Wavelet Galerkin Algorithms for Boundary Integral Equations

Wavelet Galerkin Algorithms for Boundary Integral Equations

Christian Lage and Christoph Schwab

Seminar für Angewandte Mathematik
ETH Zürich
Rämistrasse 101
CH-8092 Zürich, Switzerland

October 24, 1997

Abstract

The implementation of a fast, wavelet-based Galerkin discretization of second kind integral equations on piecewise smooth surfaces $\Gamma \subset \mathbb{R}^3$ is described. It allows meshes consisting of triangles as well as quadrilaterals. The algorithm generates a sparse, approximate stiffness matrix with $\mathcal{N} = O(N(\log N)^2)$ nonvanishing entries in $O(N(\log N)^4)$ operations where N is the number of degrees of freedom on the boundary while essentially retaining the asymptotic convergence rate of the full Galerkin scheme. A new proof of the matrix-compression estimates is given based on derivative-free kernel estimates. The condition number of the sparse stiffness matrices is bounded independently of the meshwidth. The data structure containing the compressed stiffness matrix is described in detail: it requires $O(\mathcal{N})$ memory and can be set up in $O(\mathcal{N})$ operations. Numerical experiments show that the asymptotic performance estimates apply for moderate N. Problems with $N = 10^5$ degrees of freedom were computed in core on a workstation. The impact of various parameters in the compression scheme on the performance and the accuracy of the algorithm is studied.

1 Introduction

Wavelets have, since their appearance in the mid 80ies, influenced many areas of applied mathematics. After the fundamental work in signal processing and in harmonic analysis (see [10] and the references there) it was pointed out in the paper [1] that wavelet bases might also be very advantageous for the discretization of integral operators. This was later made more precise in [3] where it was shown that a wide class of classical pseudodifferential operators could be discretized by Galerkin-Petrov schemes using test- and trial spaces spanned by (biorthogonal) wavelet bases in such a way that a) the asymptotic convergence rate of the numerical solution was optimal and b) the work and the storage of the discretization scheme was bounded by $O(N(\log N)^a)$ (with a either zero or a small positive integer), see also [12]. This pointed the way to a new class of fast algorithms for integral equations, in addition to the slightly older multipole and panel-clustering algorithms. Wavelet-based schemes for integral equations promise moreover preconditioning of first kind integral equations [3, 12] and a new approach to adaptivity and a-posteriori error estimation [2]. In particular, wavelet based boundary element methods do not require explicit kernel expansions for each problem under consideration. Some technical issues in the construction of practically viable algorithms, such as numerical quadrature, were addressed in [16] where it was shown that fully discrete schemes preserving the above benefits are possible. However, no implementation of an order N wavelet-based scheme for boundary integral equations on general surfaces in $\mathbb{R}^3$ seems to be available yet (the computations reported [12] were based on applying a wavelet transform to the full $N \times N$ stiffness matrix).

To present an object oriented C++ implementation of order $N(\log N)^a$ algorithms and data structures of a wavelet-Galerkin discretization for boundary integral equations on polyhedra and to analyze the performance is the purpose of the present paper. Some of the developments presented here were reported in [9]. We confine the exposition to second kind boundary integral equations or, more generally, to boundary integral operators of order zero and to the fully orthogonal discontinuous multiwavelets introduced in [16, 15]. We give a detailed description of the algorithms necessary to locate the $\mathcal{N} = O(N(\log N)^2)$ nonzero entries in the compressed stiffness matrix and of the data structure to manipulate it. We show that this requires only $O(\mathcal{N})$ operations and $O(\mathcal{N})$ memory. We further elaborate in detail on the implementation of the numerical quadrature scheme introduced in [16] We find in particular that it is essential for a good performance of the scheme to reuse some elementary integrals on various levels judiciously and present a caching-strategy to this end. Based on the theoretical analysis of the compression strategy, our algorithm has various parameters governing the compression threshold. In numerical experiments, we investigate in detail the impact of these parameters upon the performance of the algorithm. We find that the compression rate can be strongly influenced without compromising stability, but that the accuracy of the solution does suffer under too severe compression. In our experiments, problems with $N = O(10^5)$ were tractable without accessing external memory. In addition, we present a new approach to the compression estimates which are based on sharp, derivative-free kernel estimates in the complex domain. They allow to make predictions on the compression of Helmholtz type kernels in dependence on the (nondimensional) wavenumber ω. We find that the well-known reduced compressibility at large wavenumbers ω can be compensated by overcompression, i.e. by using wavelets with a correspondingly high number of vanishing moments.

The outline of the paper is as follows: in Section 2, we present some notation and the class of boundary integral equations which we will consider. Section 3 is then devoted to the Galerkin boundary element method and the construction of the multiwavelets is presented. Further, we give derivative-free decay estimates and explain the compression strategy. In Section 4, we address the numerical integration. Section 5 is devoted to a detailed explanation of the main algorithms used in the implementation of the multiscale scheme. Finally, in Section 6 we discuss numerical results obtained with the presented algorithms.

2 Boundary Integral Equations

Let $D \subset \mathbb{R}^3$ be a bounded domain with a piecewise analytic boundary $\Gamma = \partial D$ which is globally Lipschitz. More precisely, Γ is partitioned into N_Γ open pieces Γ_k which are smooth images of either the triangle $\mathcal{T} = \{ (x_1, x_2) \mid 0 < x_2 < x_1 < 1 \}$ or the square $\mathcal{S} = \{ (x_1, x_2) \mid 0 < x_1, x_2 < 1 \}$ in $\mathbb{R}^2$, i.e. there exist bijective maps χ_k which are analytic in $\overline{\mathcal{U}}$, $\mathcal{U} \in \{\mathcal{T}, \mathcal{S}\}$ such that

$$\Gamma_k = \chi_k(\mathcal{U}).$$

The partition $\{\Gamma_k\}_{k=0}^{N_\Gamma - 1}$ is assumed to be regular, i.e. for $k \neq k'$ the set $\overline{\Gamma}_k \cap \overline{\Gamma}_{k'}$ is either empty, a vertex, or an entire edge. Under these assumptions there exists a Lipschitz atlas of Γ with charts $\tilde{\chi}_k$ which, when restricted to Γ_k, coincide with χ_k.

By $L^2(\Gamma)$ we denote the space of functions $u : \Gamma \to \mathbb{R}$ that are square integrable with respect to the surface measure ds_x. The space $L^2(\Gamma)$, equipped with the inner product

$$\langle u, v \rangle = \int_\Gamma uv ds_x, \tag{2.1}$$

is a Hilbert space. An inner product $(\cdot , \cdot)$ which is equivalent to $\langle \cdot , \cdot \rangle$ (i.e. giving rise to equivalent norms) in $L^2(\Gamma)$ can be defined by

$$(u, v) = \sum_{k=0}^{N_\Gamma - 1} \int_{\mathcal{U}} (u \circ \chi_k)\, (v \circ \chi_k)\, dx_1\, dx_2 \tag{2.2}$$

For $s > 0$, we consider also the Sobolev spaces $H^s(\Gamma_k)$ of functions with pullback in $H^s(\mathcal{U})$ endowed with the norm $\| \cdot \|_{H^s(\Gamma_k)}$. The space of functions $u \in L^2(\Gamma)$ with $u|_{\Gamma_j} \in H^s(\Gamma_k)$ for $s > 0$ is denoted by $\prod_{k=0}^{N_\Gamma - 1} H^s(\Gamma_k)$. Evidently, the expression

$$\|u\|_s := \left(\sum_{k=0}^{N_\Gamma - 1} \|u\|^2_{H^s(\Gamma_k)} \right)^{1/2} \tag{2.3}$$

is a norm in $\prod_{k=0}^{N_\Gamma - 1} H^s(\Gamma_k)$.

For a given $f \in L^2(\Gamma)$, we consider the variational boundary integral equation

$$u \in L^2(\Gamma) \qquad \langle v, Au \rangle = \langle v, f \rangle \qquad \forall v \in L^2(\Gamma). \tag{2.4}$$

For the integral operator A, we suppose the following.

Assumption 1 *$A : L^2(\Gamma) \to L^2(\Gamma)$ is continuous and injective.*

Assumption 2 *A admits the representation via a kernel k:*

$$(Au)(x) := c(x)u(x) + \int_\Gamma k(x,y)u(y)ds_y \qquad x \in \Gamma, \tag{2.5}$$

where $k(x,y) \in \mathrm{Lip}((\Gamma \times \Gamma)\backslash\{x = y\})$ *is an analytic function of* $x \in \overline{\Gamma}_k$, $y \in \overline{\Gamma}_{k'}$, $0 \leq k, k' < N_\Gamma$ *which admits an analytic extension for* x *in some neighborhood of* $\overline{\Gamma}_k$ *resp. for* y *in some neighborhood of* $\overline{\Gamma}_{k'}$. *We furthermore assume that the kernel satisfies the bound*

$$|k(x,y)| \leq \frac{C}{|x-y|^2} \tag{2.6}$$

for every x, y *in some complex neighborhood of each boundary piece* $\overline{\Gamma}_k$.

The integral in (2.5) must be taken in the Cauchy Principal Value sense, in general, and $c(x)$ is assumed to be positive and analytic on Γ_k, i.e. A is formally a Fredholm integral operator of the second kind. Further, D_x^α, D_y^β are Cartesian derivatives in $\mathbb{R}^3$ acting on a smooth extension of $k(x,y)$ to tubular neigborhoods in $\mathbb{R}^3$ of each surface piece Γ_k.

Remark 3 Since the charts χ_k are assumed to be bijective and analytic, there exists a constant $\gamma > 0$ depending only on χ_k and the shape of Γ such that

$$0 < \gamma^{-1} \leq \frac{|\chi_k(u) - \chi_k(u')|}{|u - u'|} \leq \gamma \tag{2.7}$$

for $u, u' \in \overline{\mathcal{U}}$, $u \neq u'$.

Remark 4 We point out that in our compression estimates we only require the analyticity of k and the charts χ_k and the estimate (2.6), but *not* the estimates

$$\left|D_x^\alpha D_y^\beta k(x,y)\right| \leq \frac{C_{\alpha\beta}}{|x-y|^{2+|\alpha|+|\beta|}}$$

for any multiindices $\alpha, \beta \in \mathbb{N}_0^3$ which are used, for example, in [3, 16, 12]. The analyticity assumptions are usually satisfied in applications for boundary integral equations and are also underlying the quadrature error analysis, see [16].

3 Wavelet Galerkin Discretization

We present now a fully discrete wavelet Galerkin discretization of (2.4) together with its properties. Full proofs for all assertions can be found in [15, 16].

3.1 Galerkin Discretization

We will be interested in Galerkin discretizations of (2.4). Given a dense subspace sequence $\{V^L\}_{L=0}^\infty$ in $L^2(\Gamma)$, the Galerkin approximations u^L of u are defined by

$$u^L \in V^L \qquad \left\langle v, Au^L \right\rangle = \langle v, f \rangle \qquad \forall v \in V^L. \tag{3.1}$$

By introducing a basis $\{\varphi_j^L\}$ for the subspace V^L the vector of coefficients $\mathbf{u}^L = (u_j^L)$ of the discrete solution u^L is determined by the linear system

$$\mathbf{A}^L \mathbf{u}^L = \left(\mathbf{M}^L + \mathbf{K}^L\right) \mathbf{u}^L = \mathbf{f}^L \tag{3.2}$$

where mass- and stiffness matrix are given by

$$(\mathbf{M}^L)_{jj'} := \left\langle \varphi_j^L, c\, \varphi_{j'}^L \right\rangle = \int_\Gamma c(x) \varphi_j^L(x) \varphi_{j'}^L(x)\, ds_x, \tag{3.3}$$

$$(\mathbf{K}^L)_{jj'} := \left\langle \varphi_j^L, K \varphi_{j'}^L \right\rangle = \int_\Gamma \int_\Gamma k(x,y) \varphi_j^L(x) \varphi_{j'}^L(y)\, ds_y ds_x \tag{3.4}$$

and the right hand side by $\mathbf{f}^L := \left(\left\langle \varphi_j^L, f \right\rangle\right)$.

