A Multiscale Method for the Double Layer Potential Equation on a Polyhedron

W. Dahmen, B. Kleemann, S. Prößdorf and R. Schneider *

Abstract

This paper is concerned with the numerical solution of the double layer potential equation on polyhedra. Specifically, we consider collocation schemes based on multiscale decompositions of piecewise linear finite element spaces defined on polyhedra. An essential difficulty is that the resulting linear systems are not sparse. However, for uniform grids and periodic problems one can show that the use of multiscale bases gives rise to matrices that can be well approximated by sparse matrices in such a way that the solutions to the perturbed equations exhibits still sufficient accuracy. Our objective is to explore to what extent the presence of corners and edges in the domain as well as the lack of uniform discretizations affects the performance of such schemes. Here we propose a concrete algorithm, describe its ingredients, discuss some consequences, future perspectives, and open questions, and present the results of numerical experiments for several test domains including non-convex domains.

1 Introduction

When dealing with constant coefficient elliptic partial differential equations an alternative to the classical finite element approach is offered by the *boundary element method* and consists in the reduction to boundary integral equations via Green's identities. For instance, the Dirichlet problem for Laplace's equation on a bounded and simply connected polyhedron $\mathcal{P} \subseteq I\!R^3$ or the Neumann problem for the same equation on $I\!R^3 \setminus \mathcal{P}$ can be reduced to the second kind integral equation

$$Au = f \tag{1.1}$$

over the boundary $\Omega := \partial\mathcal{P}$ (cf. e. g. [29]), where $A = I + 2W$ and W is the double layer potential operator (cf. (2.2)). In particular, this *exterior domain problem* is an example for potential advantages of the boundary element method since it avoids discretizing unbounded domains.

*The second author has been supported by a grant of Deutsche Forschungsgemeinschaft under grant number Pr 336/2-1.

The most popular and frequently used schemes for the numerical solution of $Au = f$ are based on Galerkin, collocation, and quadrature (= Nyström) methods. However, all these schemes share one common drawback. They require solving linear systems $\mathbf{A}^j_\varphi \mathbf{u}^j = \mathbf{f}^j$ with matrices $\mathbf{A}^j_\varphi$ that are *dense*, i.e, whose entries are essentially all different from zero. Resulting storage requirements as well as the fact that this kind of problem doesn't fit the usual type of solvers for large scale problems therefore restrict the allowable number of unknowns, say $N = \mathcal{O}(4^j)$, drastically, since the computational complexity for generating the matrix $\mathbf{A}^j_\varphi$ and computing the solution $\mathbf{u}^j$ is at least $\mathcal{O}(N^2)$. Since one still expects iterative schemes to be best suited the bulk of computations lies in matrix generation and matrix vector multiplication. Rokhlin [36] and independently Hackbusch and Nowak [22] proposed a way for significantly reducing the computational complexity of these tasks. Their algorithms are based on explicit multipole expansions of the underlying potential. In particular, the algorithm in [22] concerns boundary element methods. An intriguing alternative to this approach was recently proposed by Beylkin, Coifman and Rokhlin [5] based on *multiscale decompositions* of appropriate trial spaces spanned by compactly supported wavelets. A multigrid algorithm for fast matrix vector multiplication which was developed independently by Brandt/Lubrecht and Harten [6, 23] is actually closely related to the wavelet method. The crucial point here is that the particular structure of the wavelet basis makes the entries of the corresponding stiffness matrices decay away from the diagonal so that the computational complexity can be reduced by discarding entries that stay below a given threshold. This is quite similar in spirit to the classical field of applications for wavelets, namely image compression, in that wavelets provide here sparse approximations to the underlying Schwartz kernel.

These facts as well as recent results on preconditioning (see e.g. [11]) have encouraged studying wavelet methods not only for signal and image analysis but also for the numerical treatment of operator equations. Without trying to be exhaustive we briefly mention a few corresponding attempts in this direction. Alpert [1] proposed a multiwavelet approach dealing with discontinuous piecewise polynomials (p-method) and their orthogonalization. This approach looks quite attractive. But its practical realization is rather complicated since higher order polynomials must be implemented. Moreover these functions are not appropriate for the discretization of the normal derivative of the double layer potential. Another idea is based on Cliffords analysis [2]. We had already mentioned the startling work [5] dealing with Galerkin approximations. Motivated by these results a systematic study of a wide class of *generalized Petrov-Galerkin schemes*, for a similarly wide class of pseudo-differential equations was undertaken in [12, 13, 14]. This covers collocation as well as classical Galerkin and Petrov-Galerkin methods. The class of operators contains all classical pseudo-differential operators, e.g. those contained in the Hörmander class as well as those arising in connection with the boundary element method. Here one main objective was to develop a general stability analysis that could be applied to convergence and compression issues. In fact, in this framework the general interplay between ingredients like regularity and degree of exactness of trial spaces, as well as of certain dual spaces can be clarified to a large extent. In particular, this suggests some advantages of biorthogonal wavelets over orthonormal ones. However, rigorous results could be only obtained at the expense of confining the discussion to periodic problems.

The objective of this paper is therefore to study the performance of analogous multiscale versions of the boundary element method for closed polyhedral surfaces. However, instead of treating Galerkin schemes as in [5] we focus here essentially on collocation which is used quite frequently in applications. In principle, this is covered by the general analysis in [13] which, however, as mentioned above, is confined to periodic problems, i.e., smooth domains, and uniform grids. The present setting is essentially different. Firstly, we have to deal with domains that are not smooth but have corners and edges causing strong singularities of the kernel functions. Secondly, the topology of the domain excludes the use of uniform grids and forces us to consider functions on closed surfaces. Consequently, we have to dispense with employing classical wavelets and most parts of the corresponding classical machinary, but have to work with suitable more flexible multiscale decompositions of sequences of nested trial spaces on closed surfaces. Our primary goal is to get a better idea to what extent the above mentioned adverse facts affect the performance of multiscale techniques similar to those studied in [13]. Moreover, we wish to see whether possibly simple and robust ingredients suffice in practice to provide sufficiently accurate results. Therefore we confine the discussion here to piecewise linear finite elements although higher order elements could be handled analogously. Our procedure can roughly be described as follows. As mentioned before, the initial matrix $\mathbf{A}^j_\varphi$ is not sparse. And so is $\mathbf{A}^j \approx \mathbf{A}^j_\varphi$ which is merely an approximation of the exact collocation matrix $\mathbf{A}^j_\varphi$ obtained by evaluating the involved integrals by means of a suitable quadrature formula. As a next step we construct a multiscale decomposition of our finite element spaces. This gives rise to a certain pyramid-type algorithm which transforms $\mathbf{A}^j$ into a matrix $\mathbf{A}^j_\psi$ corresponding to the respective multiscale basis. Most of the entries of this matrix will be seen to decay more rapidly away from the diagonal than those in $\mathbf{A}^j_\varphi$ (and in $\mathbf{A}^j$). The idea is then to replace all entries whose modulus stays below a given threshold by zero ending up with a sparse approximation $\tilde{\mathbf{A}}^j_\psi$ of $\mathbf{A}^j_\psi$. This step is called *matrix compression* or *sparsification*. It significantly reduces storage requirements and facilitates application of efficient sparse matrix solvers. For the setting considered in [13] it could be shown that a proper balance of compression and consistency errors gives rise to solutions of the compressed scheme which meet prescribed error tolerances and in some cases even exhibit optimal asymptotical convergence rates. Lacking most of the analytic tools from the setting in [13] we are therefore mostly interested at this stage in the interplay between compression and accuracy of the final numerical solution. Therefore we have not tried to optimize all parts of the algorithm yet. For instance, we presently still compute initially the full matrix $\mathbf{A}^j$ which is an order N^2 procedure. Ultimately one would want to compute only essentially those entries that are needed to obtain the compressed matrix $\tilde{\mathbf{A}}^j_\psi$. Also the use of graded grids near the edges would be appropriate. For the sake of simplicity we have kept the discretization uniform on each facet of the polyhedron but are very pleased to observe that this simple version already produces surprisingly good results. In that sense this paper is to be understood mainly as a progress report on studying as we believe very promising concepts. To our knowledge this is the first application of multiscale decomposition techniques in connection with the boundary element method for $3D$-problems. Therefore we focus here on describing and discussing the ingredients of the algorithm and the numerical tests conducted so far. A more detailed theoretical treatment will be postponed to a forthcoming paper.

The paper is organized as follows. In Section 2 we pose the analytical problem and describe the discretized collocation method. In Section 3 we develop the ingredients of a corresponding multiscale decomposition of the trial spaces and the associated transformations. Section 4 is devoted to matrix compression while some results on numerical tests are reported on in Section 5. We conclude with some discussion of the results and of future perspectives in Section 6.

2 Discretization of the Double Layer Potential Equation

2.1 The Double Layer Potential Operator

It is well known that the Dirichlet problem for Laplace's equation on some domain $\mathcal{P} \subset I\!R^3$ can be reduced to solving the double layer potential equation

$$Au = f \tag{2.1}$$

over the boundary $\Omega = \partial\mathcal{P}$ (cf. e. g. [29]) where we will assume in the following that $\mathcal{P}$ is a polyhedron. Here $A = I + 2W$ and W is given by

$$Wu(x) := [1/2 - \theta_\Omega(x)]u(x) + \frac{1}{4\pi}\int\limits_\Omega \frac{n_y \cdot (x-y)}{|y-x|^3} u(y) d_y\Omega, \tag{2.2}$$

where $\theta_\Omega(x)$ is the inner solid angle of Ω at $x \in \Omega$ and n_y denotes the unit vector of the interior normal to $\mathcal{P}$ at y. The fact that the boundary Ω is not smooth entails certain theoretical and practical complications. One needs nonuniform meshes and the operator W is not compact. The kernel function $k(x,y) := \frac{1}{4\pi} n_y \cdot (x-y)|y-x|^{-3}$ vanishes if x and y are located on the same face of Ω. However, if y and x tend to a point on some edge of Ω but remain on different faces, then $k(x,y)$ is of order $|x-y|^{-2}$. Thus the kernel function of W has fixed strong singularities along the edges.

The result of [31] about the boundedness of Cauchy singular integral operators on Lipschitz curves implies the L^2- boundedness of the double layer potential operator on Lipschitz domains. Indeed the double layer potential operator is a Calderón-Zygmund operator. In order to make use of the available theory for double layer potential operators on Lipschitz-surfaces, we will assume throughout this paper that the polyhedron is a Lipschitz-domain with boundary $\Omega \subset I\!R^3$ (cf. [32]).

All examples of polyhedra which are treated in our numerical experiments fulfil this condition. Figure 1 shows one example.

We conclude this subsection with a brief overview over the various existing numerical schemes for the approximate solution of $Au = f$. Probably the first method was the so called *panel method*, based on piecewise constant collocation (cf.[42, 26, 43, 3]). This method was proved to converge in supremum norm provided that $\mathcal{P}$ satisfies a certain condition introduced by Wendland in [42]. Moreover, Kral and Wendland [28] showed that the panel method is stable for the case of certain rectangular domains $\mathcal{P}$ including those used in our tests. The proof of the stability for the piecewise linear collocation is completely analogous. However, the proof of stability for the discretized collocation is still open. We only remark that stability can be enforced also for discretized collocation by means of certain modifications of the discretization near the edges which was shown for a second order method in [25]. Furthermore, collocation with piecewise quadratic trial functions was considered in [4]. Elschner [19] analysed the Galerkin method with piecewise polynomial trial functions over arbitrary polyhedra, and the Galerkin method combined with an approximation of the Lipschitz boundary by smooth surfaces was investigated by Dahlberg and Verchota [10].

2.2 Mesh Generation

To keep the implementation as simple as possible we consider here only regular triangulations of the domain, i.e., through the process of mesh refinement all triangles remain geometrically similar to a finite number of initial triangles so that, in particular, the ratios of the radius of circumscribed and inscribed circles remain bounded for all the triangles involved. Nevertheless this will be seen to yield already sufficiently accurate results. Here, of course, a collection of triangles is called a triangulation of Ω if the intersection of any two of the triangles is either a common edge or vertex or empty and if the union of all triangles is Ω. The corresponding simplicial complex consisting of triangles, edges and corners will be referred to as *mesh*.

All meshes to be considered in the sequel are built through successive regular refinements of an initial mesh Ω^0 which corresponds to some initial triangulation of Ω. Such a triangulation typically has only the extreme points of $\mathcal{P}$ as vertices. An example for the L-block is depicted in Figure 1. Subdividing each triangle in Ω^0 into four congruent subtriangles generates Ω^1 and iterating this process gives rise to a sequence of nested meshes Ω^j of *depth* or *level* $j \in I\!N_0$ (see Figure 2). Clearly the *meshsize* h on the level j is proportional to 2^{-j}. Of course, this construction is frequently used in finite element and boundary element schemes.

The vertices x_K of the triangles are called the *knots* or *knot points* of the mesh. Their indices K will be collected in a corresponding *grid* $\square^j = \{K : x_K \text{ is a knot in } \Omega^j\}$. It is convenient to introduce a distance between $K, K' \in \square^j$ as the rescaled distance between the corresponding knots

$$|K - K'| := 2^j |x_K - x_{K'}| \ , \ \ K, K' \in \square^j \ . \tag{2.3}$$

By construction the grids are nested as well

$$\square^0 \subset \square^1 \subset \cdots \subset \square^{j-1} \subset \square^j, \tag{2.4}$$

and the set of additional knots obtained when passing from level $j-1$ to level j is denoted by

$$\Delta^{j-1} := \square^j \backslash \square^{j-1} \ . \tag{2.5}$$

By

$$N = N_j = \# \square^j \tag{2.6}$$

we will always denote the number of knots on the $j--$th level and therefore the number of unknowns of the linear systems studied below. N_j is proportional to 2^{2j} with constants depending only on the initial triangulation Ω^0.

