Journal of Computational Mathematics
Vol.36, No.1, 2018, 90–109.

http://www.global-sci.org/jcm
doi:10.4208/jcm.1610-m2016-0496

ON ADAPTIVE WAVELET BOUNDARY ELEMENT METHODS *

H. Harbrecht and M. Utzinger
Universität Basel, Departement Mathematik und Informatik, Spiegelgasse 1, 4051 Basel, Schweiz
Email: helmut.harbrecht@unibas.ch, manuela.utzinger@unibas.ch

Abstract

The present article is concerned with the numerical solution of boundary integral equations by an adaptive wavelet boundary element method. This method approximates the solution with a computational complexity that is proportional to the solution's best N-term approximation. The focus of this article is on algorithmic issues which includes the crucial building blocks and details about the efficient implementation. By numerical examples for the Laplace equation and the Helmholtz equation, solved for different geometries and right-hand sides, we validate the feasibility and efficiency of the adaptive wavelet boundary element method.

Mathematics subject classification: 41A25, 65N38, 65T60.
Key words: Boundary element method, wavelets, adaptivity.

1. Introduction

In science and engineering, one often comes across partial differential equations, some of which can be formulated as boundary integral equations on the boundary of the domain of interest. Solving the original problem would result in having to discretize the problem in a domain (e.g. with finite element methods), which would lead to a sparse but extremely large system of linear equations, especially in the three-dimensional situation. Rewriting the problem by a boundary integral equation not only reduces the dimensionality by one, but does give the possibility to solve also exterior boundary value problems. Particularly for such problems, this approach brings many advantages, since it is not necessary to find a way (e.g. by introducing artificial boundaries) to handle the infinite expansion of the domain. Of course, this advantage does not come entirely without cost. Since the involved operators are not local, the resulting matrices are dense and the complexity to solve the linear system by the boundary element method is at least $\mathcal{O}(N^2)$, with N denoting the degrees of freedom.

Modern approaches like the *fast multipole method* [1,2], the *panel clustering* [3], the *adaptive cross approximation* [4, 5], or *hierarchical matrices* [6, 7] are known to reduce the complexity to log-linear or even linear cost. Another approach is wavelet matrix compression [8]. The wavelets' vanishing moments lead, in combination with the fact that the kernel of the integral operator becomes smoother when getting farther away from the diagonal, to a quasi-sparse system matrix. As shown in [9], only $\mathcal{O}(N)$ matrix entries are relevant for maintaining the convergence rate of the underlying Galerkin scheme.

A further issue to be addressed for the efficient solution of boundary integral equations is the one of adaptivity. For non-smooth geometries or right-hand sides, it is necessary being able to resolve specific parts of the geometry, while other parts could stay coarse. In contrast to

* Received January 5, 2016 / Revised version received June 28, 2016 / Accepted October 24, 2016 / Published online October 11, 2017 /

uniform refinement, an adaptive refinement reduces the degrees of freedom drastically without compromising the accuracy. This means that not only a lot of computation power can be saved, but also a lot of memory, making the computation of certain problems possible in the first place. Efficient and reliable a posteriori error estimators have first been introduced in [10] and convergence of adaptive refinements for traditional boundary element methods has been established in [11, 12]. But we are not aware of an implementation which combines these error estimators with fast boundary element methods.

We thus follow here an different approach which has been proposed in [13, 14] for local operators and in [9,15] for nonlocal operators. Namely, we cast the boundary integral equation into an infinite system of linear equations and solve it then approximately by an iterative method. As the application of the infinite system matrix has to be approximated during the solution process, we have to choose a certain portion out of this infinite matrix. Hence, refinement rather means that more wavelets are added. In fact, we aim at choosing the N wavelets which will contribute most to the approximate solution. This concept is referred to as the *best N-term approximation*, see e.g. [16].

Where the efficient computation of the matrix entries is the most demanding and time consuming part of the whole implementation, it is not possible to achieve efficiency without using the appropriate adaptive structures. It is absolutely necessary to work with element and wavelet trees as already proposed in [17, 18]. We will introduce related building blocks RHS, COARSE, APPLY, and SOLVE, which are already known in theory from e.g. [13, 14, 18, 19]. The implementation of these routines is discussed in the present context of boundary element methods. The numerical method is able to compute the solution of the boundary integral equation in asymptotically optimal complexity. This means that any target accuracy can be achieved at a computational expense that stays proportional to the number of degrees of freedom (within the setting determined by an underlying wavelet basis) that would ideally be necessary for realizing that target accuracy if full knowledge about the unknown solution were given.

In this article, besides presenting results for the single-layer operator of the Laplacian, which is symmetric and positive definite, we also present results for the Brackhage-Werner formulation of the (low-frequency) Helmholtz equation. We thus arrive at a linear combination of the acoustic single-layer operator and the acoustic double-layer operator. Here, the theory of [9,15] does not hold anymore. Nevertheless, the arguments of [20] are applicable for proving optimality of the underlying adaptive scheme since the operator under consideration is a compact perturbation of a symmetric and positive definite operator.

The outline of this article is as follows. At first, in Section 2, we introduce the boundary integral equation and the surface representation under consideration. Then, in Section 3, we present the piecewise constant wavelet basis which we will employ to cast the boundary integral equation into an equivalent, bi-infinite system of linear equations. Section 4 is dedicated to the realization of an adaptive algorithm of optimal complexity. Numerical results are given in Section 5. Finally, in Section 6, we state concluding remarks.

2. Boundary Integral Equations and Surface Representation

Let $\Omega \subset \mathbb{R}^3$ be a bounded and simply connected domain. Its boundary $\Gamma := \partial\Omega$ is assumed to be composed by a union of smooth, four-sided patches Γ_i:

$$\Gamma = \bigcup_{i=1}^{M} \Gamma_i, \quad \Gamma_i = \boldsymbol{\gamma}_i(\square) \quad i = 1, \ldots, M.$$

The composition is regular, i.e., the intersection $\Gamma_i \cap \Gamma_j$, $i \neq j$, of two different patches is either empty, a common edge, or a common vertex. The mappings $\boldsymbol{\gamma}_i : \square \to \Gamma_i$ are smooth diffeomorphisms from the unit square $\square = [0,1]^2$ to the patch Γ_i.

A mesh of level j on Γ is induced by dyadic subdivisions of depth j of the unit square into 4^j squares

$$\square_{j,\boldsymbol{k}} \coloneqq [2^{-j}k_1, 2^{-j}(k_1+1)] \times [2^{-j}k_2, 2^{-j}(k_2+1)] \subseteq \square,$$

where $\boldsymbol{k} = (k_1, k_2)$ with $0 \leq k_1, k_2 < 2^j$. This generates $4^j M$ *elements* (or elementary domains) $\Gamma_{i,j,\boldsymbol{k}} \coloneqq \gamma_i(\square_{j,\boldsymbol{k}}) \subseteq \Gamma_i$, $i = 1, \ldots, M$, see Figure 2.1 for an illustration.

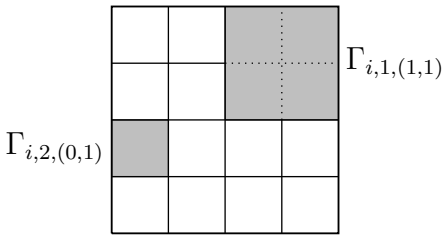

Fig. 2.1. Element $\Gamma_{i,1,(1,1)}$ and element $\Gamma_{i,2,(0,1)}$.

In order to ensure that the collection of elements $\{\Gamma_{i,j,\boldsymbol{k}}\}$ on the level j forms a regular mesh on Γ, the parametric representation is subjected to the following *matching condition*: For each $\boldsymbol{x}$ in the intersection of two patches $\Gamma_i \cap \Gamma_{i'}$, there exists an affine map $\Xi : \square \to \square$ such that $\boldsymbol{x} = \boldsymbol{\gamma}_i(\boldsymbol{s}) = (\boldsymbol{\gamma}_{i'} \circ \Xi)(\boldsymbol{s})$. In other words, the diffeomorphisms $\boldsymbol{\gamma}_i$ and $\boldsymbol{\gamma}_{i'}$ coincide on the common edge except for orientation. This setting is frequently used and well understood in Computer Aided Geometric Design and is the topic of recent studies in isogeometric analysis, cf. [21].

In what follows, we shall be concerned with the numerical solution of a boundary integral equation

$$(\mathcal{A}u)(\boldsymbol{x}) = \int_\Gamma k(\boldsymbol{x},\boldsymbol{y})u(\boldsymbol{y})\,\mathrm{d}\sigma_{\boldsymbol{y}} = f(\boldsymbol{x}) \quad \text{for } \boldsymbol{x} \in \Gamma. \tag{2.1}$$

The boundary integral operator $\mathcal{A} : H^q(\Gamma) \to H^{-q}(\Gamma)$ is assumed to be a bijective operator of order $2q$, where $H^q(\Gamma)$ denotes the Sobolev space of functions of smoothness q and $H^{-q}(\Gamma)$ denotes its dual. Unique solvability follows thus for arbitrary right-hand side $f \in H^{-q}(\Gamma)$. The kernel $k(\boldsymbol{x},\boldsymbol{y})$ under consideration has for our purpose to be analytically standard.

Definition 2.1. *Consider a kernel $k(\boldsymbol{x},\boldsymbol{y})$ of order $2q$, the multi indices $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)$, $\boldsymbol{\beta} = (\beta_1, \beta_2)$ and $|\boldsymbol{\alpha}| = \alpha_1 + \alpha_2$. Then, the transported kernel functions*

$$k_{i,i'}(\boldsymbol{s},\boldsymbol{t}) \coloneqq k\big(\boldsymbol{\gamma}_i(\boldsymbol{s}), \boldsymbol{\gamma}_{i'}(\boldsymbol{t})\big)\mu_i(\boldsymbol{s})\mu_{i'}(\boldsymbol{t}), \quad 1 \leq i, i' \leq M$$

are called analytically standard of order $2q$ if the partial derivatives are bounded by

$$|\partial_{\boldsymbol{s}}^{\boldsymbol{\alpha}}\partial_{\boldsymbol{t}}^{\boldsymbol{\beta}} k_{i,i'}(\boldsymbol{s},\boldsymbol{t})| \lesssim \frac{(|\boldsymbol{\alpha}| + |\boldsymbol{\beta}|)!}{(r\|\boldsymbol{\gamma}_i(\boldsymbol{s}) - \boldsymbol{\gamma}_{i'}(\boldsymbol{t})\|)^{(2+2q+|\boldsymbol{\alpha}|+|\boldsymbol{\beta}|)}} \tag{2.2}$$

for some $r > 0$ provided that $2 + 2q + |\boldsymbol{\alpha}| + |\boldsymbol{\beta}| > 0$.

