boldsymbol{\lambda}} : \boldsymbol{\lambda}=(\mathbf{j},\mathbf{k})\in\mathbf{J},\ j_i\ge j_0^{(i)},\ 1\le i\le n\} \tag{2.6}$$

with the so called scaling functions $\Phi_{\mathbf{j}_0} := \{\bigotimes_{i=1}^n\varphi_{j_0^{(i)},k_i} : \mathbf{k}\in\mathbb{Z}^n\}$, [23]. 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 $|\operatorname{supp}\psi_{\boldsymbol{\lambda}}|\sim 2^{-|\boldsymbol{\lambda}|_1}$, $|\boldsymbol{\lambda}|_1 := |\mathbf{j}|_1 := j_1+\cdots+j_n$. Another important feature is the polynomial exactness of order d of $\Phi_{\mathbf{j}_0}$ and the number $\widetilde{d}\ge d$ of vanishing moments of the wavelets. Both parameters depend on the particular choice of φ and ψ. Following [10, 22], we remark that $d\ge 2$ is a sufficient condition in our case for Ψ and $\bar\Psi$ to be Riesz bases for $H^1(\mathbb{R}^n)$.

2.3. **Wavelet discretization.** Now we use a tensor wavelet basis $\Psi := \{\psi_{\boldsymbol{\lambda}} : \boldsymbol{\lambda}\in\mathbf{J}\}$ to transform (2.1) into a well-conditioned discrete operator equation. Note that this definition includes both types of wavelet basis defined above, i.e. without (cf. (2.4)) and with scaling functions on a fixed level $\mathbf{j}_0$ (cf. (2.6)). From the Riesz basis property of Ψ, we infer that for the solution u to (2.1) there exists a unique

$\mathbf{u} \in \ell_2(\mathbf{J})$ with $u = \mathbf{u}^T\Psi$. This means that $\mathbf{u}$ is the (unknown) sequence of wavelet coefficients of u. Thus, (2.1) is equivalent to the infinite linear system

$$\mathbf{A}\mathbf{u} = \mathbf{f}, \tag{2.7}$$

where $\mathbf{A} := (\langle\, \mathcal{A}\,\Psi, \Psi\,\rangle)^T$ and $\mathbf{f} := \langle\, f, \Psi\,\rangle$. Note, that (2.7) is well-posed on $\ell_2(\mathbf{J})$ since by (2.2), (2.3) and (2.5), the symmetric bilinear form $\mathbf{a}(\mathbf{v},\mathbf{v}) := \langle\, \mathbf{A}\mathbf{v}, \mathbf{v}\,\rangle_{\ell_2} = a(\mathbf{v}^T\Psi, \mathbf{v}^T\Psi)$ satisfies

$$c_1\|\mathbf{v}\|_{\ell_2}^2 \le \mathbf{a}(\mathbf{v},\mathbf{v}) \le c_2\|\mathbf{v}\|_{\ell_2}^2, \qquad \mathbf{v} \in \ell_2(\mathbf{J}), \tag{2.8}$$

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 reasoning using (2.3), also continuous. For this reason, the operator $\mathbf{A} : \ell_2(\mathbf{J}) \to \ell_2(\mathbf{J})$ is symmetric, continuous (with continuity constant c_2) and coercive (with coercivity constant c_1). Moreover, from (2.8) follows that $\mathbf{A} : \ell_2(\mathbf{J}) \to \ell_2(\mathbf{J})$ is boundedly invertible with operator norms on $\ell_2(\mathbf{J})$

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

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

$$\|\mathbf{v}\|_{\mathbf{A}} \sim \|\mathbf{v}\|_{\ell_2} \sim \|\mathbf{v}^T\Psi\|_{H^1(\mathbb{R}^n)}, \quad \forall \mathbf{v} \in \ell_2(\mathbf{J}). \tag{2.9}$$

To avoid the use of various constants, we write $C \lesssim D$ if there exists a constant $c > 0$ such that $C \le cD$. 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 $\mathbf{\Lambda} \subset \mathbf{J}$. For this purpose, we introduce for $\mathbf{v} \in \ell_2(\mathbf{J})$ the projection $\mathbf{P}_{\mathbf{\Lambda}}\mathbf{v} := \mathbf{v}|_{\ell_2(\mathbf{\Lambda})}$. Analogously, we set $\mathbf{A}_{\mathbf{\Lambda}} := (\mathbf{P}_{\mathbf{\Lambda}}\mathbf{A})|_{\ell_2(\mathbf{\Lambda})}$, $\mathbf{f}_{\mathbf{\Lambda}} := \mathbf{P}_{\mathbf{\Lambda}}\mathbf{f}$ and $\mathbf{v}_{\mathbf{\Lambda}} := \mathbf{P}_{\mathbf{\Lambda}}\mathbf{v}$. Thus, we obtain the finite Galerkin system

$$\mathbf{A}_{\mathbf{\Lambda}}\mathbf{u}_{\mathbf{\Lambda}} = \mathbf{f}_{\mathbf{\Lambda}}. \tag{2.10}$$

One possible interpretation of many adaptive schemes is to find a sequence of indices $\mathbf{\Lambda}^{(0)}, \mathbf{\Lambda}^{(1)}, \mathbf{\Lambda}^{(2)}, \ldots$ so that the corresponding Galerkin solutions $\mathbf{u}^{(k)}$ of (2.10) for $\mathbf{\Lambda} = \mathbf{\Lambda}^{(k)}$ converge possibly fast towards $\mathbf{u}$ 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 $\#\operatorname{supp}\mathbf{u}_N = N$, i.e., an argument of

$$\sigma_N(\mathbf{u}) := \inf_{\mathbf{w}\in\Sigma_N} \|\mathbf{u} - \mathbf{w}\|_{\ell_2}, \tag{2.11}$$

where $\Sigma_N := \{\mathbf{v} \in \ell_2(\mathbf{J}) : \#\operatorname{supp}\mathbf{v} \le N\}$ is a nonlinear manifold in $\ell_2(\mathbf{J})$.

In order to define (quasi-)optimality, we collect all sequences whose best N-term approximation converges with rate $s > 0$ in the so called *approximation class*

$$\mathcal{A}^s := \{\mathbf{v} \in \ell_2 : \sigma_N(\mathbf{v}) \lesssim N^{-s}\}, \quad s \ge 0.$$

A (quasi-)norm on $\mathcal{A}^s$ is given by $|\mathbf{v}|_{\mathcal{A}^s} := \sup_{N\geq 0}(N+1)^s\|\mathbf{v}-\mathbf{v}_N\|_{\ell_2}$, where $\mathbf{v}_N$ is a best N-term approximation of $\mathbf{v}$. It can be shown, that if $\mathbf{u}\in\mathcal{A}^s$ for some $s>0$, then $\mathbf{u}$ can be approximated for a given tolerance $\varepsilon>0$ by some $\mathbf{v}=\mathbf{v}(\varepsilon)$ such that

$$\|\mathbf{u}-\mathbf{v}\|_{\ell_2}\leq\varepsilon,\qquad \#\operatorname{supp}\mathbf{v}\lesssim\varepsilon^{-1/s}\,|\mathbf{u}|_{\mathcal{A}^s}^{1/s}. \tag{2.12}$$

This sets the benchmark and we call an adaptive wavelet method *(quasi-)optimal* if for $u=\mathbf{u}^T\Psi$ with $\mathbf{u}\in\mathcal{A}^s$, $s\in(0,s^*]$, and a given accuracy $\varepsilon>0$ the scheme produces an output $\mathbf{v}$ with (2.12) with linear complexity in arithmetic operations and storage requirements.

The next question is of course, under which conditions on u or $\mathbf{u}$, one actually has $\mathbf{u}\in\mathcal{A}^s$. As far as the sequence $\mathbf{u}$ of the wavelet coefficients is concerned, it is well-known that certain decay rates are needed in order to ensure a certain rate of approximation. This decay is expressed by the so called weak $\ell_\tau(\mathbf{J})$-spaces defined as follows (cf. [12]). For each $0<\tau<2$ and $\mathbf{v}\in\ell_2(\mathbf{J})$, we define $|\mathbf{v}|_{\ell_\tau^{\mathrm{w}}}:=\sup_{n\geq 1}n^{1/\tau}v_n^*$, 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 $\ell_\tau^{\mathrm{w}}(\mathbf{J}):=\{\mathbf{v}\in\ell_2(\mathbf{J}):|\mathbf{v}|_{\ell_\tau^{\mathrm{w}}}<\infty\}$ with the corresponding norm $\|\mathbf{v}\|_{\ell_\tau^{\mathrm{w}}}:=|\mathbf{v}|_{\ell_\tau^{\mathrm{w}}}+\|\mathbf{v}\|_{\ell_2}$, $\mathbf{v}\in\ell_\tau^{\mathrm{w}}$. Then, it is known that $\ell_\tau\hookrightarrow\ell_\tau^{\mathrm{w}}\hookrightarrow\ell_{\tau+\delta}$ for any $\delta\in(0,2-\tau]$ explaining the notion 'weak'-ℓ_τ. With this notation at hand, $\sigma_N(\mathbf{u})$ decays with a fixed rate $s>0$ if $\mathbf{u}\in\ell_\tau^{\mathrm{w}}(\mathbf{J})$ for

$$\frac{1}{\tau}=s+\frac{1}{2}. \tag{2.13}$$

In this case, there exists a constant $C_\tau>0$ depending on τ such that (cf. [5])

$$\sigma_N(\mathbf{u})\leq C_\tau\|\mathbf{u}\|_{\ell_\tau^{\mathrm{w}}}N^{-s}. \tag{2.14}$$

Hence, the best possible value s^* for which (2.14) holds for all $s<s^*$, is the optimal rate. It turns out that s^* is related to the Besov regularity of the underlying function (and properties of the wavelets Ψ).

For anisotropic (tensor product) wavelets, it is known from [24, 27] that if $u\in\bigcap_{k=1}^n\bigotimes_{\tau\ i=1}^{\ n}B_\tau^{s+\delta_{i,k}}(L_\tau(\mathbb{R}))$, then $\mathbf{u}\in\ell_\tau^{\mathrm{w}}(\mathbf{J})$ for $0<s<d-1$, d being the order of the wavelets. Here, $\otimes_\tau$ is a so called τ-tensor product introduced in [24]. Thus, we obtain $s^*=d-1$ provided that we use anisotropic wavelets (of order d) 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 with the same rate 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. A symmetric operator $\mathbf{A}:\ell_2(\mathbf{J})\to\ell_2(\mathbf{J})$ 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

$$\|\mathbf{A}-\mathbf{A}_j\|\leq\beta_j 2^{-js}. \tag{2.15}$$

Compression estimates which fit into the setting of (2.15) have been discussed in detail for different types of operators for example in [25, 28]. We consider such estimates for operators and wavelet bases on unbounded domains later in Section

3.1. 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 $\mathbf{v} \in \ell_\tau^{\mathrm{w}}(\mathbf{J})$ (e.g. the first 2^j entries of $\mathbf{v}^*$), then it holds

$$\|\mathbf{v} - \mathbf{v}_{[j]}\|_{\ell_2} \leq 2^{-js} \, |\mathbf{v}|_{\mathcal{A}^s}, \tag{2.16}$$

if τ is chosen as in (2.13). One can use this observation to show that if $\mathbf{A} \in \mathcal{B}_s$, then it is a bounded operator on $\ell_\tau^{\mathrm{w}}(\mathbf{J})$ and also derive a method for approximating an infinite matrix-vector product $\mathbf{Av}$ (cf. [5]).

2.6. **An optimal adaptive wavelet algorithm.** Now, we describe the adaptive wavelet solver **ADWAV** from [17] which we used as a basis for our extension to unbounded domains. The core scheme is shown in Algorithm 1. We start with a real number ν_{-1} (whose meaning will be explained below) and a desired tolerance $\varepsilon > 0$. Finally, we need to choose constants $\alpha, \gamma, \theta, \omega$ such that:

- $0 < \omega < \alpha < 1$ 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 $[\mathbf{u}(\varepsilon), \mathbf{\Lambda}(\varepsilon)] = \mathbf{ADWAV}[\nu_{-1}, \varepsilon]$

1: $\mathbf{\Lambda}^{(0)} = \emptyset$, $k := 0$, $\mathbf{w}^{(0)} := \mathbf{0}$
2: **repeat**
3: $\quad [\mathbf{\Lambda}^{(k+1)}, \nu_k] = \mathbf{GROW}[\mathbf{w}^{(k)}, \theta\nu_{k-1}, \varepsilon]$
4: $\quad \mathbf{g}^{(k+1)} = \mathbf{P}_{\mathbf{\Lambda}^{(k+1)}}(\mathbf{RHS}[\gamma\nu_k])$
5: $\quad \mathbf{w}^{(k+1)} = \mathbf{GALSOLVE}[\mathbf{\Lambda}^{(k+1)}, \mathbf{g}^{(k+1)}, \mathbf{w}^{(k)}, (1+\gamma)\nu_k, \gamma\nu_k]$
6: $\quad k = k + 1$
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 **ADWAV**, let us recall the properties of this adaptive wavelet scheme.

Theorem 2.1 ([17, Theorem 2.7])**.** *The output* $\mathbf{w} = \mathbf{u}(\varepsilon)$ *of the routine* **ADWAV**$[\nu_{-1}, \varepsilon]$ *satisfies* $\|\mathbf{Aw} - \mathbf{f}\|_{\ell_2} \leq \varepsilon$. *If* $\nu_{-1} \sim \|\mathbf{f}\|_{\ell_2} \gtrsim \varepsilon$, *and* $\mathbf{u} \in \ell_\tau^{\mathrm{w}}(\mathbf{J})$ *for some* $s < s^*$, $\frac{1}{\tau} = s + \frac{1}{2}$, *then* $\#\operatorname{supp} \mathbf{w} \lesssim \varepsilon^{-1/s}|\mathbf{u}|_{\ell_\tau^{\mathrm{w}}}^{1/s}$ *and the number of arithmetic operations and storage locations is bounded by some absolute multiple of the same expression.* □

Now, we are going to detail all subroutines involved in **ADWAV**. Within **ADWAV**, the routine **GROW** shown in Algorithm 2 enlarges the current index set $\mathbf{\Lambda}^{(k)}$ in such a way that the new index set $\mathbf{\Lambda}^{(k+1)}$ guarantees

$$\|\mathbf{P}_{\mathbf{\Lambda}^{(k+1)}}(\mathbf{A}\mathbf{u}_{\mathbf{\Lambda}^{(k)}} - \mathbf{f})\|_{\ell_2} \geq \beta\|\mathbf{A}\mathbf{u}_{\mathbf{\Lambda}^{(k)}} - \mathbf{f}\|_{\ell_2}, \tag{2.17}$$

for some $0 < \beta < 1$ (this is sometimes also called *saturation property*). Then, due to Galerkin orthogonality (cf. [5, Lemma 4.1]), one has the following error reduction $\|\mathbf{u} - \mathbf{u}_{\mathbf{\Lambda}^{(k+1)}}\|_{\mathbf{a}} \leq (1 - \frac{c_2}{c_1}\beta^2)^{1/2} \, \|\mathbf{u} - \mathbf{u}_{\mathbf{\Lambda}^{(k)}}\|_{\mathbf{a}}$ with the constants c_1, c_2 from (2.8).