2.3 Finite Element Spaces

A little care has to be taken when talking about functions defined on Ω since Ω is not topologically equivalent to a planar domain. Thus whenever explicit representations of a function are needed it seems most convenient to express its restriction to any triangle in Ω^0 (or in fact to any subtriangle of some triangle in Ω^0) in terms of the respective barycentric coordinates. In particular, this readily allows us to define for each triangle τ of Ω^0 the canonical *Courant hat functions* φ_K^j on τ relative to $\Omega^j \cap \tau$, i.e., each such function restricted to any $\tau \in \Omega^j$ is piecewise linear and continuous on τ satisfying the nodal conditions

$$\varphi_K^j(x_{K'}) = 2^j \delta_{K,K'}, \tag{2.7}$$

for all $K, K' \in \square^j|_\tau$. When x_K is a knot in the relative interior of a common edge e of two neighboring triangles $\tau, \tau' \in \Omega^0$ the two corresponding hat functions on these two triangles assemble automatically to a single nodal function φ_K^j which is now defined on $\tau \cup \tau'$ and is continuous on all of Ω. Analogously, when x_K is an extreme point of $\mathcal{P}$, i.e., a vertex in Ω^0 the function φ_K^j is composed of several pieces coming from the adjacent triangles. This uniquely defines φ_K^j on all of Ω as a piecewise linear continuous function satisfying (2.7) for all $K, K' \in \square^j$ and

$$\operatorname{supp} \varphi_K^j = \cup\{\tau \in \Omega^j : x_K \in \tau\}. \tag{2.8}$$

Likewise integration on Ω is reduced to integration on the planar faces of $\mathcal{P}$ and hence on $I\!R^2$. Thus the norm $\|\cdot\|_{L_2(\Omega)}$ may be defined by reduction to norms on planar domains via

$$\|u\|^2_{L_2(\Omega)} := \sum_{\tau \in \Omega^0} \|u\|^2_{L_2(\tau)}. \tag{2.9}$$

The normalization in (2.7) then ensures the existence of positive constants c_1, c_2 independent of j such that

$$c_1 \leq \|\varphi_K^j\|_{L_2(\Omega)} \leq c_2\ . \tag{2.10}$$

Moreover, with this normalization one has *stability* (Riesz-basis), i.e., there exist positive constants C_1, C_2 such that

$$C_1(\sum_{K\in\square^j} |u_K^j|^2)^{\frac{1}{2}} \leq \|\sum_{K\in\square^j} u_K^j \varphi_K^j\|_{L^2(\Omega)} \leq C_2(\sum_{K\in\square^j} |u_K^j|^2)^{\frac{1}{2}} \tag{2.11}$$

holds uniformly in $j \in I\!N_0$ and any coefficient vector $\mathbf{u}^j = (u_K^j : K \in \square^j)$. Since obviously, by (2.7),

$$\varphi_K^j = 2^{-j-1} \sum_{K'\in\square^{j+1}} \varphi_K^j(x_{K'})\varphi_{K'}^{j+1} \tag{2.12}$$

the spaces

$$V^j := \text{ span } \{\varphi_K^j : K \in \square^j\} \tag{2.13}$$

of piecewise linear continuous functions on Ω are nested

$$V^0 \subset V^1 \subset \cdots \subset V^{j-1} \subset V^j. \tag{2.14}$$

Finally, restriction to the planar faces of $\mathcal{P}$ as above readily yields second order accuracy of the trial spaces

$$\inf_{u^j \in V^j} \|u^j - u\|_2 = \mathcal{O}(h^2), \tag{2.15}$$

where h is the meshsize of Ω^j, provided that the restrictions of $u \in C(\Omega)$ to any $\tau \in \Omega^0$ has second order weak derivatives in $L_2(\tau)$.

So far we have confirmed the validity of the basic properties of what would be called *multiresolution analysis* in a classical wavelet context for the present setting of nested trial spaces on a polyhedral manifold. Of course, piecewise polynomial trial functions of higher degree could be defined along similar lines. As in the case of B-splines one would not have to insist on interpolatory basis functions as above.

2.4 Discretization

The knot collocation method on the finest grid $\square^j$ consists in finding a piecewise linear and continuous function $u^j \in V^j$ such that

$$Au^j(x_K) = f(x_K) \quad , \quad K \in \square^j \ , \tag{2.16}$$

which requires solving a linear system with coefficient matrix whose entries have the form

$$(A\varphi^j_{K'})(x^j_K) = \frac{1}{2\pi} \int\limits_{\operatorname{supp}\varphi^j_{K'}} \frac{n_y \cdot (x^j_K - y)}{|y - x^j_K|^3} \varphi^j_{K'}(y) d_y\Omega, \quad K, K' \in \square^j, K \neq K'. \tag{2.17}$$

To avoid analytic integration, when computing these entries, we approximate the involved integrals with the aid of a quadrature formula which is known to be exact for polynomials p of degree at most two (e.g. [39]). To describe this, let $\tau = [x_{K_1}, x_{K_2}, x_{K_3}] \in \Omega^j$ and let $x_{K'_1}, x_{K'_2}, x_{K'_3}$ denote the midpoints of its edges $[x_{K_1}, x_{K_2}]$, $[x_{K_2}, x_{K_3}]$,$[x_{K_3}, x_{K_1}]$, respectively (cf. Figure 2). The formula then reads

$$\int\limits_\tau v(x) \simeq \frac{1}{3} \sum_{i=1}^{3} v(x_{K'_i}) |\tau| dx. \tag{2.18}$$

Moreover, to ensure a proper treatment of the singularities we employ a regularization technique, which is sometimes called singularity subtraction (cf. [25]). In fact, taking into account that the constant function is an eigenfunction of W corresponding to the eigenvalue 1/2 (cf. [29], Sect.1.3), the equation $Au = f$ may be written as

$$2u(x) + \frac{1}{2\pi} \int\limits_\Omega \frac{n_y \cdot (x - y)}{|y - x|^3} [u(y) - u(x)] d_y\Omega = f(x), \quad x \in \Omega. \tag{2.19}$$

The subtraction technique for discretizing the collocation method consists now in applying the above quadrature rule to (2.19) for $x = x_K, K \in \square^j$ and u replaced by a piecewise linear trial function $u^j \in V^j$. Now it is not hard to see that this procedure leads to approximations $a^j_{K,K'}$ for the entries $(A\varphi^j_{K'})(x^j_K)$ of the exact collocation matrix, which are the usual quadrature approximations of the quantities in (2.17) when $K \neq K'$, while the diagonal entries are given by

$$a^j_{K,K} = 2 - \sum_{K' \in \square^j, K' \neq K} a^j_{K,K'}. \tag{2.20}$$

The quadrature method described above is relatively easy to implement and the computation of entries for the corresponding stiffness matrix is quite fast. It leads to the linear system

$$\mathbf{A}^j \mathbf{u}^j = \mathbf{f}^j, \quad \mathbf{A}^j = (a^j_{K,K'})_{K,K' \in \square^j}. \tag{2.21}$$

In view of (2.15) the overall convergence of the scheme is expected to be at most quadratic i.e. $\mathcal{O}(h^2)$. In fact, due to the lack of regularity of the solution and the presence of corners and edges in the domain Ω, the order of convergence should actually be lower in practice. However, in spite of these adverse facts we observe surprisingly good accuracy and convergence rates (see Table 1).

Former excerpts from our numerical experiments mentioned in [14, 38] were obtained without singularity subtraction. These results were observed to be sufficiently accurate when $\mathcal{P}$ is a cube. But the results turned out to be visibly worse for nonconvex domains. Therefore we have incorporated the above singularity subtraction in all of the present examples.

3 Multiscale Decomposition

The concept of *multiresolution analysis*, mainly due to Mallat and Meyer, plays a fundamental role for the construction, analysis, as well as for applications of wavelets. While initially orthogonal decompositions of shift invariant spaces were considered, more general multiscale decomposition techniques have emerged from this concept (see e.g. [7, 23]). Similar techniques such as the frequency decomposition multigrid method [20, 21] came up in different contexts. In this section we describe the type of general multiscale decomposition techniques which will be used in our algorithm. Some theoretical background can be found in [7, 23].

3.1 Decomposition of Function Spaces

The basic philosophy of multiscale techniques is to view objects on a high discretization level as successive updates of coarser versions. This is usually realized through corresponding decompositions of underlying function spaces serving as continuous models of the data under consideration. Such decompositions typically rest on certain *two scale relations* which we describe now for the present setting. First we recall from (2.12) the refinement equation

$$\varphi^j_K = \sum_{K' \in \square^{j+1}} m^j_{K',K} \varphi^{j+1}_{K'} \ , \quad K \in \square^j \ , \tag{3.1}$$

where the so called *mask* or *filter* coefficients $m^j_{K',K}$ are, in view of (2.12), given by

$$m^j_{K',K} = 2^{-j-1}\varphi^j_K(x_{K'}), \quad K \in \square^j,\ K' \in \square^{j+1}. \tag{3.2}$$

The $N_{j+1} \times N_j$ matrix $\mathbf{M}^j = (m^j_{K',K})_{K' \in \square^{j+1}, K \in \square^j}$ is therefore sparse. It is often called *subdivision matrix.* In the context of multigrid methods its adjoint usually serves a canonical restriction. Clearly on those grid points located in the interior of the triangles $\tau \in \Omega^0$ it can be represented by the following 7 point stencil

$$\frac{1}{4}\begin{pmatrix} 0 & 1 & 1 \\ 1 & \mathbf{2} & 1 \\ 1 & 1 & 0 \end{pmatrix}. \tag{3.3}$$

Here the bold value indicates a position K in the coarse grid $\square^j$. In addition one needs now a relation which complements the basis in V^j to one for V^{j+1}, i.e., one needs additional functions

$$\psi^j_K := \sum_{K' \in \square^{j+1}} c^j_{K',K}\varphi^{j+1}_{K'}, \quad K \in \Delta^j, \tag{3.4}$$

such that

$$\varphi^{j+1}_K = \sum_{K' \in \square^j} \overline{d^j_{K,K'}}\varphi^j_{K'} + \sum_{K' \in \Delta^j} \overline{g^j_{K,K'}}\psi^j_{K'}. \tag{3.5}$$

More precisely, the objective is to find for a given $\mathbf{M}^j$ additional matrices $\mathbf{C}^j = (c^j_{K',K})_{K' \in \square^{j+1}, K \in \Delta^j}$, $\mathbf{D}^j = (d^j_{K',K})_{K' \in \square^j, K \in \square^{j+1}}$, and $\mathbf{G}^j = (g^j_{K',K})_{K' \in \Delta^j, K \in \square^{j+1}}$ such that the relations (3.4) and (3.5) hold. Of course, (3.5) means that

$$V^{j+1} = V^j \oplus W^j \ , \ \ (\,W^j \cap V^j = \{0\}\,) \tag{3.6}$$

where

$$W^j := \operatorname{span}\{\psi^j_K : K \in \Delta^j\}. \tag{3.7}$$

It is easy to see that this decomposition is equivalent to the matrix relation

$$\mathbf{M}^j(\mathbf{D}^j)^* + \mathbf{C}^j(\mathbf{G}^j)^* = I, \tag{3.8}$$

which is the pivotal ingredient of the subsequent discrete decomposition schemes. Relation (3.8) means that the composed $N_{j+1} \times N_{j+1}$ matrices $(\mathbf{M}^j, \mathbf{C}^j)$ and $(\mathbf{D}^j, \mathbf{G}^j)^*$ are inverses of each other

$$(\mathbf{M}^j, \mathbf{C}^j)(\mathbf{D}^j, \mathbf{G}^j)^* = I = (\mathbf{D}^j, \mathbf{G}^j)^*(\mathbf{M}^j, \mathbf{C}^j). \tag{3.9}$$

Consequently one derives the identities

$$(\mathbf{D}^j)^*\mathbf{M}^j = I\ ,\ (\mathbf{G}^j)^*\mathbf{C}^j = I\ ,\ (\mathbf{D}^j)^*\mathbf{C}^j = 0\ ,\ (\mathbf{G}^j)^*\mathbf{M}^j = 0\ . \tag{3.10}$$

We remark that the stability of the φ_K^j combined with the relation (3.8) or (3.9) implies the stability of the functions $\psi_K^j, K \in \Delta^j$ (cf. [7]), for fixed $j \in IN$. A familiar possible choice for the matrices $\mathbf{C}^j, \mathbf{D}^j, \mathbf{G}^j$ is

$$\begin{aligned} c_{K',K}^j &= \delta_{K',K}, \quad K' \in \Box^{j+1}, K \in \Delta^j, \\ d_{K',K}^j &= 2\delta_{K,K'}, \quad K \in \Box^{j+1}, K' \in \Box^j, \\ g_{K',K}^j &= \begin{cases} -2^{-j}\varphi^j(x_{K'}) & \text{if} \quad K \in \Box^j \\ \delta_{K',K} & \text{if} \quad K \in \Delta^j, \end{cases} \end{aligned} \tag{3.11}$$

which corresponds to so called hierarchical bases for the spaces V^j (cf. [44]). Unfortunately, due to the lack of moment conditions (see (3.28) below), this very handy decomposition is not quite appropriate for our purposes. However, it can be used to derive from it other decompositions which are better suited. More information about this issue can be found in [7] where these questions are discussed in much more generality.

We postpone the actual specification of the matrices $\mathbf{C}^j$ and assume at this point that they are given and that they are sparse, i.e., are defined through stencils with finite support. We remark that according to (3.9) an appropriate choice of $\mathbf{M}^j$ together with $\mathbf{C}^j$ determines the matrices $\mathbf{D}^j$ and $\mathbf{G}^j$.