We make the *assumption* that the Galerkin scheme (3.1) is *stable* in the following sense:

Assumption 5 *For L_0 sufficiently large, there exists $c > 0$ such that*

$$\forall L \geq L_0 : \qquad \inf_{u \in V^L/\{0\}} \sup_{v \in V^L/\{0\}} \frac{\langle v, Au \rangle}{\|u\|_0 \|v\|_0} \geq c > 0. \tag{3.5}$$

The stability (3.5) implies that for sufficiently large L, the Galerkin solutions u^L of (3.1) exist and are quasioptimal [6], i.e.

$$\|u - u^L\|_0 \leq C \inf_{v \in V^L} \|u - v\|_0. \tag{3.6}$$

Remark 6 The stability (3.5) of the Galerkin discretization (3.1) is not self-evident, since we consider equations of the second kind. There are two basic situations in which (3.5) does hold, however. In the first case, we assume that the surface Γ is smooth, i.e. C^∞. In this case, the operators A arising from the boundary reduction of elliptic boundary value problems in Ω are in fact classical pseudodifferential operators and the stability (3.5) follows from a Gårding inequality in $L^2(\Gamma)$ for these operators. The validity of a Gårding inequality has been obtained for several second kind integral operators of mathematical physics [17].

The second case occurs for Γ being the boundary of a polyhedron. Here a Gårding inequality in $L^2(\Gamma)$ does in general not hold. Nevertheless, stability of (3.1) has been shown by other means in [5] under the provision that V^L is constrained to be zero in an $O(h)$ vicinity of the edges of Γ.

It has been observed in numerous numerical experiments, however, that on polyhedra (3.5) apparently holds even without the zero constraint, although a proof does not seem to be available. We assume (3.5) in our analysis.

3.2 Multiwavelet Basis

In order to define a dense subspace sequence $\{V^L\}_{L=0}^\infty$ by a multiscale scheme we construct a hierarchy of meshes on the manifold Γ as follows: divide $\mathcal{U}$ into four congruent subdomains $\mathcal{U}^i$, $1 \leq i \leq 4$, by successively halving the sides. The subdomains can be expressed by translations and scalings τ_i of the reference domain $\mathcal{U}$:

$$\tau_i : \mathbb{R}^2 \mapsto \mathbb{R}^2, \quad \text{with} \quad \mathcal{U}^i = \tau_i(\mathcal{U}). \tag{3.7}$$

Applying τ_i recursively yields a hierarchy of meshes $\{\mathcal{M}^L\}_{L=0}^{\infty}$ on the reference domain $\mathcal{U}$ which is lifted on the manifold Γ in order to define meshes M^L on the boundary:

$$\mathcal{M}^0 := \{\mathcal{U}\}, \qquad \mathcal{M}^L := \left\{\tau_i(\mathcal{U}) : \mathcal{U} \in \mathcal{M}^{L-1}, 1 \le i \le 4\right\}$$

$$M^L := \left\{U_j^L : 0 \le j < 4^L N_\Gamma\right\} := \left\{\chi_k(\mathcal{U}) : \mathcal{U} \in \mathcal{M}^L, 0 \le k < N_\Gamma\right\} \tag{3.8}$$

The subspaces V^L are constructed in the same manner namely by defining the structure in local coordinates and lifting them on the boundary. Let

$$\mathbb{P}_d(\mathcal{U}) := \operatorname{span}\{\xi^\nu : \xi \in \mathcal{U}, \nu \in \mathbb{N}_0^2 \text{ with } |\nu| \le d\}$$

be the space of polynomials of total degree d. Then define the spaces:

$$\mathcal{V}^0 := \mathbb{P}_d(\mathcal{U}), \quad \mathcal{V}^L := \left\{\varphi \in L^2(\mathcal{U}) : \forall\, i \in \{1,\dots,4\} \quad \varphi \circ \tau_i \in \mathcal{V}^{L-1}\right\} \quad \text{for } L > 0,$$

$$\mathcal{W}^0 := \mathcal{V}^0, \quad \mathcal{W}^L := \left\{\psi \in \mathcal{V}^L : \forall\, \varphi \in \mathcal{V}^{L-1} \quad \langle\varphi,\psi\rangle = 0\right\} \quad \text{for } L > 0.$$

The space $\mathcal{V}^L$ is derived by a replication of the space $\mathcal{V}^{L-1}$ in each of the four subdomains $\mathcal{U}^i$. Therefore, the spaces $\mathcal{V}^L$ form a hierarchy, i.e.

$$\mathcal{V}^0 \subset \mathcal{V}^1 \subset \cdots \subset \mathcal{V}^{L-1} \subset \mathcal{V}^L \subset \cdots \tag{3.9}$$

with $\mathcal{N}_L := \dim \mathcal{V}^L = 4^L \dim \mathbb{P}_d(\mathcal{U})$.

The subspaces $\mathcal{W}^L$ are for $L > 0$ the $L^2(\mathcal{U})$-orthogonal complements of $\mathcal{V}^{L-1}$ in $\mathcal{V}^L$ such that we obtain the multilevel splitting

$$\mathcal{V}^L := \mathcal{W}^0 \oplus \mathcal{W}^1 \oplus \cdots \oplus \mathcal{W}^L. \tag{3.10}$$

To construct an orthonormal basis for $\mathcal{V}^L$ we proceed in the following way: obtain, for example by applying the Gram-Schmidt process, orthonormal bases $\tilde{\Psi}^0$ and $\tilde{\Psi}^1$ for the subspaces $\mathcal{W}^0$ and $\mathcal{W}^1$ respectively:

$$\tilde{\Psi}^0 := \left\{\tilde{\psi}_j^0 : 0 \le j < \mathcal{N}_0\right\}, \quad \tilde{\Psi}^1 := \left\{\tilde{\psi}_j^1 : 0 \le j < \mathcal{N}_1 - \mathcal{N}_0\right\}.$$

Subsequently, derive via translation and scaling orthonormal bases $\tilde{\Psi}^L$ for the remaining spaces $\mathcal{W}^L$, $L > 1$:

$$\begin{aligned}\tilde{\Psi}^L &:= \left\{\tilde{\psi}_j^L : 0 \le j < \mathcal{N}_L - \mathcal{N}_{L-1}\right\} \\ &= \left\{2^{L-1}\tilde{\psi} \in L^2(\mathcal{U}) : \exists\, 1 \le i \le 4 \quad \tilde{\psi} \circ \tau_i \in \tilde{\Psi}^{L-1} \text{ and } \operatorname{supp}\tilde{\psi} \subset \overline{\mathcal{U}^i}\right\}.\end{aligned}$$

Then

$$\mathcal{V}^L = \operatorname{span} \bigcup_{l=0}^{L} \tilde{\Psi}^l$$

defines an orthonormal basis of $\mathcal{V}^L$.

Finally, we lift the subspaces and the multiwavelet bases onto the manifold Γ and obtain the desired dense sequence of subspaces $\{V^L\}_{L=0}^{\infty}$ in $L^2(\Gamma)$:

$$V^L := \left\{\varphi \in L^2(\Gamma) : \varphi \circ \chi_k \in \mathcal{V}^L \quad \forall\, k \in \{0,\dots,N_\Gamma - 1\}\right\} \tag{3.11}$$

$$W^L := \left\{\psi \in L^2(\Gamma) : \psi \circ \chi_k \in \mathcal{W}^L \quad \forall\, k \in \{0,\dots,N_\Gamma - 1\}\right\} \tag{3.12}$$

and

$$\begin{aligned}\Psi^L &:= \left\{\psi_j^L :\ 0 \le j < N_L - N_{L-1}\right\} \\ &= \left\{\psi \in L^2(\Gamma) :\ \exists\, 0 \le k < N_\Gamma \quad \psi \circ \chi_k \in \tilde{\Psi}^L \text{ and } \operatorname{supp}\psi \subset \overline{\Gamma_k}\right\}\end{aligned} \tag{3.13}$$

where $N_L := N_\Gamma \mathcal{N}_L$, $N_{-1} := 0$. The spaces W^L remain orthogonal complements of V^{L-1} in V^L but now with respect to the inner product (2.2). In this sense the multilevel splitting (3.10) is preserved such that $(\psi_j^l, \psi_{j'}^{l'}) = \delta_{ll'}\delta_{jj'}$ for all $l, l' \in \mathbb{N}_0$, i.e.

$$\bigcup_{l=0}^{L} \Psi^l = \left\{\psi_j^l :\ 0 \le j < N_l - N_{l-1},\ 0 \le l \le L\right\} \tag{3.14}$$

is an orthonormal basis of V^L. Accordingly, $\|u\|_{L^2(\Gamma)}$ can be characterized by wavelet expansion coefficients:

Proposition 7 *For every $u \in L^2(\Gamma)$, there holds*

$$\|u\|^2_{L^2(\Gamma)} \sim \sum_{l=0}^{\infty} \sum_{j=0}^{N_l - N_{l-1} - 1} \left|(u, \psi_j^l)\right|^2 \tag{3.15}$$

and

$$\sum_{l=0}^{L} \sum_{j=0}^{N_l - N_{l-1} - 1} \left|(u, \psi_j^l)\right|^2 \sim \sum_{l=0}^{L} \sum_{j=0}^{N_l - N_{l-1} - 1} \left|u_j^l\right|^2 \tag{3.16}$$

where $\sim$ denotes the equivalence of norms and (u_j^l) the vector of coefficients of the $L^2(\Gamma)$-orthogonal projection of u on V^L with respect to the basis (3.14)*. The equivalence constants depend on the geometry Γ and are independent of the level L.*

Remark 8 The equivalence (3.16) depends on the polynomial degree d in general. If D is a straight-sided polyhedron in $\mathbb{R}^3$, there is obviously no dependence on d. If, on the other hand, the Γ_k are curved, the d-dependence can be eliminated if the inner product $\langle\cdot,\cdot\rangle$ in (3.1) is changed to $(\cdot,\cdot)$. The equivalence (3.15) is always independent of d.

Moreover, due to the multiscale decomposition (3.10) the basis functions ψ_j^l, which we will call *multiwavelets* in what follows, have vanishing moments of order $d+1$ in local coordinates:

$$\int_{\mathcal{U}} (\psi_j^l \circ \chi_k)(\xi)\, \xi^\nu \, d\xi = 0 \quad \text{for } |\nu| \le d,\ l > 0 \tag{3.17}$$

which implies the smallness of certain entries in the stiffness matrix (see Section 3.3).

We employ the multiwavelet basis (3.14) in the Galerkin equation (3.1) to determine the discrete solutions $u^L \in V^L$ where the stiffness matrix $\mathbf{K}^L$ is given by

$$(\mathbf{K}^L)_{(l,j)(l',j')} := \int_\Gamma \int_\Gamma k(x,y)\psi_j^l(x)\psi_{j'}^{l'}(y)\, ds_y ds_x \quad \text{for } 0 \le l, l' \le L.$$

Note that $\mathbf{K}^L$ is not symmetric in general. It follows from the stability (3.5) and the norm equivalence (3.15) that the condition numbers of the sequence $\{\mathbf{A}^L\}$ of matrices are bounded:

Proposition 9 *There exists $\kappa \in \mathbb{R}$ such that for all L*

$$\operatorname{cond}_2(\mathbf{A}^L) \le \kappa. \tag{3.18}$$

3.3 Compression

The vanishing moment property (3.17) of the wavelet basis $\{\psi_I\}$ defined in (3.14) implies with (2.6) that many of the entries in $\mathbf{K}^L$ are small.

Proposition 10 *Let $S_j^l := \operatorname{supp}\psi_j^l$. The entries $(\mathbf{K}^L)_{(l,j)(l',j')}$ with $d_{jj'}^{ll'} := \operatorname{dist}(S_j^l, S_{j'}^{l'}) > 0$ satisfy*

$$\left|(\mathbf{K}^L)_{(l,j)(l',j')}\right| \le C\, M(k,\Gamma)\,(c(d))^2\,\left(\gamma d_{jj'}^{ll'}\right)^{-2(d+2)}\,2^{-(d+2)(l+l')} \tag{3.19}$$

where C is independent of $d, l, l', j, j', k, \Gamma$, $c(d) = (d+1)(1+\log(d+1))$ and $M(k,\Gamma)$ denotes the maximum of the analytic continuation of $k(\chi_k(\xi), \chi_{k'}(\eta))$ into a neighborhood of $\xi \in \mathcal{U}$ and $\eta \in \mathcal{U}$ in $\mathbb{C}^2$.