Remark 2.2. For geometries that are piecewise smooth, the kernels of the boundary integral operators of order $2q$ are known to be in general analytically standard of order $2q$.

The $L^2(\Gamma)$-inner product with respect to Γ is given by

$$\langle u, v\rangle = \int_\Gamma u(\boldsymbol{x})\, v(\boldsymbol{x})\, \mathrm{d}\sigma_{\boldsymbol{x}} = \sum_{i=1}^{M} \int_\square u\big(\boldsymbol{\gamma}_i(\boldsymbol{s})\big) v\big(\boldsymbol{\gamma}_i(\boldsymbol{s})\big) \mu_i(\boldsymbol{s})\, \mathrm{d}\boldsymbol{s}. \tag{2.3}$$

Here, the expression μ_i stands for the surface measure which is defined as

$$\mu_i(\boldsymbol{s}) \coloneqq \left\| \frac{\partial \boldsymbol{\gamma}_i(\boldsymbol{s})}{\partial s_1} \times \frac{\partial \boldsymbol{\gamma}_i(\boldsymbol{s})}{\partial s_2} \right\| > 0. \tag{2.4}$$

By extending the inner product (2.3) continuously to dualities as $u \in H^{-q}(\Gamma)$ and $v \in H^q(\Gamma)$, we arrive at the variational formulation of (2.1):

$$\text{seek } u \in H^q(\Gamma) \text{ such that } \langle \mathcal{A}u, v\rangle = \langle f, v\rangle \text{ for all } v \in H^q(\Gamma).$$

3. Wavelet Bases on Surfaces

3.1. Wavelets on the interval

We shall consider piecewise constant wavelets with three vanishing moments. To that end, let

$$V_j^{[0,1]} = \operatorname{span}\big\{\varphi_{j,k}^{[0,1]} : k \in \Delta_j^{[0,1]} \coloneqq \{0, 1, \ldots, 2^j - 1\}\big\}$$

be the space of piecewise constant ansatz functions relative to a subdivision of the interval $[0, 1]$ into 2^j equidistant intervals. We assume that the basis functions are L^2-normalized, leading to

$$\varphi_{j,k}^{[0,1]}(s) = \begin{cases} 2^{j/2}, & \text{if } s \in [2^{-j}k, 2^{-j}(k+1)], \\ 0, & \text{otherwise.} \end{cases}$$

The sequence of ansatz spaces is nested, i.e., $V_{j-1}^{[0,1]} \subset V_j^{[0,1]}$ for all $j \in \mathbb{N}$. To keep track of the increment of information between subsequent spaces, one decomposes $W_j^{[0,1]} = V_j^{[0,1]} \ominus V_{j-1}^{[0,1]}$ with some complementary space W_j, which is generated by so-called wavelets

$$W_j^{[0,1]} = \operatorname{span}\Big\{\psi_{j,k}^{[0,1]} : k \in \nabla_j^{[0,1]} \coloneqq \{0, 1, \ldots, 2^{j-1} - 1\}\Big\}.$$

According to [22], compactly supported, biorthogonal, piecewise constant wavelets with three vanishing moments on the interval are given by

$$\psi_{j,0}^{[0,1]} = \frac{1}{\sqrt{2}}\left(-\frac{5}{8}\varphi_{j,0}^{[0,1]} + \frac{11}{8}\varphi_{j,1}^{[0,1]} - \frac{1}{2}\varphi_{j,2}^{[0,1]} - \frac{1}{2}\varphi_{j,3}^{[0,1]} + \frac{1}{8}\varphi_{j,4}^{[0,1]} + \frac{1}{8}\varphi_{j,5}^{[0,1]}\right),$$

$$\psi_{j,2^{j-1}-1}^{[0,1]} = \frac{1}{\sqrt{2}}\left(-\frac{1}{8}\varphi_{j,2^j-6}^{[0,1]} - \frac{1}{8}\varphi_{j,2^j-5}^{[0,1]} + \frac{1}{2}\varphi_{j,2^j-3}^{[0,1]} + \frac{1}{2}\varphi_{j,2^j-4}^{[0,1]} - \frac{11}{8}\varphi_{j,2^j-2}^{[0,1]} + \frac{5}{8}\varphi_{j,2^j-1}^{[0,1]}\right),$$

$$\psi_{j,k}^{[0,1]} = \frac{1}{\sqrt{2}}\left(-\frac{1}{8}\varphi_{j,2k-2}^{[0,1]} - \frac{1}{8}\varphi_{j,2k-1}^{[0,1]} + \varphi_{j,2k}^{[0,1]} - \varphi_{j,2k+1}^{[0,1]} + \frac{1}{8}\varphi_{j,2k+2}^{[0,1]} + \frac{1}{8}\varphi_{j,2k+3}^{[0,1]}\right),$$

where $k \in \{1, 2, \ldots, 2^{j-1} - 2\}$. Thus, we have only two wavelets (one at each boundary) which do not coincide with the interior wavelets, see Figure 3.1 for an illustration. All wavelets satisfy

$$\int_0^1 s^i \psi_{j,k}^{[0,1]}(s)\, \mathrm{d}s = 0 \text{ for all } i = 0, 1, 2. \tag{3.1}$$

On the levels $j = 1$ and $j = 2$, we will just use Haar wavelets.

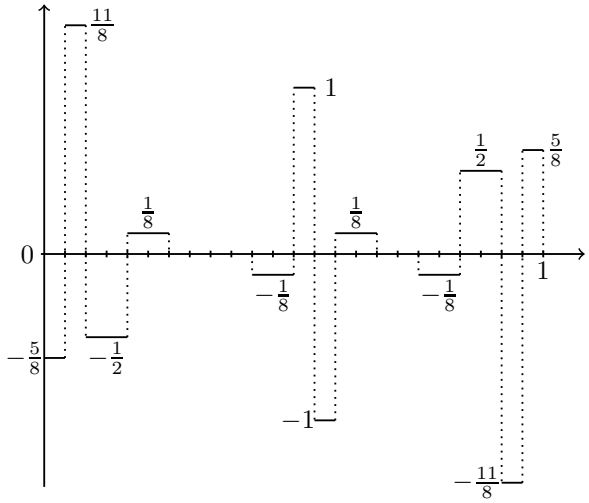

Fig. 3.1. Wavelet with three vanishing moments with the left and right boundary modification.

3.2. Wavelet bases on the unit square

To obtain wavelets on the unit square, we start with the definition of the sets of ansatz functions which are defined as tensor products of the one-dimensional ansatz functions

$$\varphi_{j,\boldsymbol{k}}^{\square}(\boldsymbol{s}) = \varphi_{j,k_1}^{[0,1]}(s_1) \cdot \varphi_{j,k_2}^{[0,1]}(s_2),$$

where $\boldsymbol{s} = (s_1, s_2)$ and $\boldsymbol{k} = (k_1, k_2) \in \Delta_j^{\square} := \Delta_j^{[0,1]} \times \Delta_j^{[0,1]}$. This leads to the ansatz spaces

$$V_j^{\square} = \operatorname{span}\big\{\varphi_{j,\boldsymbol{k}}^{\square} : \boldsymbol{k} \in \Delta_j^{\square}\big\},$$

which are again nested $V_{j-1}^{\square} \subset V_j^{\square}$. To construct the wavelets in the complement spaces $W_j^{\square} = V_j^{\square} \ominus V_{j-1}^{\square}$, we use the splitting

$$\begin{aligned} V_j^{\square} &= V_j^{[0,1]} \otimes V_j^{[0,1]} \\ &= \big(V_{j-1}^{[0,1]} \oplus W_j^{[0,1]}\big) \otimes \big(V_{j-1}^{[0,1]} \oplus W_j^{[0,1]}\big) \\ &= V_{j-1}^{\square} \oplus \big(W_j^{[0,1]} \otimes V_{j-1}^{[0,1]}\big) \oplus \big(V_j^{[0,1]} \otimes W_j^{[0,1]}\big). \end{aligned}$$

We hence arrive at two types of wavelets, namely

$$\begin{aligned} \psi_{j,\boldsymbol{k},1}^{\square}(\boldsymbol{s}) &= \psi_{j,k_1}^{[0,1]}(s_1) \cdot \varphi_{j-1,k_2}^{[0,1]}(s_2), \quad \boldsymbol{k}_1 = (k_1, k_2) \in \nabla_{j,1}^{\square} := \nabla_j^{[0,1]} \times \Delta_{j-1}^{[0,1]}, \\ \psi_{j,\boldsymbol{k},2}^{\square}(\boldsymbol{s}) &= \varphi_{j,k_1}^{[0,1]}(s_1) \cdot \psi_{j,k_2}^{[0,1]}(s_2), \qquad \boldsymbol{k}_2 = (k_1, k_2) \in \nabla_{j,2}^{\square} := \Delta_j^{[0,1]} \times \nabla_j^{[0,1]}, \end{aligned}$$

see Figure 3.2 for an illustration of these wavelets.

3.3. Wavelet bases on surfaces

Having the bases for the unit square at hand, we can then construct the bases on the surface Γ by using the mappings $\boldsymbol{\gamma}_i$ to lift functions from the unit square to the according patch Γ_i, i.e.,

$$\varphi_{j,\boldsymbol{k}}^{\Gamma_i}(\boldsymbol{x}) = \varphi_{j,\boldsymbol{k}}^{\square}\big(\boldsymbol{\gamma}_i^{-1}(\boldsymbol{x})\big), \quad \boldsymbol{k} \in \triangle_j^{\square},$$

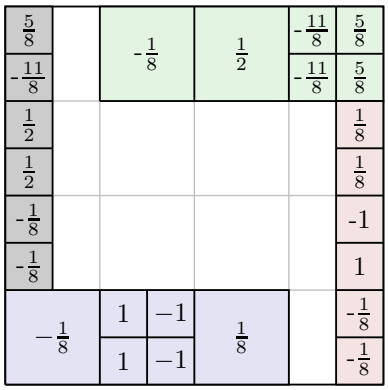

Fig. 3.2. Different wavelets on the unit square of type one (boundary wavelet: green / interior wavelet: violett) and of type 2 (boundary wavelet: grey / interior wavelet: red).

and likewise

$$\psi^{\Gamma_i}_{j,\boldsymbol{k},1}(\boldsymbol{x}) = \psi^{\square}_{j,\boldsymbol{k},1}\big(\boldsymbol{\gamma}_i^{-1}(\boldsymbol{x})\big), \quad \boldsymbol{k} \in \nabla^{\square}_{j,1},$$
$$\psi^{\Gamma_i}_{j,\boldsymbol{k},2}(\boldsymbol{x}) = \psi^{\square}_{j,\boldsymbol{k},2}\big(\boldsymbol{\gamma}_i^{-1}(\boldsymbol{x})\big), \quad \boldsymbol{k} \in \nabla^{\square}_{j,2},$$

for all $\boldsymbol{x} \in \Gamma_i$. We get the complete set of ansatz functions on the surface by collecting these functions for all patches $i = \{0, 1, \ldots, M\}$, where we will from now on use the index $\boldsymbol{\lambda}$ to encode the level, the location, and the type of the particular basis function. Specifically, $|\boldsymbol{\lambda}|$ decodes the level j.