Under the same assumptions as in Theorem 2.1 and if $\mathbf{w} \in \ell_\tau^{\mathrm{w}}(\mathbf{J})$, then the number of operations and storage locations required by $[\mathbf{\Lambda}, \nu] = \mathbf{GROW}[\mathbf{w}, \bar{\nu}, \varepsilon]$ is

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

Define $\zeta := 2\frac{\omega\bar{\nu}}{1-\omega}$.
repeat
  $\zeta := \zeta/2$, $\mathbf{r} :=$**RHS**$[\zeta/2] -$**APPLY**$[\mathbf{w}, \zeta/2]$,
until $\nu := \|\mathbf{r}\|_{\ell_2} + \zeta \le \varepsilon$ or $\zeta \le \omega\|\mathbf{r}\|_{\ell_2}$
if $\nu > \varepsilon$ **then**
  determine a minimal set $\mathbf{\Lambda} \supset \operatorname{supp} \mathbf{w}$ such that $\|\mathbf{P}_{\mathbf{\Lambda}}\mathbf{r}\| \ge \alpha\|\mathbf{r}\|_{\ell_2}$.
else
  set $\mathbf{\Lambda} := \emptyset$.
end if

bounded by some absolute multiple of $\min\{\bar{\nu}, \nu\}^{-1/s}[|\mathbf{w}|^{1/s}_{\ell^{\mathrm{w}}_\tau} + |\mathbf{u}|^{1/s}_{\ell^{\mathrm{w}}_\tau} + \bar{\nu}^{1/s}(\#\operatorname{supp} \mathbf{w} + 1)]$. Moreover, we have $\nu \ge \|\mathbf{A}\mathbf{w} - \mathbf{f}\|_{\ell_2}$ and, if $\nu > \varepsilon$, the saturation property

$$\frac{\alpha - \omega}{1 + \omega}\nu \le \|\mathbf{P}_{\mathbf{\Lambda}}(\mathbf{A}\mathbf{w} - \mathbf{f})\|_{\ell_2}, \qquad \#(\mathbf{\Lambda}\backslash\operatorname{supp} \mathbf{w}) \lesssim \nu^{-1/s}|\mathbf{u}|_{\ell^{\mathrm{w}}_\tau} \tag{2.18}$$

holds with the constants α and ω described above.

The routine **RHS**$[\delta]$ produces an approximation $\mathbf{g}$ to $\mathbf{f}$ such that $\|\mathbf{f} - \mathbf{g}\|_{\ell_2} \le \delta$. To preserve the linear complexity of **ADWAV**, the length of $\mathbf{g}$ must be bounded by some constant multiple of $\delta^{-1/s}|u|^{1/s}_{\ell^{\mathrm{w}}_\tau}$.

Finally, **GALSOLVE** (short for Galerkin solver) produces an approximate solution $\widetilde{\mathbf{w}}_{\mathbf{\Lambda}}$ with $\|\mathbf{A}_{\mathbf{\Lambda}}\widetilde{\mathbf{w}}_{\mathbf{\Lambda}} - \mathbf{f}_{\mathbf{\Lambda}}\| \le \varepsilon$ starting with an initial guess $\mathbf{w}_{\mathbf{\Lambda}}$ satisfying $\|\mathbf{A}_{\mathbf{\Lambda}}\mathbf{w}_{\mathbf{\Lambda}} - \mathbf{f}_{\mathbf{\Lambda}}\| \le \delta$.

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

Determine $\mathbf{A}_J$ in the sense of (2.15) with $J = J(\varepsilon)$ as small as possible and $\|\mathbf{A} - \mathbf{A}_J\| \le \frac{\varepsilon}{3}$.
Assemble $\mathbf{B} := \mathbf{P}_{\mathbf{\Lambda}}[\frac{1}{2}(\mathbf{A}_J + \mathbf{A}_J^*)]|_{\ell_2(\mathbf{\Lambda})}$ with $\mathbf{A}_J^*$ being the adjoint of $\mathbf{A}_J$.
Compute $\mathbf{r}_0 := \mathbf{f}_{\mathbf{\Lambda}} - \mathbf{P}_{\mathbf{\Lambda}}($**APPLY**$[\mathbf{w}_{\mathbf{\Lambda}}, \frac{\varepsilon}{3}])$.
Determine $\mathbf{x}$ as the solution of $\mathbf{B}\mathbf{x} = \mathbf{r}_0$ and set $\widetilde{\mathbf{w}}_{\mathbf{\Lambda}} = \mathbf{w}_{\mathbf{\Lambda}} + \mathbf{x}$.

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 $\mathbf{w} =$ **APPLY**$[\mathbf{v}, \eta]$ satisfies $\|\mathbf{A}\mathbf{v} - \mathbf{w}\|_{\ell_2} \le \eta$ as well as $\operatorname{supp} \mathbf{w} \lesssim \|\mathbf{v}\|^{1/s}_{\ell^{\mathrm{w}}_\tau}\eta^{-\frac{1}{s}}$ provided that $\mathbf{v} \in \ell^{\mathrm{w}}_\tau(\mathbf{J})$, $\frac{1}{\tau} = s + \frac{1}{2}$, see [5, 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, 13] without destroying the approximation properties.

3. An optimal adaptive wavelet algorithm on unbounded domains

Having reviewed all ingredients of the asymptotically optimal scheme **ADWAV**, we can now identify the modifications that are necessary in order to treat problems on unbounded domains. In particular, we have to verify that $\mathbf{A} \in \mathcal{B}_s$ and that a realization of **RHS** for both basis types, namely Ψ (cf. (2.4)) and $\bar{\Psi}$ (cf. (2.6)) is available. Here, we shall focus on the case $n = 1$. Extensions to higher space dimensions can then be derived analogously (cf. eg. [25]). In order to facilitate

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

1: Set $N:=\#\operatorname{supp}\mathbf{v}$ and $k(\eta)$ as the smallest integer such that $2^{k(\eta)}\geq\eta^{-\frac{1}{s}}\|\mathbf{v}\|_{\ell^{\mathrm{w}}_\tau}^{\frac{1}{s}}$.
2: 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 $i>\log N$.
3: **for** $k=1$ **to** $k(\eta)$ **do**
4: $\quad R_k:=c_2\{\|\mathbf{v}-\mathbf{v}_{[k]}\|_{\ell_2}+\alpha_k 2^{-ks}\|\mathbf{v}_{[0]}\|_{\ell_2}+\sum_{\ell=0}^{k-1}\alpha_\ell 2^{-\ell s}\|\mathbf{v}_{[k-\ell]}-\mathbf{v}_{[k-\ell-1]}\|_{\ell_2}\}$
5: $\quad$ **if** $R_k\leq\eta$ **exit**
6: **end for**
7: Compute $\mathbf{w}:=\mathbf{w}_k:=\mathbf{A}_k\mathbf{v}_{[0]}+\mathbf{A}_{k-1}(\mathbf{v}_{[1]}-\mathbf{v}_{[0]})+\cdots+\mathbf{A}_0(\mathbf{v}_{[k]}-\mathbf{v}_{[k-1]})$.

the distinction between the two basis types Ψ and $\bar\Psi$, in the sequel a minimal level $j_0=-\infty$ refers to the first type and $j_0>-\infty$ to the second type. Note that the remaining routines **GROW**, **GALSOLVE** and **APPLY** can be used without modifications. Optimality and convergence of the scheme on unbounded domains then follow directly from Theorem 2.1.

3.1. **Compressibility.** To show the property $\mathbf{A}\in\mathcal{B}_s$, one assumes that there exists $\sigma>0$ such that the operator $\mathcal{A}:H^{1+\sigma}(\mathbb{R})\to H^{1-\sigma}(\mathbb{R})$ is bounded and uses the Bernstein inequality

$$\|w_\ell\|_{H^r(\mathbb{R})}\lesssim 2^{l(r-s)}\|w_\ell\|_{H^s(\mathbb{R})},\quad \forall w_l\in W_\ell:=\operatorname{span}_{L_2(\mathbb{R})}\{\psi_\lambda:|\lambda|=\ell\},\tag{3.1}$$

which holds for $\ell\geq 0$, $s<d-\frac{1}{2}$ and $r\in[-\widetilde{d},d-\frac{1}{2})$ (cf., e.g. [28]).

The case $j_0>-\infty$. If we use a wavelet basis with scaling functions on a minimal level $j_0>-\infty$, then, by rescaling the underlying mother wavelet ψ and mother scaling function φ, we can assume w.l.o.g. that $j_0=0$. In this setting we can use (3.1) and apply the compression scheme for differential operators from [28].

The case $j_0=-\infty$. Conversely, without a lower bound for the levels, $\mathbf{A}$ from (2.7) can be subdivided into four blocks corresponding to the sign of the level of the wavelets corresponding to rows and columns. Here, (3.1) cannot be applied to the blocks of the subdivision corresponding to negative row and/or column levels.

Instead, let $\lambda=(j,k),\lambda'=(j',k')\in\mathcal{J}$ with (w.l.o.g) $j'\leq j$ as well as $0\leq\nu,\nu'\leq 1$. Then, by transforming the variables, we get

$$\begin{aligned}
&\int_{\mathbb{R}}\psi_\lambda^{(\nu)}(x)\psi_{\lambda'}^{(\nu')}(x)\,\mathrm{d}x\\
&=\int_{\mathbb{R}}\frac{\mathrm{d}^\nu}{\mathrm{d}x^\nu}(1+2^{2j})^{-\frac{1}{2}}\,2^{\frac{1}{2}j}\,\psi(2^jx-k)\frac{\mathrm{d}^{\nu'}}{\mathrm{d}x^{\nu'}}(1+2^{2j'})^{-\frac{1}{2}}\,2^{\frac{1}{2}j'}\,\psi(2^{j'}x-k')\,\mathrm{d}x\\
&=(1+2^{2j})^{-\frac{1}{2}}(1+2^{2j'})^{-\frac{1}{2}}\,2^{j\nu}\,2^{j'\nu'}\,2^{-j'}\,2^{\frac{1}{2}(j+j')}\int_{\mathbb{R}}\psi(2^{j-j'}y-k+2^{j-j'}k')\psi(y)\mathrm{d}y\\
&=\frac{\sqrt{1+2^{2(j-j')}}}{\sqrt{(1+2^{2j})(1+2^{2j'})}}\,2^{2j'\nu'}\int_{\mathbb{R}}\psi^{(\nu)}_{j-j',k-2^{j-j'}k'}(y)\psi^{(\nu')}(y)\,\mathrm{d}y.
\end{aligned}$$

Thus, we can now apply (3.1) and use the results from [28] for all matrix blocks.

Remark 3.1. We restrict ourselves to the case of differential operators with constant coefficients as in this case, the so called s-computability (cf. [18]) of $\mathbf{A}$ follows directly from the compressibility. The s-computability property says that for

$\mathbf{A} \in \mathcal{B}_s$, the compressed matrix $\mathbf{A}_j$ can also be computed with at most $\mathcal{O}(2^j)$ operations such that $\|\mathbf{A}-\mathbf{A}_j\| \lesssim 2^{-js}$. Thus, for differential operators with non-constant coefficients, we can apply the results from [18] if $j_0 > -\infty$, but not if $j_0 = -\infty$ since the supports of the wavelets can get arbitrary large with decreasing level.

3.2. **RHS on unbounded domains.** We describe one possible realization for the set up of **RHS** which consists of constructing a priori an index set of active right-hand side wavelet coefficients ∇_η of size $\mathcal{O}(\eta^{-1/s})$ for a given accuracy η such that $\|\mathbf{f}-\mathbf{f}|_{\nabla_\eta}\|_{\ell_2} \le \eta$. Here, we focus on the case $j_0 = -\infty$ since for $j_0 > -\infty$, analogous estimates can even easier be derived.

As opposed to the case of a bounded domain, we do not only have to control the maximal level of wavelet indices in such a set ∇_ε but also a minimal level and bounds for the translation indices on each level. For this purpose, besides additional smoothness conditions on f, we require that f satisfies certain decay conditions. Moreover, we assume that $f = f_1 + f_2$ can be splitted into a smooth part $f_1 \in L_2(\mathbb{R})$ and into a singular part which is a sum of delta distributions $f_2 = \sum_{i=1}^n c_i\, \delta_{x_i}$ for $c_1, \dots, c_n \in \mathbb{R}$ and $x_1, \dots, x_n \in \mathbb{R}$.

Singular part. Due to the $H^1(\mathbb{R})$-normalization of the wavelets $\psi_\lambda \in \Psi$, it holds

$$\|\psi_\lambda\|_{L_\infty} = \frac{2^{|\lambda|/2}\|\psi\|_{L_\infty}}{\sqrt{1+2^{2|\lambda|}}} \le 2^{-|\lambda|/2}\,\|\psi\|_{L_\infty}, \quad \forall \lambda \in \mathcal{J}. \tag{3.2}$$

Moreover, due to the locality of the wavelets, on each level ℓ there is only a uniformly bounded number of wavelets whose supports contain the singular points $\{x_1, \dots, x_n\}$ and we can proceed as for bounded Ω (cf., e.g. [17]).