Decay estimates like (3.19) are well-known (see, e.g., [10, 3, 12]) to hold for all classical pseudodifferential operators on smooth manifolds. We give here a new proof a) yielding stronger estimates than (3.19) for kernels $k(x,y)$ which are analytic off the diagonal $\{x = y\}$, b) which is *derivative-free*, i.e. the proof does not involve derivatives of $k(x,y)$ and of the surface parametrization. The estimates are therefore more explicit in kernel parameters such as the wavenumber for time-harmonic vibrations and c) yielding bounds which are explicit in the polynomial degree d and the number of vanishing moments.

3.3.1 Some Approximation Results

The proof of the decay estimate (3.19) is based on polynomial approximation results for analytic functions. We start with some notation.

For the interval $I = (-1,1) \subset \mathbb{R}$, we consider the ellipse $\mathcal{E}_\rho \subset \mathbb{C}$ with foci in $z = \pm 1$ and semiaxis sum $\rho > h$. In dimension $n > 1$, we denote by $\mathcal{E}_\rho^n \subset \mathbb{C}^n$ the polycylinder corresponding to $(-1,1)^n$.

The following approximation result is essential.

Lemma 11 *Let $f(x)$ be analytic in $I = [-1,1]$ and let f admit an analytic extension (again denoted by $f(z)$) to the closed ellipse $\mathcal{E}_{r_0} \subset \mathbb{C}$ with foci at $x = \Re z = \pm 1$ and with semiaxis sum $r_0 > 1$. Then there holds the error estimate*

$$\inf_{\varphi \in P_d(I)} \|f - \varphi\|_{L^\infty(I)} \le c(d+1) r^{-(d+1)} \left(1 - r^{-2}\right)^{-1} \max_{z \in \partial\mathcal{E}_r} |f(z)| \tag{3.20}$$

for $1 < r \le r_0$ and $d = 0, 1, 2, \ldots$. The constant c is independent of r, r_0, f and d.

Proof:

This follows from classical approximation results, see, e.g., [14] and by using the embedding theorem. □

For the kernel approximation we require a two-dimensional version of this result which we derive by tensor product construction. To this end, let $\Pi_d f(x)$ denote the polynomial of degree d in I interpolating $f(x)$ at the Lobatto points. For bivariate functions f of $x = (x_1, x_2)$, denote by $\Pi_d^i f$ the interpolant with respect to x_i, $i = 1, 2$.

Lemma 12 *Let $f(x_1, x_2)$ be analytic in $[-1,1]^2$ and admit, for fixed $x_1 \in I$, an analytic continuation with respect to $x_2 \in \mathcal{E}_{r_0} \subset \mathbb{C}$ and vice versa. Then, for $1 < r \le r_0$ there holds*

$$\left\| f - \Pi_d^1 \Pi_d^2 f \right\|_{L^\infty(I\times I)} \le c(d) r^{-(d+1)} (1 - r^{-2})^{-1} M_r(f) \tag{3.21}$$

where $c(d) := c(d+1)(1+\log(d+1))$ *with* c *independent of* r_0, f *and* d *and with*

$$M_r(f) := \max_{x_1 \in I} \max_{x_2 \in \partial\mathcal{E}_r} |f(x_1,x_2)| + \max_{x_2 \in I} \max_{x_1 \in \partial\mathcal{E}_r} |f(x_1,x_2)| \tag{3.22}$$

Proof:

We write

$$\left\| f - \Pi_d^1 \Pi_d^2 f \right\|_{L^\infty(I\times I)} \leq \left\| f - \Pi_d^1 f \right\|_{L^\infty(I\times I)} + \left\| \Pi_d^1 \left(f - \Pi_d^2 f \right) \right\|_{L^\infty(I\times I)}.$$

For the first term, we use Lemma 11 directly. For the second term, we use the L^∞-stability of Π_d, i.e.

$$\forall g \in C^0(I): \quad \|\Pi_d g\|_{L^\infty(I)} \leq C\,(1+\log(d+1))\,\|g\|_{L^\infty(I)}$$

and again Lemma 11 to conclude. □

Below we denote by $\hat{\Pi}_d = \Pi_d^1 \Pi_d^2$ the tensor product interpolant and write $\hat{\Pi}_d^{(x)} f$ when $f(x,y)$ is interpolated with respect to x etc.

3.3.2 Proof of the decay estimates

In the remainder of this section, we assume that

$$d_{jj'}^{ll'} := \operatorname{dist}\left(S_j^l, S_{j'}^{l'}\right) > 0 \tag{3.23}$$

with $S_j^l := \operatorname{supp} \psi_j^l$. We will prove the decay estimate (3.19) the above approximation results to the kernel $k(x,y)$ in local coordinates, i.e. to $\tilde{k}(\xi,\eta)$. We recall that

$$\begin{aligned} \left(\mathbf{K}^L\right)_{(l,j)(l',j')} &= \int_\Gamma \int_\Gamma k(x,y)\psi_j^l(x)\psi_{j'}^{l'}(y)ds_x ds_y \\ &= 2^{l+l'-2-2(l-1)-2(l'-1)} \int_{\mathcal{U}} \int_{\mathcal{U}} \tilde{k}_{(k,l)(k',l')}(\xi,\eta)\tilde{\psi}_j^1(\xi)\tilde{\psi}_{j'}^1(\eta)d\xi d\eta \end{aligned} \tag{3.24}$$

Here the kernel $\tilde{k}_{(k,l)(k',l')}(\xi,\eta)$ is the composition of $k(x,y)\,|ds_x|\,|ds_y|$ with χ_k and $\chi_{k'}$, respectively, and with a translation in the parameter domain $\mathcal{U}$ and a dilation by 2^{l-1} resp. by $2^{l'-1}$ (the scaling factor in (3.24) results from the normalization of the wavelets and the Jacobian of this scaling). Henceforth we omit the subscripts and write simply $\tilde{k}$.

We use now the vanishing moment property (3.17) and Fubini's theorem and write

$$\begin{aligned} \left(\mathbf{K}^L\right)_{(l,j)(l',j')} &= \int_\Gamma \int_\Gamma k(x,y)\psi_j^l(x)\psi_{j'}^{l'}(y)ds_y ds_x \\ &= C2^{-l-l'} \int_{\mathcal{U}} \tilde{\psi}_j^1(\xi) \int_{\mathcal{U}} \tilde{\psi}_{j'}^1(\eta) \left[\tilde{k}(\xi,\eta) - \varphi'(\xi;\eta)\right] d\eta d\xi \\ &= C2^{-l-l'} \int_{\mathcal{U}} \tilde{\psi}_{j'}^1(\eta) \int_{\mathcal{U}} \tilde{\psi}_j^1(\xi) \left[\tilde{k}(\xi,\eta) - \varphi'(\xi;\eta) - \varphi(\xi;\eta)\right] d\xi d\eta \end{aligned} \tag{3.25}$$

where $\varphi(\xi;\eta)$ is an *arbitrary* polynomial of ξ in $\mathbb{P}_d(\mathcal{U})$ with coefficients depending on η and $\varphi'(\xi;\eta)$ is a polynomial of η in $\mathbb{P}_d(\mathcal{U})$ with coefficients depending on ξ. The decay estimates will follow from the following approximation error estimates.

Proposition 13 *Assume* (3.23). *Then there exists* $\gamma > 0$ *depending only on* Γ *and the domains of analyticity of* $k(x,y)$ *and of the parametric representations* χ_k *such that*

i) for every $\xi \in \mathcal{U}$, $\tilde{k}(\xi,\eta)$ in (3.25) is a real analytic function of $\eta \in \overline{\mathcal{U}}$ and it admits an analytic continuation to the bicylinder $\mathcal{E}^2_{\rho'} \subset \mathbb{C}^2$ with semiaxis sum ρ' satisfying the bound

$$\rho' \geq \rho(S^{l'}_{j'}, S^l_j) := 1 + \gamma 2^{l'} d^{ll'}_{jj'}. \tag{3.26}$$

ii) Conversely, for every $\eta \in \overline{\mathcal{U}}$, $\tilde{k}(\xi,\eta)$ in (3.25) is real analytic in $\xi \in \overline{\mathcal{U}}$ and admits an analytic continuation to $\mathcal{E}^2_{\rho} \subset \mathbb{C}^2$ with semiaxis sum ρ satisfying

$$\rho \geq \rho(S^l_j, S^{l'}_{j'}) := 1 + \gamma 2^{l} d^{ll'}_{jj'}. \tag{3.27}$$

Moreover, there hold the estimates

$$M_{\rho,\rho'}(\tilde{k}) := \max_{\xi \in \partial\mathcal{E}^2_\rho} \max_{\eta \in \partial\mathcal{E}^2_{\rho'}} \left|\tilde{k}(\xi,\eta)\right| \leq M \left(d^{ll'}_{jj'}\right)^{-2} \tag{3.28}$$

where the constant M depends only on the analytic continuation of the kernel $\tilde{k}$.

Proof:

We first establish the analyticity of the kernel in local coordinates, i.e. of $k(\chi_k(\cdot), \chi_{k'}(\cdot))$. This is immediate from the assumed analyticity of the kernel k, Assumption 2, and from the analyticity of the charts χ_k. The size of the domains of analyticity is $\mathcal{E}_\rho$ with $\rho = 1 + \gamma d^{ll'}_{jj'}$ for some $\gamma > 0$ by Assumption 2 on the kernel (cf. also Remark 3). Now $\tilde{k}$ is obtained from $k(\chi_k(\cdot), \chi_{k'}(\cdot))$ by scaling the variables ξ and η by 2^l resp. by $2^{l'}$ which implies (3.26), (3.5), since the domains of analyticity scale analogously. The estimate (3.28) follows as Lemma 5 in [16]. □

We are now in position to prove the main result of this section which will imply Proposition 10.

Theorem 14 *With ρ, ρ' as in (3.26), (3.27), there holds the decay estimate*

$$\left|\left(\mathbf{K}^L\right)_{(l,j)(l',j')}\right| \leq C 2^{-l-l'} (c(d))^2 \left(\rho\rho'\right)^{-(d+1)} \left(1-\rho^{-2}\right)^{-1} \left(1-(\rho')^{-2}\right)^{-1} M_{\rho,\rho'}(\tilde{k}). \tag{3.29}$$

Here the constant C is independent of ρ, ρ', d and $\tilde{k}$.

Proof:

We apply Lemma 12 to (3.25) and select

- for every ξ the function $\varphi'(\xi,\eta) := \hat{\Pi}^{(\eta)}_d \tilde{k}(\xi,\eta)$ as tensor product polynomial interpolant of degree d of $\tilde{k}(\xi,\eta)$ with respect to η, and
- for every η the function $\varphi(\xi,\eta) := \hat{\Pi}^{(\xi)}_d \left(\tilde{k}(\xi,\eta) - \varphi'(\xi,\eta)\right)$ as tensor product polynomial interpolant of degree d of $\tilde{k}(\xi,\eta) - \varphi'(\xi,\eta)$ with respect to ξ.

Since $\varphi'(\xi;\eta)$ is, for every η, a linear combination of kernel evaluations in sampling points η_{ij}, $\varphi'(\xi;\eta)$ is an analytic function of ξ with the same domain of analyticity in ξ as $\tilde{k}$, and so is therefore $\tilde{k} - \varphi'$. We may therefore apply Lemma 12 to the difference

$$\tilde{k}(\cdot,\eta) - \varphi'(\cdot;\eta),$$

resulting in the bound

$$\begin{aligned} &\left\| \left(\tilde{k}(\cdot,\eta) - \varphi'(\cdot;\eta) \right) - \varphi(\cdot,\eta) \right\|_{L^\infty(I\times I)} \\ &\leq c(d)\rho^{-(d+1)}(1-\rho^2)^{-1} M_\rho\left(\tilde{k}(\cdot,\eta) - \varphi'(\cdot;\eta) \right) \quad \forall \eta \in \mathcal{U} \end{aligned} \tag{3.30}$$

with M_ρ as in (3.22).

Next, we consider for every $\xi \in \partial\mathcal{E}_\rho^2$:

$$\left\| \tilde{k}(\xi,\cdot) - \varphi'(\xi;\cdot) \right\|_{L^\infty(I\times I)}.$$

Since $\tilde{k}(\xi,\eta)$ is seperately analytic in each variable by Proposition 13 and continuous, it is jointly analytic in (ξ,η).