Hence, we arrive at ansatz spaces $V_j^\Gamma = \{\varphi_{\boldsymbol{\lambda}} : \boldsymbol{\lambda} \in \triangle_j^\Gamma\}$ which are nested $V_{j-1}^\Gamma \subset V_j^\Gamma$ for all $j \in \mathbb{N}$. The differences $W_j^\Gamma = V_j^\Gamma \ominus V_{j-1}^\Gamma$ are generated by the wavelets, i.e., $W_j^\Gamma = \{\psi_{\boldsymbol{\lambda}} : \boldsymbol{\lambda} \in \nabla_j^\Gamma\}$. Recursively, we obtain the multiscale decomposition

$$L^2(\Gamma) = \bigoplus_{j=0}^{\infty} W_j^\Gamma, \text{ where } W_0^\Gamma := V_0^\Gamma.$$

3.4. Bi-infinite system of linear equations

Denoting the index set of all wavelets by $\mathcal{I} := \bigcup_{j=0}^{\infty} \nabla_j^\Gamma$ and scaling the wavelets such that it holds $\|\psi_{\boldsymbol{\lambda}}\|_{H^s(\Gamma)} \sim 1$ for all $\boldsymbol{\lambda} \in \mathcal{I}$, the wavelets $\boldsymbol{\Psi} := \{\psi_{\boldsymbol{\lambda}} : \lambda \in \mathcal{I}\}$ constitute a Riesz basis for the Sobolev spaces $H^s(\Gamma)$ for all $s \in (-1/2, 1/2)$, meaning that

$$\|\boldsymbol{f}\| \sim \left\| \sum_{\boldsymbol{\lambda} \in \mathcal{I}} f_{\boldsymbol{\lambda}} \psi_{\boldsymbol{\lambda}} \right\|_{H^s(\Gamma)}$$

for all $\boldsymbol{f} = [f_{\boldsymbol{\lambda}}]_{\boldsymbol{\lambda} \in \mathcal{I}} \in \ell^2(\mathcal{I})$, see e.g. [23]. Setting $s := q$, we can thus cast the boundary integral equation (2.1) into a well-posed, bi-infinite system of linear equations: seek $u = \sum_{\boldsymbol{\lambda} \in \mathcal{I}} u_{\boldsymbol{\lambda}} \psi_{\boldsymbol{\lambda}} \in H^q(\Gamma)$ such that

$$\boldsymbol{A}\boldsymbol{u} = \boldsymbol{g}, \quad \boldsymbol{A} = \big[\langle \mathcal{A}\psi_{\boldsymbol{\lambda}'}, \psi_{\boldsymbol{\lambda}} \rangle\big]_{\boldsymbol{\lambda}, \boldsymbol{\lambda}' \in \mathcal{I}}, \quad \boldsymbol{u} = \big[u_{\boldsymbol{\lambda}}\big]_{\boldsymbol{\lambda} \in \mathcal{I}}, \quad \boldsymbol{g} = \big[\langle f, \psi_{\boldsymbol{\lambda}} \rangle\big]_{\boldsymbol{\lambda} \in \mathcal{I}}. \tag{3.2}$$

Of course, this bi-infinite system of linear equations cannot be solved numerically. So, we should restrict the index set $\mathcal{I}$ to some finite subset $\mathcal{T} \subset \mathcal{I}$ of cardinality N.

The following question arises: How small can the approximation error become when choosing N terms of the expansion with respect to the wavelet basis $\boldsymbol{\Psi}$? It is obvious that, having full

information on the solution $\boldsymbol{u}$, the best approximation by N terms is obtained by choosing the N largest coefficients (and setting all the others to zero). This is called the *best N-term approximation*. In particular, due to Céa's lemma, by restricting the system (3.2) of linear equation to this index set

$$\boldsymbol{A}_{\mathcal{T}}\boldsymbol{u}_{\mathcal{T}}=\boldsymbol{g}_{\mathcal{T}},\quad \boldsymbol{A}_{\mathcal{T}}=\big[\langle\mathcal{A}\psi_{\boldsymbol{\lambda}'},\psi_{\boldsymbol{\lambda}}\rangle\big]_{\boldsymbol{\lambda},\boldsymbol{\lambda}'\in\mathcal{T}},\quad \boldsymbol{u}_{\mathcal{T}}=\big[u_{\boldsymbol{\lambda}}\big]_{\boldsymbol{\lambda}\in\mathcal{T}},\quad \boldsymbol{g}_{\mathcal{T}}=\big[\langle f,\psi_{\boldsymbol{\lambda}}\rangle\big]_{\boldsymbol{\lambda}\in\mathcal{T}},\tag{3.3}$$

it holds the estimate

$$\big\|\boldsymbol{u}-\boldsymbol{u}_{\mathcal{T}}\big\|\lesssim\big\|\boldsymbol{u}-\boldsymbol{u}_{\mathcal{T}'}\big\|$$

for any other index set $\mathcal{T}'$ of cardinality N. The goal of computation is thus the following. Find an index set $\mathcal{T}$ of cardinality N such that the error of the numerical solution of (3.3) stays proportional to the error of the best N-term approximation when full information on $\boldsymbol{u}$ is given. We should quantify this in the following.

3.5. Approximation spaces

Due to the multiscale hierarchy of the basis construction, a natural quad-tree structure underlies the wavelet basis. This is easily derived from observing that the ansatz function $\varphi_{j,k}^{[0,1]}$ has the two sons $\varphi_{j+1,2k}^{[0,1]}$ and $\varphi_{j+1,2k+1}^{[0,1]}$. Likewise, the wavelet $\psi_{j,k}^{[0,1]}$ has the two sons $\psi_{j+1,2k}^{[0,1]}$ and $\psi_{j+1,2k+1}^{[0,1]}$. Consequently, each wavelet $\psi_{\boldsymbol{\lambda}}$ has exactly four sons. This tree structure is exploited for the efficient implementation of the wavelet boundary element method, see e.g. [9]. Therefore, we should restrict ourselves to index sets such that the associated wavelet bases are arranged in a tree. To that end, we define the approximation classes

$$\mathfrak{A}_q^s:=\left\{\boldsymbol{v}\in\ell^2(\mathcal{I}):|\boldsymbol{v}|_{\mathfrak{A}_q^s}:=\sup_{N\in\mathbb{N}_0}\min_{\substack{\mathcal{T}\subset\mathcal{I}\text{ is a tree}\\ \#\mathcal{T}=N}}N^s\big\|\boldsymbol{v}-\boldsymbol{v}|_{\mathcal{T}}\big\|<\infty\right\}$$

of vectors $\boldsymbol{v}\in\ell^2(\mathcal{I})$ which can be approximated at rate N^{-s}. The optimal approximation is called the *best N-term tree approximation*, which is essentially as good as the pure best N-term approximation, see e.g. [24].

On the continuous level, functions from the approximation classes $\mathfrak{A}_q^s$ can be characterized by *Besov spaces*. If the function $v=\sum_{\boldsymbol{\lambda}\in\mathcal{I}}v_{\boldsymbol{\lambda}}\psi_{\boldsymbol{\lambda}}$ is a member of $B_{\tau,\tau}^{q+2s}(\Gamma)$, then it follows $\boldsymbol{v}\in\mathfrak{A}_q^s$ for its discrete counterpart provided that $1/\tau<s+\frac{1}{2}$, cf. [25,26]. Note that the highest possible convergence rate, attainable for approximations in the energy norm $\|\cdot\|_{H^q(\Gamma)}$ from the multiresolution spaces generated by piecewise constant wavelets, is $N^{-\bar{s}}$ with

$$\bar{s}:=\frac{1-q}{2},$$

see e.g. [16].

With these preparations at hand, we can precisely formulate our goal of computation. If $\boldsymbol{u}\in\mathfrak{A}_q^s$ for some $s\leq\bar{s}$, we aim at computing an approximation $\boldsymbol{u}_{\mathcal{T}}$ with support size $\#\mathcal{T}\lesssim\epsilon^{-1/s}|\boldsymbol{u}|_{\mathfrak{A}_q^s}$ such that $\|\boldsymbol{u}-\boldsymbol{u}_{\mathcal{T}}\|\leq\epsilon$ for any target accuracy $\epsilon>0$, where the computational over-all complexity scales linearly, i.e., like $\mathcal{O}(\#\mathcal{T})$.

4. Adaptive Wavelet Boundary Element Methods

4.1. About the adaptive algorithm

We shall first summarize the ingredients which are needed to arrive at an adaptive algorithm which achieves optimal complexity. The building blocks COARSE, APPLY, RHS, and SOLVE have been introduced in [13, 14], where our particular implementation is based on [9], see also [15] for related results. The adaptive algorithm itself has been proposed in [19, 20] and coincides with classical methods which consist of the following steps:

$$\boxed{\text{SOLVE}} \longrightarrow \boxed{\text{ESTIMATE}} \longrightarrow \boxed{\text{MARK}} \longrightarrow \boxed{\text{REFINE}}$$

For a given index set $\mathcal{T} \subset \mathcal{I}$, we *solve* the Galerkin system (3.3) via $\boldsymbol{u}_{\mathcal{T}} = \text{SOLVE}[\mathcal{T}]$. Then, we *estimate* the residuum $\boldsymbol{r} = \boldsymbol{f} - \mathbf{A}\boldsymbol{u}_{\mathcal{T}}$ with sufficient accuracy $\eta > 0$ by computing

$$\boldsymbol{r}_{\mathcal{T}'} = \text{RHS}[\eta/2] - \text{APPLY}[\eta/2, \boldsymbol{u}_{\mathcal{T}}]$$

relative to a finite index set $\mathcal{T} \subset \mathcal{T}' \subset \mathcal{I}$ such that

$$\|\boldsymbol{r} - \boldsymbol{r}_{\mathcal{T}'}\| \leq \eta.$$

Herein, $\text{RHS}[\eta/2]$ produces a finitely supported approximation of the right-hand side with accuracy $\eta/2$ and $\text{APPLY}[\eta/2, \boldsymbol{u}_{\mathcal{T}}]$ approximates the matrix-vector product $\boldsymbol{A}\boldsymbol{u}_{\mathcal{T}}$ with accuracy $\eta/2$. In order to have $\|\boldsymbol{r}_{\mathcal{T}'}\|$ proportional to $\|\boldsymbol{r}\|$, i.e.,

$$(1-\omega)\|\boldsymbol{r}_{\mathcal{T}'}\| \leq \|\boldsymbol{r}\| \leq (1+\omega)\|\boldsymbol{r}_{\mathcal{T}'}\| \tag{4.1}$$

for fixed $0 < \omega < 1$, we apply the following iteration for some initial precision η_{init}:

$$\left.\begin{array}{l} \text{set } \eta = \eta_{\text{init}} \\ \text{do} \\ \quad \text{set } \eta = \eta/2 \\ \quad \text{calculate } \boldsymbol{r}_{\mathcal{T}'} = \text{RHS}[\eta/2] - \text{APPLY}[\eta/2, \boldsymbol{u}_{\mathcal{T}}] \\ \text{until } \eta \leq \omega\|\boldsymbol{r}_{\mathcal{T}'}\| \end{array}\right\} \tag{4.2}$$

The until-clause $\eta \leq \omega\|\boldsymbol{r}_{\mathcal{T}'}\|$ causes that this iteration terminates when (4.1) holds.