Smooth part. For the smooth part of f we need estimates for the wavelet coefficients $\langle\, f_1, \psi_\lambda \,\rangle$ for both negative and non-negative levels. To ease notation, we set $f \equiv f_1$.

The starting point for compression estimates for non-negative levels is the vanishing moment property of the wavelets which yields $|\langle\, f, \psi_\lambda \,\rangle| \le \|\psi_\lambda\|_{L_1} \inf_{p\in\mathcal{P}_{\widetilde{d}-1}} \|f-p\|_{L_\infty(\operatorname{supp}\psi_\lambda)}$. As $\|\psi_\lambda\|_{L_1} \le 2^{-\frac{3}{2}|\lambda|}\|\psi\|_{L_1}$, a Whitney type estimate (cf. [4, Theorem 25.1]) gives

$$|\langle\, f, \psi_\lambda \,\rangle| \le C_{\psi,f}\, 2^{-(\widetilde{d}+\frac{3}{2})|\lambda|}, \quad \forall \lambda \in \mathcal{J} \text{ with } |\lambda| \ge 0, \tag{3.3}$$

where $C_{\psi,f} := C_W(\operatorname{supp}\psi_\lambda)^{\widetilde{d}+\frac{1}{2}}\|\psi\|_{L_1}|f|_{W^{\widetilde{d},\infty}(\mathbb{R})}$ and $C_W > 0$ is a constant arising from the Whitney estimate. Obviously, (3.3) is not a estimate if $|\lambda| < 0$. Instead, assuming that $f \in L_1(\mathbb{R})$, the Hölder inequality yields

$$|\langle\, f, \psi_\lambda \,\rangle| \le \|f\|_{L_1(\mathbb{R})}\|\psi\|_{L_\infty}\, 2^{|\lambda|/2}, \quad \forall \lambda \in \mathcal{J} \text{ with } |\lambda| < 0. \tag{3.4}$$

Now we are ready to prove the announced approximation for the right-hand side.

Proposition 3.2. *Let $s > 0$ and $f \in L_1(\mathbb{R}) \cap W^{\widetilde{d},\infty}(\mathbb{R})$. Assume that there exist $R_0 > 0$ and $C_f > 0$ such that for $\beta \ge \frac{s(\widetilde{d}+\frac{3}{2})}{1+\widetilde{d}-s}$:*

$$\|f - f_R\|_{L_2(\mathbb{R})} \le C_f R^{-\beta}, \quad \forall R > R_0, \quad f_R := f|_{[-R,R]}. \tag{3.5}$$

Then, for $\varepsilon \lesssim 1$, there exist $J_\varepsilon^+, J_\varepsilon^- \in \mathbb{N}$ and $R_\varepsilon > R_0$ such that $\mathbf{g}_\varepsilon := \mathbf{g}_\varepsilon^+ + \mathbf{g}_\varepsilon^-$ with

$$\mathbf{g}_\varepsilon^+ := \left\{\langle\, f, \psi_\lambda \,\rangle : |\operatorname{supp}\psi_\lambda \cap [-R_\varepsilon, R_\varepsilon]| > 0,\, 0 \le |\lambda| \le J_\varepsilon^+\right\}, \tag{3.6}$$

$$\mathbf{g}_\varepsilon^- := \left\{\langle\, f_{R_\varepsilon}, \psi_\lambda \,\rangle : |\operatorname{supp}\psi_\lambda \cap [-R_\varepsilon, R_\varepsilon]| > 0,\, -J_\varepsilon^- \le |\lambda| < 0\right\}, \tag{3.7}$$

satisfies $\|\mathbf{f} - \mathbf{g}_\varepsilon\|_{\ell_2} \le \varepsilon$ and $\#\operatorname{supp}\mathbf{g}_\varepsilon \sim \varepsilon^{-1/s}$.

Proof. Let $\varepsilon > 0$ and set $R_\varepsilon := D_1\,\varepsilon^{-1/\beta}$ with $D_1 := (C_f\,c_2\,(c_\Psi c_{\mathcal{A}})^{-1})^{1/\beta}$. Then, (3.5) yields $\|f - f_{R_\varepsilon}\|_{L_2(\mathbb{R})} \le C_f R_\varepsilon^{-\beta} \le \frac{c_\Psi c_{\mathcal{A}}}{c_2}\cdot\varepsilon$. Denote by $u_{R_\varepsilon} = \mathbf{u}_{R_\varepsilon}^T\Psi$ the unique solution of the operator equation $\mathcal{A}[u_{R_\varepsilon}] = f_{R_\varepsilon}$. We set $\mathbf{f}_\varepsilon := \mathbf{f}_\varepsilon^+ + \mathbf{f}_\varepsilon^-$ where

$$\mathbf{f}_\varepsilon^+ := \{\langle\, f, \psi_\lambda\,\rangle : |\operatorname{supp}\psi_\lambda \cap [-R_\varepsilon, R_\varepsilon]| > 0,\ |\lambda| \ge 0\},$$
$$\mathbf{f}_\varepsilon^- := \{\langle\, f_{R_\varepsilon}, \psi_\lambda\,\rangle : |\operatorname{supp}\psi_\lambda \cap [-R_\varepsilon, R_\varepsilon]| > 0,\ |\lambda| < 0\}.$$

Then, we have $\mathbf{f}_\varepsilon = \langle\, f_{R_\varepsilon}, \Psi\,\rangle$ and because of $\|g\|_{H^{-1}(\mathbb{R})} \le \|g\|_{L_2(\mathbb{R})}$, $g \in L_2(\mathbb{R})$:

$$\|\mathbf{f} - \mathbf{f}_\varepsilon\|_{\ell_2} = \|\mathbf{A}(\mathbf{u} - \mathbf{u}_{R_\varepsilon})\|_{\ell_2} \le c_2\|\mathbf{u} - \mathbf{u}_{R_\varepsilon}\|_{\ell_2} \le \frac{c_2}{c_\Psi}\|\mathbf{u}^T\Psi - \mathbf{u}_{R_\varepsilon}^T\Psi\|_{H^1(\mathbb{R})}$$
$$\le \frac{c_2}{c_\Psi c_{\mathcal{A}}}\|f - f_{R_\varepsilon}\|_{H^{-1}(\mathbb{R})} \le \frac{c_2}{c_\Psi c_{\mathcal{A}}}\|f - f_{R_\varepsilon}\|_{L_2(\mathbb{R})} \le \varepsilon.$$

Due to the compact support of ψ_λ, there is a constant $c > 1$ independent of j and R_ε such that

$$\#\{\lambda \in \mathcal{J} : |\lambda| = j,\ |\operatorname{supp}\psi_\lambda \cap [-R_\varepsilon, R_\varepsilon]| > 0\} \le \max\{c\,2^{j+1}\,R_\varepsilon, M\},$$

with $M := \lceil|\operatorname{supp}\psi|\rceil = \#\{k \in \mathbb{Z} : |\operatorname{supp}\psi_{0,k} \cap \operatorname{supp}\psi| > 0\}$. Then, we have for $R_\varepsilon > \frac{M}{2}$ with $D_2 := 2\,c\,C_{\psi,f}^2$ by (3.3)

$$\|\mathbf{f}_\varepsilon^+ - \mathbf{g}_\varepsilon^+\|_{\ell_2}^2 \le \sum_{\substack{|\lambda| > J_\varepsilon^+ \\ |\operatorname{supp}\,\psi_\lambda \cap [-R_\varepsilon, R_\varepsilon]| > 0}} |\langle\, f, \psi_\lambda\,\rangle|^2 \le D_2\,R_\varepsilon\,2^{-2(1+\widetilde{d})J_\varepsilon^+}.$$

Thus, we can determine

$$J_\varepsilon^+ = \frac{1}{2(1+\widetilde{d})}\left[\left(2 + \frac{1}{\beta}\right)|\log_2\varepsilon| + |\log_2(D_1D_2)|\right] \sim 1 + \frac{2 + 1/\beta}{2(1+\widetilde{d})}|\log_2\varepsilon|,$$

such that $\|\mathbf{f}_\varepsilon^+ - \mathbf{g}_\varepsilon^+\|_{\ell_2}^2 \le \varepsilon^2$. As the length of $\mathbf{g}_\varepsilon^+$ is of order $R_\varepsilon \cdot 2^{J_\varepsilon^+} \sim \varepsilon^{-\frac{\widetilde{d}+\beta+\frac{3}{2}}{\beta(1+\widetilde{d})}}$ and $\beta \ge \frac{s(\widetilde{d}+\frac{3}{2})}{1+\widetilde{d}-s}$, the length of $\mathbf{g}_\varepsilon^+$ is bounded by some constant multiple of $\varepsilon^{-1/s}$.

For negative levels, we now take (3.4) into account. Moreover, we know that for a fixed level $|\lambda| < 0$, there are only $\max\{2cD_1\varepsilon^{-1/\beta}\,2^{|\lambda|}, M\}$ non-zero entries $\mathbf{f}_\lambda^-$ in $\mathbf{f}^-$. Thus, by choosing $J_\varepsilon^- \ge 1 + |\frac{1}{\beta}\log_2\varepsilon| + |\log_2\frac{M}{2cD_1}|$, there are only M non-zero entries in $\mathbf{f}_\varepsilon^-$ on each level $|\lambda| < -J_\varepsilon^-$. As $f \in L_1(\mathbb{R})$, it holds by (3.4):

$$\|\mathbf{f}_\varepsilon^- - \mathbf{g}_\varepsilon^-\|_{\ell_2}^2 \le \sum_{\substack{|\lambda| < -J_\varepsilon^- \\ |\operatorname{supp}\,\psi_\lambda \cap [-R_\varepsilon, R_\varepsilon]| > 0}} |\langle\, f, \psi_\lambda\,\rangle|^2 \le D_3\,2^{-J_\varepsilon^-},$$

where $D_3 := M\|f\|_{L_1(\mathbb{R})}^2\|\psi\|_{L_\infty}^2$. Thus, we can choose

$$J_\varepsilon^- = \max\left\{2|\log_2\varepsilon| + |\log_2 D_3|,\ 1 + \left|\frac{1}{\beta}\log_2\varepsilon\right| + \left|\log_2\frac{M}{2c}\right|\right\},$$

such that $\|\mathbf{f}_\varepsilon^- - \mathbf{g}_\varepsilon^-\|_{\ell_2}^2 \le \varepsilon^2$. Moreover, $\#\operatorname{supp}\mathbf{g}_\varepsilon^- \lesssim R_\varepsilon + MJ_\varepsilon^- \sim \varepsilon^{-1/\beta} + (1 + \max\{2, \frac{1}{\beta}\}|\log_2\varepsilon|)$. As $J_\varepsilon^- \sim 1 + \max\{2, \frac{1}{\beta}\}|\log_2\varepsilon| \lesssim \varepsilon^{-1/\beta}$ and since $\beta \ge s$ such as $\varepsilon \lesssim 1$, it follows that $\#\operatorname{supp}\mathbf{g}_\varepsilon^- \lesssim \varepsilon^{-1/s}$. Finally, we only have to collect the above results to obtain:

$$\|\mathbf{f} - \mathbf{g}\|_{\ell_2}^2 = \|\mathbf{f} - \mathbf{f}_\varepsilon\|_{\ell_2}^2 + \|\mathbf{f}^+ - \mathbf{g}_\varepsilon^+\|_{\ell_2}^2 + \|\mathbf{f}^- - \mathbf{g}_\varepsilon^-\|_{\ell_2}^2 \le 3\varepsilon^2.$$

Replacing ε by $\frac{\varepsilon}{\sqrt{3}}$ then yields the claim. □

Remark 3.3.

a) Note that both parts $\mathbf{g}_\varepsilon^+$ and $\mathbf{g}_\varepsilon^-$ can be computed efficiently. This is due to the compact support of ψ_λ and f_R.

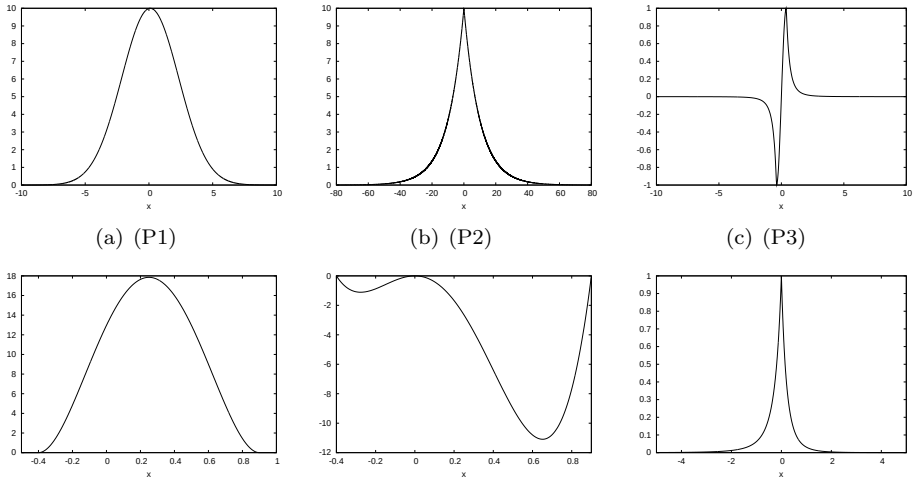
b) Moreover, the construction of $\mathbf{g}_\varepsilon^+$ in (3.6) did not make use of the fact, that $f \in W^{\widetilde{d},\infty}(\mathbb{R}\backslash[-R_\varepsilon, R_\varepsilon])$. Moreover, by defining decay conditions on $f^{(\widetilde{d})}$, one can relax (3.5) and a different construction of $\mathbf{g}_\varepsilon^+$ is possible.

3.3. **Numerical examples in one dimension.** We give some examples in 1D, namely instances of the following reaction-diffusion problem:

$$-u''(x) + u(x) = f(x), x \in \mathbb{R}, \quad \lim_{|x|\to\infty} u(x) = 0,\ u \in H^1(\mathbb{R}), \tag{3.8}$$

for $f \in H^{-1}(\mathbb{R})$. Note that the operator fulfills all required assumptions and that $c_{\mathcal{A}} = C_{\mathcal{A}} = 1$. All examples (also those presented in Section 4 below) are realized in C++ using the software libraries FLENS and LAWA, [21, 29]. 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 3.1. We have chosen these particular examples due to the following reasons:

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



(a) (P1) (b) (P2) (c) (P3)

(d) (P4) (e) (P5) (f) (P6)

FIGURE 3.1. Solutions u_i for (P1)-(P6).