We may therefore apply Lemma 12 once more to estimate the polynomial approximation error in η of $\tilde{k}(\xi,\eta) - \hat{\Pi}_d^{(\eta)}\tilde{k}(\xi,\eta)$ for $\xi \in \partial\mathcal{E}_\rho^2$:

$$\left\| \tilde{k}(\xi,\cdot) - \varphi'(\xi;\cdot) \right\|_{L^\infty(I\times I)} \leq c(d)\,(\rho')^{-(d+1)} \left(1-(\rho')^{-2}\right)^{-1} M_{\rho'}(\tilde{k}(\xi,\cdot)) \tag{3.31}$$

Combining (3.31) and (3.30) gives the assertion. □

In order to deduce (3.19) from Theorem 14 we note that

$$\rho^{-(d+1)}\left(1-\rho^{-2}\right)^{-1} = \frac{\rho^{1-d}}{(\rho+1)(\rho-1)} \leq \frac{\rho^{-d}}{\gamma 2^l d_{jj'}^{l\,l'}} \leq \left(\gamma 2^l d_{jj'}^{l\,l'}\right)^{-(d+1)}. \tag{3.32}$$

Combining (3.32) and an analogous bound with respect to ρ' with (3.28) in (3.29) proves (3.19).

Remark 15 In the case of the double layer potential of the Helmholtz-kernel

$$k(x,y) = \frac{\partial}{\partial n_y}\left(\frac{e^{i\omega|x-y|}}{4\pi\,|x-y|}\right)$$

where $\omega \in \mathbb{R}$ is the dimensionless wavenumber, we have in (3.28) the bound

$$M_{\rho,\rho'}(\tilde{k}) \leq C(\Gamma)\,|\omega|\exp(c\,|\omega|)$$

Here c is an absolute constant, depending only on the global regularity of the surface Γ. Inserting this into (3.19), we find after possibly changing c the compression estimates

$$\left|(\mathbf{K}^L)_{(l,j)(l',j')}\right| \leq C\,M(\Gamma)\,(c(d))^2\,\left(\gamma d_{jj'}^{l\,l'}\right)^{-2(d+2)}\,2^{c|\omega|-(d+2)(l+l')}$$

Assuming $c \sim 1$, we see that the compression can only take place when $(d+2)(l+l') \geq |\omega|$. This underlines the advantage of a higher number of vanishing moments for such problems. For example, if the number of vanishing moments is equal to $|\omega|$, one can expect to obtain again the usual compression behavior (notice that a large number of vanishing moments does not cause larger supports in our setting and that a high polynomial degree is also advantageous for the solution approximation in Helmholtz problems).

3.3.3 Matrix truncation

The truncation strategy for the stiffness matrix in the wavelet basis is now as follows:

$$(\tilde{\mathbf{K}}^L)_{(l,j)(l',j')} := \begin{cases} (\mathbf{K}^L)_{(l,j)(l',j')} & \text{if } \operatorname{dist}(S_j^l, S_{j'}^{l'}) \le \tau_{ll'} \\ 0 & \text{otherwise} \end{cases} \tag{3.33}$$

where $(\tau_{ll'})$ is a matrix of truncation parameters at our disposal.

For a given selection of the truncation parameters $\tau_{ll'}$, we solve instead of (3.2) the perturbed system

$$\tilde{\mathbf{A}}^L \tilde{\mathbf{u}}^L = \left(\mathbf{M}^L + \tilde{\mathbf{K}}^L\right) \tilde{\mathbf{u}}^L = \mathbf{f}^L \tag{3.34}$$

and denote by

$$\tilde{u}^L = \sum_{l=0}^{L} \sum_j \tilde{u}_j^L \psi_j^l \in V^L$$

the corresponding approximate solution. The multiwavelets introduced above exist for any approximation order d. They are fully orthogonal and their support does not increase with d. This is paid for by their discontinuity which implies an increased dimension of V^L as compared to, for example, smoothest splines of the same degree d on the same mesh. For example, for $d = 1$, i.e. piecewise (bi)linear functions, the space V^L of discontinuous functions has, roughly speaking, six times as many degrees of freedom as the corresponding continuous trial space while giving the same asymptotic convergence rates (even though the constant in the convergence estimate is certainly smaller). It is therefore not clear if a Galerkin BEM based on continuous trial spaces and, for example, the panel-clustering approach, will outperform the multiwavelet Galerkin scheme with $d = 1$ in terms of accuracy versus CPU-time. If so, special, biorthogonal wavelet bases for spaces of continuous functions must be employed.

A special case occurs for $d = 0$, i.e. for piecewise constants. This corresponds to the classical panel method, resp. to the Haar wavelet. Here the dimension of the trial space is not increased by using the wavelet basis. Notice, however, that we have only vanishing mean then and equation (3.19) indicates that higher order vanishing moments imply corresponding faster decay. It is therefore often argued that the piecewise constant Haar multiwavelets ($d = 0$) are unsuitable as basis functions since the decay (3.19) resulting from (3.17) with $d = 0$ is too weak. This argument arose in the context of the so-called ϵ-truncation advocated in [1], but is, as we shall see, completely unjustified. What is essential is the proper number of vanishing moments with respect to the approximation order of V^l and the order of the operator A (see [3] for an analysis of operators of nonzero order). Indeed, in our case the decay (3.19) is sufficient to allow the approximation of the full Galerkin matrix $\mathbf{A}^L$ by the compressed one $\tilde{\mathbf{A}}^L$ with $\mathcal{N} = O(N_L(\log N_L)^2)$ essential, nonvanishing elements so that the asymptotic convergence rate of the Galerkin scheme (2.4) is not reduced, as we will now show.

3.4 Asymptotic Complexity and Convergence

In the following Theorem we collect the main results on convergence and complexity of the compressed Galerkin scheme.

Theorem 16 *Let $s, s' \in [0, d+1]$ and assume that the truncation parameters $\tau_{ll'}$ in (3.33) are given by*

$$\tau_{ll'} = a\, 2^{\alpha\frac{L-l}{2}+\alpha'\frac{L-l'}{2}-\frac{l+l'}{2}} \tag{3.35}$$

with $a > 0$ to be selected and

$$\frac{s}{d+1} \le \alpha \le 1, \quad \frac{s'}{d+1} \le \alpha' \le 1. \tag{3.36}$$

Then the following holds:

1. *For sufficiently large a there exists a level L_0 such that for sufficiently large a in (3.35) the compressed Galerkin scheme is stable for every $\alpha, \alpha' \in [0,1]$, i.e.*

$$\forall L \ge L_0 : \qquad \inf_{0\neq v^L\in V^L}\sup_{0\neq u^L\in V^L} \frac{\left\langle v^L, \tilde{A}^L u^L\right\rangle}{\|u^L\|_0\|v^L\|_0} \ge c > 0 \tag{3.37}$$

   where $\tilde{A}^L : V^L \to (V^L)'$ denotes the finite-dimensional operator associated to the compressed system matrix $\tilde{\mathbf{A}}^L := \mathbf{M}^L + \tilde{\mathbf{K}}^L$.

2. *Given the regularity $u, f \in H^s(\Gamma)$ and L sufficiently large yields for any $\alpha' \in [0,1]$ the error estimate*

$$\|u - \tilde{u}^L\|_0 \le C N_L^{-\frac{s}{2}} (\log N_L)^\nu \|u\|_s = C h^s |\log h|^\nu \|u\|_s \tag{3.38}$$

   with

$$\nu = \begin{cases} 0 & \frac{s}{d+1} < \alpha \le 1, \\ \frac{1}{2} \quad \textit{for} & \frac{s}{d+1} = \alpha < 1, \\ 1 & \frac{s}{d+1} = \alpha = 1. \end{cases} \tag{3.39}$$

3. *Let $g \in H^{s'}(\Gamma)$ and denote by $\varphi \in L^2(\Gamma)$ the solution of the adjoint equation $A^*\varphi = g$. Assume the regularity $\varphi \in H^{s'}(\Gamma)$ and $u \in H^s(\Gamma)$ and that (3.36) holds. Then, for sufficiently large L we have the error estimate*

$$\begin{aligned} \left|\left\langle g, u - \tilde{u}^L\right\rangle\right| &\le C N_L^{-\frac{s+s'}{2}} (\log N_L)^{\nu+\nu'} \|u\|_s \|g\|_{s'} \\ &= C h^{s+s'} |\log h|^{\nu+\nu'} \|u\|_s \|g\|_{s'} \end{aligned} \tag{3.40}$$

   with ν as in (3.39) and ν' defined analogously in terms of s'.

4. *The number $\mathcal{N}$ of nonzero entries in $\tilde{\mathbf{K}}^L$ is of order $O(N_L(\log N_L)^2)$ if $\alpha = \alpha' = 1$ and $O(N_L \log N_L)$ otherwise.*

The proof of this theorem is analogous to the results in [15, 3, 16] and therefore omitted. We note, analogous to (3.18), that as an immediate consequence of the stability (3.37) and the norm equivalence (3.15) we obtain the boundedness of the condition numbers of the compressed stiffness matrices:

$$\mathrm{cond}_2(\tilde{\mathbf{A}}^L) \le \tilde{\kappa}^*. \tag{3.41}$$

Remark 17 We observe that for $\alpha = \alpha'$ in (3.35) we have $\tau_{ll'} = \tau_{l'l}$ which implies a smmetric compression pattern. In the Galerkin setting, the superconvergence estimate (3.40) implies the choice $\alpha = \alpha' = 1$ for a maximal convergence rate at an interior point (regularity of the exact solution provided). We therefore assume below that $\alpha = \alpha'$. Note, however, that for collocation methods, also nonsymmetric compression patterns corresponding to $\alpha = 0, \alpha' = 1$ are more efficient. The algorithms below can be straightforwardly adapted to this situation.

Remark 18 We point out that the compression strategy (3.33) together with (3.35) is quite different from the so-called "ϵ-truncation" proposed in [1]. In fact, this compression criterion is analysis-based and keeps matrix elements of different size in different blocks. Dropping all matrix entries below a certain treshhold will not, in general, lead to a compressed scheme with proper asymptotic convergence rates.

4 Numerical Integration

The preceding analysis still assumes that the entries

$$(\mathbf{K}^L)_{(l,j)(l',j')} := \int_\Gamma \int_\Gamma k(x,y)\psi_j^l(x)\psi_{j'}^{l'}(y)\,ds_y ds_x \;=\; \int_\mathcal{U} \int_\mathcal{U} \tilde{k}(\xi,\eta)\tilde{\psi}_j^l(\xi)\tilde{\psi}_{j'}^{l'}(\eta)\,d\eta d\xi \qquad (4.1)$$

of the stiffness matrix are evaluated exactly. This is not possible, in general, and approximations by means of numerical integration must be used. Theorem 16 estimated the impact of the consistency error introduced by the matrix compression (3.33) on the asymptotic convergence rates of the Galerkin scheme. In the same framework the consistency error due to numerical integration has been analyzed in [16].

It is clear that with sufficient effort arbitrarily accurate approximations can be computed so that Theorem 16 will hold for the resulting, fully discrete scheme as well. It is less trivial, however, to find a quadrature scheme that will achieve this without undue increase in complexity. In our implementation we use the quadrature scheme that was proposed and analyzed in [16]. We now describe this scheme and present its principal properties.

For ease of exposition, we focus only on the basic case $\mathcal{U} = \mathcal{S}$, i.e. the supports of the wavelets are images of the unit square (see Remark 20 below for modifications for triangles). Since each multiwavelet is polynomial in the four subdomains $\mathcal{U}^i$ of $\mathcal{U}$ the integral (4.1) may be assembled from the integrals related to the 16 combinations of subdomains. For these subintegrals tensor product Gaussian quadrature formulas converge exponentially with the rate depending on the size of the integrand's domain of analyticity:

Lemma 19 *Let $\varepsilon > 0$ and let G_U^n denote the quadrature rule that employs in each of the four subdomains of U a two dimensional tensor product Gaussian quadrature with n Gauss points in each direction. Then for all wavelets ψ, ψ' with $U := \operatorname{supp}\psi \in M^l$, $U' := \operatorname{supp}\psi' \in M^{l'}$ satisfying $\delta := \operatorname{dist}(U,U') > 0$ the error estimate*

$$\left| \int_\Gamma \int_\Gamma k(x,y)\psi(x)\psi'(y)\,ds_y ds_x - G_{U,x}^n G_{U',y}^{n'} k(x,y)\psi(x)\psi'(y) \right| \;=\; O(\varepsilon) \qquad (4.2)$$

holds, provided the number of Gauss points is selected according to

$$n \;\geq\; \frac{d}{2} + \frac{n_\varepsilon}{\log \rho(U,U')^2}, \quad n' \;\geq\; \frac{d}{2} + \frac{n_\varepsilon}{\log \rho(U',U)^2}, \quad n_\varepsilon := -\log(2^{l+l'}\varepsilon) - \log \delta^2 \tag{4.3}$$

with ρ *defined in* (3.26) *and* (3.27)*, respectively. Moreover, for*

$$l \geq l' > 0 \quad \textit{and} \quad \rho(U',U) \geq 2 \tag{4.4}$$

(4.2) *is satisfied if*

$$n_\varepsilon := -\log(2^{l-l'}\,\varepsilon) + \log \gamma^2. \tag{4.5}$$

Remark 20 The cases of (4.1) involving triangular panels are easily reduced to the considered case of quadrilaterals by applying the surjective mapping

$$\mathcal{D} : \mathcal{S} \mapsto \mathcal{T},\ \mathcal{D}(\xi) := \begin{pmatrix} \xi_1 \\ \xi_1\xi_2 \end{pmatrix}$$

sometimes called the *Duffy transformation*. Then Lemma 19, in particular (4.2), remains valid if the number of Gauss points selected exceeds $n + \frac{1}{2}$ and $n' + \frac{1}{2}$, respectively.