3.2 Discrete Transformations

The reason for considering the above space decompositions is to facilitate a multiscale decomposition of vectors of the form

$$\mathbf{u}^j = (u_K^j)_{K \in \Box^j}. \tag{3.12}$$

Interpreting these coefficients u_K^j as coefficients of an expansion of some function

$$u^j = \sum_{K \in \Box^j} u_K^j \varphi_K^j \ \in V^j, \tag{3.13}$$

we conclude from (3.5)

$$\begin{aligned} u^j &= \sum_{K \in \Box^j} u_K^j \varphi_K^j = \sum_{K \in \Box^j} u_K^j \left(\sum_{K' \in \Box^{j-1}} \overline{d_{K,K'}^{j-1}} \varphi_{K'}^{j-1} + \sum_{K' \in \Delta^{j-1}} \overline{g_{K,K'}^{j-1}} \psi_{K'}^{j-1} \right) \\ &= \sum_{K' \in \Box^{j-1}} ((\mathbf{D}^{j-1})^* \mathbf{u}^j)_{K'} \varphi_{K'}^{j-1} + \sum_{K' \in \Delta^{j-1}} ((\mathbf{G}^{j-1})^* \mathbf{u}^j)_{K'} \psi_{K'}^{j-1} \\ &= \sum_{K' \in \Box^{j-1}} u_{K'}^{j-1} \varphi_{K'}^{j-1} + \sum_{K' \in \Delta^{j-1}} w_{K'}^{j-1} \psi_{K'}^{j-1} \end{aligned} \tag{3.14}$$

i.e.,

$$\begin{pmatrix} \mathbf{u}^{j-1} \\ \mathbf{w}^{j-1} \end{pmatrix} = \begin{pmatrix} (\mathbf{D}^{j-1})^* \\ (\mathbf{G}^{j-1})^* \end{pmatrix} \mathbf{u}^j. \tag{3.15}$$

Thus, the change of bases taking $u^j \in V^j$ from a *nodal basis* representation relative to the basis $\mathcal{B}^j_\varphi := \{\varphi^j_K : K \in \square^j\}$ into a representation in terms of the *multiscale basis* $\mathcal{B}^j_\psi := \{\varphi^0_K : K \in \square^0\} \cup \{\psi^l_K : K \in \Delta^l,\ l = 0, \ldots, j-1\}$ can be performed with the aid of a *pyramid type scheme*:

$$\begin{array}{ccccccccc} & (\mathbf{D}^{j-1})^* & & (\mathbf{D}^{j-2})^* & & & (\mathbf{D}^0)^* & \\ \mathbf{u}^j & \rightarrow & \mathbf{u}^{j-1} & \rightarrow & \mathbf{u}^{j-2} & \cdots & \rightarrow & \mathbf{u}^0 \\ & \searrow & (\mathbf{G}^{j-1})^* & \searrow & (\mathbf{G}^{j-2})^* & & \searrow & (\mathbf{G}^0)^* \\ & & \mathbf{w}^j & & \mathbf{w}^{j-1} & \cdots & & \mathbf{w}^0\ . \end{array} \tag{3.16}$$

Conversely, since by (3.1) and (3.4),

$$\sum_{K' \in \square^{j-1}} f^{j-1}_{K'} \varphi^{j-1}_{K'} + \sum_{K' \in \Delta^{j-1}} w^{j-1}_{K'} \psi^{j-1}_{K'} = \sum_{K \in \square^j} \left((\mathbf{M}^{j-1}\mathbf{u}^{j-1})_K + (\mathbf{C}^{j-1}\mathbf{w}^{j-1})_K \right) \varphi^j_K, \tag{3.17}$$

one has

$$\mathbf{u}^j = (\mathbf{M}^{j-1}, \mathbf{C}^{j-1}) \begin{pmatrix} \mathbf{u}^{j-1} \\ \mathbf{w}^{j-1} \end{pmatrix}. \tag{3.18}$$

Thus, given the representation

$$u^j = \sum_{K \in \square^0} u^0_K \varphi^0_K + \sum_{l=0}^{j-1} \sum_{K \in \Delta^l} w^l_K \psi^l_K, \tag{3.19}$$

the coefficient vector $\mathbf{u}^j$ of u^j relative to the nodal basis $\mathcal{B}^j_\varphi$ is obtained by the following reverse process

$$\begin{array}{ccccccccc} \mathbf{w}^0 & & \mathbf{w}^1 & & \mathbf{w}^2 & \cdots & \mathbf{w}^{j-1} & & \\ & \searrow & \mathbf{C}^0 & \searrow & \mathbf{C}^1 & & & \searrow & \mathbf{C}^{j-1} \\ \mathbf{u}^0 & \rightarrow & \mathbf{u}^1 & \rightarrow & \mathbf{u}^2 & \cdots & \mathbf{u}^{j-1} & \rightarrow & \mathbf{u}^j \\ & \mathbf{M}^0 & & \mathbf{M}^1 & & & & \mathbf{M}^{j-1} & \end{array} \tag{3.20}$$

which involves now the matrices $\mathbf{M}^l, \mathbf{C}^l$ (see also [7]). Only those matrices will be needed in the present context.

One has to keep in mind that our primary task is to generate from the matrix $\mathbf{A}^j_\varphi$ the matrix $\mathbf{A}^j_\psi$ relative to the multiscale basis $\mathcal{B}^j_\psi$ and this procedure also requires only the matrices $\mathbf{M}^l, \mathbf{C}^l$. To see this denote by Φ^l, Ψ^l the row vectors consisting of the functions $\varphi^l_{K'}, K' \in \square^l$ and $\psi^l_{K'}, K' \in \Delta^l$, respectively. In these terms (3.1) and (3.4) can be rewritten as

$$\Phi^{j-1} = \Phi^j \mathbf{M}^{j-1}, \quad \Psi^{j-1} = \Phi^j \mathbf{C}^{j-1}, \tag{3.21}$$

so that

$$A(\Psi^{j-1}, \Phi^{j-1}) = A\Phi^j (\mathbf{C}^{j-1}, \mathbf{M}^{j-1}). \tag{3.22}$$

Since $(A\Phi^j)(x_K)$ is the Kth row of the matrix $\mathbf{A}^j_\varphi$, iterating the above transformation yields

$$\mathbf{A}^j_\varphi \mathbf{L}^{j-1}\cdots\mathbf{L}^0 =: \mathbf{A}^j_\varphi \mathbf{T}^j, \tag{3.23}$$

where

$$\mathbf{L}^l := \left(\begin{array}{c|c} (\mathbf{C}^l, \mathbf{M}^l) & 0 \\ \hline 0 & \mathbf{I} \end{array}\right), \quad l = 0, \cdots, j-1. \tag{3.24}$$

Thus whenever also the matrices $\mathbf{C}^l$ are sparse the transformation (3.23) requires only $\mathcal{O}(N)$ nontrivial operations where as before $N = \#\Box^j$.

We will see below (cf. (4.5)) that the entries of $\mathbf{A}^j_\varphi \mathbf{T}^j$ decay more rapidly away from the diagonal than those of $\mathbf{A}^j_\varphi$. Our experiments show that this remains true for $\mathbf{A}^j\mathbf{T}^j$ where $\mathbf{A}^j$ is the approximation of $\mathbf{A}^j_\varphi$ obtained by the quadrature (2.18). This suggests approximating $\mathbf{A}^j\mathbf{T}^j$ by a sparse matrix. However, our numerical tests have revealed that such a sparsification works even better an the equivalent system

$$(\mathbf{T}^j)^*\mathbf{A}^j\mathbf{T}^j u^j_\psi = (\mathbf{T}^j)^* f \tag{3.25}$$

where $f_K = f(x_K), K \in \Box^j$, and $\mathbf{u}^j_\psi$ is given by

$$\mathbf{u}^j = \mathbf{L}^{j-1}\cdots\mathbf{L}^0\mathbf{u}^j_\psi =: \mathbf{T}^j u^j_\psi. \tag{3.26}$$

Note that the transformation yielding

$$\mathbf{A}^j_\psi := (\mathbf{T}^j)^*\mathbf{A}^j\mathbf{T}^j \tag{3.27}$$

corresponds to a change of bases arising in connection with Galerkin schemes. In [13] it is referred to as *wavelet representation*. There the decay of the entries was studied systematically for periodic problems. Below in Section 4 we will therefore be content with briefly indicating the basic idea.

We now turn to specify the complementary matrices $\mathbf{C}^l$. To this end, recall from [5, 13] that the compression rate for the matrices $\mathbf{A}^j_\psi$ is expected to depend on the *order* d of vanishing moments of the functions ψ^l_K. By this we mean that

$$\int_\tau \psi^l_K(x)p(x)dx = 0, \quad K \in \Delta^l|_\tau,\ l = 0, \ldots, j-1, \tag{3.28}$$

holds for each $\tau \in \Omega^0$ and every polynomial p of degree less than d. Since the exactness of the quadrature (2.18) limits the overall convergence rate to two we expect vanishing moments of order $d = 2$ to be sufficient. It is shown in [7] how to construct from (3.11) matrices $\mathbf{C}^l, \mathbf{D}^l, \mathbf{G}^l$ which are all sparse such that the corresponding spaces W^l in (3.6) have the form $W^l = (Q^{l+1} - Q^l)V^{l+1}$ where the Q^l are local projectors onto the spaces V^l possessing uniformly bounded L_2-norms. However, here our objective is to explore the efficiency of possibly simple schemes, i.e, masks with possibly small support. Therefore for the present numerical studies we have chosen $\mathbf{C}^l$ to be of the following form.

Each knot point x_L, $L \in \Delta^j$, is on a line connecting two neighboring points on the coarse grid x_{K^l} and x_{K^r}, $K^l, K^r \in \Box^j$. In general we define for $a := |x_L - x_{K^l}|$ and $b := |x_L - x_{K^r}|$ the entries of $\mathbf{C}^j$ by

$$c^j_{K,L} := \begin{cases} 2 & \text{when} \quad x_K = x_L, \\ \frac{-2b}{(a+b)} & \text{when} \quad x_K = x_{K^l}, \\ \frac{-2a}{(a+b)} & \text{when} \quad x_K = x_{K^r}, \\ 0 & \qquad\quad \text{else.} \end{cases} \tag{3.29}$$

Since in our construction $x_L = \frac{1}{2}(x_{K^r} + x_{K^l})$ we have $a = b$ so that in the interior of each $\tau \in \Omega^0$ the matrix $\mathbf{C}^j$ corresponds to the following 3 point stencils

$$\begin{pmatrix} 0 & 0 & 0 \\ -1 & \mathbf{2} & -1 \\ 0 & 0 & 0 \end{pmatrix} \frac{1}{2} c_b \ , \quad \begin{pmatrix} 0 & -1 & 0 \\ 0 & \mathbf{2} & 0 \\ 0 & -1 & 0 \end{pmatrix} \frac{1}{2} c_b \ , \quad \begin{pmatrix} 0 & 0 & -1 \\ 0 & \mathbf{2} & 0 \\ -1 & 0 & 0 \end{pmatrix} \frac{1}{2} c_b \ , \quad c_b = 1 \ . \tag{3.30}$$

Again the boldface values correspond to indices $L \in \Delta^j$. In our numerical tests we have varied the normalizing constant c_b. Its choice affects the normalization of the matrix $\mathbf{G}^j$ which is not specified here (and which may not be sparse but have entries that decay exponentially away from the diagonal). The above choice of $\mathbf{C}^j$ ensures that (3.28) holds for $d = 2$ whenever ψ^j_K corresponds to a knot $x_K, K \in \Delta^j$ located in the interior of some triangle $\tau \in \Omega^0$. When K corresponds to a point on an edge of $\mathcal{P}$ (3.28) holds only for $d = 1$. However, since these points correspond to a lower dimensional set one should not expect this to affect the overall accuracy in any significant way.

One should mention that the univariate counterpart of the matrix $\mathbf{C}^j$ already appears in [6, 23]. There it is used for a different purpose, namely to complement the refinement matrix corresponding to the relation (3.1) for the delta-distribution to an invertible matrix analogous to (3.9). In the same way it arises also in [44] in a bivariate setup when converting the hierarchical basis representation into the nodal basis representation.

Of course, one could use other masks from [6, 23] to increase d in (3.28) but we dispense here with this option for the reasons mentioned before.

3.3 Stability, Biorthogonality

If the matrices $(\mathbf{M}^j, \mathbf{C}^j)$ and $(\mathbf{D}^j, \mathbf{G}^j)$ have uniformly bounded L_2-norms, which means that the matrices $(\mathbf{M}^j, \mathbf{C}^j)$ have uniformly bounded condition numbers, for any $u^j = \sum_{K \in \Box^j} u^j_K \varphi^j_K$ the estimates

$$c_1 |\mathbf{u}^j|_{l_2(\Box^j)} \leq \|u^j\|_{L_2(\Omega)} \leq c_2 |\mathbf{u}^j|_{l_2(\Box^j)} \tag{3.31}$$

as well as

$$c_1 \|u^j\|_{L_2(\Omega)} \leq \left(|\mathbf{u}^{j-1}|^2_{l_2(\Box^{j-1})} + |\mathbf{w}^{j-1}|^2_{l_2(\Delta^{j-1})}\right)^{1/2} \leq c_2 \|u^j\|_{L_2(\Omega)} \tag{3.32}$$

hold uniformly in $j \in IN$. For the above choice of $\mathbf{M}^j$ and $\mathbf{C}^j$ this is indeed the case as is already suggested by the following simple observation. By the above remarks concerning its role in [6, 23, 44], $\mathbf{C}^j$ must have full rank. Since its columns are orthogonal to those in $\mathbf{M}^j$ which is also known to have full rank the matrix $(\mathbf{M}^j, \mathbf{C}^j)$ must be invertible. However, this stability between two levels is not quite sufficient. In addition the transformation $\mathbf{T}^j$ in (3.23) should be *stable* which means that the estimate

$$c_1 \|u^j\|_{L_2(\Omega)} \leq \left(|\mathbf{u}^0|_{l_2(\Box^0)} + \sum_{l=0}^{j-1} |\mathbf{w}^l|_{l_2(\Delta^l)} \right)^{1/2} \leq c_2 \|u^j\|_{L_2(\Omega)} \tag{3.33}$$

holds uniformly in $j \in IN$, or equivalently that

$$\operatorname{cond}_2 \mathbf{T}^j = \mathcal{O}(1), \quad j \to \infty \ . \tag{3.34}$$

In this case the basis $\{\psi_L^j : L \in \Delta^j : j \in IN_0\} \cup \{\varphi_K^0 : K \in \Box^0\}$ is an unconditional Schauder basis in $L_2(\Omega)$. As mentioned before the transformation defined by (3.11), which is also used in [6], is not stable.