3.3.1. *Normalization of the wavelets.* It has turned out that quantitatively, it is favorable to replace the definition of $H^1(\mathbb{R})$-normalized wavelets in (1.1) by the following equivalent version:

$$\psi_{j,k}(x) := \Big(\|\psi\|_{L_2} + 2^{2j}\|\psi^{(1)}\|_{L_2}\Big)^{-\frac{1}{2}} 2^{\frac{j}{2}}\psi(2^j x - k), \quad \forall x \in \mathbb{R},\ \forall j,k \in \mathbb{Z}, \tag{3.9}$$

which we used for our numerical experiments (including those in Section 4).

3.3.2. *Parameters.* For the realization of **ADWAV**, good estimates of the constants c_1 and c_2 in the norm equivalence (2.8) are necessary for different polynomial orders d, vanishing moments $\widetilde{d}$ and different minimal levels. The values we used can be found in Table 3.1.

In order to ensure a good performance of **ADWAV** the choice of the parameters α, ω, γ and θ (cf. Subsection 2.6) is not trivial. We fixed the parameter $\omega = 0.01$ and chose the parameter $\alpha < (1-\omega)\kappa(\mathbf{A})^{-\frac{1}{2}} - \omega$ as large as possible. Moreover, we used $\gamma = \frac{1}{12}\kappa(\mathbf{A})^{-\frac{1}{2}}\frac{\alpha-\omega}{1+\omega}$ and $\theta = \frac{2}{7}$. Note that this set of parameters satisfies the necessary optimality and convergence condition stated in [17].

j_0		0	-1	-2	-4	-6	-20	$-\infty$
$d=2, \widetilde{d}=2$	c_1	0.37	0.58	0.58	0.46	0.36	0.22	0.19
	c_2	2.10	1.86	1.86	1.86	1.89	1.98	2.10
$d=3, \widetilde{d}=3$	c_1	0.43	0.39	0.29	0.15	0.11	0.04	0.03
	c_2	1.94	2.03	2.24	2.55	2.61	2.65	2.70
$d=3, \widetilde{d}=5$	c_1	0.45	0.41	0.32	0.19	0.17	0.16	0.16
	c_2	1.96	2.07	2.32	2.66	2.71	2.73	2.80

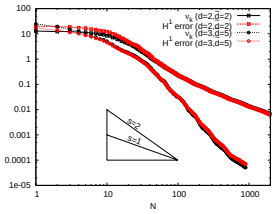
TABLE 3.1. Estimated bounds for c_1, c_2 from (2.8) for **ADWAV**.

Remark 3.4. Concerning the computation of the estimates in Table 3.1, to our knowledge, there is no method to compute these values analytically. Nevertheless, bounds for c_1, c_2 can be computed numerically. We describe the case $j_0 = -\infty$ since $j_0 > -\infty$ can be treated analogously. It suffices to consider the finite collections $\Psi_{R,J^-,J^+} := \{\psi_\lambda : \operatorname{supp}\psi_\lambda \cap [-R,R] \neq \emptyset,\ J^- \le |\lambda| < J^+\}$ for $R > 0$, $J^- \le 0$ and $J^+ \ge 0$ and to compute c_1, c_2 for the finite matrices $\mathbf{A}_{R,J^-,J^+}$ obtained by replacing Ψ by Ψ_{R,J^-,J^+} in (2.7). For $R \to \infty$, $J^+ \to \infty$, $J^- \to -\infty$, one can then observe that the computed eigenvalues converge.

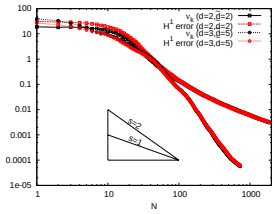
3.3.3. *Choice of a minimal level.* The bounds for c_1 and c_2 given in Table 3.1 already indicate that the choice of a minimal level j_0 is not trivial as we might have two conflicting goals: On the one hand, we want to be free in the choice of a minimal level j_0 to capture both small and very large supports of the numerical solution. On the other hand, we clearly see that the condition number $\kappa(\mathbf{A})$ depends strongly on j_0 and for **ADWAV**, it is favorable to choose $\kappa(\mathbf{A})$ as small as possible. For our first experiments in this section, we choose the level $j_0 = |\lambda|$ of the largest wavelet coefficient $\langle f, \psi_\lambda \rangle$ in the right-hand side vector that can be derived analytically. We will further discuss this issue below.

3.3.4. *Convergence rates.* The results of our experiments concerning the convergence are shown in Figure 3.2 for $j_0 = -\infty$ and Figure 3.3 for $j_0 > -\infty$ where we used wavelet bases with $d = 2$, $\widetilde{d} = 2$ and $d = 3$, $\widetilde{d} = 5$. The latter choice is due to the much better condition number $\kappa(\mathbf{A})$ if $j_0 = -\infty$ (cf. Table 3.1). In both cases

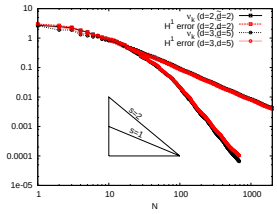
we see the error estimator ν_k. To be able to compare different bases with regard to their performance, we measure the error in $H^1(\mathbb{R})$. We observe in both cases that the optimal rate is asymptotically attained and also that the error estimates are quite sharp.



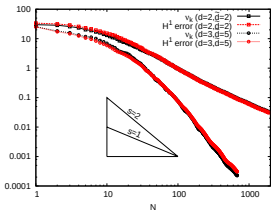
(a) (P1) with $j_0 = -\infty$.



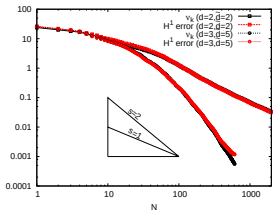
(b) (P2) with $j_0 = -\infty$.



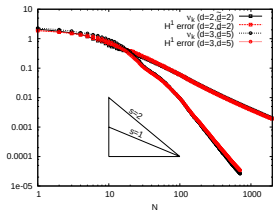
(c) (P3) with $j_0 = -\infty$.



(d) (P4) with $j_0 = -\infty$.

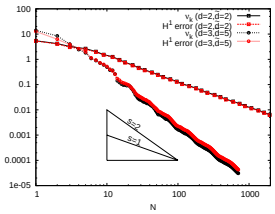


(e) (P5) with $j_0 = -\infty$.

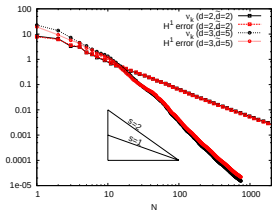


(f) (P6) with $j_0 = -\infty$.

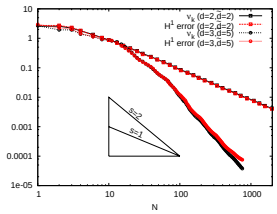
FIGURE 3.2. Residual estimator ν_k and $H^1(\mathbb{R})$-error for **ADWAV** over the degrees of freedom N.



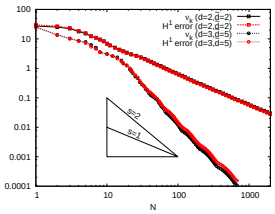
(a) (P1) with $j_0 = -2$.



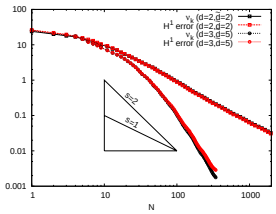
(b) (P2) with $j_0 = -4$.



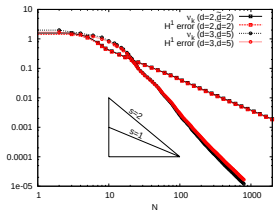
(c) (P3) with $j_0 = -1$.



(d) (P4) with $j_0 = 0$.



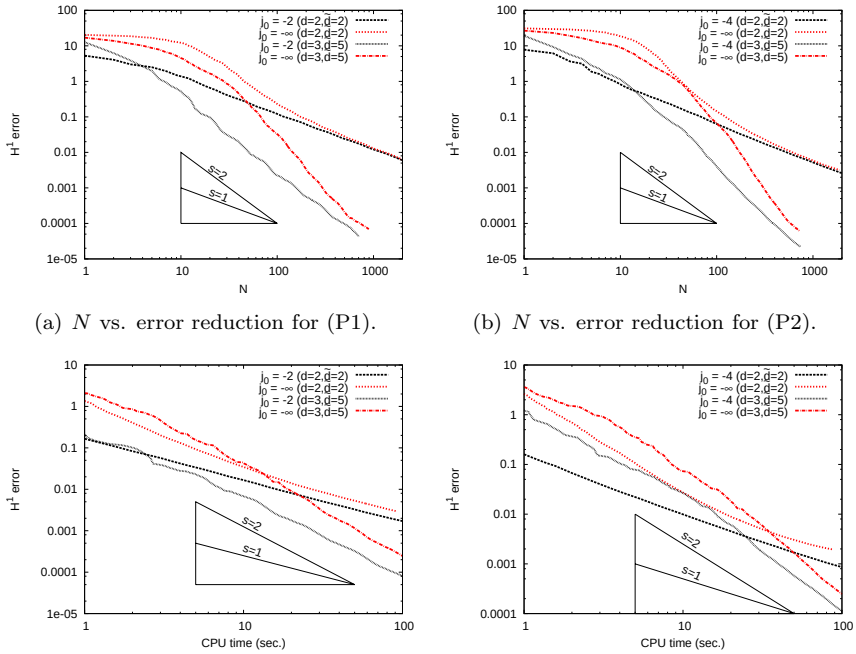
(e) (P5) with $j_0 = 0$.



(f) (P6) with $j_0 = -1$.

FIGURE 3.3. Residual estimator ν_k and $H^1(\mathbb{R})$-error for **ADWAV** over the degrees of freedom N.

3.3.5. *Discussion of the numerical results.* Despite the same asymptotic convergence rates for $j_0 = -\infty$ and $j_0 > -\infty$, there are some important quantitative differences between the two approaches. As an example, we consider (P1) and (P2). In Figure 3.4, we observe that the use of scaling functions on a minimal level j_0 significantly reduces the number of degrees of freedom required to attain a given target accuracy. This is due to the fact that few scaling functions suffice to approximate the polynomial part of the solution, whereas, in the case $j_0 = -\infty$, in order to obtain the same approximation using only wavelets, we also need wavelets on very low levels $j \leq 0$ which results in a higher number of degrees of freedom. Moreover, although we have a very simple structure in the basis for $j_0 = -\infty$ (we do not have to distinguish between wavelet and scaling functions), this advantage does not payoff as we can see from the computation times in Figure 3.4 c) and d).



(a) N vs. error reduction for (P1).

(b) N vs. error reduction for (P2).

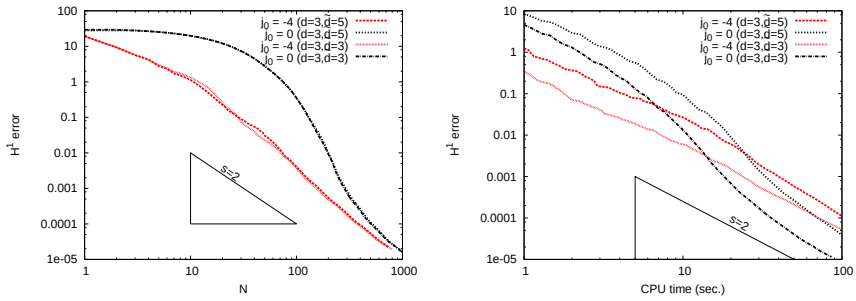
(c) Time vs. error reduction for (P1).

(d) Time vs. error reduction for (P2).

FIGURE 3.4. Comparison of the approximation errors ($H^1(\mathbb{R})$-error) produced by **ADWAV** for $j_0 = \infty$ and $j_0 > -\infty$.

Next, we compare the influence of the choice of the minimal level j_0 and the choice of the number of vanishing moments to the error reduction and the CPU times. Representatively, we consider (P2) for which the result can be found in Figure 3.5. We see in Figure 3.5 a) that the rate of convergence with respect to the number of degrees of freedom does not depend on $\widetilde{d}$. Observe that the minimal level $j_0 = 0$ results in a worse rate of convergence compared to $j_0 = -4$. But if we take the required computation time into account (cf. Figure 3.5 b)), we observe that, due a better condition number, the scheme converges asymptotically faster for $j_0 = 0$. Moreover, due to the fact that wavelets with $\widetilde{d} = 3$ have shorter support lengths

and less singular points than the one with $\widetilde{d} = 5$, the convergence of **ADWAV** is faster for $\widetilde{d} = 3$. At this point, note that the wavelet basis with $d = 3$, $\widetilde{d} = 3$ provides less favorable condition numbers for $j_0 \to -\infty$ (cf. Table 3.1).



(a) N vs. error reduction. (b) Time vs. error reduction.

FIGURE 3.5. Influence of the minimal level j_0 and comparison between $d = 3$, $\widetilde{d} = 3$ and $d = 3$, $\widetilde{d} = 5$ for (P2).

Our numerical results indicate that the use of scaling functions on a coarse level j_0 is favorable. Nevertheless, as we can see in Figures 3.2 and 3.3, both versions of **ADWAV** require a lot of iterations. In order to reduce this number, one can increase the value for α which mainly determines the number of added indices per iteration in **GROW**. But doing so, we loose the guaranteed convergence. Alternatively, we present a heuristic algorithm in the next section.

4. 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, 30]. 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 $\mathbf{\Lambda}^{(k)}$ by a heuristic approach. Moreover, we do not need the assumption that $\mathcal{A}$ is self-adjoint.