Unfortunately, the domain of analyticity may degenerate such that the complexity of the quadrature according to Lemma 19 cannot be bounded logarithmically: let $l = L$, $l' = 0$ and $\delta = O(2^{-L})$, then for n' in (4.3) it follows that

$$n' = \Omega(2^L |\log \varepsilon|) = \Omega(\sqrt{N_L}),$$

i.e. n' grows at least as $\sqrt{N_L}$. Hence, the number of kernel evaluations for such an entry is at best bounded by $O(N_L)$. Since there are $O(\sqrt{N_L})$ entries of this kind and since all of them occur in the compressed stiffness matrix the reduction of the complexity by compression is almost cancelled.

In order to recover the almost optimal complexity estimates of the previous Section a dyadic subdivision of the larger panel in the quadrature scheme is introduced (see [13][16]).

Lemma 21 *Let* $l \geq l' \geq 0$ *and* $U \in M^l$, $U' \in M^{l'}$. *Then there exists a subdivision*

$$\Lambda(U,U') \subset M^{l'} \cup \cdots \cup M^l \tag{4.6}$$

of U' *such that*

$$\tilde{U} \in \Lambda(U,U') \implies \rho(\tilde{U},U) \geq 2 \quad \textit{or} \quad \left(\operatorname{dist}(U,\tilde{U}) = 0 \quad \textit{and} \quad \tilde{U} \in M^l\right) \tag{4.7}$$

and

$$\left|\Lambda(U,U') \cap M^{\tilde{l}}\right| \leq C_\gamma \qquad \textit{for } l' \leq \tilde{l} \leq l, \tag{4.8}$$

i.e. the number of panels contained in the subdivision is bounded by $O(\log N_L)$.

Note that due to the subdivision the singular cases, i.e. $\operatorname{dist}(U,U') = 0$, that may occur are reduced to three basic situations: U equal to U', U and U' sharing a common edge or U and U' sharing a common vertex. They can be handled by special quadrature schemes [11][7] to provide an error of order $O(\varepsilon)$ in at most $O(|\log \varepsilon|^3)$ operations.

Combining the results of the previous lemmas constitutes the following strategy, which in effect is a variable order, composite quadrature rule (see [13]):

Theorem 22 *Let $\varepsilon_r > 0$, $\varepsilon_s > 0$ and $U \in M^l$, $U' \in M^{l'}$ with $l \geq l' \geq 0$. Then for all wavelets ψ, ψ' with $\operatorname{supp}\psi = U$ and $\operatorname{supp}\psi' = U'$ the following error estimate holds:*

$$\left| \int_\Gamma \int_\Gamma k(x,y)\psi(x)\psi'(y)\,ds_y ds_x - \sum_{\tilde{U}\in\Lambda(U,U')} Q_{U,\tilde{U}}\, k(x,y)\psi(x)\psi'(y) \right| \;=\; O(\varepsilon_s + \varepsilon_r) \tag{4.9}$$

where

$$Q_{U,\tilde{U}} := \begin{cases} G^n_{U,x} G^{\tilde{n}}_{\tilde{U},y} & \text{if } \operatorname{dist}(U,\tilde{U}) > 0 \\ \text{special quadrature scheme} & \text{otherwise} \\ \text{of accuracy } O(\varepsilon_s) \text{ [11]} & \end{cases} \tag{4.10}$$

and

$$n \;\geq\; \frac{d}{2} + \frac{n_{\varepsilon_r}}{\log \rho(U,\tilde{U})^2}, \quad \tilde{n} \;\geq\; \frac{d}{2} + \frac{n_{\varepsilon_r}}{\log \rho(\tilde{U},U)^2}, \tag{4.11}$$

with

$$n_{\varepsilon_r} := -\log(2^{l-l'}\,\varepsilon_r) + \log(l - l' + 1) + \log \gamma^2. \tag{4.12}$$

In particular, the choice $\varepsilon_s = 2^{-(d+1)(2L-l-l')+l'-l}$ and $\varepsilon_r = 2^{-2(L-l)}\varepsilon_s$, i.e.

$$n_{\varepsilon_r} := ((d+2)(2L - l - l') + l' - l)\log 2 + \log(l - l' + 1) + \log \gamma^2, \tag{4.13}$$

preserves all assertions of Theorem 16. Moreover, the total cost of the numerical integration in order to generate the compressed stiffness matrix can be estimated to be of order $O(N_L(\log N_L)^4)$ operations.

Theorem 22 cites the results shown in [16] which hold for any polynomial degree d. For $d = 0$ in fact small improvements are possible such that the time to assemble the stiffness matrix in our numerical experiments could be reduced by a factor of three:

Remark 23 The term $\log(l - l' + 1)$ in (4.12) compensates the influence of the number of panels in a subdivision on the quadrature error. If $\operatorname{dist}(U,U') >> 0$ then $\Lambda(U,U') \cap M^{\tilde{l}} = \emptyset$, $l \geq \tilde{l} \geq l'$, in most of the cases. Hence, a subdivision consists of less than $O(l - l' + 1)$ panels. Replacing $\log(l - l' + 1)$ by $\epsilon \log(\tilde{l} - l' + 1)$ with $\epsilon > 1$ attends to these situations while (4.9) still holds.

Furthermore, the choice $\varepsilon_r = 2^{-2(L-l)}\varepsilon_s$ may be refined:

$$\varepsilon_r = \begin{cases} \varepsilon_s & \text{if } \rho(U',U) < 2, \\ 2^{-2(L-l)}\varepsilon_s & \text{if } \rho(U',U) \geq 2 \end{cases} \tag{4.14}$$

without changing the assertion of Theorem 22.

5 Implementation

The implementation of our method is based on the library *Concepts-1.2*. This library provides an object-oriented framework for boundary element methods where basic concepts of Petrov-Galerkin schemes such as subspaces, functions, operators or dualforms are captured in class definitions [8].

5.1 Basic Structures

For an efficient evaluation of the stiffness matrix by element matrices an abstract data type is required which associates each panel U_j^l, $0 \leq l < L$ with basis functions $\psi \in \bigcup \Psi^l$ satisfying $\operatorname{supp} \psi = U_j^l$. We call each association of this kind an *element* represented in the existing framework by the abstract base class Element. The interface of this base class provides the access to the underlying panel and the corresponding basis functions represented by a global index $b \in \mathbb{N}_0$. The implementation of the class by means of derived classes depends on the shape of the panel and the basis functions involved. For the algorithm under consideration two classes become necessary to cover triangular and quadrangular panels. In addition to the standard interface of the base class Element the interface of the wavelet classes must offer the possibility to access the elements corresponding to the four subdomains of the particular panel U_j^l, which we will denote by $U_{4j}^{l+1}, \ldots, U_{4j+3}^{l+1}$ in what follows. This guarantees the efficient implementation of the composite quadrature scheme discussed in the previous section. Moreover, the information about referencing the subelements can be used to build element trees. The roots of the trees are the elements of level zero, whereas the leafs consist of level $L-1$ elements. Accordingly, we introduce the following ordering relation of the panels U_j^l which reflects a preorder depth-first traversal of the element trees:

$$U_j^l \prec U_{j'}^{l'} \quad :\Longleftrightarrow \quad j < j'4^{l-l'} \text{ or } \left(j = j'4^{l-l'} \text{ and } l < l'\right). \tag{5.1}$$

The construction of the wavelet elements with respect to a given polyhedra as well as the generation of the global index for the basis functions is the task of the class Multiscale a specialization of the base class Space already defined in the framework. This class is used to represent test and trial spaces of the discretization scheme. It provides the operation to scan all elements of the particular space used, for example, to assemble the stiffness matrix. In the case of the class Multiscale the elements are listed using preorder depth-first traversal of the element trees. This ordering is the natural choice to assemble the compressed stiffness matrix. Due to the discontinuity of our multiwavelets the generation of a global index is easily accomplished by scanning the elements and assigning each associated basis function the number of functions visited so far. Thus, when using preorder depth-first traversal of the elements subsequent indices in general reflect a clustering of basis functions. This is used to determine and compress blocks of zeros in the stiffness matrix easily.

5.2 Discrete Operator

The essential characteristic of operators, i.e. the mapping of functions, is represented by the abstract base class Operator in *Concepts–1.2*. In the case of standard boundary element methods, i.e. when no compression of the system matrix is applied, the obvious implementation of an operator is a two-dimensional array. In this case the mapping of the operator is just an ordinary matrix-vector product. This situation is covered by the class *OP_Allpurpose* in the framework *Concepts–1.2*.

In the case of the presented multiscale scheme we have to choose a more sophisticated implementation to meet the objectives of the method. According to Theorem 16, the truncation criterion (3.33) yields a compressed stiffness matrix with $O(N_L(\log N_L)^2)$ nonzero entries. Thus, it is essential that the nonzero entries can be localized also in $O(N_L(\log N_L)^s)$ operations, i.e. without an exhaustive search of the N_L^2 combinations. Furthermore, a storage scheme for the compressed matrix with a memory consumption proportional to the

number of nonzero entries and a small overhead for the management must be provided. In addition, an efficient access to the entries of the stiffness matrix when processing a matrix-vector product is essential.

As it is well known from the implementation of finite elements, the most efficient and natural way to exploit the quadrature is to perform the assembly be means of evaluating element matrices for each combination of panels U and U':

$$(E^{\Psi}_{UU'})_{\psi\psi'} := \int_U \int_{U'} k(x,y)\psi(x)\psi'(y)\,ds_y ds_x, \quad \operatorname{supp}\psi = U,\ \operatorname{supp}\psi' = U'. \tag{5.2}$$

Therefore, element matrices form the basic structure of the following algorithms. Besides it is ensured that the algorithms can also be applied in the case of continuous wavelets, since the constraints due to the continuity are handled by an appropriate generation of the global index which does not affect element matrices.

5.2.1 Localization

The algorithm for the localization of nonzero entries is based on the obvious implication

$$\operatorname{dist}(U^l_j, U^{l'}_{j'}) > \tau_{ll'} \text{ and } U^{\tilde{l}}_{\tilde{j}} \subset U^{l'}_{j'} \ \Rightarrow\ \operatorname{dist}(U^l_j, U^{\tilde{l}}_{\tilde{j}}) > \tau_{l\tilde{l}} \tag{5.3}$$

where the threshold values $\tau_{ll'}$ and $\tau_{l\tilde{l}}$ are chosen according to (3.35). Hence, if the elements are scanned in depth-first order with respect to the element tree described in the previous Section the elements that do not contribute to the stiffness matrix are located in subtrees that can be skipped in $O(1)$ operations. This strategy yields the following recursive algorithm:

Algorithm 24

assemble$(U^l_j, U^{l'}_{j'})$ {
  if $\operatorname{dist}(U^l_j, U^{l'}_{j'}) \le \tau_{ll'}$ *or* $l = 0$ *or* $l' = 0$ {
    evaluate $E^{\Psi}_{U^l_j, U^{l'}_{j'}}$ *and update* $\mathbf{K}^L$
    if $(l' < L-1)$
      for $i = 0$ *to* 3 *assemble*$(U^l_j, U^{l'+1}_{4j'+i})$
  }
}

For given panels U and U' Algorithm 1 determines all entries in the compressed stiffness matrix that correspond to the wavelets $\psi, \psi' \in \bigcup_{l=0}^{L} \Psi^l$ with $\operatorname{supp}\psi = U$ and $\operatorname{supp}\psi' \subset U'$. Therefore, the calling sequence

$$\text{for} \quad U \in \bigcup_{l=0}^{L-1} M^l,\ U' \in M^0 \quad \operatorname{assemble}(U, U') \tag{5.4}$$

generates the complete compressed stiffness matrix $\mathbf{K}^L$ row by row. In particular, rows related to the wavelets ψ with $\psi \subset U$ are treated simultaneously.