The supporting index set $\mathcal{T}'$ of the approximate residuum $\boldsymbol{r}_{\mathcal{T}'}$ enlarges the original index set $\mathcal{T}$ enough to ensure that the Galerkin solution with respect to $\mathcal{T}'$ would reduce the error by a constant factor. Nevertheless, to control the complexity, we have to coarsen the index set $\mathcal{T}'$ such that

$$\|\boldsymbol{r}_{\mathcal{T}''}\| \leq \theta\|\boldsymbol{r}_{\mathcal{T}'}\|$$

for fixed $0 < \theta < 1$ sufficiently small. This is done by calling

$$\boldsymbol{r}_{\mathcal{T}''} = \text{COARSE}[\theta, \boldsymbol{r}_{\mathcal{T}'}].$$

It combines the steps *mark* and *refine* since the new index set $\mathcal{T}''$ is a refined version of the original index set $\mathcal{T}$. We emphasize that $\mathcal{T}''$ is still large enough to guarantee the convergence of the algorithm when starting the procedure again with $\mathcal{T} := \mathcal{T}''$.

In the subsequent sections, we provide the routines which have been mentioned before. They are all of optimal computational complexity, leading to the desired linear over-all complexity. For more details on the analysis and the choice of the constants, we refer the reader to [19, 20].

4.2. The routine Coarse

A coarsening algorithm, whose output is a near-best tree, has been established in [27]. The idea of the algorithm is not to delete any wavelet coefficients from an input tree, but only to cut whole branches, i.e., a node and all its successors.

To measure the error, we assign the ℓ^2-norm of the branch at an index $\boldsymbol{\lambda} \in \mathcal{T} \subset \mathcal{I}$. This is recursively performed bottom-up by setting

$$e(\boldsymbol{\lambda}) \coloneqq |v_{\boldsymbol{\lambda}}|^2 + \sum_{\substack{\boldsymbol{\lambda}' \text{ is a} \\ \text{successor of } \boldsymbol{\lambda}}} |v_{\boldsymbol{\lambda}'}|^2 = |v_{\boldsymbol{\lambda}}|^2 + \sum_{\substack{\boldsymbol{\lambda}' \text{ is a} \\ \text{son of } \boldsymbol{\lambda}}} e(\boldsymbol{\lambda}').$$

Thus, the error induced by removing the branch at the node $\boldsymbol{\lambda}$ is simply given by $\sqrt{e(\boldsymbol{\lambda})}$.

Since the coarsened tree is supposed to be minimal in size, it is imperative to find deep branches with a small error $e(\boldsymbol{\lambda})$. Hence, to incorporate the depth of the branches, we consider the error functional $\tilde{e}(\boldsymbol{\lambda})$ by setting $\tilde{e}(\boldsymbol{\lambda}) \coloneqq e(\boldsymbol{\lambda})$ for the root node $\boldsymbol{\lambda}$ of $\mathcal{T}$. For all the descendant nodes, we compute

$$q(\boldsymbol{\lambda}) \coloneqq \frac{\widetilde{e}(\boldsymbol{\lambda})}{e(\boldsymbol{\lambda}) + \widetilde{e}(\boldsymbol{\lambda})} \sum_{\substack{\boldsymbol{\lambda}' \text{ is a} \\ \text{son of } \boldsymbol{\lambda}}} e(\boldsymbol{\lambda}')$$

and set $\tilde{e}(\boldsymbol{\lambda}') \coloneqq q(\boldsymbol{\lambda})$ for the four sons $\boldsymbol{\lambda}'$ of $\boldsymbol{\lambda}$. Note that it holds $\tilde{e}(\boldsymbol{\lambda}') < \tilde{e}(\boldsymbol{\lambda})$ if $\boldsymbol{\lambda}'$ is a successor of $\boldsymbol{\lambda}$. We can thus traverse the tree from the root node to the leaves in such a way that the functional $\widetilde{e}(\boldsymbol{\lambda})$ decreases monotonically. We stop if the error is sufficiently small, yielding a near-best subtree $\mathcal{T}' \subset \mathcal{T}$ such that

$$\big\|\boldsymbol{v}_{\mathcal{T}} - \boldsymbol{v}_{\mathcal{T}'}\big\| \leq \epsilon.$$

Clearly, the construction of the subtree $\mathcal{T}'$ is performed within optimal complexity $\mathcal{O}(\#\mathcal{T})$. In addition, we have the following result in accordance with [27].

Theorem 4.1 ([27]) *For any given, finitely supported input $\boldsymbol{v}_{\mathcal{T}}$, the computational cost of the output $\boldsymbol{v}_{\mathcal{T}'}$, produced by* Coarse$[\epsilon, \boldsymbol{v}_{\mathcal{T}}]$*, remains proportional to $\#\mathcal{T}$. The output tree $\mathcal{T}' \subset \mathcal{T}$ is near-best, i.e., it holds $\|\boldsymbol{v}_{\mathcal{T}} - \boldsymbol{v}_{\mathcal{T}'}\| \leq \epsilon$ and $\#\mathcal{T}'$ is uniformly bounded by the size of the optimal tree.*

4.3. The routine Rhs

The construction of a sequence of approximations for the solution $\boldsymbol{u}$, which converge with a certain rate, requires the availability of a sequence of approximations for the right-hand side $\boldsymbol{f}$ that converge with at least that rate. It can be shown that for any $s < s^\star$, if $\boldsymbol{u} \in \mathfrak{A}_q^s$, then $\boldsymbol{f} \in \mathfrak{A}_q^s$ with $|\boldsymbol{f}|_{\mathfrak{A}_q^s} \lesssim |\boldsymbol{u}|_{\mathfrak{A}_q^s}$. Nevertheless, this property does not tell us how to construct an approximation which is qualitatively as good as the best N-term tree approximation of the right-hand side and has a comparable support size. We will assume the availability of the following routine, whose realization depends on the right-hand side at hand.

Assumption 4.1. *Given a target accuracy $\epsilon > 0$, the output of the routine $\boldsymbol{f}_{\mathcal{T}} =$* Rhs$[\epsilon]$ *satisfies $\|\boldsymbol{f} - \boldsymbol{f}_{\mathcal{T}}\| \leq \epsilon$ and*

$$\#\mathcal{T} \lesssim \epsilon^{-1/s} |\boldsymbol{u}|_{\mathfrak{A}_q^s}^{1/s}$$

provided that $\boldsymbol{u} \in \mathfrak{A}_q^s$ for some $s < s^\star$. The cost to compute this output is proportional to $\epsilon^{-1/s} |\boldsymbol{u}|_{\mathfrak{A}_q^s}^{1/s} + 1$.

4.4. The routine Apply

In view of (3.1), the wavelets have $\tilde{m} = 3$ vanishing moments, which amounts to the *cancellation property*

$$|\langle v, \psi_{\boldsymbol{\lambda}} \rangle| \lesssim 2^{-|\boldsymbol{\lambda}|(q+\tilde{m}+1)} |v|_{W^{\tilde{m},\infty}(\operatorname{supp}(\psi_{\boldsymbol{\lambda}}))}.$$

Here, the term $|v|_{W^{\tilde{m},\infty}(\Omega)} := \sup_{|\boldsymbol{\alpha}|=\tilde{m},\, \boldsymbol{x}\in\Omega} |\partial^{\boldsymbol{\alpha}} v(\boldsymbol{x})|$ denotes the semi-norm in $W^{\tilde{m},\infty}(\Omega)$. By combining this estimate with (2.2), we can derive the following decay estimate for the matrix entries, cf. [18, 28–31],

$$|\langle \mathcal{A}\psi_{\boldsymbol{\lambda}}, \psi_{\boldsymbol{\lambda}'} \rangle| \lesssim \frac{2^{-(|\boldsymbol{\lambda}|+|\boldsymbol{\lambda}'|)(q+\tilde{m}+1)}}{\big(\operatorname{dist}\operatorname{supp}(\psi_{\boldsymbol{\lambda}}), \operatorname{supp}(\psi_{\boldsymbol{\lambda}'})\big)^{2+2q+2\tilde{m}}}. \tag{4.3}$$

Hence, wavelets having a larger distance from each other produce a larger value in the denominator of the term, giving a smaller matrix coefficient in total. The same holds for two wavelets having a larger sum of refinement scales $|\boldsymbol{\lambda}| + |\boldsymbol{\lambda}'|$. Note that (4.3) gives rise to the so-called *first compression*, cf. [18, 30].

In addition to (4.3), we can state a second estimate, cf. [18, 30, 31],

$$|\langle \mathcal{A}\psi_{\boldsymbol{\lambda}}, \psi_{\boldsymbol{\lambda}'} \rangle| \lesssim \frac{2^{-|\boldsymbol{\lambda}|(q+\tilde{m}+1)} 2^{|\boldsymbol{\lambda}'|(1-q)}}{\operatorname{dist}\big(\operatorname{supp}(\psi_{\boldsymbol{\lambda}}), \operatorname{sing}\operatorname{supp}(\psi_{\boldsymbol{\lambda}'})\big)^{2q+\tilde{m}}}, \tag{4.4}$$

where we assumed that $|\boldsymbol{\lambda}'| < |\boldsymbol{\lambda}|$. This estimate leads to the so-called *second compression*, cf. [18, 30]. The term $\operatorname{sing}\operatorname{supp}(\psi_{\boldsymbol{\lambda}})$ stands for the singular support of a wavelet, i.e., the points on which the wavelet is not smooth. The singular support coincides always with certain element boundaries, see also Figure 4.1, which can be exploited to compute the particular distance in accordance with [17].