Motivated by the numerical results from Section 3, we shall use from now on exclusively the collection $\bar{\Psi}$ defined in (2.6) for a fixed coarse level $\mathbf{j}_0 = (j_0^{(1)}, \dots, j_0^{(n)})$. To simplify notations, we set $\psi_{j_0^{(i)}-1,k_i} := \varphi_{j_0^{(i)},k_i}$ for $k_i \in \mathbb{Z}$ and define $\bar{\mathcal{J}}_i := \{\lambda_i := (j_i, k_i) : \ j_0^{(i)} - 1 \le j_i, k_i \in \mathbb{Z}\}$ for $i = 1, \dots, n$ such that with $\bar{\mathbf{J}} := \bar{\mathcal{J}}_1 \times \cdots \times \bar{\mathcal{J}}_n \subset \mathbf{J}$, we have $\bar{\Psi} = \{\psi_{\boldsymbol{\lambda}} : \ \boldsymbol{\lambda} \in \bar{\mathbf{J}}\}$.

4.1. Algorithm.

We start by describing the main components of the algorithm.

4.1.1. *Numerical solution of the Galerkin system.* In each iteration of the algorithm, we have to solve a Galerkin system (2.10) for an index set $\mathbf{\Lambda} \subset \bar{\mathbf{J}}$. But in general one only solves a perturbed linear system

$$\widetilde{\mathbf{A}}_{\mathbf{\Lambda}} \widetilde{\mathbf{u}}_{\mathbf{\Lambda}} = \widetilde{\mathbf{f}}_{\mathbf{\Lambda}}, \quad \widetilde{\mathbf{f}}_{\mathbf{\Lambda}} := \mathbf{f}_{\mathbf{\Lambda}} + \widetilde{\mathbf{s}}_{\mathbf{\Lambda}}, \tag{4.10}$$

with $\|\widetilde{\mathbf{s}}_\Lambda\|_{\ell_2} < \eta_{\mathbf{f}}$, $\eta_{\mathbf{f}} > 0$ a given tolerance, instead of solving (2.10) 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 4.1. *Let $\boldsymbol{\Lambda}$ be a finite subset of $\bar{\mathbf{J}}$ and assume that $\|\mathbf{A}_{\boldsymbol{\Lambda}} - \widetilde{\mathbf{A}}_{\boldsymbol{\Lambda}}\| < \eta_{\mathbf{A}}$ with $\eta_{\mathbf{A}} < c_1$ and $\|\mathbf{f}_{\boldsymbol{\Lambda}} - \widetilde{\mathbf{f}}_{\boldsymbol{\Lambda}}\|_{\ell_2} < \eta_{\mathbf{f}}$. Then, $\widetilde{\mathbf{A}}_{\boldsymbol{\Lambda}}$, $\widetilde{\mathbf{A}}_{\boldsymbol{\Lambda}}^{-1}$ are uniformly bounded and it holds that*

$$\|\mathbf{u}_{\boldsymbol{\Lambda}} - \widetilde{\mathbf{u}}_{\boldsymbol{\Lambda}}\|_{\ell_2} \leq c_1^{-1}\Big(\frac{\eta_{\mathbf{A}}}{c_1 - \eta_{\mathbf{A}}}\big(\eta_{\mathbf{f}} + \|\mathbf{f}_{\boldsymbol{\Lambda}}\|_{\ell_2}\big) + \eta_{\mathbf{f}}\Big), \tag{4.11}$$

where $\mathbf{u}_{\boldsymbol{\Lambda}}$ is the solution of (2.10) and $\widetilde{\mathbf{u}}_\Lambda$ is the solution of (4.10).

Proof. First, we prove continuity and coercivity of the bilinear form induced by $\widetilde{\mathbf{A}}_{\boldsymbol{\Lambda}}$ on $\ell_2(\boldsymbol{\Lambda})$. From the assumptions, we infer from (2.8) that for all $\mathbf{w}_{\boldsymbol{\Lambda}}, \mathbf{v}_{\boldsymbol{\Lambda}} \in \ell_2(\boldsymbol{\Lambda})$

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

with c_2 from (2.8). Moreover, for all $\mathbf{v}_{\boldsymbol{\Lambda}} \in \ell_2(\boldsymbol{\Lambda})$ we have

$$(c_1 - \eta_{\mathbf{A}})\|\mathbf{v}_{\boldsymbol{\Lambda}}\|_{\ell_2}^2 \leq \langle\, \widetilde{\mathbf{A}}_{\boldsymbol{\Lambda}}\mathbf{v}_{\boldsymbol{\Lambda}}, \mathbf{v}_{\boldsymbol{\Lambda}}\,\rangle_{\ell_2}, \tag{4.12}$$

with c_1 from (2.8). Now, the following estimate is straightforward:

$$\begin{aligned}
\|\mathbf{u}_{\boldsymbol{\Lambda}} - \widetilde{\mathbf{u}}_{\boldsymbol{\Lambda}}\|_{\ell_2}^2 &\leq c_1^{-1}\langle\, \mathbf{A}_{\boldsymbol{\Lambda}}(\mathbf{u}_{\boldsymbol{\Lambda}} - \widetilde{\mathbf{u}}_{\boldsymbol{\Lambda}}), \mathbf{u}_{\boldsymbol{\Lambda}} - \widetilde{\mathbf{u}}_{\boldsymbol{\Lambda}}\,\rangle_{\ell_2} \\
&\leq c_1^{-1}\langle\, \mathbf{A}_{\boldsymbol{\Lambda}}\mathbf{u}_{\boldsymbol{\Lambda}} - \widetilde{\mathbf{A}}_{\boldsymbol{\Lambda}}\widetilde{\mathbf{u}}_{\boldsymbol{\Lambda}} + \widetilde{\mathbf{A}}_{\boldsymbol{\Lambda}}\widetilde{\mathbf{u}}_{\boldsymbol{\Lambda}} - \mathbf{A}_{\boldsymbol{\Lambda}}\widetilde{\mathbf{u}}_{\boldsymbol{\Lambda}}, \mathbf{u}_{\boldsymbol{\Lambda}} - \widetilde{\mathbf{u}}_{\boldsymbol{\Lambda}}\,\rangle_{\ell_2} \\
&\leq c_1^{-1}\big(\|(\mathbf{A}_{\boldsymbol{\Lambda}} - \widetilde{\mathbf{A}}_{\boldsymbol{\Lambda}})\widetilde{\mathbf{u}}_{\boldsymbol{\Lambda}}\|_{\ell_2}\|\mathbf{u}_{\boldsymbol{\Lambda}} - \widetilde{\mathbf{u}}_{\boldsymbol{\Lambda}}\|_{\ell_2} + \langle\, \mathbf{f}_{\boldsymbol{\Lambda}} - \widetilde{\mathbf{f}}_{\boldsymbol{\Lambda}}, \mathbf{u}_{\boldsymbol{\Lambda}} - \widetilde{\mathbf{u}}_{\boldsymbol{\Lambda}}\,\rangle_{\ell_2}\big) \\
&\leq c_1^{-1}\|\mathbf{u}_{\boldsymbol{\Lambda}} - \widetilde{\mathbf{u}}_{\boldsymbol{\Lambda}}\|_{\ell_2}\left(\eta_{\mathbf{A}}\|\widetilde{\mathbf{u}}_{\boldsymbol{\Lambda}}\|_{\ell_2} + \eta_{\mathbf{f}}\right).
\end{aligned}$$

From estimate (4.12) we finally get

$$\|\widetilde{\mathbf{u}}_{\boldsymbol{\Lambda}}\|_{\ell_2} \leq (c_1 - \eta_{\mathbf{A}})^{-1}\|\mathbf{f}_{\boldsymbol{\Lambda}} + \widetilde{\mathbf{s}}_{\boldsymbol{\Lambda}}\|_{\ell_2} \leq (c_1 - \eta_{\mathbf{A}})^{-1}\left(\eta_{\mathbf{f}} + \|\mathbf{f}_{\boldsymbol{\Lambda}}\|_{\ell_2}\right). \tag{4.13}$$

□

Remark 4.2. We remark that ellipticity is not a necessary condition for estimate (4.11). The proof of Proposition 4.1 remains essentially the same if we only require that the norms of $\mathbf{A}$, $\mathbf{A}^{-1}$ and their perturbations are uniformly bounded.

From Proposition 4.1, we infer that there is no gain if one of the tolerances $\eta_{\mathbf{f}}$ or $\eta_{\mathbf{A}}$ is much smaller than the other. For this reason, we assume from now on that for a given tolerance $\mathrm{tol}_{\,\mathrm{iter}}$,

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

so that (4.11) can be replaced by $\|\mathbf{u}_\Lambda - \widetilde{\mathbf{u}}_\Lambda\|_{\ell_2} \lesssim \mathrm{tol}_{\,\mathrm{iter}}\|\mathbf{f}_\Lambda\|_{\ell_2}$. In particular, we have in mind to use a compressed matrix $\mathbf{A}_{J,\boldsymbol{\Lambda}} = \mathbf{P}_{\boldsymbol{\Lambda}}\mathbf{A}_J|_{\ell_2(\boldsymbol{\Lambda})}$ for $\mathbf{A} \in \mathcal{B}_s$ to solve the Galerkin system (2.10) approximately using the compression results from [28]. Under the assumption that this is possible, we propose the routine **LINSOLVE** (Algorithm 5) to solve appearing linear systems.

Remark 4.3. Concerning the complexity of **LINSOLVE**, observe that for a given tolerance $\mathrm{tol}_{\,\mathrm{iter}}$, there exists an increasing function $C(\mathrm{tol}_{\,\mathrm{iter}})$ such that the assembling of $\widetilde{\mathbf{f}}_{\boldsymbol{\Lambda}}$ can be obtained in $\mathcal{O}(C(\mathrm{tol}_{\,\mathrm{iter}})\cdot\#\boldsymbol{\Lambda})$ operations where $C(\mathrm{tol}_{\,\mathrm{iter}}) \to \infty$ if $\mathrm{tol}_{\,\mathrm{iter}} \to 0$. This is due to the fact that for a decreasing target tolerance $\mathrm{tol}_{\,\mathrm{iter}}$, we require better approximations $\widetilde{\mathbf{f}}_{\boldsymbol{\Lambda}}$ to $\mathbf{f}_{\boldsymbol{\Lambda}}$ which leads to a increasing complexity. The same can be found for the assembling of $\mathbf{A}_{J,\boldsymbol{\Lambda}}$. The compression level J is actually

Algorithm 5 LINSOLVE$[\boldsymbol{\Lambda}, \mathbf{w}_{\boldsymbol{\Lambda}}, \mathrm{tol}_{\mathrm{iter}}] \to \widetilde{\mathbf{u}}_{\boldsymbol{\Lambda}}$

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

an increasing function of $\mathrm{tol}_{\mathrm{iter}}$ such that $J(\mathrm{tol}_{\mathrm{iter}}) \to \infty$ if $\mathrm{tol}_{\mathrm{iter}} \to 0$. Therefore, the complexity for the assembling of $\mathbf{A}_{J,\boldsymbol{\Lambda}}$ is of order $\mathcal{O}(\min\{\#\boldsymbol{\Lambda}, J(\mathrm{tol}_{\mathrm{iter}})\} \cdot \#\boldsymbol{\Lambda})$ and thus asymptotically not optimal. This is in contrast to the method **GALSOLVE** where the maximal compression level $J(\varepsilon)$ is uniformly bounded for all $\varepsilon > 0$. Moreover, it is a priori not clear how the number of iterations of the inner linear solver depends on $\mathrm{tol}_{\mathrm{iter}}$. We shall come back to this issue in Section 4.3.

4.1.2. *Residual computation.* A large part of the computation time of the **ADWAV** algorithm is used by **APPLY**. Alternatively, starting from an index set $\boldsymbol{\Lambda}$ and the corresponding Galerkin system (2.10), one can compute a so called *security zone* $\widehat{\boldsymbol{\Lambda}} \supset \boldsymbol{\Lambda}$ using Algorithm 6 (cf. [30, p.235]). For a constant $c > 0$, we set for $\lambda_i \in \bar{\mathcal{J}}_i$ with $\operatorname{supp} \psi_{\lambda_i} = [l_{\lambda_i}, u_{\lambda_i}]$ and $i = 1, \dots, n$:

$$\mathcal{C}(\lambda_i, c) := \{\mu \in \bar{\mathcal{J}}_i : \ |\operatorname{supp} \psi_\mu \cap c \cdot \operatorname{supp} \psi_{\lambda_i}| > 0, \ |\lambda_i| \le |\mu| \le |\lambda_i| + 1\}, \tag{4.14}$$

where $c \cdot \operatorname{supp} \psi_{\lambda_i} := [c\, l_{\lambda_i} + (1-c) z_{\lambda_i}, c\, u_{\lambda_i} + (1-c) z_{\lambda_i}]$, $z_{\lambda_i} = \frac{1}{2}(l_{\lambda_i} + u_{\lambda_i})$, is the contracted support of ψ_{λ_i} around its barycenter. Due to the locality of the basis function ψ_{λ_i}, the cardinality of $\mathcal{C}(\lambda_i, c)$ for $i = 1, \dots, n$ is uniformly bounded by a constant $M(c) > 0$. Observe that not only wavelets on higher levels are inserted into the security zone, but also further scaling function indices are added for $|\lambda_i| = j_0^{(i)} - 1$. This will later on permit an adaptive truncation of the formerly unbounded domain.

Remark 4.4. We emphasize that the security zone $\mathcal{C}(\lambda_i, c)$ in (4.14) is constructed such that no indices λ_i with levels $|\lambda_i| < j_0^{(i)} - 1$ are inserted. Since both scaling functions and wavelets on level $j_0^{(i)} - 1$ are linear combinations of scaling functions on level $j_0^{(i)}$, this would a result in an over-determined system.

Algorithm 6 C$[\boldsymbol{\Lambda}, c] \to \widehat{\boldsymbol{\Lambda}}$

1: $\widehat{\boldsymbol{\Lambda}} := \emptyset$.
2: **for** $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_n) \in \boldsymbol{\Lambda}$ **do**
3: **for** $i = 1, \dots, n$ **do**
4: Compute $\mathcal{C}(\lambda_i, c)$.
5: **end for**
6: Define $\mathcal{C}(\boldsymbol{\lambda}, c) := \mathcal{C}(\lambda_1, c) \times \lambda_2 \times \cdots \times \lambda_n \cup \cdots \cup \lambda_1 \times \cdots \times \lambda_{n-1} \times \mathcal{C}(\lambda_n, c)$.
7: Set $\widehat{\boldsymbol{\Lambda}} := \widehat{\boldsymbol{\Lambda}} \cup \mathcal{C}(\boldsymbol{\lambda}, c)$.
8: **end for**

As seen above, for any $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_n) \in \bar{\mathbf{J}}$, $\#\mathcal{C}(\lambda_i, c) \le M(c)$ for $i = 1, \dots, n$. Thus, the cardinality of $\mathcal{C}(\boldsymbol{\lambda}, c)$ defined in Algorithm 6 is bounded by $n \cdot M(c)$. For this reason, the cardinality of the output of $\mathbf{C}[\boldsymbol{\Lambda}, c]$ such as the complexity of this routine are both of order $\mathcal{O}(n \cdot M(c) \cdot \#\boldsymbol{\Lambda})$.