Remark 25 Note that in general the basis functions of Ψ^0 do not have the vanishing moment property (3.17) such that the corresponding entries are not compressible. In Algorithm 24, we take this into account by evaluating the element matrix in the cases $l = 0$ or $l' = 0$ additionally. However, in this way more entries than necessary are evaluated, but it is doubtful if a special treatment of Ψ^0 in the implementation is more efficient, especially when a subdivision of the elements may become necessary such that element matrices corresponding to Ψ^1 must be evaluated anyway.

Lemma 26 *The nonzero entries of the compressed stiffness matrix* $\mathbf{K}^L$ *are localized in* $O(\mathcal{N})$ *operations, where* $\mathcal{N}$ *denotes the number of nonzero entries.*

Proof:
With respect to the localization each call of the function *assemble()* is of order $O(1)$. Hence, the complexity of localizing the nonzero entries is proportional to the number of calls of *assemble()* which itself is bounded by the number of nonzero entries. □

Since the evaluation of the exact distance of two panels is an expensive operation, even for triangles or quadrilaterals, the distance of their related bounding boxes, i.e. the smallest box $[c_l, c_u] := \{x \in \mathbb{R}^3 : c_l \leq x \leq c_u\}$ containing the considered panel, is used instead. This approximation is conservative in the sense that it is a lower bound for the exact value. Thus, more entries of the stiffness matrix (cf. 24) and more Gauss points (cf. (4.3)) than necessary are employed.

5.2.2 Compression

Our compression technique is based on runlength encoding, where we only encode sequences of zero entries in the stiffness matrix. In particular, for a given row of the matrix we replace sequences of zeros by control tags that specify the length and the starting index of the following block of nonzero values. In contrast to standard data structures such as lists of entries to represent sparse matrices, this approach preserves blocks of nonzero entries in the stiffness matrix. Thus, the amount of informations necessary to decode the compressed matrix, in our case the control tags, are reduced and matrix-vector products could be realized efficiently, since the entries are not randomly distributed in the memory.

However, we cannot affort to assemble a complete row by Algorithm 24 and compress afterwards since scanning the row amounts to $O(N_L)$ operations. Therefore, we relate the global index $b \in \mathbb{N}_0$ of the wavelets to the calling sequence of the elements implied by Algorithm 24, i.e. the depth-first ordering (5.1) such that a subtree of elements is aligned with a block of subsequent entries in a row of the stiffness matrix. Then, the subtrees skipped during the assembly correspond to sequences of zeros. Hence, each zero block can be matched in $O(1)$ operations making a compression in $O(\mathcal{N})$ operations available.

In order to improve the ratio of compression we exploit the symmetry $\tau_{ll'} = \tau_{l'l}$ of the threshold defined in (3.35) which implies a symmetric pattern of zero and nonzero entries of the stiffness matrix (cf. Remark 17). Hence, assembling rows and columns simultaneously offers the possibility to reuse the control tags of a row for the corresponding column saving each time the storage of one tag. Obviously only half of the entries, say the lower triangular matrix with respect to an arbitrary ordering, should be evaluated in row mode whereas the rest is to be evaluated in column mode. To distinguish row and column mode we chose a

different ordering of the entries as implied by (5.1), namely an ordering corresponding to lexicographically ordered panels:

$$U_j^l < U_{j'}^{l'} \quad :\Longleftrightarrow \quad l < l' \ \text{ or } \ (l = l' \text{ and } j < j')\,. \tag{5.5}$$

In regard of the quadrature algorithm of Section 4, this ordering ensures the assumption $l \geq l'$. We finally get the following modification of Algorithm 24, which we use in our implementation:

Algorithm 27

assemble$(U_j^l, U_{j'}^{l'})$ {
  if $\left(\operatorname{dist}(U_j^l, U_{j'}^{l'}) \leq \tau_{ll'}\right)$ *or* $l' = 0\big)$ *and* $\left(l > l' \text{ or } (l = l' \text{ and } j \leq j')\right)$ {
    evaluate $E^{\Psi}_{U_j^l, U_{j'}^{l'}}$, $E^{\Psi}_{U_{j'}^{l'}, U_j^l}$ *and update* $\mathbf{K}^L$
    if $(l' < L - 1)$
      for $i = 0$ *to* 3 *assemble*$(U_j^l, U_{4j'+i}^{l'+1})$
  }
}

5.3 Quadrature

The general structure of the quadrature strategy used is already described in Section 4. However, there are two topics concerning an efficient implementation that should be discussed in detail:

- construction of subdivisions and
- recycling of temporary results of the composite quadrature.

5.3.1 Quadrature Algorithm

To construct a subdivision consistent with (4.6) - (4.8) one obviously has to exploit the element trees. This yields a recursive reformulation of the quadrature Q (cf. Theorem 22) with an appropriate subdivision incorporated:

$$\tilde{Q}_{U_j^l, U_{j'}^{l'}} := \begin{cases} \displaystyle\sum_{i=0}^{3} \tilde{Q}_{U_j^l, U_{4j'+i}^{l'+1}} & \rho(U_{j'}^{l'}, U_j^l) < 2 \text{ and } l > l' \\ Q_{U_j^l, U_{j'}^{l'}} & \text{otherwise} \end{cases} \tag{5.6}$$

for $l \geq l'$. Since the recursion terminates at least for $l = l'$ and since $\rho(U_{j'}^l, U_j^l) < 2$ implies $\operatorname{dist}(U_{j'}^l, U_j^l) = 0$ the properties (4.6) and (4.7) are satisfied. In particular, they are satisfied with a minimal number of recursion steps yielding a minimal number of elements in the implicit subdivision of (5.6) such that (4.8) holds due to Lemma 21.

We interlace the evaluation of all entries of an element matrix with the quadrature shown above to derive the following algorithm:

Algorithm 28

$$
\begin{aligned}
&\mathit{integrate}(U_j^l, U_{j'}^{l'}, U_{\tilde{j}}^{\tilde{l}})\ \{\\
&\quad \mathit{if}\ \ \rho(U_j^l, U_{\tilde{j}}^{\tilde{l}}) \geq 2 \ \ \mathit{or}\ \ l = \tilde{l}\\
&\qquad E\ :=\ (Q_{U_j^l, U_{\tilde{j}}^{\tilde{l}}}\ k(x,y)\psi(x)\psi'(y))_{\operatorname{supp}\psi = U_j^l,\ \operatorname{supp}\psi' = U_{j'}^{l'}}\\
&\quad \mathit{else}\\
&\qquad E\ :=\ \sum_{i=0}^{3}\ \mathit{integrate}(U_j^l, U_{j'}^{l'}, U_{4\tilde{j}+i}^{\tilde{l}+1})\\
&\quad \mathit{return}\ E\\
&\}
\end{aligned}
$$

By calling $\mathit{integrate}(U, U', U')$ the element matrix $E_{UU'}^{\Psi}$ is returned, provided $U \in M^l$ and $U' \in M^{l'}$ with $l \geq l'$. The remaining entries, i.e. $E_{UU'}^{\Psi}$ with $l < l'$, could be determined by invoking $\mathit{integrate}(U, U', U)$. However, in order to exploit the construction of a subdivision exhaustively, the evaluations of $E_{UU'}^{\Psi}$ and its symmetric equivalent $E_{U'U}^{\Psi}$ should be embedded in one function:

Algorithm 29

$$
\begin{aligned}
&\mathit{integrate}(U_j^l, U_{j'}^{l'}, U_{\tilde{j}}^{\tilde{l}})\ \{\\
&\quad \mathit{if}\ \ \rho(U_j^l, U_{\tilde{j}}^{\tilde{l}}) \geq 2 \ \ \mathit{or}\ \ l = \tilde{l}\\
&\qquad E\ :=\ (Q_{U_j^l, U_{\tilde{j}}^{\tilde{l}}}\ k(x,y)\psi(x)\psi'(y))_{\operatorname{supp}\psi = U_j^l,\ \operatorname{supp}\psi' = U_{j'}^{l'}}\\
&\qquad E'\ :=\ (Q_{U_{\tilde{j}}^{\tilde{l}}, U_j^l}\ k(x,y)\psi(x)\psi'(y))_{\operatorname{supp}\psi = U_{j'}^{l'},\ \operatorname{supp}\psi' = U_j^l}\\
&\quad \mathit{else}\\
&\qquad (E, E')\ :=\ \sum_{i=0}^{3}\ \mathit{integrate}(U_j^l, U_{j'}^{l'}, U_{4\tilde{j}+i}^{\tilde{l}+1})\\
&\quad \mathit{return}\ (E, E')\\
&\}
\end{aligned}
$$

This way the subdivision, which is identical in both cases, is only generated once. Besides, during the assembly according to Algorithm 27 always such a pair of element matrices is requested.

5.3.2 Cache

Generating the entries of the stiffness matrix by means of the described dyadic subdivision scheme imposes the addition of several temporary quantities. Chances are that these quantities could be reused for subsequent calculations. However, this is not possible in the formulation of Algorithm 28. Therefore, we have to introduce a slight modification. We consider the following bases for the subspaces $\mathcal{V}^L$ and V^L, respectively:

$$
\begin{aligned}
\tilde{\Phi}^0 &:= \tilde{\Psi}^0,\\
\tilde{\Phi}^L &:= \left\{\tilde{\varphi} \in L^2(\Gamma) :\ \exists\ 1 \leq i \leq 4 \quad \tilde{\varphi} \circ \tau_i \in \tilde{\Phi}^{L-1} \text{ and } \operatorname{supp}\tilde{\varphi} \subset \overline{\mathcal{U}^i}\right\},
\end{aligned}
$$

$$\begin{aligned}\Phi^L &:= \left\{\varphi_j^L :\ 0 \le j < N_L\right\} \\ &= \left\{\varphi \in L^2(\Gamma) :\ \exists\, 0 \le k < N_\Gamma \quad \varphi \circ \chi_k \in \tilde{\Phi}^L \text{ and } \operatorname{supp}\varphi \subset \overline{\Gamma_k}\right\}.\end{aligned}$$

Then, the two scale relation

$$\varphi_j^L = \sum_{j' \in J(j)} a_{jj'} \varphi_{j'}^{L+1} \tag{5.7}$$

is satisfied with coefficients $a_{jj'}$ independent of L, provided an appropriate ordering of the basis functions is used. Moreover, (5.7) implies a corresponding relation for element matrices with respect to the new bases:

$$E^{\Phi}_{U_j^l U_{j'}^{l'}} = \sum_{i=0}^{3} E^{\Phi}_{U_j^l U_{4j'+i}^{l'+1}} A_i = \sum_{i=0}^{3} A_i^T E^{\Phi}_{U_{4j+i}^{l+1} U_{j'}^{l'}} \tag{5.8}$$

where A_i are certain $\mathcal{N}_1 \times \mathcal{N}_1$–matrices and

$$(E^{\Phi}_{U_j^l U_{j'}^{l'}})_{\varphi\varphi'} := \int\limits_{U_j^l} \int\limits_{U_{j'}^{l'}} k(x,y)\varphi(x)\varphi'(y)\, ds_y ds_x \tag{5.9}$$

with $\varphi \in \Phi^{l+1}$, $\varphi' \in \Phi^{l'+1}$ such that $\operatorname{supp}\varphi \subset U_j^l$, $\operatorname{supp}\psi' \subset U_{j'}^{l'}$. Similarly, since $\Psi^L \subset \operatorname{span}\Phi^L$, there exists a $\mathcal{N}_1 \times \mathcal{N}_1$–matrix B_0 and a $\mathcal{N}_1 \times (\mathcal{N}_1 - \mathcal{N}_0)$–matrix B_1 such that

$$E^{\Psi}_{U_j^l U_{j'}^{l'}} = \begin{cases} B_0^T\ E^{\Phi}_{U_j^l U_{j'}^{l'}}\ B_0 & \text{if } l = 0 \text{ and } l' = 0, \\ B_0^T\ E^{\Phi}_{U_j^l U_{j'}^{l'}}\ B_1 & \text{if } l = 0 \text{ and } l' > 0, \\ B_1^T\ E^{\Phi}_{U_j^l U_{j'}^{l'}}\ B_0 & \text{if } l > 0 \text{ and } l' = 0, \\ B_1^T\ E^{\Phi}_{U_j^l U_{j'}^{l'}}\ B_1 & \text{if } l > 0 \text{ and } l' > 0. \end{cases} \tag{5.10}$$

The modification of Algorithm 28 consists in evaluating E^{Φ} using recursion (5.8) instead of E^{Ψ}. Hence, in every step of the recursion a complete element matrix with respect to the bases Φ^l is generated. Due to (5.10) these element matrices can be used to retrieve possibly necessary information in subsequent calls. Note that the accuracy of the recycled element matrices is sufficient since n_{ϵ_r} in (4.13) decreases with respect to l'.