There are several strategies for establishing stability. Of course, the two-level stability implies global stability if the functions $\psi_L^j, L \in \Delta^j$ are *orthogonal* to the $\varphi_K^j, K \in \Box^j$. Moreover, it has been observed in [38] that the boundedness of a certain integral operator with Calderón-Zygmund kernel is a necessary condition for the stability. After completing this paper we learned that [33, 24] have found an equivalent result phrased in terms of a Carleson condition. In the classical setting of multiresolution analysis for uniform grids on IR^n stability can be guaranteed, even when dealing with non-orthogonal complements W^j, by constructing *biorthogonal* systems of wavelets [9]. A resulting necessary and sufficient condition for stability is, however, not easy to verify in practice.

In the present context this approach amounts to the following task. Given the matrix $\mathbf{M}^j$ find a *completion* $\mathbf{C}^j, \mathbf{D}^j, \mathbf{G}^j$ satisfying (3.8) such that the compounded matrices $(\mathbf{M}^j, \mathbf{C}^j)$ and $(\mathbf{D}^j, \mathbf{G}^j)$ have uniformly bounded L_2-norms. Moreover, the *dual* matrices $\mathbf{D}^j$ have to be chosen so that there exists stable systems of L_2-functions $\eta_K^j, K \in \Box^j$, which are refinable relative to $\mathbf{D}^j$, i.e.,

$$\eta_K^j = \sum_{K' \in \Box^{j+1}} d_{K',K}^j \eta_{K'}^{j+1}, \quad K \in \Box^j. \tag{3.35}$$

If this is the case the functions

$$\zeta_L^j := \sum_{K' \in \Box^{j+1}} g_{K',L}^j \eta_{K'}^{j+1}, \quad L \in \Delta^j, \tag{3.36}$$

satisfy the biorthogonality relations

$$\begin{aligned} (\varphi_K^j, \eta_{K'}^j) &= \delta_{K,K'}, \quad K, K' \in \Box^j, \\ (\psi_L^j, \zeta_{L'}^j) &= \delta_{L,L'}, \quad L, L' \in \Delta^j, \\ (\varphi_K^j, \zeta_L^j) &= (\psi_L^j, \eta_K^j) = 0, \quad K \in \Box^j, \ L \in \Delta^j, \end{aligned} \tag{3.37}$$

where the ψ_L^j are given by (3.4) and $(f,g) := \int\limits_\Omega f\overline{g}$. One should emphasize that the transformation $\mathbf{T}^j$ and hence the whole algorithm requires only the matrices $\mathbf{M}^j, \mathbf{C}^j$ while the dual pair $\mathbf{D}^j, \mathbf{G}^j$ is only needed to construct a complete biorthogonal system of functions as in (3.37), which in turn can be used to establish stability of $\mathbf{T}^j$ (cf. [9]).

4 Matrix Compression

Wavelet approximations to operators have gained considerable importance because of several reasons. They support efficient preconditioning for non zero order operators (cf. [11, 13]), regularization of linear ill-posed problems [18], and matrix compression [5, 13]. The latter issue is closely related to atomic decompositions of Calderón-Zygmund operators. These properties are based on the fact, that wavelets provide unconditional schauder bases for a wide scale of Sobolev, Besov and Triebel-Lizorkin spaces.

One expects these features to remain valid also in a more general multiscale setting of the present type. In particular, the results in [13] for the periodic case suggest that, not only when dealing with Galerkin approximations but also for collocation schemes, the entries in the corresponding matrices decay away from block diagonals (cp. e.g. [5]).

This should facilitate an efficient *compression*, i.e., an approximation of $\mathbf{A}_\varphi^j\mathbf{T}^j$ or $\mathbf{A}_\psi^j$ by a sparse matrix, so that the solution to the perturbed system is still sufficiently accurate. For periodic boundary conditions this could be rigorously confirmed in [13] for a wide class of operators as well as numerical schemes based on uniform grids. Following the lines of [13] entries of $\mathbf{A}_\varphi^j\mathbf{T}^j$ may be estimated (similarly as in the case of Galerkin schemes [5]) as follows. First assume that x_L is some vertex such that the support of ψ_L^j is contained in some triangle $\tau \in \Omega_0$. Exploiting the fact that ψ_L^j has compact support, we obtain, on account of (3.28),

$$
\begin{aligned}
|(A\psi_L^j)(x_K)| &= |\int\limits_\Omega k(x_K,y)\psi_L^j(y)dy| \\
&= |\int\limits_\Omega (k(x_K,y)-p(y))\psi_L^j(y)dy|, \qquad (4.1)
\end{aligned}
$$

where p is any polynomial of degree at most one. Thus, using Schwarz' inequality, taking $\|\psi_L^j\|_{L_2(\Omega)} \sim 1$ into account, and choosing p as the Taylor polynomial of $k(x_K,\cdot)$ at x_L, one obtains

$$
\begin{aligned}
|(A\psi_L^j)(x_K)| &\leq \|k(x_K,\cdot)-p\|_{L_2(\operatorname{supp}\psi_L^j)}\|\psi_L^j\|_{L_2(\Omega)} \\
&\leq c\, 2^{-3j} \max_{|\alpha|=2, y\in\operatorname{supp}\psi_L^j} |D_y^\alpha k(x_K,y)|. \qquad (4.2)
\end{aligned}
$$

The well known estimates for the double layer potential kernel

$$
\begin{aligned}
|k(x,y)| &\leq c|x-y|^{-2} \qquad (4.3)\\
|D_x^\alpha k(x,y)| + |D_y^\alpha k(x,y)| &\leq c|x-y|^{-2-|\alpha|} \ , \quad |\alpha| \leq 2 \ , \qquad (4.4)
\end{aligned}
$$

therefore yield

$$2^{-j}|(A\psi^j_L)(x_K)| \le c\ (\ 2^j|x_K - x_L|)^{-4}. \tag{4.5}$$

In particular, when x_L and x_K are knots in the same mesh one obtains

$$2^{-j}|(A\psi^j_L)(x_K)| \le c\,(1 + |K - L|)^{-4}, \quad K, L \in \Box^j. \tag{4.6}$$

Taking the slightly different normalization in [13] into account, this is in full agreement with the results in [13] for the periodic case.

When ψ^j_L is not supported on only one triangle $\tau \in \Omega_0$ the order of moment conditions (3.28) drops down from two to one so that the decay is also lowered by one order. Since this affects only $\mathcal{O}(N^{3/2})$ elements the overall accuracy should not deteriorate significantly. On account of the order of exactness of our quadrature rule (2.18), we expect the entries of the corresponding approximations to behave similarly. One should note that the entries $(A\varphi^j_{K'})(x_K)$ of $\mathbf{A}^j_\varphi$ instead can only be expected to decay like $2^j(1 + |K - L|)^{-2}$.

Let us turn next to the matrix $\mathbf{A}^j_\psi = (\mathbf{T}^j)^*\mathbf{A}^j\mathbf{T}^j$ (3.27). To clarify the basic idea it is helpful to consider again for a moment the exact collocation matrix $\mathbf{A}^j_\varphi$ instead of $\mathbf{A}^j$. The first step of the analogous formation (3.27) is then given by

$$(\mathbf{L}^{j-1})^*\mathbf{A}^j_\varphi\mathbf{L}^{j-1} = \begin{pmatrix} \mathbf{A}^{j-1}_{\varphi,\varphi} & \mathbf{A}^{j-1}_{\varphi,\psi} \\ \mathbf{A}^{j-1}_{\psi,\varphi} & \mathbf{A}^{j-1}_{\psi,\psi} \end{pmatrix} \tag{4.7}$$

where

$$\begin{aligned} \mathbf{A}^{j-1}_{\varphi,\varphi} &:= (\mathbf{M}^{j-1})^*\mathbf{A}^j_\varphi\mathbf{M}^{j-1}, \quad & \mathbf{A}^{j-1}_{\varphi,\psi} &:= (\mathbf{M}^{j-1})^*\mathbf{A}^j_\varphi\mathbf{C}^{j-1} \\ \mathbf{A}^{j-1}_{\psi,\varphi} &:= (\mathbf{C}^{j-1})^*\mathbf{A}^j_\varphi\mathbf{M}^{j-1}, \quad & \mathbf{A}^{j-1}_{\psi,\psi} &:= (\mathbf{C}^{j-1})^*\mathbf{A}^j_\varphi\mathbf{C}^{j-1}. \end{aligned} \tag{4.8}$$

More generally, defining

$$\begin{gathered} \mathbf{A}^l_{\varphi,\varphi} := (\mathbf{M}^l)^*\cdots(\mathbf{M}^{j-1})^*\mathbf{A}^j_\varphi\mathbf{M}^{j-1}\cdots\mathbf{M}^l, \\ \mathbf{A}^l_{\varphi,\psi} := (\mathbf{M}^l)^*\mathbf{A}^{l+1}_{\varphi,\varphi}\mathbf{C}^l, \quad \mathbf{A}^l_{\psi,\varphi} := (\mathbf{C}^l)^*\mathbf{A}^{l+1}_{\varphi,\varphi}\mathbf{M}^l, \\ \mathbf{A}^l_{\psi,\psi} := (\mathbf{C}^l)^*\mathbf{A}^{l+1}_{\varphi,\varphi}\mathbf{C}^l \end{gathered} \tag{4.9}$$

and compounding these matrices yields the so called *atomic decomposition* of $\mathbf{A}^j_\varphi$ [5, 13, 23]. The estimation of the entries of the atomic decomposition can be used for ultimately estimating the entries of $(\mathbf{T}^j)^*\mathbf{A}^j_\varphi\mathbf{T}^j$. A typical entry in the atomic decomposition is

$$a^l_{L,K'} = 2^{-l-1}\sum_{K\in\Box^j}\varphi^l_{K'}(x_K)(A\psi^l_L)(x_K). \tag{4.10}$$

Exploiting that $\varphi^l_{K'}, \psi^l_L$ have compact support and assuming first that $\mathrm{supp}\varphi^l_{K'} \cap \mathrm{supp}\psi^l_L = \emptyset, \mathrm{supp}\psi^l_L \subset \tau$ for some $\tau \in \Omega^0$, we obtain, on account of (3.28)

$$|a^l_{L,K'}| \le \sup_{x\in\mathrm{supp}\varphi^l_K}\left|\int_\Omega k(x,y)\psi^l_L(y)dy\right|. \tag{4.11}$$

The right hand side may now be estimated as above yielding

$$|a^l_{L,K'}| \le c\, 2^l (1 + |K' - L|)^{-4}. \tag{4.12}$$

Similar remarks as before apply when ψ^l_L is not supported on a single $\tau \in \Omega^0$.

The essential steps of our algorithm may now be described as follows:

- The discretized collocation method gives rise to a linear system (2.21)

$$\mathbf{A}^j \mathbf{u}^j = \mathbf{f}^j, \tag{4.13}$$

  where $\mathbf{A}^j$ is the approximation to $\mathbf{A}^j_\varphi$ described in Section 2.

- The transformation $\mathbf{T}^j$ from (3.23) gives the matrix

$$\mathbf{A}^j_\psi := (\mathbf{T}^j)^* \mathbf{A}^j \mathbf{T}^j, \tag{4.14}$$

  so that the system (2.21) is equivalent to

$$\mathbf{A}^j_\psi \mathbf{u}^j_\psi = (\mathbf{T}^j)^* \mathbf{f}^j, \tag{4.15}$$

  where $\mathbf{u}^j = \mathbf{T}^j \mathbf{u}^j_\psi$.

- In view of the order of our quadrature rule, the above estimates (4.5) predict a corresponding decay of the entries in $\mathbf{A}^j_\psi$. Replacing all entries in $\mathbf{A}^j_\psi$ by zero whose modulus stays below a given threshold th yields the *compressed* system

$$\tilde{\mathbf{A}}^j_\psi \tilde{\mathbf{u}}^j_\psi = (\mathbf{T}^j)^* \mathbf{f}^j. \tag{4.16}$$

- This latter system is solved with the aid of a sparse iterative solver.

The decay estimates (4.5) allow us to estimate the deviation of the solution of (4.16) from the solution to the full system (2.21) and, in some cases, also from the exact solution of $Au = f$. For periodic problems and generalized Petrov-Galerkin schemes on uniform grids this has been done in [13]. In fact, it is shown there that for every $\epsilon > 0$ there exists a sparse matrix $\tilde{\mathbf{A}}^j_\psi$ which differs in the spectral norm from $\mathbf{A}^j_\psi$ only by ϵ. A similar result can be shown also for the so called atomic decomposition [13]. Furthermore, compression rate and consistency error can be balanced in a way that the solutions to the compressed systems exhibit under certain circumstances still asymptotically optimal convergence rates relative to the exact solutions of $Au = f$. An analogous analysis for the present setting would go beyond the scope of this paper and is deferred to a forthcoming paper.