As an estimate of the residual $\mathbf{A}\mathbf{u}_{\boldsymbol{\Lambda}} - \mathbf{f}$, we now take

$$\mathbf{r}_{\widehat{\boldsymbol{\Lambda}}} := \mathbf{P}_{\widehat{\boldsymbol{\Lambda}}}(\mathbf{A}\mathbf{u}_{\boldsymbol{\Lambda}} - \mathbf{f}), \tag{4.15}$$

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

$$\widetilde{\mathbf{r}}_{\widehat{\boldsymbol{\Lambda}}} := \mathbf{P}_{\widehat{\boldsymbol{\Lambda}}}(\mathbf{A}_{J,\boldsymbol{\Lambda}}\widetilde{\mathbf{u}}_{\boldsymbol{\Lambda}} - \widetilde{\mathbf{f}}), \tag{4.16}$$

using the compressed matrix $\mathbf{A}_{J,\boldsymbol{\Lambda}}$ and an approximate solution $\widetilde{\mathbf{u}}_{\boldsymbol{\Lambda}}$ from (4.10). For the next result we need the following notation: If $\mathbf{v}_\Lambda$ is a vector in $\ell_2(\boldsymbol{\Lambda})$ with finite support $\boldsymbol{\Lambda}$, then $\mathbf{v}_{\widehat{\boldsymbol{\Lambda}}}$ denotes its extension by zeros to $\widehat{\boldsymbol{\Lambda}}$.

Proposition 4.5. *Let $\boldsymbol{\Lambda}, \widehat{\boldsymbol{\Lambda}}$ be finite subsets of $\bar{\mathbf{J}}$ with $\widehat{\boldsymbol{\Lambda}} \supset \boldsymbol{\Lambda}$ and assume that the assumptions from Proposition 4.1 hold. Then, by setting $\widetilde{\mathbf{A}}_{\widehat{\boldsymbol{\Lambda}}} := \mathbf{A}_{J,\widehat{\boldsymbol{\Lambda}}}$ for sufficiently large J, we have*

$$\|\widetilde{\mathbf{r}}_{\widehat{\boldsymbol{\Lambda}}} - \mathbf{r}_{\widehat{\boldsymbol{\Lambda}}}\|_{\ell_2} \lesssim \mathrm{tol}_{\mathrm{iter}}\, \|\mathbf{f}_{\widehat{\boldsymbol{\Lambda}}}\|_{\ell_2}. \tag{4.17}$$

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

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

From Proposition 4.1 we get that $\|\widetilde{\mathbf{u}}_{\boldsymbol{\Lambda}} - \mathbf{u}_{\boldsymbol{\Lambda}}\|_{\ell_2} \lesssim \mathrm{tol}_{\mathrm{iter}}\, \|\mathbf{f}_{\boldsymbol{\Lambda}}\|_{\ell_2}$. Moreover, as in (4.13), we see that $\|\mathbf{u}_{\boldsymbol{\Lambda}}\|_{\ell_2}\|\mathbf{A}_{\widehat{\boldsymbol{\Lambda}}} - \mathbf{A}_{J,\widehat{\boldsymbol{\Lambda}}}\| \lesssim \mathrm{tol}_{\mathrm{iter}}\, \|\mathbf{f}_{\boldsymbol{\Lambda}}\|_{\ell_2}$. □

At this point we recall that replacing the exact, infinite residual $\mathbf{A}\mathbf{u}_{\boldsymbol{\Lambda}} - \mathbf{f}$ by $\widetilde{\mathbf{r}}_{\widehat{\boldsymbol{\Lambda}}}$ is an heuristic approach. On the one hand, there is no proof of the existence of $0 < \beta < 1$ such that $\|\mathbf{P}_{\widehat{\boldsymbol{\Lambda}}}(\mathbf{A}\mathbf{u}_{\boldsymbol{\Lambda}} - \mathbf{f})\|_{\ell_2} \ge \beta\|\mathbf{A}\mathbf{u}_{\boldsymbol{\Lambda}} - \mathbf{f}\|_{\ell_2}$. On the other hand, the advantage is that we do not need **APPLY** or **RHS**.

Algorithm 7 RESIDUAL$[\widehat{\boldsymbol{\Lambda}}, \widetilde{\mathbf{u}}_{\boldsymbol{\Lambda}}, \mathrm{tol}_{\mathrm{iter}}] \to \widetilde{\mathbf{r}}_{\widehat{\boldsymbol{\Lambda}}}$

1: Estimate $J \in \mathbb{N}$ such that $\|\mathbf{A}_{\widehat{\boldsymbol{\Lambda}}} - \mathbf{A}_{J,\widehat{\boldsymbol{\Lambda}}}\|_{\ell_2} < \min\{c_1, \mathrm{tol}_{\mathrm{iter}}\}$.
2: Compute $\widetilde{\mathbf{r}}_{\widehat{\boldsymbol{\Lambda}}}$ according to (4.16).

4.1.3. *Coefficient thresholding.* Obviously, if we call iteratively $\boldsymbol{\Lambda}^{(k+1)} = \mathbf{C}(\boldsymbol{\Lambda}^{(k)}, c)$ starting with some initial set $\boldsymbol{\Lambda}^{(0)}$, the sizes of the index sets $(\boldsymbol{\Lambda}^{(k)})_{k\in\mathbb{N}}$ may grow exponentially fast. For this reason, we have to keep the index sets for which we call $\mathbf{C}$ small. This is realized by thresholding the wavelet coefficients in $\widetilde{\mathbf{u}}_{\boldsymbol{\Lambda}^{(k)}}$ and in the estimated residual $\widetilde{\mathbf{r}}_{\boldsymbol{\Lambda}^{(k)}}$. For this purpose, Algorithm 8 realizes a threshold on finitely supported vectors $\mathbf{v}$ and returns a vector $\bar{\mathbf{v}}$ such that $\|\mathbf{v} - \bar{\mathbf{v}}\|_{\ell_2} \le \delta$ for a given tolerance $\delta > 0$. Here, also approximate sorting procedures from [1, 13] can be used, so that **THRESH** can be realized in linear complexity.

Let us now investigate what the effect of **THRESH** used to threshold the output of **LINSOLVE**. Therefore, let $\mathbf{u}_{\boldsymbol{\Lambda}}$ be the solution of (2.10) and $\widetilde{\mathbf{u}}_{\boldsymbol{\Lambda}} :=$ **LINSOLVE**$[\boldsymbol{\Lambda}, \mathbf{w}_{\boldsymbol{\Lambda}}, \mathrm{tol}_{\mathrm{iter}}]$ for some starting point $\mathbf{w}_{\boldsymbol{\Lambda}}$. Then, as $\|\mathbf{u}_{\boldsymbol{\Lambda}} - \widetilde{\mathbf{u}}_{\boldsymbol{\Lambda}}\|_{\ell_2} \lesssim \mathrm{tol}_{\mathrm{iter}}\|\mathbf{f}_{\boldsymbol{\Lambda}}\|_{\ell_2}$, it holds for $\bar{\mathbf{u}} =$**THRESH**$[\widetilde{\mathbf{u}}_{\boldsymbol{\Lambda}}, \mathrm{tol}_{\mathrm{iter}}]$ that

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

Algorithm 8 THRESH$[\mathbf{v}, \delta] \to \bar{\mathbf{v}}$

1: Sort the vector $\mathbf{v} = (\mathbf{v}_{\boldsymbol{\lambda}})_{\boldsymbol{\lambda}\in\operatorname{supp}\mathbf{v}}$ by decreasing order which yields the sorted vector $\mathbf{v}^* = (\mathbf{v}^*_{(i,\boldsymbol{\lambda}_i)})_{i=1,\dots,M}$ where $M := \#(\operatorname{supp}\mathbf{v})$ and $(i,\boldsymbol{\lambda}_i)$ for $i = 1,\dots,M$ indicate the ordering in $\mathbf{v}^*$ such as the corresponding index in $\operatorname{supp}\mathbf{v}$.
2: Compute $\|\mathbf{v}\|_{\ell_2}$. The vector $\overline{\mathbf{v}}$ and its support $\boldsymbol{\Lambda}$ are given by $\overline{\mathbf{v}} := (\mathbf{v}_{\boldsymbol{\lambda}_i})_{i=1,\dots,K}$ and $\boldsymbol{\Lambda} := \{\boldsymbol{\lambda}_1,\dots,\boldsymbol{\lambda}_K\}$ where K is the smallest integer such that $\sum_{i=1}^{K}|\mathbf{v}^*_{(i,\boldsymbol{\lambda}_i)}|^2 \geq \|\mathbf{v}\|^2_{\ell_2} - \delta^2$.

where $\operatorname{supp}\bar{\mathbf{u}} \subseteq \boldsymbol{\Lambda}$. Hence, we get an approximation of order $\operatorname{tol}_{\mathrm{iter}}$ to the Galerkin solution $\mathbf{u}_{\boldsymbol{\Lambda}}$ but in general with a much smaller support. This observation can also be made for the residual computation. Let $\mathbf{r}_{\boldsymbol{\Lambda}}$ be the residual defined in (4.15) and $\widetilde{\mathbf{r}}_{\widehat{\boldsymbol{\Lambda}}} = \mathbf{RESIDUAL}[\widehat{\boldsymbol{\Lambda}}, \widetilde{\mathbf{u}}_{\boldsymbol{\Lambda}}, \operatorname{tol}_{\mathrm{iter}}]$. Then, by Proposition 4.5, $\|\mathbf{r}_{\widehat{\boldsymbol{\Lambda}}} - \widetilde{\mathbf{r}}_{\widehat{\boldsymbol{\Lambda}}}\|_{\ell_2} \lesssim \operatorname{tol}_{\mathrm{iter}}\|\mathbf{f}_{\widehat{\boldsymbol{\Lambda}}}\|_{\ell_2}$. Thus, for $\bar{\mathbf{r}} = \mathbf{THRESH}[\widetilde{\mathbf{r}}_{\widehat{\boldsymbol{\Lambda}}}, \operatorname{tol}_{\mathrm{iter}}]$, it holds $\|\mathbf{r}_{\widehat{\boldsymbol{\Lambda}}} - \bar{\mathbf{r}}\|_{\ell_2} \lesssim \operatorname{tol}_{\mathrm{iter}}(1 + \|\mathbf{f}_{\widehat{\boldsymbol{\Lambda}}}\|_{\ell_2})$, where, as above, the support of $\bar{\mathbf{r}}$ is in general much smaller than $\widehat{\boldsymbol{\Lambda}}$.

4.1.4. *The simplified algorithm* **S-ADWAV**. Now that we have all necessary routines together, we can describe the complete algorithm. **S-ADWAV** (cf. Algorithm 9 below) computes in each iteration an approximate solution $\mathbf{w}^{(k)}$ to the Galerkin system $\mathbf{A}_{\boldsymbol{\Lambda}_k^{\mathrm{cand.}}}\mathbf{w}^{(k)} = \mathbf{f}_{\boldsymbol{\Lambda}_k^{\mathrm{cand.}}}$ where $\boldsymbol{\Lambda}_k^{\mathrm{cand.}}$ 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 $\operatorname{tol}_{\mathrm{iter}}$. 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 $\|\mathbf{u}^{(k)} - \mathbf{w}^{(k)}\|_{\ell_2} \leq \operatorname{tol}_{\mathrm{iter}}$. Its support $\boldsymbol{\Lambda}^{(k)} := \operatorname{supp}\mathbf{u}^{(k)}$ is referred to as the set of *active* indices. Around the support of $\mathbf{u}^{(k)}$, the security zone $\widehat{\boldsymbol{\Lambda}}_k$ is constructed using the routine **C** and the residual $\mathbf{r}_{\widehat{\boldsymbol{\Lambda}}_k} = \mathbf{P}_{\widehat{\boldsymbol{\Lambda}}_k}(\mathbf{A}\mathbf{u}^{(k)} - \mathbf{f})$ is computed. If $\|\mathbf{r}_{\widehat{\boldsymbol{\Lambda}}_k}\|_{\ell_2}$ is smaller than the given target tolerance, then we accept $\mathbf{u}^{(k)}$ as solution. Otherwise, a new candidate set of activable indices $\boldsymbol{\Lambda}_{k+1}^{\mathrm{cand.}}$ is constructed by thresholding the residual $\mathbf{r}_{\widehat{\boldsymbol{\Lambda}}}$. As we always use the same tolerance $\operatorname{tol}_{\mathrm{iter}}$ for thresholding and the numerical solution of the Galerkin system, the approximation errors we generate are all of order $\operatorname{tol}_{\mathrm{iter}}$ (see Propositions 4.1 and 4.5 such as Paragraph 4.1.3). 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 $\boldsymbol{\Lambda}^{(k)} = \boldsymbol{\Lambda}^{(k+1)}$. Therefore, in addition to the algorithm described in [2] and [30], we decrease the threshold tolerance $\operatorname{tol}_{\mathrm{iter}}$ by a factor $\frac{1}{2}$ if the difference of the relative residuals of two iterations is too close to zero. Thus, in case of $\boldsymbol{\Lambda}^{(k+1)} = \boldsymbol{\Lambda}^{(k)}$, the threshold tolerance is decreased and therefore, we obtain $\mathbf{w}^{(k)} = \mathbf{w}^{(k+1)}$ but $\operatorname{supp}\mathbf{u}^{(k+1)} \supseteq \operatorname{supp}\mathbf{u}^{(k)}$ 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 $\boldsymbol{\Lambda}_k = \boldsymbol{\Lambda}_{k+m}$ for some $m \geq 2$ cannot not be excluded, we add an inner loop with a maximal number M of iterations which prevents such loops and ensures that the tolerance $\operatorname{tol}_{\mathrm{iter}}$ decreases.