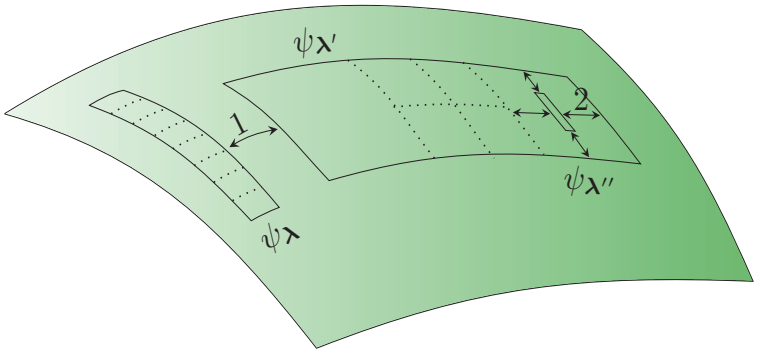

Fig. 4.1. Illustration of the situation which is affected by the first compression (1) and by the second compression (2).

Estimates (4.3) and (4.4) are the key estimates to derive the sparse representation of the operator $\boldsymbol{A}$.

Definition 4.1. *The operator $\boldsymbol{A}$ is called $s^\star$-compressible if, for any $s < s^\star$, there exist submatrices $\boldsymbol{A}_j$ with at most the order of $\alpha_j 2^j$ non-zero entries per row and column such that*

$$\|\boldsymbol{A} - \boldsymbol{A}_j\| \leq \alpha_j 2^{-sj},$$

with $(\alpha_j)_{j\in\mathbb{N}}$ being a summable sequence of positive numbers satisfying $\sum_{j\in\mathbb{N}} \alpha_j \leq 1$.

Theorem 4.2 ([31]) *Suppose that the operator $\mathcal{A} : H^{q+\sigma}(\Gamma) \to H^{-q+\sigma}(\Gamma)$ is bounded for a sufficiently large range of positive σ. Then, $\boldsymbol{A}$ is $s^\star$-compressible with $s^\star > \bar{s}$ provided that $\tilde{m} > 1 - 2q$.*

Setting $\hat{j} \coloneqq j + \log_2(\alpha_j)$, the corresponding compression pattern has also been identified in [31]. To obtain $\boldsymbol{A}_j$ from $\boldsymbol{A}$ replace $\langle \mathcal{A}\psi_{\boldsymbol{\lambda}}, \psi_{\boldsymbol{\lambda}'}\rangle$ by zero according to the following compression rule:

$$\big[\boldsymbol{A}_j\big]_{\boldsymbol{\lambda},\boldsymbol{\lambda}'} = \begin{cases} 0, & \text{if } \big||\boldsymbol{\lambda}| - |\boldsymbol{\lambda}'|\big| > \hat{j}, \\ 0, & \text{if } 2^{\min\{|\boldsymbol{\lambda}|,|\boldsymbol{\lambda}'|\}} \operatorname{dist}\big(\operatorname{supp}(\psi_{\boldsymbol{\lambda}}), \operatorname{supp}(\psi_{\boldsymbol{\lambda}'})\big) \geq \mathcal{B}^{(j,1)}_{||\boldsymbol{\lambda}|-|\boldsymbol{\lambda}'||}, \\ 0, & \text{if } |\boldsymbol{\lambda}| - |\boldsymbol{\lambda}'| > \frac{\hat{j}}{2} \text{ and } 2^{|\boldsymbol{\lambda}'|} \operatorname{dist}\big(\operatorname{supp}(\psi_{\boldsymbol{\lambda}}), \operatorname{sing\,supp}(\psi_{\boldsymbol{\lambda}'})\big) \geq \mathcal{B}^{(j,2)}_{||\boldsymbol{\lambda}|-|\boldsymbol{\lambda}'||}, \\ 0, & \text{if } |\boldsymbol{\lambda}'| - |\boldsymbol{\lambda}| > \frac{\hat{j}}{2} \text{ and } 2^{|\boldsymbol{\lambda}|} \operatorname{dist}\big(\operatorname{sing\,supp}(\psi_{\boldsymbol{\lambda}}), \operatorname{supp}(\psi_{\boldsymbol{\lambda}'})\big) \geq \mathcal{B}^{(j,2)}_{||\boldsymbol{\lambda}|-|\boldsymbol{\lambda}'||}, \\ \langle \mathcal{A}\psi_{\boldsymbol{\lambda}}, \psi_{\boldsymbol{\lambda}'}\rangle, & \text{otherwise,} \end{cases} \tag{4.5}$$

where the cut-off parameters $\mathcal{B}^{(j,1)}_\ell$ and $\mathcal{B}^{(j,2)}_\ell$ are given by

$$\mathcal{B}^{(j,1)}_\ell \coloneqq a \max\Big\{1, 2^{(\hat{j}-2\ell)b}\Big\}, \qquad \mathcal{B}^{(j,2)}_\ell \coloneqq a \max\Big\{\gamma_{\ell-\frac{\hat{j}}{2}} 2^{\hat{j}-2\ell}, 2^{-\ell}\Big\}. \tag{4.6}$$

Here, $a > 1$ and $1/4 \leq b < 1/2$ are suitable fixed parameters, and $(\gamma_\ell)_{\ell\in\mathbb{N}}$ is a polynomially decreasing sequence such that $\sum_{\ell=\mathbb{N}} \gamma_\ell < \infty$. Specifically, we take $\gamma_\ell \coloneqq \ell^{-2}$.

Given the target accuracy $\epsilon > 0$ and a finitely supported vector $\boldsymbol{v}$ with the associated index set $\mathcal{T}$ forming a tree, we can compute a sequence of nested trees $\emptyset = \mathcal{T}_J \subset \cdots \subset \mathcal{T}_1 \subset \mathcal{T}_0 \coloneqq \mathcal{T}$ such that

$$\big\|\boldsymbol{v} - \boldsymbol{v}|_{\mathcal{T}_j}\big\| \leq 2^{j\bar{s}}\epsilon, \quad j = 1, \dots, J. \tag{4.7}$$

Here, J is found by

$$J \coloneqq \left\lceil \frac{\log_2(\|\boldsymbol{v}\|/\epsilon)}{\bar{s}} \right\rceil.$$

The difference sets $\Delta_j \coloneqq \mathcal{T}_j \setminus \mathcal{T}_{j-1}$ are referred to as *layers*. We denote the according portions of $\boldsymbol{v}$ by $\boldsymbol{v}|_{\Delta_j}$ and write $\boldsymbol{v} = \big[\boldsymbol{v}|_{\Delta_J}, \boldsymbol{v}|_{\Delta_{J-1}}, \dots, \boldsymbol{v}|_{\Delta_1}\big]^\top$. Using estimate (4.7), we deduce that

$$\big\|\boldsymbol{v}|_{\Delta_j}\big\| = \big\|\boldsymbol{v}|_{\mathcal{T}_{j-1}} - \boldsymbol{v}|_{\mathcal{T}_j}\big\| \leq \big\|\boldsymbol{v} - \boldsymbol{v}|_{\mathcal{T}_j}\big\| \leq 2^{j\bar{s}}\epsilon.$$

Finally, we compute $\boldsymbol{w}_\epsilon = \sum_{j=1}^J \boldsymbol{A}_j \boldsymbol{v}|_{\Delta_j}$ to approximate the infinite matrix-vector product $\boldsymbol{A}\boldsymbol{v}$ up to an error ϵ:

$$\|\boldsymbol{A}\boldsymbol{v} - \boldsymbol{w}_\epsilon\| = \left\|\boldsymbol{A}\boldsymbol{v} - \sum_{j=1}^J \boldsymbol{A}_j \boldsymbol{v}|_{\Delta_j}\right\| \leq \sum_{j=1}^J \|\boldsymbol{A} - \boldsymbol{A}_j\| \big\|\boldsymbol{v}|_{\Delta_j}\big\| \leq \sum_{j=1}^J \alpha_j 2^{-j\bar{s}} 2^{j\bar{s}}\epsilon \leq \epsilon.$$

This is the first part of the following theorem.

Theorem 4.3 ([18]) *Let $\epsilon > 0$ and $\boldsymbol{v}$ be finitely supported with tree structured index set. Then, the approximation $\boldsymbol{w}_\epsilon = \textsc{Apply}(\epsilon, \boldsymbol{v})$ is also a finitely supported tree and satisfies $\|\mathbf{A}\boldsymbol{v} - \boldsymbol{w}_\epsilon\| \leq \epsilon$ with, for any $s < s^\star$, $\#\operatorname{supp} \boldsymbol{w}_\epsilon \lesssim \epsilon^{-1/s} |\boldsymbol{v}|^s_{\mathfrak{A}^s_q}$, where the number of arithmetic operations and storage locations is bounded by some absolute multiple of $\epsilon^{-1/s}|\boldsymbol{v}|^s_{\mathfrak{A}^s_q} + 1$.*

In our implementation, we employ the following modification of the approximate matrix-vector product to obtain symmetric matrix pattern. Then, we can compute the system matrix

basically along the lines of [17]. We first *predict* the supporting tree $\mathcal{T}'$ of the output at hand by using (4.5). Then, setting $\Delta_{-1} := \mathcal{T}' \setminus \mathcal{T}$, the approximation $\boldsymbol{w}_\epsilon = \sum_{j=1}^{J} \boldsymbol{A}_j \boldsymbol{v}|_{\Delta_j}$ can be schematically written as

$$\boldsymbol{w}_\epsilon = \left[\begin{array}{c|c|c|c|c|c} \boldsymbol{A}_J & \boldsymbol{A}_{J-1} & \boldsymbol{A}_{J-2} & \cdots & \boldsymbol{A}_1 & \boldsymbol{A}_0 \end{array}\right] \cdot \left[\begin{array}{c} \boldsymbol{v}|_{\Delta_J} \\ \hline \boldsymbol{v}|_{\Delta_{J-1}} \\ \hline \boldsymbol{v}|_{\Delta_{J-2}} \\ \hline \vdots \\ \hline \boldsymbol{v}|_{\Delta_1} \\ \hline \boldsymbol{0}|_{\Delta_0} \end{array}\right].$$

This involves a quadratic matrix, where we can just set $\boldsymbol{A}_0 := \boldsymbol{0}$. We arrive at a symmetric matrix pattern, if, whenever a coefficient $\langle \mathcal{A}\psi_{\boldsymbol{\lambda}}, \psi'_{\boldsymbol{\lambda}} \rangle$ is relevant according to (4.5), we also let the transposed coefficient $\langle \mathcal{A}\psi'_{\boldsymbol{\lambda}}, \psi_{\boldsymbol{\lambda}} \rangle$ be relevant. This formally leads to the matrix-vector product

$$\boldsymbol{w}_\epsilon = \left[\begin{array}{c|c|c|c} \boldsymbol{A}_{J,J} & \boldsymbol{A}_{J,J-1} & \cdots & \boldsymbol{A}_{J,0} \\ \hline \boldsymbol{A}_{J-1,J} & \boldsymbol{A}_{J-1,J-1} & \cdots & \boldsymbol{A}_{J-1,0} \\ \hline \vdots & \vdots & & \vdots \\ \hline \boldsymbol{A}_{0,J} & \boldsymbol{A}_{0,J-1} & \cdots & \boldsymbol{A}_{0,0} \end{array}\right] \cdot \left[\begin{array}{c} \boldsymbol{v}|_{\Delta_J} \\ \hline \boldsymbol{v}|_{\Delta_{J-1}} \\ \hline \vdots \\ \hline \boldsymbol{0}|_{\Delta_0} \end{array}\right] \tag{4.8}$$

and gives rise to an even more accurate approximation, while the computational complexity is at most doubled. We refer to Figure 4.2 for an illustration of this matrix compression.