We refer to Figure 3 for a pseudocode of the whole transformation (4.14) and of the decomposition (3.16).

5 Results

For our numerical experiments we consider a Dirichlet problem for the Laplace's equation $\Delta U(x) = 0 \quad , \quad x \in \mathcal{P}$, and choose here smooth Dirichlet data, $U(x)|_\Omega = f(x)$, $x = ({}_1x, {}_2x, {}_3x) \in \Omega$. In spite of the smoothness of the Dirichlet data the solution will generally not be smooth because the boundary is not smooth.

We have tested the method described above for three different polyhedra, a cube, a tetrahedron and an L-block (bench). The last example deals with a non-convex domain (see Picture 1).

If U is the solution of the Dirichlet problem in $\mathcal{P}$ with the boundary value $f(x),\ x \in \Omega$ then U admits the representation

$$U(x) = \frac{1}{4\pi} \int\limits_\Omega \frac{n_y \cdot (x-y)}{|x-y|^3} u(y) d_y\Omega, \quad x \in \mathcal{P}, \tag{5.1}$$

where u is the solution of the double layer potential equation $Au = f$. In particular we choose the harmonic function $U(x) := \sqrt{\frac{11}{4}}(({}_1x+1)^2 + {}_2x^2 + {}_3x^2)^{-\frac{1}{2}}, x \in \mathcal{P}$, $f(x) := U(x)|_\Omega$.

5.1 The Uncompressed Collocation Scheme

In order to have a control for our compression error, we have to monitor the error between the exact solution u, and the approximate solution u^j obtained by the given (uncompressed) discretized collocation method on a grid $\square^j$ with $N = N_j$ knots. Since we are often interested in the potential U, or in some functionals of the boundary data u, rather than in the values of u, we directly go ahead and determine first an approximation of U as follows. Inserting u^j into the representation formula above and computing the integral via the same quadrature rule used for the computation of the entries of the stiffness matrix, we derive the following formula for the approximate solution of the boundary value problem (cf. Figure 2):

$$U^j(x) := \frac{1}{4\pi} \sum_{\tau \in \Omega^j} \frac{1}{3} \sum_{i=1}^{3} \frac{n_\tau \cdot (x - y^\tau_{K_i^!})}{|x - y^\tau_{K_i^!}|^3} u^j(y^\tau_{K_i^!})|\tau|, \tag{5.2}$$

where for $\tau = [y_{K_1}, y_{K_2}, y_{K_3}]$ the points $y^\tau_{K_i^!}$ are defined by $y^\tau_{K_i^!} = \frac{1}{2}(y_{K_i} + y_{K_k}) \in \Delta^j, i = 1, 2, 3$ with $k = i+1$, for $i = 1, 2$ and with $k = 1$ if $i = 3$, are the midpoints of the edges of the triangle τ and $u^j(y^\tau_{K_i^!}) := \frac{1}{2}(u^j(y^\tau_{K_i}) + u^j(y^\tau_{K_k}))$.

The error $ERR_{x^i} := |U(x^i) - U^j(x^i)|$ at the points $x^1 = (0.5, 0.5, 0.5)$, $x^2 = (0.6, 0.4, 0.3), x^3 = (0.3, 0.3, 0.25)$ is displayed in Table 1. As a further control we replace the exact solution u of (2.19) by the numerical solution u^J on a very fine grid and compute $ERR^j_e = \frac{\|u^j - u^J\|_0}{\|u^J\|_0}$ where here $\|u\|_0 := (\sum_{K \in \square^j} |u(x_K)|^2)^{1/2}$. This error is equivalent to the relative L^2-error and is also displayed in Table 1. The three values of ERR^j_e standing in brackets are extrapolated from the former ones using the convergence order α as given below.

Since the solution of the integral equation has a singular behavior in the vicinity of the corners and edges, the convergence order of the L^2–error ERR_e^j of the discrete equation (2.21) is expected to be worse than 7/12, i.e. $ERR_e^j N^{\frac{7}{12}} \to \infty$. In our computations, however, we have observed that

$$\alpha = -\frac{\log(ERR_e^{j+1}) - \log(ERR_e^j)}{\log(N_{j+1}) - \log(N_j)}, \tag{5.3}$$

turns out to be somewhat better than expected (cp. α in Table 1).

For the convergence of the approximate solution to the exact solution of the Dirichlet problem in the interior of $\mathcal{P}$, we expect that the rate of convergence coincides with the order of approximation by linear functions, i.e. $\max_{K\in\square^j}(ERR_{x_K}) = \mathcal{O}(N^{-1})$. This is confirmed by checking

$$\beta = -\frac{\log(\max_{i=1,2,3}(ERR_{x^i}^{j+1})) - \log(\max_{i=1,2,3}(ERR_{x^i}^j))}{\log(N_{j+1}) - \log(N_j)}, \tag{5.4}$$

$ERR_{x^i}^j = ERR_{x^i}$ for certain levels j (cp. β in Table 1). Note that because of $N^{-1} \sim h^2$ this is in agreement with the remarks at the end of Section 2.

5.2 The Compressed Scheme

In the multiscale representation $\mathbf{A}_\psi^j$ we discard those elements which are below a given threshold th and end up with a compressed or sparsified matrix $\tilde{\mathbf{A}}_\psi^j$. The ratio between the total number N^2 of matrix elements of the full matrix and the number nze of nonzero elements after thresholding defines the *compression rate*= $cpr := \frac{N^2}{nze}$. By compression we obtain a perturbed system. The solution of this perturbed system (4.16) is denoted by $\tilde{u}_\psi^j$. Substituting $\tilde{u}_\psi^j$ into the discrete representation formula (5.2), we compute

$$U_\psi^j(x) := \frac{1}{4\pi}\sum_{\tau\in\Omega^j}\frac{1}{3}\sum_{i=1}^{3}\frac{n_\tau\cdot(x-y_{K_i'}^\tau)}{|x-y_{K_i'}^\tau|^3}\tilde{u}_\psi^j(y_{K_i'}^\tau)|\tau|, \tag{5.5}$$

with τ and K_i' as above.

Of course, the compression causes an additional error. An acceptable compression should have only a negligable influence on the precision of the final approximate solution. To monitor $\tilde{u}_\psi^j$ we again replace the exact solution u of (2.19) by the numerical solution u^J on a very fine level J and compute $ERR_e^\psi = \frac{||\tilde{u}_\psi^j - u^J||_0}{||u^J||_0}$. In order to determine which threshold results in a suitable compression, we compare this error with the error ERR_e^j of the discretization scheme. Similarly we compare ERR_{x^i} with the corresponding error $|U(x^i) - U_\psi^j(x^i)|$. To estimate this error we compute the maximum $MERR_x := max_{i=1,2,3}|U(x^i) - U_\psi^j(x^i)|$. This is the second quantity that should help determining which compression rate has still a negligable influence on the precision of the approximate solution. Finally, we compute the additional error arising from the compression $ERR := \frac{||u^j - \tilde{u}_\psi^j||_0}{||u^j||_0}$, which is also displayed in most of the tables.

At this place we summarize all previously defined notation in Table 2.

The main results for the geometries we have tested are presented in Tables 3, 4 and 5. The bold quantities refer to the largest threshold th and the corresponding solution $\tilde{u}^j_\psi$ for which $ERR^\psi_e \approx ERR^j_e$ on the same level j (ERR^j_e is to be found in Table 1). In this case, thresholding apparently has a *negligable* influence on the solution $\tilde{u}^j_\psi$. Furthermore for the bold quantities one has $MERR_x \approx ERR_{x^i}, i = 1,2,3$ and for $j > 4$ even $MERR_x \leq ERR_{x^i}, i = 1,2,3$ (cp. Table 1). We observe that we can choose a larger threshold, and this results in a better compression for larger N, satisfying $MERR_x \approx \inf_i ERR_{x^i},\ i = 1,2,3$.

In Figures 4 and 5 the number nze of nonzero elements of the compressed matrices $\tilde{\mathbf{A}}^j_\psi$, for which we observed acceptable precision (bold quantities in Tables 3, 4 and 5) versus the number of knot points $N = \#\square^j$ are plotted and compared with the N^2 elements of the dense matrix. Figure 5 exhibits a nearly linear increase of nze for greater N. The compression reduces storage significantly to $1/cpr \cdot N^2$. Likewise the CPU time for the matrix-vector multiplications of the iterative solution of the linear system decreases substantially.

So far we have only made a first attempt using *level dependent thresholds* in order to fit the quadratic order of convergence. In this case of level dependent thresholding we have observed a small improvement (see Table 6). Actually we have not invested further efforts in this direction.

For the solution of the discrete and compressed scheme we use GMRES as an iterative procedure (cf. [37],[41]). This method has been proved to be an efficient tool for this kind of equation (cf. e.g. [40],[25]). For comparison and for the estimation of the condition numbers $\kappa(\mathbf{A}^j)$ and $\kappa(\tilde{\mathbf{A}}^j_\psi)$ we apply a direct method. This method is an expert driver of the well known LA-package [17]. The number of iteration steps, the estimated condition numbers and the CPU times t_{GE} for Gaussian elimination and t_{GM} for GMRES iteration on a DEC 3000 AXP 500 workstation are presented in Table 11. The termination bound for the iteration process is chosen to be about $ERR^j_e/100$. Because we have not constructed an orthogonal basis the condition of the original matrix should be better than that of the transformed matrix. Furthermore compression should have a minor influence on the condition numbers. This is confirmed by our experiments. We made some experiments choosing different constant multiples c_b, in the corresponding two scale relation (3.30). An appropriate choice of c_b does appear to improve the behavior of the condition numbers of the matrix $\tilde{\mathbf{A}}^j_\psi$ (cf. Tables 7 and 8). The condition numbers of the total transformation $\mathbf{T}^j$ are still quite large and seem to increase with N. Therefore the issue of stability requires further investigations. Nevertheless the present transformation $\mathbf{T}^j$ performs significantly better than the one obtained by substituting the stencil (3.3) for the transformation from the fine to the coarse grid by

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & \mathbf{2} & 0 \\ 0 & 0 & 0 \end{pmatrix}, \tag{5.6}$$

(cp. (3.11)). Again the bold marked value points to a knot $K \in \square^j$ on the coarser grid. For the tetrahedron this is confirmed by the condition numbers of the transformed matrix and the resulting number of iterations needed for the solution of the corresponding linear system by GMRES (see Table 9).

In addition we have simulated the atomic decomposition (see [13] for more details) by compressing the matrices of the type $\mathbf{A}^{j-1}_{\psi,\varphi}$, $\mathbf{A}^{j-1}_{\varphi,\psi}$, $\mathbf{A}^{j-1}_{\psi,\psi}$ given by (4.9). The compression and accuracy is observed to be somewhat worse than for the wavelet representation (cp. Table 10).

Our last experiment was made to answer a question we were asked by practitioners. We demonstrate the effect of compressing the original stiffness matrix in Table 12. In the case of the cube, about 1/6 of the entries in the original stiffness matrix are zero. Further small entries exist due to decay in the kernel of the operator. What will be the effect of cancelling small entries in the original matrix? Applying the same threshold as in Table 3 to the original stiffness matrix, the compression becomes very ineffective. In addition the resulting error is already unacceptable. If one wants to retain sufficient accuracy only a minor compression is admissible. Finally, compression rates which are comparable to those for the multiscale representation yields completely wrong results. This is in agreement with our theoretical predictions [13]. In fact, since a compression of $\mathbf{A}^j$ neglects the influence of the far field, one cannot expect this to work.

6 Concluding Remarks

Our aim was to develop first a possibly cheap and unsophisticated scheme. In spite of a relatively strong compression one observes acceptable accuracy. For $N \sim 3500$ such a compression rate is about 16 and we observed this rate to be more or less independent of the underlying geometry. Although we have considered only few levels, the results indicate a good asymptotic behavior. For the nonconvex body, the convergence is lower which accounts for the fact that the compression produces less precise approximations. In CPU–time solving the linear system by an iterative scheme we also observed a speed up factor of about 5-8. In view of an efficient matrix compression, iterative solvers are the only reasonable choice for large scale systems. Our method damps the coefficients away from the singularity. Thus the double layer potential operator for the Laplacian can be well compressed since it has only fixed singularities along the edges. (One should not forget the presence of the identity operator). We expect the same to persist for the double layer potential for the Stokes system. But, for the Navier–Lamé equations of elasticity and for 3-D Maxwell equation, the double layer potential operator is a singular integral operator even if the surface is smooth. In this case it is not clear whether the compression works comparatively well.

We expect that the present method works well for the normal derivative of the double layer potential. This is a hyper-singular integral operator of order $+1$ so that a multiscale representation should support preconditioning. The single layer potential operator is of order -1. In this case we expect that, for a suitable compression together with preconditioning, the present approach based on only two vanishing moments (cf. (3.28)) will not be sufficient. In the future it will be important to develop methods with higher order moment conditions. Employing nonuniform grids becomes probably unavoidable when dealing with large scale problems which would justify the significant overhead in bookkeeping the data and managing the more complicated data structure.

At this stage it is still hard to compare the present method with panel clustering. In a future project we want to compute the sparsified matrix directly which in turn may be additionally compressed by thresholding. For this purpose, the coefficients corresponding to the near field have to be computed with the accuracy for the finest grid. The far field influence which mainly affects the coarser grids, can be sufficiently handled by less accurate quadratures. We like to mention that this strategy is very similar in spirit to the clustering in the panel clustering method.