Algorithm 9 $[\mathbf{u}(\varepsilon), \mathbf{\Lambda}(\varepsilon)] = \mathbf{S\text{-}ADWAV}[\varepsilon]$

Let $h > 0$ be a control width, M a fixed number of inner loops, $c > 0$, ρ a tuning parameter, $\mathrm{tol}_{\mathrm{iter}} > 0$ an initial tolerance and $\mathbf{\Lambda}_{1,1}^{\mathrm{cand.}}$ an initial index set.

1: **for** $k = 1, 2, 3, \ldots$ **do**
2: **for** $m = 1, 2, \ldots, M$ **do**
3: $\mathbf{w}^{(k,m)} = \mathbf{LINSOLVE}[\mathbf{\Lambda}_{k,m}^{\mathrm{cand.}}, \mathbf{u}^{(k-1,m)}, \rho \cdot \mathrm{tol}_{\mathrm{iter}}]$
4: $\mathbf{u}^{(k,m)} = \mathbf{THRESH}[\mathbf{w}^{(k,m)}, \mathrm{tol}_{\mathrm{iter}}]$
5: $\mathbf{\Lambda}^{(k,m)} = \operatorname{supp} \mathbf{u}^{(k,m)}$; $\widehat{\mathbf{\Lambda}}_{k,m} = \mathbf{C}[\mathbf{\Lambda}^{(k,m)}, c]$
6: $\mathbf{r}^{(k,m)} = \mathbf{RESIDUAL}[\widehat{\mathbf{\Lambda}}_{k,m}, \mathbf{u}^{(k,m)}, \mathrm{tol}_{\mathrm{iter}}]$
7: **if** $\|\mathbf{r}^{(k,m)}\|_{\ell_2} \leq \varepsilon \|\mathbf{f}_{\widehat{\mathbf{\Lambda}}_{k,m}}\|_{\ell_2}$ **then**
8: $\mathbf{u}(\varepsilon) := \mathbf{u}^{(k,m)}$, $\mathbf{\Lambda}(\varepsilon) := \mathbf{\Lambda}^{(k,m)}$; EXIT
9: **end if**
10: $\bar{\mathbf{r}}^{(k)} = \mathbf{THRESH}[\mathbf{r}^{(k,m)}, \mathrm{tol}_{\mathrm{iter}}]$; $\mathbf{\Lambda}_{k+1,m}^{\mathrm{cand.}} = \operatorname{supp} \mathbf{u}^{(k)} \cup \operatorname{supp} \bar{\mathbf{r}}^{(k,m)}$
11: **if** $\left| \frac{\|\mathbf{r}^{(k,m)}\|_{\ell_2}}{\|\mathbf{P}_{\widehat{\mathbf{\Lambda}}_{k,m}} \mathbf{f}\|_{\ell_2}} - \frac{\|\mathbf{r}^{(k-1,m)}\|_{\ell_2}}{\|\mathbf{P}_{\widehat{\mathbf{\Lambda}}_{k-1,m}} \mathbf{f}\|_{\ell_2}} \right| < h$ **then**
12: BREAK
13: **end if**
14: **end for**
15: $\mathrm{tol}_{\mathrm{iter}} = \frac{1}{2} \mathrm{tol}_{\mathrm{iter}}$
16: **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 $\operatorname{supp} (\mathbf{u}^{(m,k)})^T \Psi$, is done implicitly. Every time $\mathbf{C}[\mathbf{\Lambda}^{(k,m)}, c]$ is called, additional scaling function indices on the coarsest level are added to the security zone $\widehat{\mathbf{\Lambda}}^{(k,m)}$. If 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 $\mathbf{\Lambda}_{k+1,m}^{\mathrm{cand.}}$ set after the call of $\mathbf{THRESH}[\mathbf{r}^{(k,m)}, \mathrm{tol}_{\mathrm{iter}}]$. This proceeding provides the possibility that in each iteration, the computational domain can be extended, but also truncated as we have another call of **THRESH** after the solution of the Galerkin system. Therefore, the call of **THRESH** in line 5 of Algorithm 9 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.

4.1.5. *Choice of a minimal level and an initial index set.* As for **ADWAV**, the choice of a minimal level is crucial. Choosing the level $|\boldsymbol{\lambda}|$ of the largest wavelet coefficient $\langle f, \psi_{\mathbf{\Lambda}} \rangle$ is one possibility to construct $\mathbf{\Lambda}_{1,1}^{\mathrm{cand.}}$ by adding the scaling function index with the same level and translation index as the largest wavelet coefficient. We emphasize that estimates for $\langle f, \psi_{\boldsymbol{\lambda}} \rangle$ are available in Section 3.2. This is the path we follow for the one-dimensional examples below. For the examples in two dimensions, we briefly describe an alternative for determining $\mathbf{j}_0$ and $\mathbf{\Lambda}_{1,1}^{\mathrm{cand.}}$.

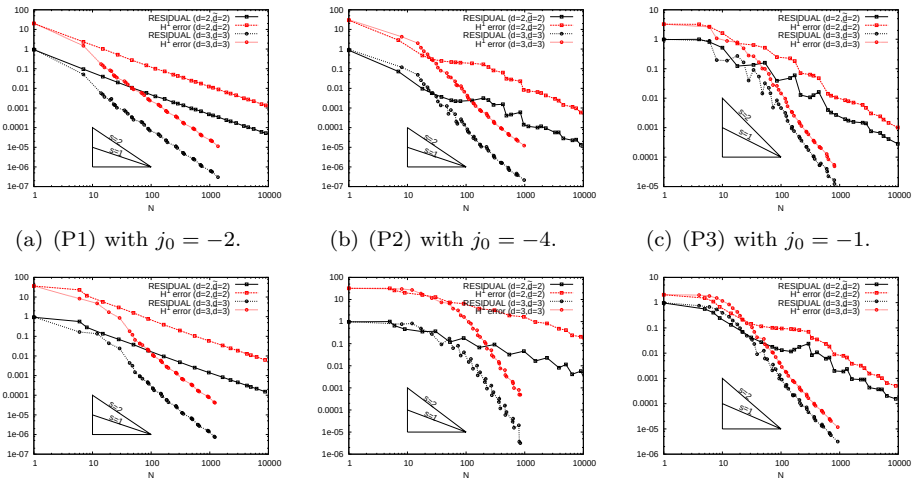
4.1.6. *Convergence and complexity.* As already mentioned at the beginning of this section, there is no proof for the convergence of **S-ADWAV** . In view of Remark 4.4 there is another issue concerning the complexity of **S-ADWAV**. For the routines

LINSOLVE and **RESIDUAL**, the constants $C(\mathrm{tol}_{\mathrm{iter}})$ (for assembling the right-hand side) and $J(\mathrm{tol}_{\mathrm{iter}})$ (for assembling the matrix) determine the complexity and are increasing if $\mathrm{tol}_{\mathrm{iter}}$ tends to zero which is actually the case in **S-ADWAV**. This shows that asymptotically, this algorithm cannot be optimal.

Nevertheless, we show in our numerical experiments that these constants do not grow fast for decreasing $\mathrm{tol}_{\mathrm{iter}}$ and still permit an efficient numerical algorithm. This is also due to the fact that we use solutions from former iterations as initial guesses in **LINSOLVE** so that the number of iterations of the linear solver only grows for very small values of $\mathrm{tol}_{\mathrm{iter}}$.

4.2. **Numerical experiments in one dimension.** In this section, we present numerical results obtained with **S-ADWAV** in one space dimension. We focus on the reaction-diffusion problems from Section 3.3 and compare the results with those obtained by **ADWAV**.

4.2.1. *Convergence rates.* Within **S-ADWAV**, we used the parameters $c = 0.125$, $h = 0.0001$, $M = 2$, $\rho = 0.1$, $\mathrm{tol}_{\mathrm{iter}} = 0.01$ and $d = \widetilde{d} = 2$ such as $d = \widetilde{d} = 3$. We observe in Figure 4.6 that both the output $\|\mathbf{r}^{(k,m)}\|_{\ell_2}$ of the heuristic residual error estimator **RESIDUAL** such as the corresponding approximation error measured in $H^1(\mathbb{R})$ converge asymptotically with the optimal rate $s = d - 1$. But we also observe that the error in **S-ADWAV** not necessarily decreases in each iteration – we observe jumps in both the residual and the $H^1(\mathbb{R})$-error (cf. Figures 4.6 c), e) and f)).



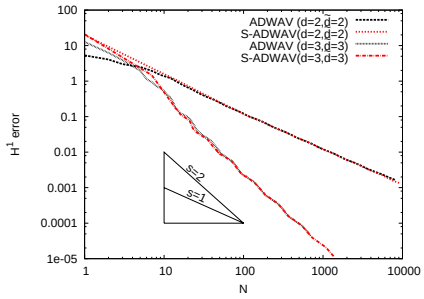
(a) (P1) with $j_0 = -2$. (b) (P2) with $j_0 = -4$. (c) (P3) with $j_0 = -1$.

(d) (P4) with $j_0 = 0$. (e) (P5) with $j_0 = 0$. (f) (P6) with $j_0 = -1$.

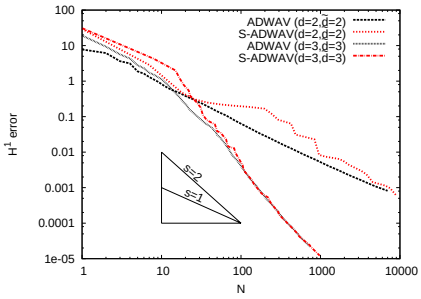
FIGURE 4.6. Output $\|\mathbf{r}^{(k,m)}\|_{\ell_2}$ of **RESIDUAL** and $H^1(\mathbb{R})$-error for **S-ADWAV** over the degrees of freedom N.

4.2.2. *Comparison of* **ADWAV** *and* **S-ADWAV**. Exemplarily, we consider (P1) and (P2) to compare the two adaptive schemes where we only consider **ADWAV** with $j_0 > -\infty$. In Figure 4.7 a), we see that for the smooth solution of (P1) there is nearly no difference between **ADWAV** and **S-ADWAV** with regard to required

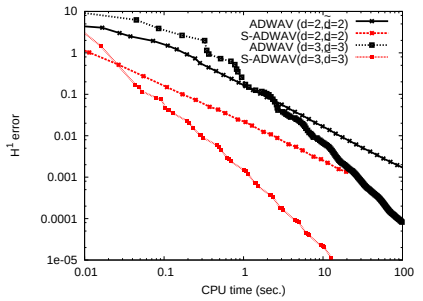
degrees of freedom to attain a certain target tolerance. In contrast, in case of the singular solution of (P2), **ADWAV** needs less degrees of freedom compared to **S-ADWAV** (cf. Figure 4.7 b)). This is due to the fact that within **GROW** higher levels for the resolution of a singularity can be added within one iteration whereas the routine **C** can add at most wavelets on the next higher level. Nevertheless, we observe from the computation times given in Figures 4.7 c) and d) that this effect is compensated by **S-ADWAV** which needs less outer iterations.



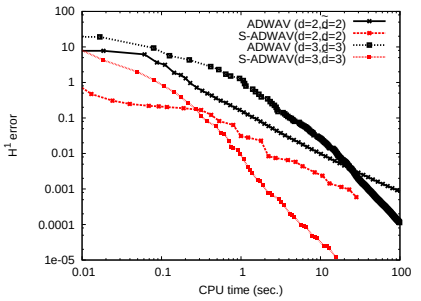
(a) N vs. error reduction (P1).



(b) N vs. error reduction (P2).



(c) CPU time vs. error reduction (P1).

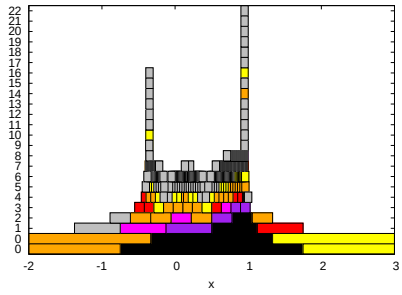


(d) CPU time vs. error reduction (P2).

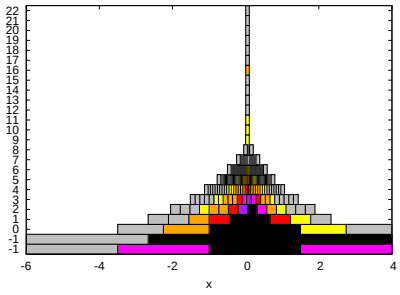
FIGURE 4.7. Output $\|\mathbf{r}^{(k,m)}\|_{\ell_2}$ of **RESIDUAL** and $H^1(\mathbb{R})$-error for **S-ADWAV** over the degrees of freedom N.

In Figure 4.8, we show the structure of the index sets produced by **ADWAV** (top row) and **S-ADWAV** (bottom row). Here, we exemplarily consider the reference solution (P5) with compact support (left column) and a singular reference solution with global support (P6). Using the example of (P5), we see that for a comparable size of index set sets, **ADWAV** uses the information provided by **RHS** to add higher levels already at early stages of the algorithm. As already said above, this is not the case for **S-ADWAV** which does not have this information. Nevertheless, we observe that both algorithms reliably detect singularities – both for the interval example (where the interval bounds are treated as singularities) and the globally supported solution of (P6)

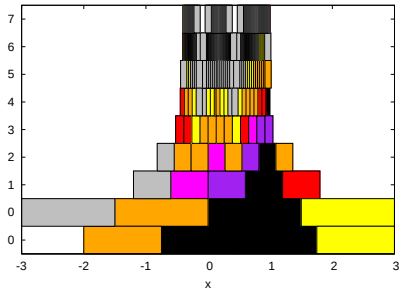
4.2.3. *A convection-diffusion problem.* For the reaction-diffusion examples one might argue that it would also be possible to a determine a computational domain a priori and then to use standard methods for PDEs on bounded domains. In order



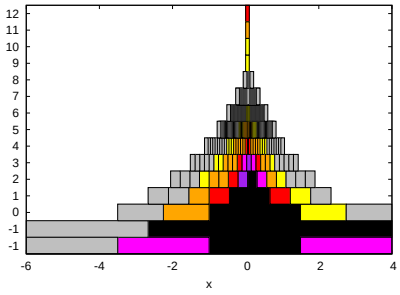
(a) (P5), **ADWAV** #supp $\mathbf{u}_{\mathbf{\Lambda}} = 345$.