Algorithm 30

integrate$(U_j^l, U_{j'}^{l'})$ {
  if $(E^{\Psi}_{U_j^l U_{j'}^{l'}}$ *not cached*$)$ *reclaim*$(U_j^l, U_{j'}^{l'}, U_{j'}^{l'})$
  return $E^{\Psi}_{U_j^l U_{j'}^{l'}}$
}

reclaim$(U_j^l, U_{j'}^{l'}, U_{\tilde{j}}^{\tilde{l}})$ {
  if $(\rho(U_j^l, U_{\tilde{j}}^{\tilde{l}}) \ge 2$ *or* $l = \tilde{l})$
    $E^{\Phi} := (Q_{U_j^l, U_{\tilde{j}}^{\tilde{l}}}\, k(x,y)\varphi(x)\varphi'(y))_{\operatorname{supp}\varphi \subset U_j^l,\ \operatorname{supp}\varphi' \subset U_{j'}^{l'}}$
  else

$$E^{\Phi} := \sum_{i=0}^{3} \mathit{reclaim}(U_j^l, U_{j'}^{l'}, U_{4\tilde{j}+i}^{\tilde{l}+1}) A_i$$

load cache with $E^{\Psi}_{U_j^l U_{\tilde{j}}^{\tilde{l}}} := B_k^T E^{\Phi} B_{\tilde{k}}$

return E^{Φ}

}

In Algorithm 30 the function $\mathit{integrate}(U, U')$ initiates a cache reclaim if the requested element matrix $E^{\Psi}_{UU'}$ is not available; otherwise a cached value of the matrix is returned immediately. Again, according to Algorithm 29, we evaluate the symmetric counterpart of every element matrix simultaneously. However, to simplify the algorithmic formulation this is not shown in Algorithm 30.

In the implementation we cache all information that is generated during one call of the function $\mathit{reclaim}(U, U', U')$, i.e. we provide a cache size of order $O(N)$. If one takes the hierarchical sequence of the element matrix generation of Algorithm 24 into account, the cache size is sufficient to reuse almost all informations generated by the subdivision scheme and to satisfy the maximum potential number of requests to $\mathit{integrate}(U, U')$ by cached values.

However, with Algorithm 30 it is not possible to prevent all element matrices from being calculated twice. In particular, for $U \in M^l$, $U' \in M^0$, $l > 0$ and $\tilde{U} \in \Lambda(U, U') \cap M^l$ the element matrices $E^{\Psi}_{U\tilde{U}}$ and $E^{\Psi}_{\tilde{U}U}$ are generated in the subdivision process. Note that, in addition, $U \in \Lambda(\tilde{U}, U'')$ for an appropriate $U'' \in M^0$ holds in most of the cases. Hence, both element matrices are evaluated anew since they are not in the hierarchy of element matrices cached. Nevertheless, numerical experiments show, that one could only save less than 20% of the kernel evaluations, i.e roughly speeking at most a speed up of 1.25, if a more sophisticated cache strategy is used, which guarantees that no element matrices are evaluated more than once for any particular calling sequence.

6 Numerical Experiments

In this Section, we present the results of three numerical experiments obtained with the described implementation of the multiscale scheme. On a polyhedron $D \subset \mathbb{R}^3$ we considered the Laplace equation with Dirichlet boundary conditions:
For given $f \in L^2(D)$ find $U \in H^1(D)$ such that

$$\begin{aligned} \Delta U &= 0 \quad \text{in} \quad D, \\ U &= f \quad \text{on} \quad \Gamma := \partial D. \end{aligned}$$

The double layer ansatz $U(x) = \langle k(x, \cdot), u \rangle$ where the double layer kernel is given by

$$k(x, y) = -\frac{1}{4\pi} \frac{\langle n(y), y - x \rangle}{\|y - x\|^3} \tag{6.1}$$

leads with the jump relations to the second kind boundary integral equation

$$u \in L^2(\Gamma): \quad \langle v, Au \rangle = \langle v, f \rangle \quad \forall v \in L^2(\Gamma) \tag{6.2}$$

level	N_L	a, α	time[s]	mem[MB]	it	cpr
2	96	0.3, 1.0	0.5	0.06	16	0.827
3	384	0.3, 1.0	5.0	0.37	17	0.333
4	1536	0.3, 1.0	37.5	2.03	17	0.113
5	6144	0.3, 1.0	224.6	9.97	17	0.035
6	24576	0.3, 1.0	1181.3	46.68	17	0.010
7	98304	0.3, 1.0	6449.6	213.44	18	0.003
8	393216	0.3, 1.0	36026.4	959.76	18	0.001

Table 1: First experiment: a, α = threshold parameter, *time* = time for assembly and solution, *mem* = memory required to store the compressed matrix inclusive management overhead, *it* = number of iterations, *cpr* = memory consumption with respect to a dense matrix.

with the integral operator

$$(Au)(x) = -\frac{1}{2}u(x) + \int_\Gamma k(x,y)u(y)\,ds_y$$

defined almost everywhere on Γ. We solved (6.2) on several polyhedral domains with quite similar performance. Here, we only report the results obtained with a polyhedron defined by six equilateral triangles and the right hand side

$$f(x) = \|x - x_0\|^{-1}, \qquad x_0 \in \mathbb{R}^3 \text{ in the exterior of } D.$$

For the discretization constant test and trial functions ($d = 0$) were used. In our experiments, we did not make use of the fact that entries in the stiffness matrix corresponding to panels located in the same face of the polyhedron D vanish. Therefore, the memory and CPU-time listings below are representative of the algorithm's performance also for boundary integral equations with kernels other than (6.1) and for curved surfaces. The following results were obtained on a *SUN Ultra-Enterprise* on a single processor with 1 GB RAM and 125 MHz clock.

In the first experiment we kept the parameters a and $\alpha = \alpha'$ of the thresholds (3.35) controlling the compression fixed and solved the problem on various levels up to about 400000 unknowns (Table 1). On the finest mesh the compressed matrix consists of only 0.1% of the entries of the dense stiffness matrix. In addition, it can be observed that the number of iterations used by the solver (GMRes without restart) is almost constant validating the bounded condition numbers of the compressed matrices.

In Figure 1 the time of assembly and compression is depicted. Here, the upper dashed line corresponds to the predicted bound $O(N_L(\log N_L)^4)$ in Theorem 22. The plot indicates that the influence of the higher order logarithmic terms on the computing time seems to be negligible compared to the $O(N_L(\log N_L)^2)$ term illustrated by the lower dashed line. Roughly speaking, on an average nearly a constant number of operations is used to evaluate an entry of the stiffness matrix.

In all numerical experiments the time for solution accounts only for less than 10% of the total time shown in the tables. Therefore, with the present method the BEM-paradigm

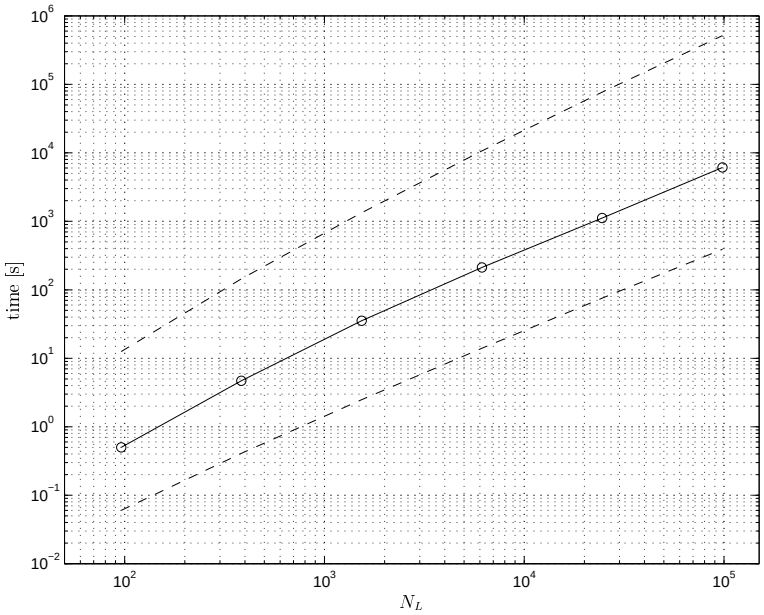

Figure 1: Time for assembly and compression of the matrix.

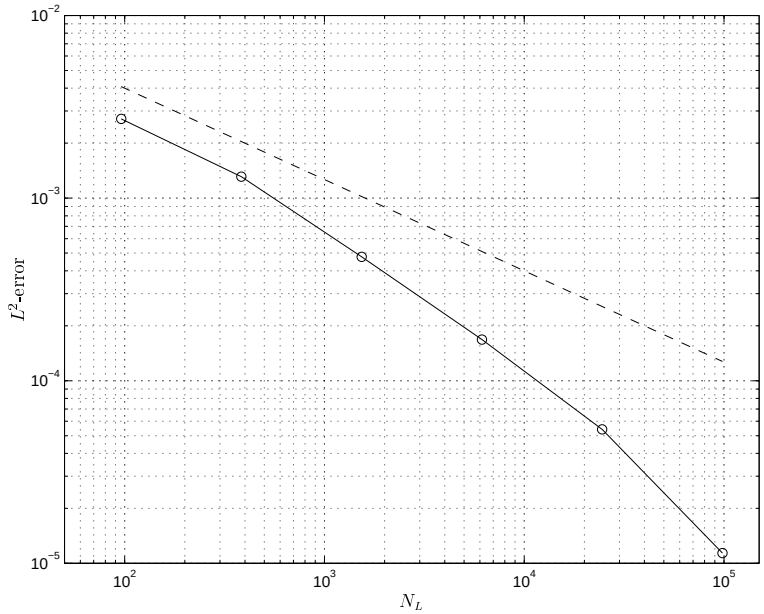

Figure 2: $\|u^l\|_0 - \|u\|_0$ versus N_L.

that most of the work is spent for quadrature is still valid and a speed up similar to the one for dense matrices can be achieved with the parallelization of the matrixassembly.

Figure 2 and Figure 3 show the behaviour of the L^2-error of the density u on the boundary and the average error in several interior points of the solution U, respectively. The L^2-error is approximated by the difference of the norm of the discrete density and the norm of the exact density. Since an exact solution is not available we have computed an

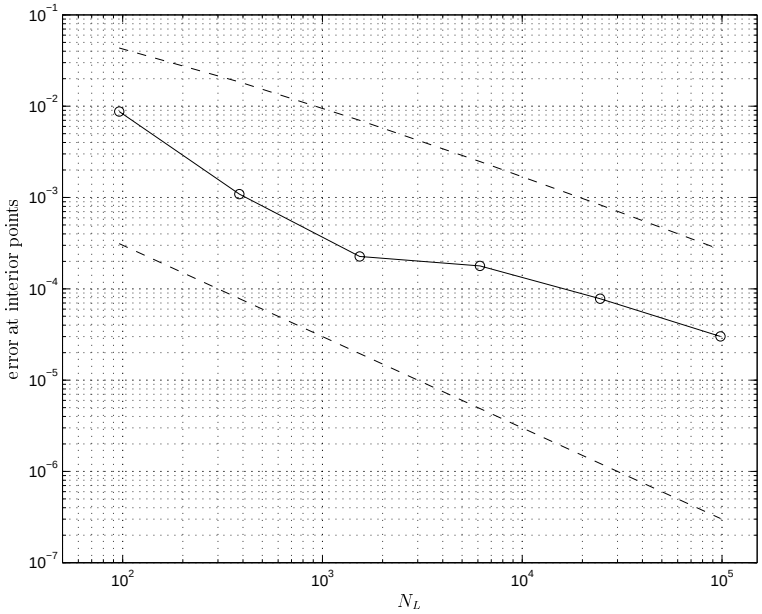

Figure 3: Error at interior points versus N_L.

approximate value by higher order quadrature and extrapolation. According to Theorem 22, the expected rate of convergence is determined by regularity properties of A and its adjoint A^*. From the known edge and vertex singularities of the Laplacean in polyhedra [4] it can be verified that in the example under consideration here both operators admit solutions belonging to $H^1(\Gamma)$ for smooth right hand sides. This means that we have Theorem 22 with $s = \tilde{s} = 1$ and expect $O(N_L^{-1/2})$ convergence in the $L^2(\Gamma)$-norm and $O(N_L^{-1})$ convergence at an interior point (note that collocation or Nyström schemes do not display this kind of superconvergence at an interior point and would require H^2-regularity on Γ and $d = 1$ to achieve $O(N_L^{-1})$ convergence at an interior point).