In essence both methods have common roots and are closely related. The multiscale compression avoids the explicit multipole expansion. But as a prize one has to deal with all coarser levels. However, the multiscale method seems to offer a more flexible and widely applicable concept, although with the presently available tools one still encounters serious difficulties when dealing with complicated geometries. One advantage of the present multiscale technique applied to boundary integral equations is that we have additionally a simple `a posteriori` criterion to decide which coefficients are essentially required. Applying the present thresholding leads often to much better compression than the `a priori` choice of coefficients. This seems to apply to our double layer potential operator. Further advantages of the multiscale algorithm are related to preconditioning nonzero order operators and regularizing ill conditioned problems.

Acknowledgement

We would like to thank A.Rathsfeld for helpful discussions and valuable suggestions concerning the contents of this paper.

References

[1] Alpert, B.: Sparse representation of smooth linear operators. Preprint, PhD Thesis, Yale University, 1990.

[2] Anderson, L., Jawerth, B., Mitrea, M.: The Cauchy singular integral operator and clifford wavelets. Preprint, MSRI reports, 1992.

[3] Atkinson, K.E.: A survey of boundary integral equation methods for the numerical solution of Laplace's equation in three dimensions, in: *Numerical solution of integral equations* (ed. M.Golberg) Plenum Press, New York, 1990.

[4] Atkinson, K.E., Chien, D.: Piecewise polynomial collocation for boundary integral equations, submitted to *SIAM J. Sci. Stat. Comp.*

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

[6] Brandt, A., Lubrecht, A.A.: Multilevel matrix multiplication and fast solution of integral equations, *J. Comp. Phys.* **90**, 348 – 370 (1991).

[7] Carnicer, J.M., Dahmen, W., Peña, J.M., Local decompositions of refinable spaces, preprint.

[8] Ciarlet, P.L.: *The Finite Element Method for Elliptic Problems.* North Holland, Amsterdam, New York, Oxford 1978.

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

[10] Dahlberg, B.E.J., Verchota,G.: Galerkin methods for the boundary integral equations of elliptic equations in non-smooth domains, in: Proc. Conf. Boca Raton Fla. 1988, Providence, 39–60 (1990).

[11] Dahmen, W., Kunoth, A.: Multilevel preconditioning, *Numerische Mathematik* **63**, 315–344 (1992).

[12] Dahmen, W., Prößdorf, S., Schneider, R.: Wavelet approximation methods for pseudodifferential equations I: Stability and convergence, IAAS-Preprint No. 7, Berlin 1992, to appear in *Math. Zeitschrift.*

[13] Dahmen, W., Prößdorf, S., Schneider, R.: Wavelet approximation methods for pseudodifferential equations II: Matrix compression and fast solution, Preprint No. 82, RWTH Aachen, 1993, to appear in *Advances in Computational Mathematics.*

[14] Dahmen, W., Prößdorf, S., Schneider, R.: Multiscale methods for pseudodifferential equations, TH FB Mathematik Preprint No. 1566, to appear in Schuhmaker & Webb (eds.) *Topics in the Theory and Applications of Wavelets.*

[15] Daubechies, I.: Orthonormal bases of compactly supported wavelets, *Comm. Pure and Appl. Math.* **41**, 909–996 (1988).

[16] Daubechies, I.: *Ten Lectures on Wavelets*, CBMS-NSF Regional Conference Series in Applied Mathematics **61**, 1992.

[17] Dongarra, J., Duff I., Sorensen, D., H.van der Vorst., *Solving Linear Systems on Vector and Shared Memory Computers.* The Society for Industrial and Applied Mathematics, University City Science Center, Philadelphia, 1991.

[18] Donoho, D.L.: Nonlinear solution of linear inverse problems by wavelet-vaguelette decomposition. Technical Report, Department of Statistics, Stanford University 1992.

[19] Elschner, J.: The double layer potential operator over polyhedral domains II: Spline Galerkin methods, *Math. Meth. Appl. Sci.* **45**, 23–37 (1992).

[20] Hackbusch, W.: The frequency decomposition multigrid method. I. Application to anisotropic equations. *Numer. Math.* **56**, 229 – 245 (1989).

[21] Hackbusch, W.: The frequency decomposition multigrid method. II. Convergence analysis based on the additive Schwarz method *Numer. Math.* **63**, 433 – 453 (1992).

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

[23] Harten, A., Yad-Shalom, I.: Fast multiresolution algorithms for matrix-vector multiplication, ICASE Report No. 92-55, October 1992.

[24] Hemker, P.W., Plantevin, F.: Wavelet bases adapted to inhomogeneous cases. to appear in Kornwinder (eds.) *Wavelets: An Elementary Treatment of Theory and Appl.*. 1993.

[25] Kleemann, B., Rathsfeld, A.: Nyström's method and iterative solvers for the solution of the double layer potential equation over polyhedral boundaries, IAAS-Berlin, Preprint No. 36 (1993).

[26] Kleinman, R.E., Wendland, W.L.: On Neumann's method for the Exterior Neumann problem for the Helmholtz equation, *Journal of Math. Anal. and Appl.* **1**, 170–202 (1977).

[27] Kral, J., Wendland, W.L.: Some examples concerning applicability of the Fredholm-Radon method in potential theory, *Aplikace matematiky* **31**, 293–308 (1986).

[28] Kral, J., Wendland, W.L.: On the applicability of the Fredholm–Radon method in potential theory and the panel method, in: Notes on Numerical Fluid Mechanics, vol. 21, Viehweg, 120–136 (1988).

[29] Mazya, V.G.: Boundary integral equations, in: V.G.Mazya and S.M.Nikol'skiǐ (eds.) *Encyclopaedia of Math. Sciences*, vol. 27, Analysis IV , Springer-Verlag, Berlin, Heidelberg 1991.

[30] Meyer, Y.: *Ondelettes et Opérateurs 1 : Ondelettes.* Hermann, Paris, 1990.

[31] Meyer, Y.: *Ondelettes et Opérateurs 2 : Opérateur de Caldéron-Zygmund*, Hermann, Paris 1990.

[32] Meyer, Y.: *Ondelettes et Opérateurs 3*, Hermann, Paris 1990.

[33] Plantevin, F.: Une application de la transformée continue en ondelettes à la méchanique quantique et analyse multi-résolution adaptive, dissertation, Centre Physique Théoretique, Marseille, 1992.

[34] Rathsfeld, A.: The invertibility of the double layer potential operator in the space of continuous functions defined on a polyhedron. The panel method, *Appl. Anal.* **45**, 135–177 (1992).

[35] Rathsfeld, A.: On quadrature methods for the double layer potential equation over the boundary of a polyhedron, to appear.

[36] Rokhlin, V.: Rapid solution of integral equations of classical potential theory. *J. Comp. Phys.* **60**, (1985).

[37] Saad, Y., Schultz, M.H.: GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.* **7**, No. 3, 856–869 (1986).

[38] Schneider, R.: Wavelets and frequency decomposition multilevel methods to appear in *Notes in Numer. Fluid Dynamics* (Vieweg Verlag).

[39] Stephan, E.P., Wendland, W.L.: A hypersingular boundary integral method for two-dimensional screen and crack problems, *Arch. Rational Mech. Anal.* **112**, 363 – 390 (1990).

[40] Vavasis, S.A.: Preconditioning for boundary integral equations, *SIAM J. Matrix Anal. Appl.* **13**, Vol. 3, 905 – 925 (1992).

[41] Walker, H.F.: Implementation of the GMRES method using Householder transformations, *SIAM J. Sci. Stat. Comput.* **9**, No. 1, 152–163 (1988).

[42] Wendland, W.L.: Behandlung von Randwertaufgaben im $\mathbf{R}^3$ mit Hilfe von Einfach- und Doppelschichtpotentialen, *Numer. Math.* **11**, 380–404 (1968).

[43] Wendland, W.L.: Boundary methods and their asymptotic convergence, in: P.Filipi (ed.), Lecture Notes of the CISM Summer- School on "Theoretical Acoustics and Numerical Techniques", International Centre for Mechanical Sciences, Udine, Notes Nb. 277, Springer-Verlag Wien, New York, 137–216 (1983).

[44] Yserentant, H.: On the multilevel splitting of finite element spaces. *Numer. Math.* **49**, 379 – 412 (1986).

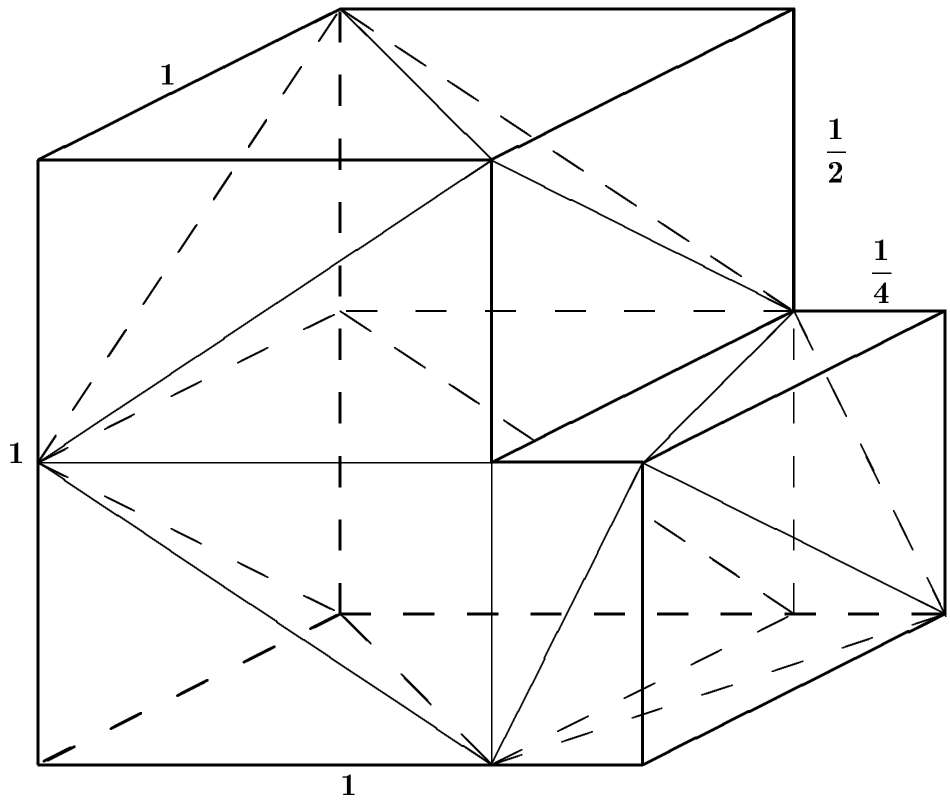

Figure 1: Bench (= L-block), in a cube of edge length 1 with initial triangulation.

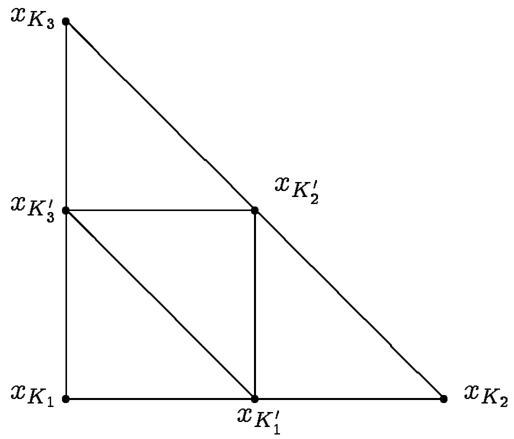

Figure 2: Triangle $[x_{K_1}, x_{K_2}, x_{K_3}]$ and its refinement. $\{K_1, K_2, K_3\} \subset \square^j \Rightarrow \{K_1, K_2, K_3, K'_1, K'_2, K'_3\} \subset \square^{j+1}$, $\{K'_1, K'_2, K'_3\} \subset \Delta^j$.

Let $N = \#\square^j$.

```
procedure vector_decomposition (in: f(1 ··· N), out: g(1 ··· N))
for l = j − 1 to 0 step -1 do
begin
    forall L ∈ Δ^l do                              % realizes g^l_L = (C^l)* f^{l+1}_K, K ∈ □^{l+1}
    begin
        g(L) = f(L)
        forall K' ∈ □^{l+1}, K' neighbor of L do   % loops twice because of only two neighbors
        begin
            g(L) = g(L) − (1/2) f(K')
        end
    end
    forall K ∈ □^l do                              % realizes f^l_K = (M^l)* f^{l+1}_K
    begin
        f(K) = (1/2) f(K)
        forall L ∈ Δ^l, L neigbor of K do          % usually loops six times
        begin
            f(K) = f(K) + (1/4) f(L)
        end
    end
end
```

Now, (4.14) reads as follows:

```
for K = 1 to N step 1 do
begin
    for K' = 1 to N step 1 do
    begin
        f(K') = a_{K,K'}
    end
    vector_decomposition(f, a_{K,1···N})
end
for K' = 1 to N step 1 do
begin
    for K = 1 to N step 1 do
    begin
        f(K) = a_{K,K'}
    end
    vector_decomposition(f, a_{1···N,K'})
end.
```

Figure 3: Pseudocode of pyramid scheme.