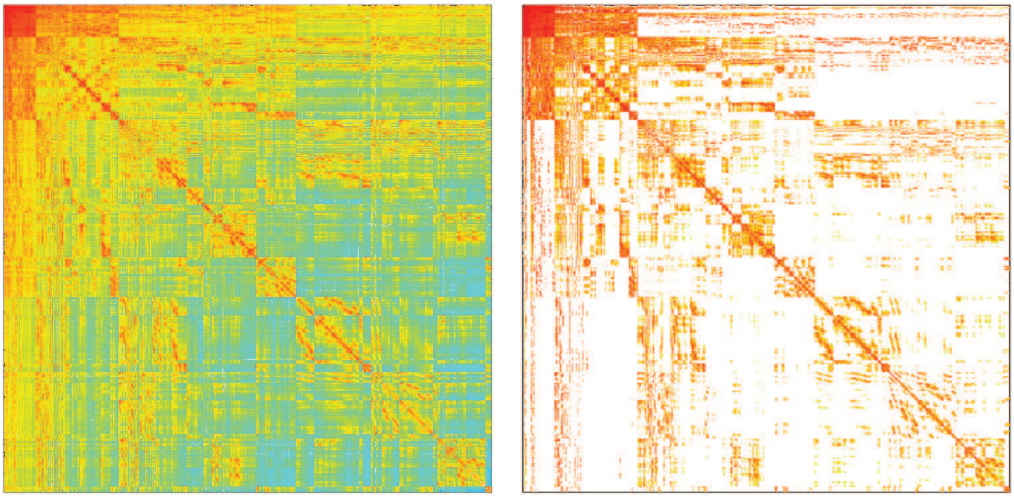

Fig. 4.2. Original matrix (left) and compressed matrix (right).

4.5. The routine SOLVE

Starting from the Galerkin solution $\boldsymbol{u}_\mathcal{T}$ which satisfies $\|\boldsymbol{u} - \boldsymbol{u}_\mathcal{T}\| \leq \epsilon$, we have extended the index set $\mathcal{T}$ to the index set $\mathcal{T}''$ which is sufficiently large to ensure the saturation property. Hence, the error of the new Galerkin solution $\boldsymbol{u}_{\mathcal{T}''} = \text{SOLVE}(\mathcal{T}'')$ decreases by a constant factor compared to the error ϵ of the old Galerkin solution $\boldsymbol{u}_\mathcal{T}$.

In order to determine $\boldsymbol{u}_{\mathcal{T}''}$, we shall solve the system (3.3) for $\mathcal{T}''$ with sufficient accuracy. In order to damp the error induced by $\boldsymbol{u}_\mathcal{T}$ by a constant factor, it suffices to take the system matrix

from the approximate matrix-vector product (4.8) for $\boldsymbol{u}_{\mathcal{T}}$ with a somewhat higher accuracy. Setting $\Delta_0 := \mathcal{T}'' \setminus \mathcal{T}$ and increasing the accuracy by $\ell := -\lfloor \log_2(\delta) \rfloor$ levels, we arrive at

$$\hat{\boldsymbol{A}}_{\mathcal{T}''} \boldsymbol{u}_{\mathcal{T}''} = \boldsymbol{f}_{\mathcal{T}''}, \text{ where } \hat{\boldsymbol{A}}_{\mathcal{T}''} = \left[\begin{array}{c|c|c|c} \boldsymbol{A}_{J+\ell,J+\ell} & \boldsymbol{A}_{J+\ell,J+\ell} & \dots & \boldsymbol{A}_{J+\ell,\ell} \\ \hline \boldsymbol{A}_{J+\ell,J+\ell} & \boldsymbol{A}_{J+\ell-1,J+\ell} & \dots & \boldsymbol{A}_{J+\ell,\ell} \\ \hline \vdots & \vdots & & \vdots \\ \hline \boldsymbol{A}_{\ell,J+\ell} & \boldsymbol{A}_{\ell,J+\ell} & \dots & \boldsymbol{A}_{\ell,\ell} \end{array}\right]. \tag{4.9}$$

Here, $0 < \delta < 1$ is the desired damping factor. In view of Theorem 4.3, the computational complexity of assembling and solving this system of linear equations is obviously $\mathcal{O}(\epsilon^{-1/s} |\boldsymbol{u}|_{\mathfrak{A}_q^s} + 1)$.

Remark 4.1. The following modification turned out to improve the efficiency of the adaptive algorithm since the loop in (4.2) for estimating the residuum already terminates after one iteration when a good initial guess η_{init} is applied. Namely, we already increase the accuracy for the routine APPLY by replacing $j \mapsto j + \ell$ in the system matrix in (4.8). This is still computationally optimal since ℓ is fixed. The main advantage is that then the system matrix in (4.9) is simply obtained by removing all lines and columns which do not belong to indices from the set $\mathcal{T}''$. Consequently, only a single system matrix has to be assembled for one step of the adaptive algorithm.

5. Numerical Results

5.1. Laplace equation in Fichera's vertex

We shall consider the Laplace equation inside a given domain Ω with Dirichlet boundary conditions:

$$\Delta U = 0 \text{ in } \Omega, \quad U = f \text{ on } \Gamma. \tag{5.1}$$

We first consider Fichera's vertex as seen in Figure 5.1, which consists of 12 patches. The Dirichlet data in (5.1) are chosen as $f \equiv 1$ such that $U \equiv 1$ is the solution of the Laplace equation.

We shall employ the indirect formulation to solve (5.1). We make the ansatz

$$U(\boldsymbol{x}) = \frac{1}{4\pi} \int_\Gamma \frac{u(\boldsymbol{y})}{\|\boldsymbol{x} - \boldsymbol{y}\|} \, \mathrm{d}\sigma_{\boldsymbol{y}}, \quad \boldsymbol{x} \in \Omega, \tag{5.2}$$

which, due to the continuity of the single-layer potential at the boundary Γ, leads to the boundary integral equation

$$\frac{1}{4\pi} \int_\Gamma \frac{u(\boldsymbol{y})}{\|\boldsymbol{x} - \boldsymbol{y}\|} \, \mathrm{d}\sigma_{\boldsymbol{y}} = f(\boldsymbol{x}), \quad \boldsymbol{x} \in \Gamma. \tag{5.3}$$

This is a boundary integral equation of the first kind, involving the single-layer operator which is of order $2q = -1$. We solve this boundary integral equation by our adaptive method and evaluate, in order to verify the solution, the potential (5.2) in many points $\{\boldsymbol{x}_i\}$ which are equi-distributed inside the Fichera vertex.

In Table 5.1, we see the output of the adaptive code. The numbers in the second and third column are (in this order) the support size $N_{\boldsymbol{r}}$ for computing the residuum $\boldsymbol{r}$ and the degrees of freedom N_{dof}, obtained after coarsening the residuum. The fourth column contains the norm

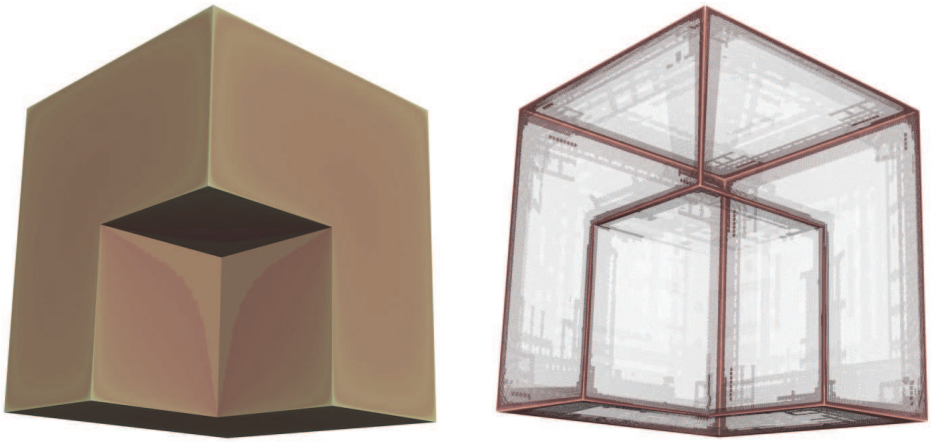

Fig. 5.1. Solution (left) and refinement (right) of the Fichera vertex.

of the approximate residuum and the fifth column contains the error of the potential. The sixth column (nnz (%)) shows us the percentage of non-zero entries in the system matrix after compression. We can clearly see that not many matrix entries remain and the matrix gets very sparse. Finally, the last column contains the computation time in seconds used for one step of the code, that is the time for growing the tree until the terminating condition is met, together with assembling and solving the system, but without evaluating the potential.