Again, the dashed line illustrates the expected behaviour of essentially $O(N_L^{-1/2})$. For the error in interior points twice the convergence rate should be observed, hence essentially $O(N_L^{-1})$ (lower dashed line) or, according to Theorem 16, $O(N_L^{-1}(\log N_L)^2)$ (upper dashed line).

Finally, we compared our method with a standard boundary element implementation generating the fully populated stiffness matrix with an optimized quadrature rule. For both methods the time used to generate a solution satisfying a given L^2 error is depicted in Figure 4 where the dashed line corresponds to the standard approach. It turns out that already for moderate errors, "moderate" with respect to our model problem, the wavelet method beats the standard approach: assuming an error between 10^{-5} and 10^{-4} the wavelet method is 10 times faster. Moreover, in this case it saves about 98% of the memory.

The second experiment investigates the behaviour of the method when the amount of compression driven by the parameter a changes (Table 2). The constant number of iterations shows that even for a high compression the algorithm remains stable. The convergence rates, in addition, are in all cases preserved as indicated by the error in interior points shown in Figure 5: when the influence of the coarser meshes, where practically no compression is possible, vanishes, the lines corresponding to different values of a fan out. Nevertheless,

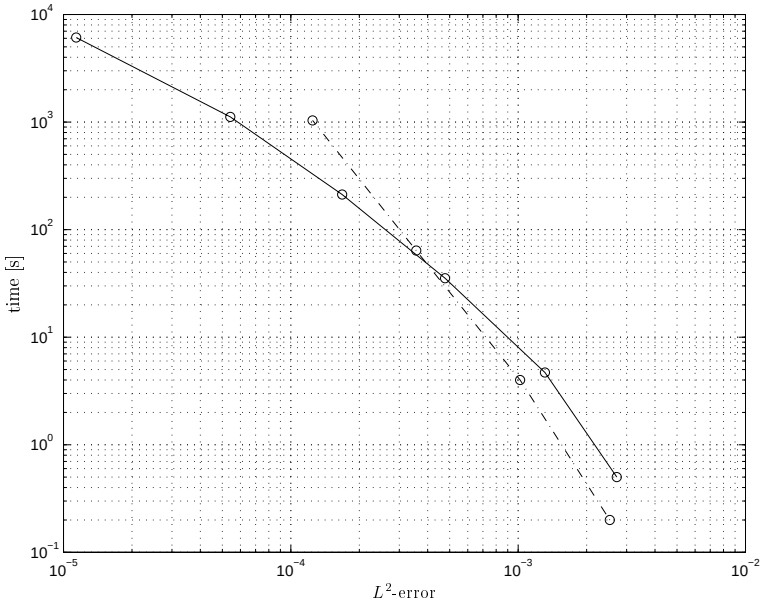

Figure 4: CPU-Time versus L^2-error.

level	N_L	a, α	time[s]	mem[MB]	it	cpr
7	98304	0.1, 1.0	6349.3	153.72	18	0.002
7	98304	0.3, 1.0	6449.6	213.44	18	0.003
7	98304	0.6, 1.0	6862.0	335.46	18	0.005
7	98304	0.9, 1.0	7286.9	467.95	18	0.006
7	98304	1.2, 1.0	7910.1	641.21	18	0.009

Table 2: Second experiment

they finally take the same slope. However, if the amount of compression is reduced by means of parameter α instead of a, this is not the case as predicted by Theorem 16 and observed in the last experiment (Table 3, Figure 6).

We point out that the influence of the amount of compression on the computing time, in particular the time of assembly, is small compared to the influence on the memory consumption (Tables 2,3). The reason for this is that the time to evaluate an entry of the stiffness matrix depends on the distance of the supports of the related wavelets whereas the amount of memory to store the value is always the same. Increasing the thresholds means adding entries to the matrix with more or less distant support, which can be computed very fast compared to the entries near the diagonal. The time of solution, however, increases as fast as the memory.

References

[1] G. Beylkin, R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithms I. *Comm. Pure and Appl. Math.*, 44:141–183, 1991.

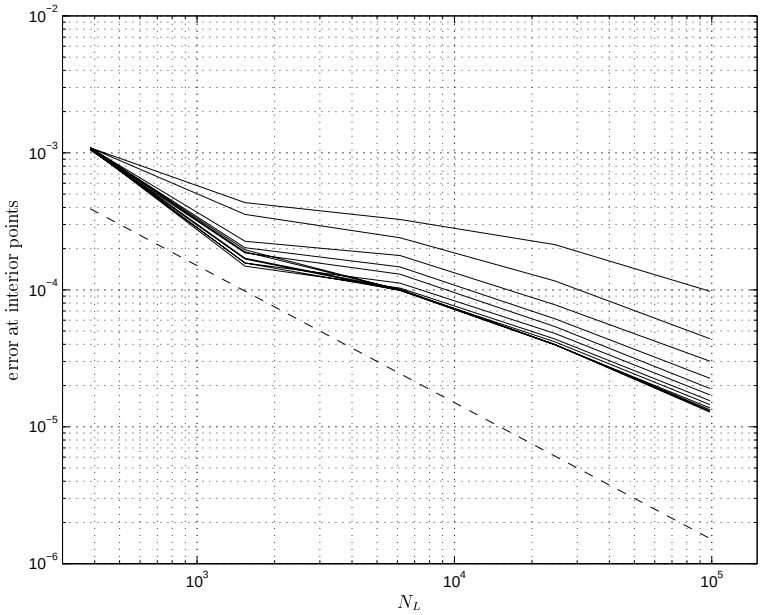

Figure 5: Error at interior points versus N_L, $a = 0.1, \ldots, 1.2$, $\alpha = 1.0$.

level	N_L	a, α	time[s]	mem[MB]	it	cpr
7	98304	1.0, 0.2	6502.3	266.23	18	0.004
7	98304	1.0, 0.4	6577.4	295.87	18	0.004
7	98304	1.0, 0.6	6682.4	345.34	18	0.005
7	98304	1.0, 0.8	7003.2	422.76	18	0.006
7	98304	1.0, 1.0	7472.4	518.97	18	0.007

Table 3: Third experiment.

[2] S. Dahlke, W. Dahmen, R. Hochmuth, and R. Schneider. Stable multiscale bases and local error estimation for elliptic problems. *Advances in Computational Mathematics*, 3, 1997.

[3] W. Dahmen, S. Prößdorf, and R. Schneider. Wavelet approximation methods for pseudodifferential equations. II: Matrix compression and fast solution. *Advances in Computational Mathematics*, 1:259–335, 1993.

[4] M. Dauge. *Elliptic Boundary Value Problems on Corner Domains*, volume 1341 of *Lecture Notes in Mathematics*. Springer-Verlag, 1988.

[5] J. Elschner. The double layer potential operator over polyhedral domains II: Spline Galerkin methods. *Math. Meth. Appl. Sci.*, 15:23–37, 1992.

[6] G.C. Hsiao and W.L. Wendland. On a boundary integral method for some exterior problems in elasticity. *Math. Mech. Astron.*, 257(18):31–60, 1985. Proc. Tiblisi University (Trudy Tibliskogo Ordena Trud. Krasn. Znam. Gosud. Univ.) UDK 539.3.

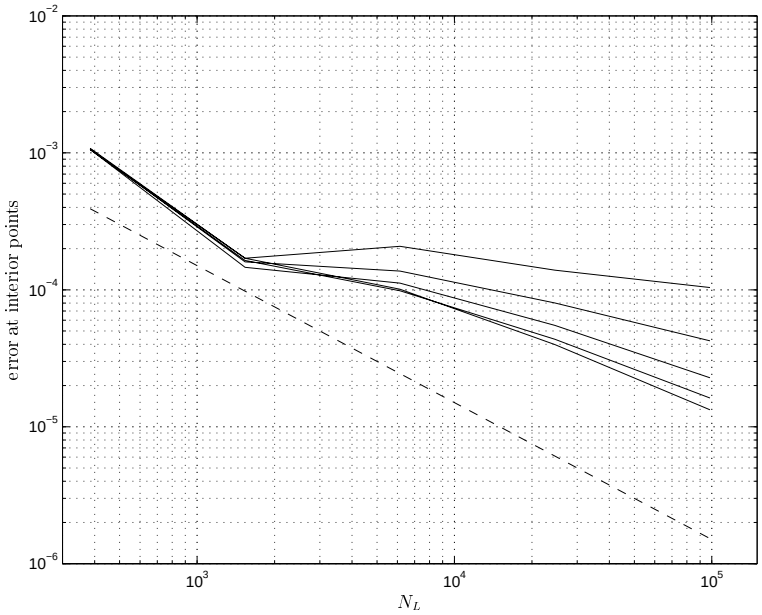

Figure 6: Error at interior points versus N_L, $a = 1.0$, $\alpha = 0.2, \ldots, 1.0$.

[7] Ch. Lage. *Softwareentwicklung zur Randelementmethode: Analyse und Entwurf effizienter Techniken*. PhD thesis, Universität Kiel, 1996. In german.

[8] Ch. Lage. The application of object oriented methods to boundary elements. *Computer Methods in Applied Mechanics and Engineering*, 1997. In press.

[9] Ch. Lage and Ch. Schwab. A Wavelet-Galerkin BEM on polyhedral surfaces in $\mathbb{R}^3$. In W. Hackbusch and G. Wittum, editors, *Numerical Treatment of Multi-Scale Problems*, Notes on Numerical Fluid Mechanics, Braunschweig, Wiesbaden, Germany, 1997. Vieweg Verlag.

[10] Y. Meyer. *Ondelettes et opérateurs Vol. 1: Ondelettes, Vol. 2: Opérateurs de Caldéron-Zygmund*. Hermann & Cie. Publ., Paris, 1990.

[11] S. Sauter. Cubature techniques for 3-D Galerkin BEM. In W. Hackbusch and G. Wittum, editors, *Boundary Elements: Implementation and Analysis of Advanced Algorithms*, volume 54 of *Notes on Numerical Fluid Mechanics*, Braunschweig, Wiesbaden, Germany, 1996. Vieweg Verlag.

[12] R. Schneider. Multiskalen- und Wavelet-Matrixkompression: Analysisbasierte Methoden zur effizienten Lösung grosser vollbesetezter Gleichungssysteme. Habilitation Thesis, Technical University Darmstadt, Germany, 1995.

[13] Ch. Schwab. Variable order composite quadrature of singular and nearly singular integrals. *Computing*, 53:173–194, 1994.

[14] E. Tadmor. The exponential accuracy of fourier and chebyshev differencing methods. *SIAM J. Numer. Anal.*, 23:1–23, 1986.

[15] T. von Petersdorff, R. Schneider, and Ch. Schwab. Multiwavelets for second kind integral equations. Technical Report BN-1153, IPST, University Maryland College Park, 1994. In press in SINUM.

[16] T. von Petersdorff and Ch. Schwab. Fully discrete multiscale Galerkin BEM. Technical Report 95-08, Seminar für Angewandte Mathematik, ETH Zürich, CH-8092 Zürich, 1995. To appear in the volume "Multiresolution Analysis and Partial Differential Equations", W. Dahmen, P. Kurdila and P. Oswald (Eds.), in the series "Wavelet Analysis and its Applications", Academic Press (1997).

[17] W.L. Wendland. Strongly elliptic boundary integral equations. In A. Iserles and M. Powell, editors, *The State of the Art in Numerical Analysis*, pages 511–561. Clarendon Press Oxford, 1987.