Discretization Method							
j	N_j	ERR_e^j	α	ERR_{x^1}	ERR_{x^2}	ERR_{x^3}	β
cube:							
1	26	$6.3 \cdot 10^{-2}$	-	$1.2 \cdot 10^{-2}$	$7.2 \cdot 10^{-3}$	$2.9 \cdot 10^{-2}$	-
2	98	$2.6 \cdot 10^{-2}$	0.67	$7.5 \cdot 10^{-4}$	$9.3 \cdot 10^{-5}$	$4.4 \cdot 10^{-3}$	1.40
3	386	$1.0 \cdot 10^{-2}$	0.70	$1.6 \cdot 10^{-4}$	$3.1 \cdot 10^{-5}$	$6.7 \cdot 10^{-4}$	1.37
4	1538	$3.5 \cdot 10^{-3}$	0.76	$4.0 \cdot 10^{-5}$	$7.4 \cdot 10^{-6}$	$1.6 \cdot 10^{-4}$	1.04
5	6146	$(1.2 \cdot 10^{-3})$	-	$1.0 \cdot 10^{-5}$	$1.9 \cdot 10^{-6}$	$4.0 \cdot 10^{-5}$	1.00
tetrahedron:							
1	14	$4.4 \cdot 10^{-2}$	-	$3.4 \cdot 10^{-3}$	$1.0 \cdot 10^{-3}$	$2.2 \cdot 10^{-2}$	-
2	50	$2.1 \cdot 10^{-2}$	0.58	$4.5 \cdot 10^{-3}$	$5.2 \cdot 10^{-3}$	$2.8 \cdot 10^{-3}$	1.13
3	194	$8.7 \cdot 10^{-3}$	0.97	$1.7 \cdot 10^{-4}$	$2.1 \cdot 10^{-4}$	$1.2 \cdot 10^{-4}$	2.37
4	770	$3.1 \cdot 10^{-3}$	0.75	$3.9 \cdot 10^{-5}$	$5.1 \cdot 10^{-5}$	$1.2 \cdot 10^{-4}$	0.41
5	3074	$(1.0 \cdot 10^{-3})$	-	$8.5 \cdot 10^{-6}$	$1.2 \cdot 10^{-5}$	$2.9 \cdot 10^{-5}$	1.03
bench:							
1	58	$4.4 \cdot 10^{-2}$	-	$8.3 \cdot 10^{-3}$	$1.2 \cdot 10^{-3}$	$9.4 \cdot 10^{-3}$	-
2	226	$1.7 \cdot 10^{-2}$	0.70	$4.4 \cdot 10^{-4}$	$3.1 \cdot 10^{-4}$	$2.2 \cdot 10^{-3}$	1.07
3	898	$5.9 \cdot 10^{-3}$	0.77	$4.2 \cdot 10^{-5}$	$1.6 \cdot 10^{-5}$	$4.2 \cdot 10^{-4}$	1.20
4	3586	$(2.0 \cdot 10^{-3})$	-	$8.5 \cdot 10^{-6}$	$5.4 \cdot 10^{-6}$	$1.0 \cdot 10^{-4}$	1.04

Table 1: Error ERR_e^j for the solution of the discrete method, errors $ERR_{x^i}, i = 1,2,3$ at the points x^1, x^2, x^3 and convergence rates 2α and 2β.

$\mathbf{A}^j$	$=$	matrix of the discretized equation (2.21) on a certain level j; corresponding solution is u^j
$\mathbf{A}_\psi^j$	$=$	matrix of the transformed equation (4.15) with j levels in the pyramid scheme (3.16)
$\tilde{\mathbf{A}}_\psi^j$	$=$	matrix of the transformed equation (4.16) after thresholding with threshold th on a certain level j; the corresponding solution is $\tilde{u}_\psi^j$
nze	$=$	number of nonzero elements of matrix $\tilde{\mathbf{A}}_\psi^j$
cpr	$=$	N^2/nze, compression rate
$\kappa(\mathbf{A})$	$=$	estimated condition number of matrix $\mathbf{A}$ determined by an LA-pack routine during Gaussian elimination
$it(\mathbf{A})$	$=$	number of iterations in GMRES applied to the linear system with $\mathbf{A}$ the system matrix which is needed to get a prescribed accuracy
$t_{GM}(\mathbf{A})$	$=$	CPU-time necessary for $it(\mathbf{A})$ iterations
$t_{GE}(\mathbf{A})$	$=$	CPU-time to solve the equation (2.21) with Gaussian elimination
u^J	$=$	solution of the discrete equation (2.21) for the highest level J; $J = 5$ for the cube, $J = 5$ for the polyhedron, $J = 4$ for the L-block
$\|\|u\|\|_0$	$=$	$(\sum_{K \in \Box^j} \|u(x_K)\|^2)^{\frac{1}{2}}$
ERR_e^j	$=$	$\|\|u^j - u^J\|\|_0 \; / \; \|\|u^J\|\|_0$
ERR_e^ψ	$=$	$\|\|\tilde{u}_\psi^j - u^J\|\|_0 \; / \; \|\|u^J\|\|_0$
ERR	$=$	$\|\|u^j - \tilde{u}_\psi^j\|\|_0 \; / \; \|\|u^j\|\|_0$
$U(x^i)$	$=$	exact (known) value of the potential U at x^i
$U^j(x^i)$	$=$	numerical calculation of the potential at x^i from the solution u^j
$U_\psi^j(x^i)$	$=$	numerical calculation of the potential at x^i from solution $\tilde{u}_\psi^j$
ERR_{x^i}	$=$	$\|U(x^i) - U^j(x^i)\|$
$MERR_x$	$=$	$\max_{i=1,2,3} \|U(x^i) - U_\psi^j(x^i)\|$

Table 2: Notation for all tables.

Cube								
j	N_j	N_j^2	th	nze	cpr	ERR_e^ψ	ERR	$MERR_x$
1	26	676	$3 \cdot 10^{-3}$	568	1.2	$6.3 \cdot 10^{-2}$	$2.0 \cdot 10^{-4}$	$2.9 \cdot 10^{-2}$
			$1 \cdot 10^{-2}$	**466**	**1.4**	$6.3 \cdot 10^{-2}$	$4.3 \cdot 10^{-3}$	$2.9 \cdot 10^{-2}$
			$3 \cdot 10^{-2}$	274	2.5	$7.7 \cdot 10^{-2}$	$8.5 \cdot 10^{-2}$	$3.0 \cdot 10^{-2}$
2	98	9.604	$1 \cdot 10^{-3}$	5176	1.8	$2.6 \cdot 10^{-2}$	$2.3 \cdot 10^{-3}$	$4.6 \cdot 10^{-3}$
			$3 \cdot 10^{-3}$	**3346**	**2.9**	$2.6 \cdot 10^{-2}$	$5.5 \cdot 10^{-3}$	$4.4 \cdot 10^{-3}$
			$1 \cdot 10^{-2}$	1990	4.8	$3.2 \cdot 10^{-2}$	$2.5 \cdot 10^{-3}$	$2.9 \cdot 10^{-3}$
3	386	148.996	$3 \cdot 10^{-4}$	36.484	4.1	$1.0 \cdot 10^{-2}$	$1.0 \cdot 10^{-3}$	$6.7 \cdot 10^{-4}$
			$1 \cdot 10^{-3}$	**21.124**	**7.0**	$1.0 \cdot 10^{-2}$	$4.4 \cdot 10^{-3}$	$5.2 \cdot 10^{-4}$
			$3 \cdot 10^{-3}$	12.646	11.8	$1.9 \cdot 10^{-2}$	$2.3 \cdot 10^{-2}$	$1.9 \cdot 10^{-3}$
4	1538	$2.4 \cdot 10^6$	$1 \cdot 10^{-4}$	401.272	5.9	$3.5 \cdot 10^{-3}$	$4.7 \cdot 10^{-4}$	$1.3 \cdot 10^{-4}$
			$3 \cdot 10^{-4}$	**235.588**	**10.3**	$3.6 \cdot 10^{-3}$	$1.3 \cdot 10^{-3}$	$1.2 \cdot 10^{-4}$
			$1 \cdot 10^{-3}$	125.524	18.8	$4.7 \cdot 10^{-3}$	$3.8 \cdot 10^{-3}$	$1.8 \cdot 10^{-4}$
5	6146	$38 \cdot 10^6$	$3 \cdot 10^{-5}$	2.098.108	18.0	-	$2.9 \cdot 10^{-4}$	$9.2 \cdot 10^{-6}$
			$1 \cdot 10^{-4}$	**1.118.296**	**33.8**	-	$8.4 \cdot 10^{-4}$	$4.0 \cdot 10^{-5}$
			$3 \cdot 10^{-4}$	647.032	58.4	-	$2.1 \cdot 10^{-3}$	$2.4 \cdot 10^{-5}$

Table 3: Number nze of nonzero elements, compression rate cpr, errors ERR_e^ψ, ERR and $MERR_x$ for the multiscale algorithm on the cube for several thresholds th and levels j.

Tetrahedron								
j	N_j	N_j^2	th	nze	cpr	ERR_e^ψ	ERR	$MERR_x$
1	14	196	$3 \cdot 10^{-3}$	188	1.0	$4.4 \cdot 10^{-2}$	$1.1 \cdot 10^{-4}$	$2.2 \cdot 10^{-2}$
			$1 \cdot 10^{-2}$	**172**	**1.1**	$4.4 \cdot 10^{-2}$	$1.8 \cdot 10^{-4}$	$2.2 \cdot 10^{-2}$
			$3 \cdot 10^{-2}$	122	1.6	$5.5 \cdot 10^{-2}$	$5.6 \cdot 10^{-2}$	$3.7 \cdot 10^{-2}$
2	50	2.500	$1 \cdot 10^{-3}$	1948	1.3	$2.1 \cdot 10^{-2}$	$7.5 \cdot 10^{-4}$	$5.2 \cdot 10^{-3}$
			$3 \cdot 10^{-3}$	**1494**	**1.7**	$2.1 \cdot 10^{-2}$	$1.5 \cdot 10^{-3}$	$5.3 \cdot 10^{-3}$
			$1 \cdot 10^{-2}$	1016	2.5	$2.3 \cdot 10^{-2}$	$1.5 \cdot 10^{-2}$	$6.3 \cdot 10^{-3}$
3	194	37.636	$3 \cdot 10^{-4}$	22.546	1.7	$8.7 \cdot 10^{-3}$	$2.2 \cdot 10^{-4}$	$2.2 \cdot 10^{-4}$
			$1 \cdot 10^{-3}$	**15.108**	**2.5**	$8.7 \cdot 10^{-3}$	$1.6 \cdot 10^{-3}$	$1.8 \cdot 10^{-4}$
			$3 \cdot 10^{-3}$	9.736	3.9	$9.4 \cdot 10^{-3}$	$5.8 \cdot 10^{-3}$	$1.5 \cdot 10^{-4}$
4	770	592.900	$1 \cdot 10^{-4}$	159.022	3.7	$3.1 \cdot 10^{-3}$	$3.1 \cdot 10^{-4}$	$1.2 \cdot 10^{-4}$
			$3 \cdot 10^{-4}$	**96.244**	**6.2**	$3.2 \cdot 10^{-3}$	$1.1 \cdot 10^{-3}$	$1.2 \cdot 10^{-4}$
			$1 \cdot 10^{-3}$	56.668	10.5	$3.9 \cdot 10^{-3}$	$3.0 \cdot 10^{-3}$	$2.3 \cdot 10^{-4}$
5	3074	$9.4 \cdot 10^{6}$	$3 \cdot 10^{-5}$	1.179.686	8.0	-	$1.8 \cdot 10^{-4}$	$2.9 \cdot 10^{-5}$
			$1 \cdot 10^{-4}$	**660.124**	**14.3**	-	$5.1 \cdot 10^{-4}$	$2.8 \cdot 10^{-5}$
			$3 \cdot 10^{-4}$	389.018	24.3	-	$1.4 \cdot 10^{-3}$	$2.5 \cdot 10^{-5}$

Table 4: Number nze of nonzero elements, compression rate cpr, errors ERR_e^ψ, ERR and $MERR_x$ for the multiscale algorithm on the tetrahedron for several thresholds th and levels j.

Bench								
j	N_j	N_j^2	th	nze	cpr	ERR_e^{ψ}	ERR	$MERR_x$
1	58	3.364	$1 \cdot 10^{-3}$	2390	1.4	$4.4 \cdot 10^{-2}$	$3.2 \cdot 10^{-3}$	$9.4 \cdot 10^{-3}$
			$3 \cdot 10^{-3}$	**1840**	**1.8**	$4.4 \cdot 10^{-2}$	$1.2 \cdot 10^{-2}$	$1.0 \cdot 10^{-2}$
			$1 \cdot 10^{-2}$	1123	3.0	$5.9 \cdot 10^{-2}$	$5.8 \cdot 10^{-2}$	$1.6 \cdot 10^{-2}$
2	226	51.076	$3 \cdot 10^{-4}$	25690	2.0	$1.7 \cdot 10^{-2}$	$1.3 \cdot 10^{-3}$	$2.2 \cdot 10^{-3}$
			$1 \cdot 10^{-3}$	**15885**	**3.2**	$1.8 \cdot 10^{-2}$	$6.2 \cdot 10^{-3}$	$2.1 \cdot 10^{-3}$
			$3 \cdot 10^{-3}$	9437	5.4	$2.3 \cdot 10^{-2}$	$2.4 \cdot 10^{-2}$	$2.3 \cdot 10^{-3}$
3	898	806.404	$1 \cdot 10^{-4}$	174.050	4.6	$6.0 \cdot 10^{-3}$	$8.5 \cdot 10^{-4}$	$4.0 \cdot 10^{-4}$
			$3 \cdot 10^{-4}$	**101.954**	**7.9**	$6.3 \cdot 10^{-3}$	$3.1 \cdot 10^{-3}$	$3.9 \cdot 10^{-4}$
			$1 \cdot 10^{-3}$	55.612	14.5	$1.5 \cdot 10^{-2}$	$1.9 \cdot 10^{-2}$	$8.6 \cdot 10^{-4}$
4	3586	$13 \cdot 10^{6}$	$3 \cdot 10^{-5}$	1.481.035	8.7	-	$3.9 \cdot 10^{-4}$	$9.9 \cdot 10^{-5}$
			$1 \cdot 10^{-4}$	**796.353**	**16.1**	-	$1.2 \cdot 10^{-3}$	$9.5 \cdot 10^{-5}$
			$3 \cdot 10^{-4}$	449.560	28.6	-	$3.8 \cdot 10^{-3}$	$1.0 \cdot 10^{-4}$

Table 5: Number nze of nonzero elements, compression rate cpr, errors ERR_e^{ψ}, ERR and $MERR_x$ for the multiscale algorithm on the bench for several thresholds th and levels j.