Table 5.1: Results for the single-layer potential and Fichera's vertex.

step	$N_{\boldsymbol{r}}$	N_{dof}	$\|\boldsymbol{r}\|$	potential error	nnz (%)	time (s)
0	—	12	—	$6.09{\cdot}10^{-2}$	100	0.01
1	48	38	$6.12{\cdot}10^{-2}$	$3.60{\cdot}10^{-2}$	100	0.06
2	183	170	$9.10{\cdot}10^{-2}$	$1.58{\cdot}10^{-2}$	92.2	0.13
3	714	564	$4.34{\cdot}10^{-2}$	$9.28{\cdot}10^{-3}$	42.3	0.47
4	2840	1502	$2.56{\cdot}10^{-2}$	$4.40{\cdot}10^{-3}$	22.7	2.71
5	8511	4502	$1.61{\cdot}10^{-2}$	$4.54{\cdot}10^{-4}$	7.91	14
6	25022	12377	$1.07{\cdot}10^{-2}$	$1.64{\cdot}10^{-4}$	3.10	48
7	47717	25068	$6.15{\cdot}10^{-3}$	$2.05{\cdot}10^{-4}$	1.37	86
8	104464	50132	$4.05{\cdot}10^{-3}$	$1.67{\cdot}10^{-4}$	0.67	238
9	239922	110464	$2.86{\cdot}10^{-3}$	$7.05{\cdot}10^{-5}$	0.30	682
10	562938	239186	$2.00{\cdot}10^{-3}$	$2.58{\cdot}10^{-5}$	0.13	2105
11	1234025	518350	$1.29{\cdot}10^{-3}$	$2.78{\cdot}10^{-5}$	0.06	6633
12	2111448	890168	$7.60{\cdot}10^{-4}$	$2.78{\cdot}10^{-5}$	0.03	11521
13	3968557	1618570	$5.79{\cdot}10^{-4}$	$1.17{\cdot}10^{-5}$	0.02	34750

The first line of Table 5.1 has to be interpreted as follows. The calculation starts with the piecewise constant ansatz functions associated with the 12 patches of the Fichera vertex, for which we assemble the system matrix, the right-hand side and subsequently solve the linear system of equations. Afterwards, we evaluate the potential and calculate its maximal error. A general line is interpreted as follows: we have a solution of the previous step $k-1$ which consists of $N_{\mathrm{dof}}^{(k-1)}$ degrees of freedom, as found in the line above. In the k-th step, the residuum is calculated for $N_{\boldsymbol{r}}^{(k)}$ degrees of freedom and then this residuum is coarsened to $N_{\mathrm{dof}}^{(k)}$ degrees

of freedom, for which the solution is finally computed. The norm of the residuum and the potential error are tabulated in the same line.

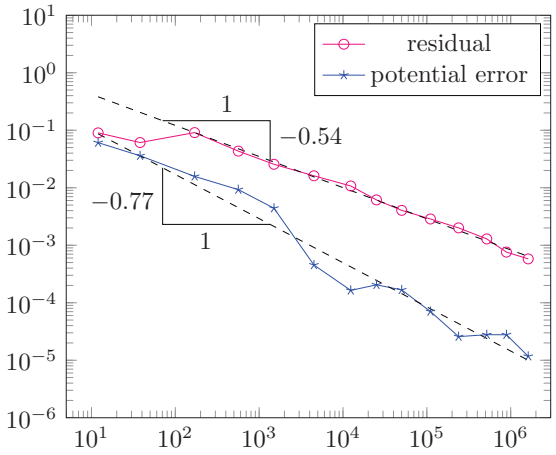

Fig. 5.2. Residual and potential error for the single-layer potential and Fichera's vertex.

Figure 5.2 shows the degrees of freedom N_{dof}, plotted against the norm $\|\boldsymbol{r}\|$ of the residuum, as well as the potential error. We observe that the convergence rate for the residuum is $N_{\mathrm{dof}}^{-0.5}$, even slightly better. This is optimal for an isotropic discretization since the anisotropic singularities which appear at the edges cannot be resolved. The convergence rate for potential is approximately $N_{\mathrm{dof}}^{-0.75}$, caused by super convergence effects of the potential evaluation.

The final solution which has been computed is shown in the left plot of Figure 5.1. In the right plot of this figure, the mesh refinement, produced by the code, is visualized. Since we would not be able to see the refinement by drawing the grid, this picture was produced in the following way: After the code terminated, we assigned to each active wavelet a point in the center of its support which is weighted with 2 to the power of the wavelet's level, achieving that a small wavelet gets assigned a large value. The picture below is thus to be interpreted as: The lighter the colour, the finer are the elements in this area. The code refines the elements along the edges and vertices of the geometry. This is the behaviour to be expected since the density u exhibits the singularities of both, the solution of the interior as well as of the exterior Laplace equation.

5.2. Laplace equation in a gearwheel

In our second example, we solve the Dirichlet problem (5.1) for the Laplacian in a nontrivial geometry, namely a gearwheel, see Figure 5.3, which is represented by 504 patches. As Dirichlet data in (5.1), we choose the restriction $f = U|_\Gamma$ of the harmonic polynomial $U(\boldsymbol{x}) = 4x_1^2 - 3x_2^2 - x_3^2$ where $\boldsymbol{x} = (x_1, x_2, x_3)$. Hence, the analytical solution of (5.1) is given by $U(\boldsymbol{x})$.

We again apply the indirect ansatz (5.2) by the single-layer potential, which again amounts to solving the boundary integral equation (5.3) for the single-layer operator. The application of the adaptive algorithm yields the results found in Table 5.2, the layout of which is the same as in the previous example. The convergence rate and the potential error are shown in Figure 5.4. Here, despite of edge singularities, we observe a convergence rate which is nearby the maximal rate $N^{-0.75}$. Compared to this, the potential error behaves this time only slightly better. The final solution is found in the left plot of Figure 5.3, while the right plot of this figure illustrates

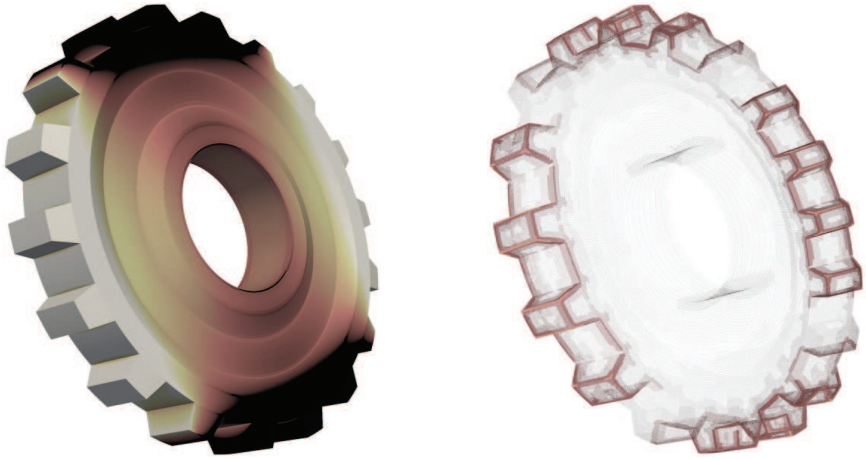

Fig. 5.3. Solution (left) and refinement (right) on the gearwheel with a smooth right-hand side.

the refinement.

Table 5.2: Results for the gearwheel with a smooth right-hand side.

step	$N_{\boldsymbol{r}}$	N_{dof}	$\|\boldsymbol{r}\|$	potential error	nnz (%)	time (s)
0	2016	1716	6.35	$8.98{\cdot}10^{-1}$	98.48	17
1	5084	4191	6.72	$1.59{\cdot}10^{-1}$	55.15	38
2	6290	5605	3.04	$7.82{\cdot}10^{-2}$	47.23	51
3	16384	10640	2.68	$2.54{\cdot}10^{-2}$	28.37	151
4	29340	15999	1.92	$1.75{\cdot}10^{-2}$	19.60	271
5	35415	20896	1.14	$1.24{\cdot}10^{-2}$	17.00	385
6	41162	33010	$3.72{\cdot}10^{-1}$	$7.13{\cdot}10^{-3}$	13.15	721
7	136946	57152	$8.54{\cdot}10^{-1}$	$2.29{\cdot}10^{-3}$	10.12	3812
8	144262	101764	$2.37{\cdot}10^{-1}$	$3.02{\cdot}10^{-3}$	4.47	3554
9	319863	142179	$4.31{\cdot}10^{-1}$	$1.69{\cdot}10^{-3}$	3.78	10234
10	366229	254617	$1.37{\cdot}10^{-1}$	$1.32{\cdot}10^{-3}$	1.73	11844
11	929944	336611	$2.86{\cdot}10^{-1}$	$4.25{\cdot}10^{-4}$	1.83	45941
12	786306	597850	$4.73{\cdot}10^{-2}$	$3.85{\cdot}10^{-4}$	0.75	35142
13	2775537	811936	$1.93{\cdot}10^{-1}$	$6.85{\cdot}10^{-5}$	0.39	228111
14	1808460	1320772	$2.97{\cdot}10^{-2}$	$6.92{\cdot}10^{-5}$	0.16	104015

5.3. A cartoon image as right-hand side

We consider next the solution of a Dirichlet problem for the exterior (low frequency) Helmholtz equation

$$\Delta U + U = 0 \text{ in } \mathbb{R} \setminus \overline{\Omega}, \quad U = f \text{ on } \Gamma.$$

In order to avoid spurious modes, we make the ansatz of Brakhage and Werner, cf. [32], and combine the acoustic double-layer potential with the acoustic single-layer potential

$$U(\boldsymbol{x}) = \int_\Gamma \left(\frac{\partial}{\partial \boldsymbol{n_y}} G(\boldsymbol{x},\boldsymbol{y}) - iG(\boldsymbol{x},\boldsymbol{y}) \right) u(\boldsymbol{y}) \, \mathrm{d}\sigma_{\boldsymbol{y}}, \quad \boldsymbol{x} \in \mathbb{R} \setminus \overline{\Omega},$$

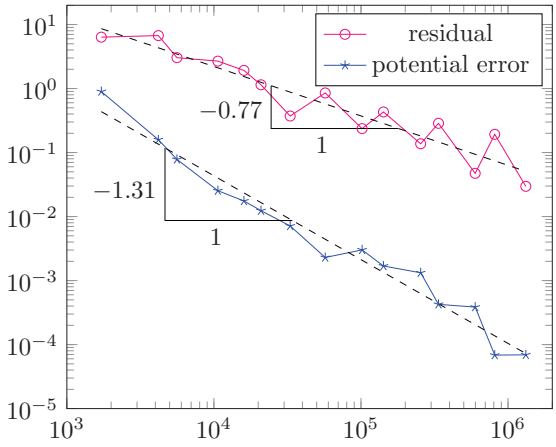

Fig. 5.4. Residual and potential error for the single-layer potential and the gearwheel geometry.

where $u \in L^2(\Gamma)$ is an unknown density and

$$G(\boldsymbol{x},\boldsymbol{y}) := \frac{1}{4\pi}\frac{e^{i\|\boldsymbol{x}-\boldsymbol{y}\|}}{\|\boldsymbol{x}-\boldsymbol{y}\|}$$

is the fundamental solution of the Helmholtz operator. Letting $\boldsymbol{x}$ tend to the boundary and observing the jump condition of the double-layer potential yields the boundary integral equation

$$\frac{1}{2}u(\boldsymbol{x}) + \int_\Gamma \left(\frac{\partial}{\partial \boldsymbol{n}_{\boldsymbol{y}}}G(\boldsymbol{x},\boldsymbol{y}) - iG(\boldsymbol{x},\boldsymbol{y})\right)u(\boldsymbol{y})\,\mathrm{d}\sigma_{\boldsymbol{y}} = f(\boldsymbol{x}), \quad \boldsymbol{x} \in \Gamma. \tag{5.4}$$

It is a Fredholm integral equation of the second kind, which means that the underlying integral operator is of order $2q = 0$.