(b) (P6), **ADWAV** #supp $\mathbf{u}_{\mathbf{\Lambda}} = 338$.



(c) (P5), **S-ADWAV** #supp $\mathbf{u}_{\mathbf{\Lambda}} = 300$.



(d) (P6), **S-ADWAV** #supp $\mathbf{u}_{\mathbf{\Lambda}} = 336$.

FIGURE 4.8. Numerical solutions $\mathbf{u}_{\mathbf{\Lambda}}$ for (P5) (left) and (P6) right obtained with **ADWAV** (top) and **S-ADWAV** (bottom) for $d = \widetilde{d} = 2$.

to treat a problem where this is not that obvious, we consider a convection diffusion problem of the form

$$-u''(x) + \beta u'(x) + u(x) = f_1(x), \quad x \in \mathbb{R}, \tag{4.18}$$

using the right-hand side from (P1) which also fulfills all required assumptions. For increasing values of β, the solution exhibits a strong layer at $x = 0$, see the left part of Figure 4.9. On the right side of Figure 4.9, we see the adaptive truncation of the computational domain. In particular, the layer is automatically detected.

4.3. **Examples in two dimensions.** Finally, we consider some bivariate problems. As above, we consider the following reaction-diffusion problem on $H^1(\mathbb{R}^2)$:

$$-\Delta u + u = f, \quad u \in H^1(\mathbb{R}^2), \tag{4.19}$$

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

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

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

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

(P10) $u_{10}(x_1, x_2) := e^{-\sqrt{(x_1-0.1)^2+(x_2-0.1)^2}}$,

The tensor product structure of the reference solutions u_7 and u_8 permits us, at least in theory, to obtain the best possible approximation $s = d - 1$ for (P7) and

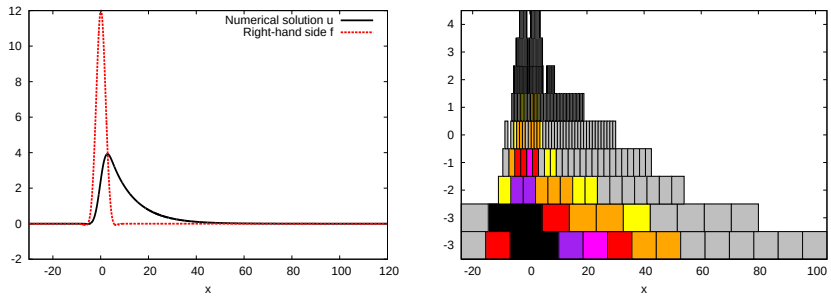


FIGURE 4.9. Solution u, right-hand side f (left) and estimated index set (right) for (4.18) with $d = \widetilde{d} = 2$, $j_0 = -2$ and $\beta = 10$.

(P8). Examples (P9) and (P10) do not have a tensor product structure. It is known that $u_{10} \in H^1(\mathbb{R}^2)\backslash H^2(\mathbb{R}^2)$. The function u_9 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.

Remark 4.6. In this section, we only consider **S-ADWAV** since, to our knowledge, heuristic adaptive schemes have not been investigated numerically with a tensor wavelet basis Ψ. Moreover, it is known (cf. eg. [13, Chapter 5]) that the condition numbers $\kappa(\mathbf{A}) = \frac{c_2}{c_1}$ and $\frac{C_\Psi}{c_\Psi}$ in (2.8) and (2.5) grow exponentially in the spatial dimension n as ϕ and ψ are not L_2-orthonormal. Thus, the requirement for a quantitative good performance of **ADWAV**, namely a small condition number $\kappa(\mathbf{A})$ being crucial for the number of degrees of freedom that are added per iteration by **GROW**, is not met. An alternative approach using L_2-orthonormal multiwavelet-bases (cf. [15, 16, 19]) which was investigated for a bounded setting in [13, 14] is on our agenda.

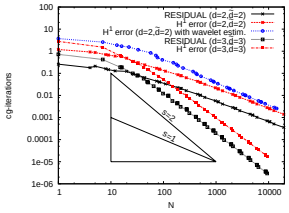
Convergence rates. We shall investigate whether it is possible to obtain the best possible approximation rate $s = d - 1$ with **S-ADWAV** using the parameters $c = 0.125$, $h = 0.0001$, $M = 3$, $\rho = 0.1$, $\mathrm{tol}_{\mathrm{iter}} = 0.4$, $d = 2, \widetilde{d} = 2$ and $d = \widetilde{d} = 3$. Here, the minimal level $\mathbf{j}_0$ has been estimated by the following proceeding. By considering the right-hand side f of each problem (given in an analytical form), we determine a finite index set ∇ such that $\|\langle f, \Psi \rangle - \langle f, \Psi \rangle|_\nabla \|_{\ell_2}\| \lesssim \varepsilon$, ε the given target tolerance. But instead of taking the level of the largest wavelet coefficient within $\langle f, \Psi \rangle|_\nabla$, we compute the vector

$$\widetilde{\mathbf{f}}_\nabla := \{\langle f, \varphi_{j_1,k_1} \cdot \varphi_{j_2,k_2} \rangle : \boldsymbol{\lambda} = ((j_1,k_1),(j_2,k_2)) \in \nabla\},$$

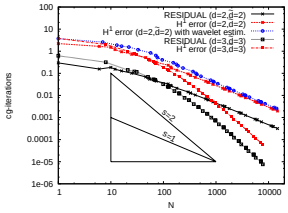
and consider the level of the largest scaling function coefficient in $\widetilde{\mathbf{f}}_\nabla$ as $\mathbf{j}_0$. Moreover, we take the corresponding scaling function index as initial candidate set.

In Figures 4.10 a)-d), we show the rate of convergence for (P7) – (P10). We observe that the asymptotically optimal rate $s = d - 1$ is realized asymptotically for the solutions of (P7), (P8) and (P9). In particular, for the reference solution u_{10} that does not permit a tensor structure, the optimal rate is not attained exactly. But we see that the modified estimate of $\mathbf{j}_0$ pays off as one can see in Figure 4.10 a) and b) where we also plot the approximation error (measured in $H^1(\mathbb{R}^2)$) for the wavelet type estimate from Section 4.1.5.

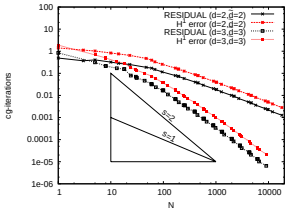
Finally, we show the number of inner iterations in **LINSOLVE** (here, a cg-method was used) for (P8). Using solutions from the previous iteration as initial guess allows to reduce this number such that for increasing number of degrees of freedom as well as for a decreasing $\mathrm{tol}_{\mathrm{iter}}$, the numbers remain small and grow very slow. We observe that the choice of the minimal level has a negligible impact.



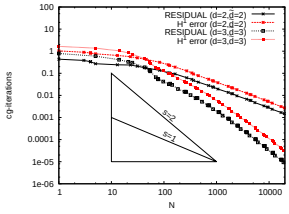
(a) (P7) with $d = \widetilde{d} = 2$, $\mathbf{j}_0 = (-2,-1)$ and $d = \widetilde{d} = 3$, $\mathbf{j}_0 = (-2,-2)$.



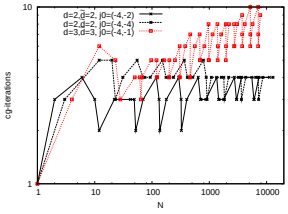
(b) (P8) with $d = \widetilde{d} = 2$, $\mathbf{j}_0 = (-4,-2)$ and $d = \widetilde{d} = 3$, $\mathbf{j}_0 = (-4,-1)$.



(c) (P9) with $d = \widetilde{d} = 2$, $\mathbf{j}_0 = (-1,-2)$ and $d = \widetilde{d} = 3$, $\mathbf{j}_0 = (0,0)$.



(d) (P10) with $d = \widetilde{d} = 2$, $\mathbf{j}_0 = (-2,-2)$ and $d = \widetilde{d} = 3$, $\mathbf{j}_0 = (-2,-2)$.



(e) Number of cg-iterations in **LINSOLVE** for (P8).

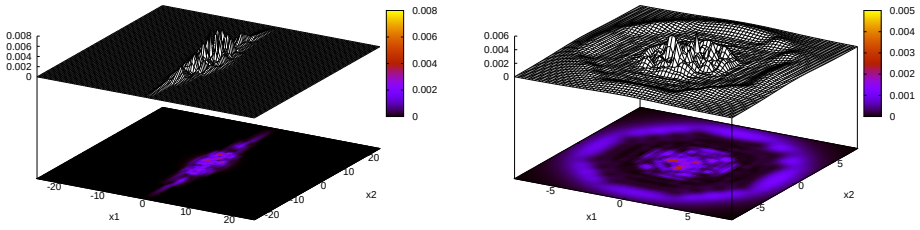
FIGURE 4.10. Output $\|\mathbf{r}^{(k,m)}\|_{\ell_2}$ of **RESIDUAL** and $H^1(\mathbb{R}^2)$-error for **S-ADWAV** over the degrees of freedom N.

Adaptive truncation of the computational domain. To get an impression of the solutions in two dimensions , we present exemplarily plots of the absolute error obtained with **S-ADWAV**. We observe that the anisotropic nature of u_8 is captured by **S-ADWAV**. The same holds true for the point singularity of u_{10} which is also detected and resolved even at an early stage of the algorithm.

5. Conclusions and Outlook

We have introduced an adaptive wavelet method for operator problems on $\mathbb{R}^n$. It was shown that the method converges and is asymptotically optimal. We have seen the benefit of using anisotropic wavelet bases in the multivariate case has been demonstrated. We have also introduced a simplified adaptive wavelet method without prove of optimality but with very 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



(a) (P8) with $\#\text{supp}\ \mathbf{u}_\Lambda = 226$.

(b) (P10) with $\#\text{supp}\ \mathbf{u}_\Lambda = 216$.

FIGURE 4.11. L_∞-error for (P8) and (P10) with $d = \widetilde{d} = 2$.

problems, etc.) has already been mentioned in the introduction. Moreover, a coupling with the space-time adaptive method proposed in [26] is on our agenda.

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, A. TABACCO, AND K. URBAN, *The wavelet element method (part I): construction and analysis*, Applied and Computational Harmonic Analysis, 6 (1999), pp. 1–52.

[4] A. COHEN, *Wavelet methods in numerical analysis*, Handbook of Numerical Analysis, 7 (2000), pp. 417–711.

[5] A. COHEN, W. DAHMEN, AND R. DEVORE, *Adaptive wavelet methods for elliptic operator equations: convergence rates*, Mathematics of Computation, 70 (2001), pp. 27–75.

[6] ———, *Adaptive wavelet methods II - beyond the elliptic case*, Foundations of Computational Mathematics, 2 (2002), pp. 203–245.

[7] ———, *Adaptive wavelet schemes for nonlinear variational problems*, SIAM Journal on Numerical Analysis, 41 (2003), pp. 1785–1823.

[8] A. COHEN, I. DAUBECHIES, AND J.-C. FEAUVEAU, *Biorthogonal bases of compactly supported wavelets*, Communications on Pure Applied Mathematics, 45 (1992), pp. 485–560.

[9] 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.

[10] W. DAHMEN, *Wavelet and multiscale methods for operator equations*, Acta Numerica, 6 (1997), pp. 55–228.

[11] 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.

[12] R. DEVORE, *Nonlinear approximation*, Acta Numerica, 7 (1998), pp. 51–150.

[13] T. DIJKEMA, *Adaptive tensor product wavelet methods for solving PDEs*, PhD thesis, Universität Utrecht, 2009.

[14] T. DIJKEMA, C. SCHWAB, AND R. STEVENSON, *An adaptive wavelet method for solving high-dimensional elliptic PDEs*, Constructive Approximation, 30 (2009), pp. 423–455.

[15] G. C. DONOVAN, J. S. GERONIMO, AND D. P. HARDIN, *Intertwining multiresolution analyses and the construction of piecewise-polynomial wavelets*, SIAM Journal on Mathematical Analysis, 27 (1996), pp. 1791–1815.

[16] ———, *Orthogonal polynomials and the construction of piecewise polynomial smooth wavelets*, SIAM Journal on Mathematical Analysis, 30 (1999), pp. 1029–1056.
[17] 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.
[18] T. Gantumur and R. Stevenson, *Computation of differential operators in wavelet coordinates*, Mathematics of Computation, 75 (2006), pp. 697–709.
[19] T. N. T. Goodman, *A class of orthogonal refinable functions and wavelets*, Constructive Approximation, 19 (2003), pp. 525–540.
[20] 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.
[21] M. Lehn, *FLENS - A Flexible Library for Efficient Nnumerical Solutions*, http://flens.sourceforge.net, (2008).
[22] R. Lorentz and P. Oswald, *Criteria for hierarchical basis in Sobolev spaces*, Applied and Computational Harmonic Analysis, 8 (2000), pp. 32–85.
[23] S. Mallat, *Multiresolution approximations and wavelet orthonormal bases of $L^2(R)$*, Transactions of the American Mathematical Society, 315 (1989), pp. 69–87.
[24] P.-A. Nitsche, *Best N-term approximation spaces for tensor product wavelet bases*, Constructive Approximation, 24 (2006), pp. 49–70.
[25] C. Schwab and R. Stevenson, *Adaptive wavelet algorithms for elliptic PDE's on product domains*, Mathematics of Computation, 77 (2008), pp. 71–92.
[26] ———, *Space-time adaptive wavelet methods for parabolic evolution problems*, Mathematics of Computation, 78 (2009), pp. 1293–1318.
[27] 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.
[28] R. Stevenson, *On the compressibility of operators in wavelet coordinates*, SIAM Journal on Mathematical Analysis, 35(5) (2004), pp. 1110–1132.
[29] A. Stippler, *LAWA - Library for Adaptive Wavelet Applications*, http://lawa.sourceforge.net, (2009).
[30] 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