We will consider in this example a smooth boundary, so that the boundary integral operator under consideration offers the full regularity, but a non-smooth right-hand side. Hence, we choose the sphere $\mathbb{S}^2$ as geometry which is represented by six patches, see also Figure 5.5. The right-hand side is given as

$$f(\boldsymbol{x}) = \begin{cases} 1, & \text{if } \|\boldsymbol{x}-(0,0,1)\|^2 \leq \frac{1}{2}, \\ 0, & \text{elsewhere.} \end{cases}$$

Such cartoon functions have been studied in [33] and can be approximated at the rate $N^{-0.5}$. Hence, the density $u \in L^2(\Gamma)$ of the boundary integral equation (5.4) is expected to converge also with the rate $N^{-0.5}$, see [26] for the details. Notice that uniform refinement would give only half the rate, i.e., the rate $N^{-0.25}$, since the right-hand side is only in the Sobolev space $H^{1/2-\delta}(\Gamma)$ with $\delta > 0$ being arbitrarily small.

The results of the algorithm are presented in Table 5.3. Since the analytical solution is not known for the Dirichlet data under consideration, we cannot compute the potential error. Nevertheless, the norm of the residuum converges, and, as seen in Figure 5.6, the optimal rate $N^{-0.5}$ is achieved. The final solution is found in the left plot of Figure 5.5, while the right plot of Figure 5.5 illustrates the refinement. It is clearly seen that refinement takes mainly place at the jump of the right-hand side.

Table 5.3: Results for the sphere with a discontinuous right-hand side.

step	$N_{\boldsymbol{r}}$	N_{dof}	$\|\boldsymbol{r}\|$	nnz (%)	time (s)
0	6	—	—	100	0.02
1	76	16	$4.04{\cdot}10^{-1}$	100	0.13
2	1496	110	$1.64{\cdot}10^{-1}$	22.7	7.51
3	3836	843	$4.21{\cdot}10^{-2}$	9.26	26
4	25574	4813	$1.99{\cdot}10^{-2}$	1.50	344
5	53156	13778	$1.02{\cdot}10^{-2}$	0.70	560
6	223563	46861	$7.56{\cdot}10^{-3}$	0.17	3813
7	636402	122534	$4.57{\cdot}10^{-3}$	0.06	12697
8	1624573	330860	$2.81{\cdot}10^{-3}$	0.02	43434

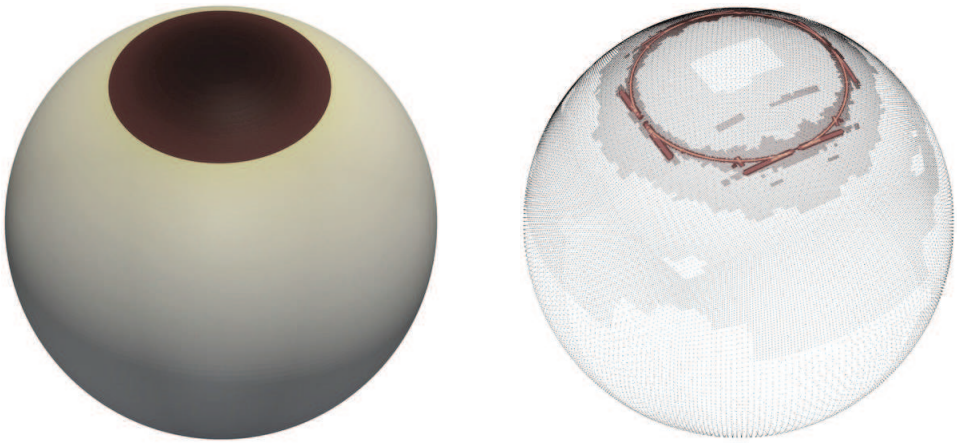

Fig. 5.5. Solution (left) and refinement (right) for the Brakhage-Werner formulation and the sphere.

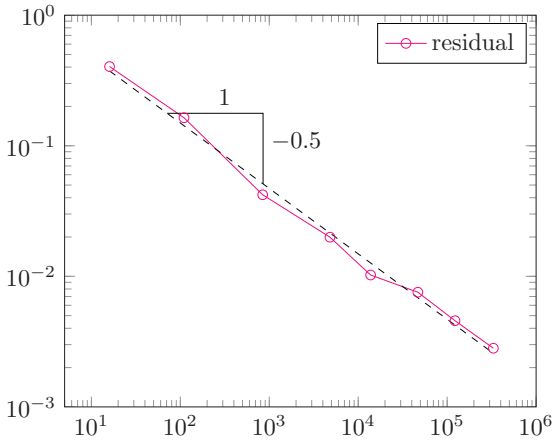

Fig. 5.6. Residual in case of a discontinuous right-hand side.

6. Conclusion

In this article, we presented an adaptive wavelet boundary element method which realizes asymptotically optimal complexity. By asymptotically optimal we mean that any target accu-

racy can be achieved at a computational expense that stays proportional to the solution's best N-term tree approximation. The layout of the algorithm mimics standard adaptive algorithms with the only difference that we *directly approximate the residuum* with sufficient accuracy instead of computing error estimators to *estimate the residuum*. The numerical results validate the feasibility and efficiency of the proposed algorithm.

Acknowledgments. This research has been supported by the Swiss National Science Foundation (SNSF) through the DACH-project "BIOTOP: Adaptive Wavelet and Frame Techniques for Acoustic BEM".

References

[1] L. Greengard and V. Rokhlin, A fast algorithm for particle simulation, *J. Comput. Phys.*, **73** (1987), 325–348.

[2] V. Rokhlin, Rapid solution of integral equations of classical potential theory, *J. Comput. Phys.*, **60** (1985), 187–207.

[3] W. Hackbusch and Z. Nowak, On the fast matrix multiplication in the boundary element method by panel clustering, *Numer. Math.*, **54** (1989), 463–491.

[4] M. Bebendorf, Approximation of boundary element matrices, *Numer. Math.*, **86** (2000), 565–589.

[5] M. Bebendorf and S. Rjasanow, Adaptive low-rank approximation of collocation matrices, *Computing*, **70** (2003), 1–24.

[6] W. Hackbusch, A sparse matrix arithmetic based on $\mathcal{H}$-matrices. Part I: Introduction to $\mathcal{H}$-matrices, *Computing*, **64** (1999), 89–108.

[7] E. Tyrtyshnikov, Mosaic sceleton approximation, *Calcolo*, **33** (1996), 47–57.

[8] G. Beylkin, R. Coifman and V. Rokhlin, The fast wavelet transform and numerical algorithms, *Comm. Pure and Appl. Math.*, **44** (1991), 141–183.

[9] W. Dahmen, H. Harbrecht and R. Schneider, Compression techniques for boundary integral equations. Asymptotically optimal complexity estimates, *SIAM J. Numer. Anal.*, **43**:6 (2006), 2251–2271.

[10] B. Faermann, Localization of the Aronszajn-Slobodeckij norm and application to adaptive boundary element methods. Part II: The three-dimensional case, *Numer. Math.*, **92**:3 (2002), 467–499.

[11] M. Feischl, M. Karkulik, J. Melenk and D. Praetorius, Quasi-optimal convergence rate for an adaptive boundary element method, *SIAM J. Numer. Anal.*, **51**:2 (2013), 1327–1348.

[12] T. Gantumur, Adaptive boundary element methods with convergence rates, *Numer. Math.*, **124** (2013), 471–516.

[13] A. Cohen, W. Dahmen and R. DeVore, Adaptive wavelet methods for elliptic operator equations. Convergence rates, *Math. Comput.*, **70** (2001), 27–75.

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

[15] T. Gantumur and R. Stevenson, Computation of singular integral operators in wavelet coordinates, *Computing*, **76** (2006), 77–107.

[16] R. DeVore, Nonlinear approximation, *Acta Numerica*, **7** (1998), 51–150.

[17] H. Harbrecht and R. Schneider, Wavelet Galerkin schemes for boundary integral equations. implementation and quadrature, *SIAM J. Sci. Comput.*, **27**:4 (2006), 1347–1370.

[18] W. Dahmen, H. Harbrecht and R. Schneider, Adaptive methods for boundary integral equations. Complexity and convergence estimates, *Math. Comput.*, **76** (2007), 1243–1274.

[19] T. Gantumur, H. Harbrecht and R. Stevenson, An optimal adaptive wavelet method for elliptic equations without coarsening, *Math. Comput.*, **76** (2007), 615–629.

[20] T. Gantumur, An optimal adaptive wavelet method for nonsymmetric and indefinite elliptic problems, *J. Comput. Appl. Math.*, **211**:1 (2008), 90–102.

[21] T. Hughes, J. Cottrell and Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, *Comput. Methods Appl. Mech. Engrg.*, **194**:39 (2005), 4135–4195.

[22] W. Dahmen, A. Kunoth and K. Urban, Biorthogonal spline-wavelets on the interval. Stability and moment conditions, *Appl. Comp. Harm. Anal.*, **6** (1999), 259–302.

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

[24] A. Cohen, W. Dahmen and R. DeVore, Sparse evaluation of compositions of functions using multiscale expansions, *SIAM J. Math. Anal.*, **35** (2003), 279–303.

[25] A. Cohen, W. Dahmen, I. Daubechies and R. DeVore, Tree-structured approximation and optimal encoding, *Appl. Comp. Harm. Anal.*, **11** (2001), 192–226.

[26] S. Dahlke and M. Weimar, Besov regularity for operator equations on patchwise smooth manifolds, *Found. Comput. Math.*, **15** (2015), 1533–1569.

[27] P. Binev and R. DeVore, Fast computation in adaptive tree approximation, *Numer. Math.*, **97** (2004), 193–217.

[28] T.von Petersdorff and C. Schwab, Wavelet approximation for first kind integral equations on polygons, *Numer. Math.*, **74** (1996), 479–519.

[29] T.von Petersdorff and C. Schwab, Fully discrete multiscale Galerkin BEM, Multiscale Wavelet Methods for PDEs, pages 287–346, Academic Press, San Diego, 1997.

[30] R. Schneider, Multiskalen- und Wavelet-Matrixkompression: Analysisbasierte Methoden zur effizienten Lösung großer vollbesetzter Gleichungssysteme, B.G. Teubner, Stuttgart, 1998.

[31] R. Stevenson, On the compressibility of operators in wavelet coordinates, *SIAM J. Math. Anal.*, **35**:5 (2004), 1110–1132.

[32] H. Brakhage and P. Werner, Über das Dirichletsche Außenraumproblem für die Helmholtzsche Schwingungsgleichung, *Arch. Math.*, **16** (1965), 325–329.

[33] D. Donoho, Sparse components of images and optimal atomic decomposition, *Constr. Approx.*, **17** (2001), 353–382.