Cube; level dependent thresholding										
j	N_j	th_0	th_1	th_2	th_3	th_4	nze	cpr	ERR_e^{ψ}	$MERR_x$
3	386	$1 \cdot 10^{-4}$	$2th_0$	$2th_1$	$4th_2$	-	17.011	**8.8**	$1.1 \cdot 10^{-2}$	$8.7 \cdot 10^{-4}$
4	1538	$1 \cdot 10^{-5}$	$2th_0$	$4th_1$	$2th_2$	$4th_3$	163.379	**14.5**	$4.0 \cdot 10^{-3}$	$1.5 \cdot 10^{-4}$

Table 6: Number nze of nonzero elements, compression rate cpr, errors ERR_e^{ψ} and $MERR_x$ for the multiscale algorithm on the cube for level dependent thresholding.

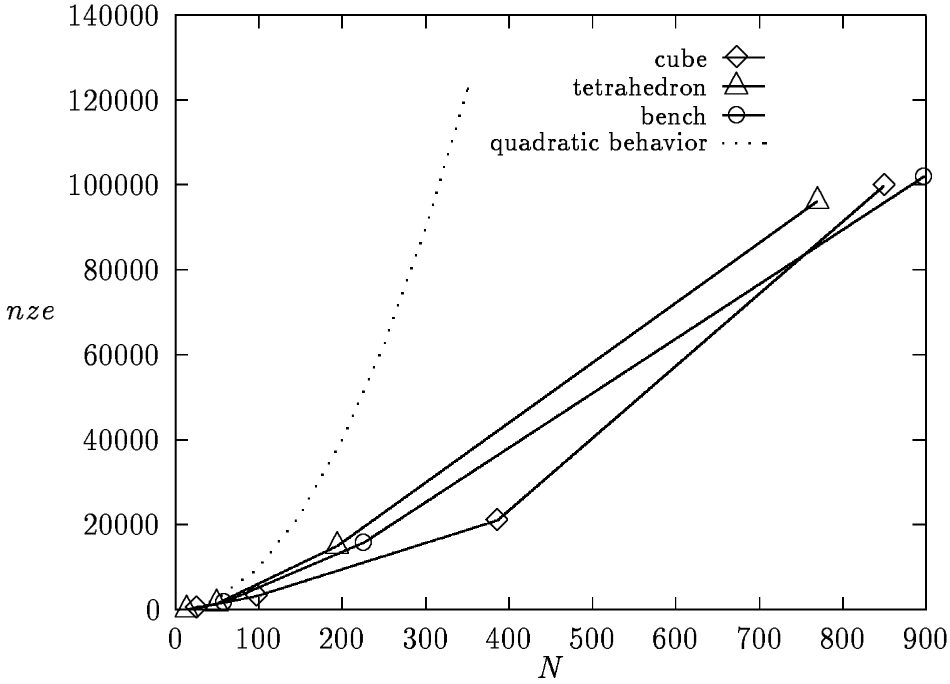

Figure 4: Number nze of nonzero elements of the transformed matrix $\tilde{\mathbf{A}}_{\psi}^{j}$ after thresholding in dependence of N_j. Dotted line for comparison: quadratic behavior of the number of elements of the original stiffness matrix.

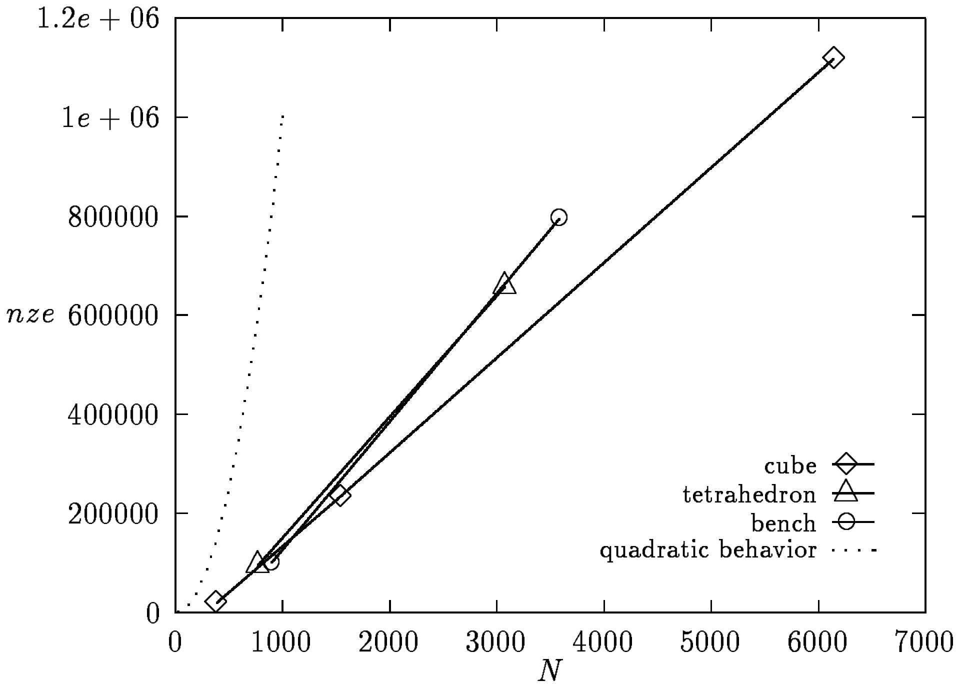

Figure 5: Number nze of nonzero elements of the transformed matrix $\tilde{\mathbf{A}}_{\psi}^{j}$ after thresholding in dependence of N_j. Dotted line for comparison: quadratic behavior of the number of elements of the original stiffness matrix.

Condition numbers					
domain	j	N_j	$\kappa(\mathbf{A}^j)$	c_b	$\kappa(\tilde{\mathbf{A}}^j_\psi)$
cube	2	98	2.1	1/2	42.5
				1.	**15.2**
				2.	66.1
	3	386	2.4	1/2	96.5
				1.	**38.8**
				2.	87.7
	4	1538	2.8	1/2	147.8
				1.	**81.6**
				2.	133.0

Table 7: Estimated condition numbers of the original stiffness matrix $\mathbf{A}^j$ and of matrix $\tilde{\mathbf{A}}^j_\psi$ in wavelet representation after thresholding for several constants c_b.

Condition numbers					
domain	j	N_j	$\kappa(\mathbf{A}^j)$	c_b	$\kappa(\tilde{\mathbf{A}}^j_\psi)$
tetrahedron	2	50	2.0	1/2	43.4
				1.	**17.5**
				2.	40.7
	3	194	2.3	1.	42.6
				3/2	**31.1**
				2.	56.3
	4	770	5.3	1.	88.8
				3/2	**63.4**
				2.	65.7
	5	3074	8.6	1.	149.7
				3/2	**114.7**
				2.	118.2
bench	1	58	4.1	1/4	60.0
				3/4	**14.0**
				1.	23.5
	2	226	6.6	1/4	200.9
				1/2	**42.8**
				1.	50.0
	3	898	11.0	1/2	106.3
				1.	**95.0**
				3/2	150.3
	4	3586	18.6	1/2	259.0
				1.	**177.2**
				3/2	208.6

Table 8: Estimated condition numbers of the original stiffness matrix $\mathbf{A}^j$ and of matrix $\tilde{\mathbf{A}}^j_\psi$ in wavelet representation after thresholding for several constants c_b.

Condition numbers					
j	N_j	$\kappa(\mathbf{A}^j)$	c_b	$\kappa(\tilde{\mathbf{A}}^j_\psi)$	$it(\tilde{\mathbf{A}}^j_\psi)$
1	14	1.4	**1.**	**12.7**	5
			3/2	12.9	-
			2.	18.4	-
2	50	2.0	1.	163.	-
			3/2	**121.**	21
			2.	163.	-
3	194	2.3	1.	1620.	-
			3/2	**1106.**	39
			2.	1459.	-
4	770	5.3	1.	14814.	-
			3/2	**9697.**	97
			2.	12596.	-
5	3074	8.6	1.	126229.	-
			3/2	**81720.**	228
			2.	105497.	-

Table 9: Estimated condition numbers of the original stiffness matrix $\mathbf{A}^j$ and of matrix $\tilde{\mathbf{A}}^j_\psi$ in wavelet representation after thresholding for the tetrahedron for several constants c_b. Transformation (5.6) rather than (3.3) was used. Number of iterations for the solution of the linear system $\tilde{\mathbf{A}}^j_\psi \tilde{\mathbf{u}}^j_\psi = \mathbf{f}^j_\psi$ with GMRES.

Atomic Decomposition							
j	N_j	th	nze	cpr	ERR_e^ψ	ERR	$MERR_x$
cube:							
1	26	$1 \cdot 10^{-2}$	466	1.4	$6.3 \cdot 10^{-2}$	$4.3 \cdot 10^{-3}$	$3.0 \cdot 10^{-2}$
2	98	$3 \cdot 10^{-3}$	3.394	2.8	$2.7 \cdot 10^{-2}$	$1.0 \cdot 10^{-2}$	$4.3 \cdot 10^{-3}$
3	386	$1 \cdot 10^{-3}$	21.892	6.8	$1.8 \cdot 10^{-2}$	$2.3 \cdot 10^{-2}$	$1.2 \cdot 10^{-3}$
4	1538	$3 \cdot 10^{-4}$	243.880	9.7	$1.0 \cdot 10^{-2}$	$1.3 \cdot 10^{-2}$	$2.5 \cdot 10^{-4}$
5	6146	$1 \cdot 10^{-4}$	1.375.108	27.5	-	-	$9.0 \cdot 10^{-5}$
tetrahedron:							
1	14	$1 \cdot 10^{-2}$	172	1.1	$4.5 \cdot 10^{-2}$	$2.1 \cdot 10^{-4}$	$2.2 \cdot 10^{-2}$
2	50	$3 \cdot 10^{-3}$	1.464	1.7	$2.1 \cdot 10^{-2}$	$3.9 \cdot 10^{-3}$	$4.8 \cdot 10^{-3}$
3	194	$1 \cdot 10^{-3}$	14.500	2.6	$9.0 \cdot 10^{-3}$	$4.0 \cdot 10^{-3}$	$1.4 \cdot 10^{-4}$
4	770	$3 \cdot 10^{-4}$	103.334	5.7	$5.3 \cdot 10^{-3}$	$5.9 \cdot 10^{-3}$	$1.3 \cdot 10^{-4}$
5	3074	$1 \cdot 10^{-4}$	794.886	11.9	-	$4.3 \cdot 10^{-3}$	$2.2 \cdot 10^{-5}$
bench:							
1	58	$3 \cdot 10^{-3}$	1.843	1.8	$4.5 \cdot 10^{-2}$	$1.1 \cdot 10^{-2}$	$1.0 \cdot 10^{-2}$
2	226	$1 \cdot 10^{-3}$	16.951	3.0	$1.9 \cdot 10^{-2}$	$1.2 \cdot 10^{-2}$	$2.4 \cdot 10^{-3}$
3	898	$3 \cdot 10^{-4}$	124.268	6.5	$1.2 \cdot 10^{-2}$	$1.5 \cdot 10^{-2}$	$6.3 \cdot 10^{-4}$
4	3586	$1 \cdot 10^{-4}$	1.065.490	12.1	-	$9.8 \cdot 10^{-3}$	$1.4 \cdot 10^{-4}$

Table 10: Number nze of nonzero elements, compression rate cpr and errors ERR_e^ψ, ERR and $MERR_x$ for the atomic decomposition. Thresholds th are chosen to be optimal for the wavelet transform (cp. Tables 3,4,5).

CPU-times									
j	N_j	$t_{GE}(\mathbf{A}^j)$	$\kappa(\mathbf{A}^j)$	$it(\mathbf{A}^j)$	$t_{GM}(\mathbf{A}^j)$	cpr	$\kappa(\tilde{\mathbf{A}}^j_\psi)$	$it(\tilde{\mathbf{A}}^j_\psi)$	$t_{GM}(\tilde{\mathbf{A}}^j_\psi)$
cube:									
3	386	1.1	2.4	6	0.2	7.0	38.9	14	**0.1**
4	1538	37.2	2.8	8	3.9	10.3	133.0	16	**0.9**
5	6146	1585.0	3.2	11	52.0	33.8	215.7	23	**6.0**
tetrahedron:									
3	194	0.3	2.3	7	0.06	3.9	31.1	13	**0.03**
4	770	6.8	5.3	8	0.9	6.2	63.4	15	**0.3**
5	3074	227.6	8.6	10	17.3	14.3	118.2	19	**2.6**
bench:									
2	226	0.4	6.6	7	0.05	3.2	50.0	17	**0.06**
3	898	9.1	11.0	9	1.2	7.9	95.0	21	**0.5**
4	3586	342.6	18.6	10	23.1	16.1	208.6	23	**3.9**

Table 11: Number of iterations it for the solution of the linear systems $\mathbf{A}^j\mathbf{u}^j = \mathbf{f}^j$ and $\tilde{\mathbf{A}}^j_\psi\tilde{\mathbf{u}}^j_\psi = (\mathbf{T}^j)^*\mathbf{f}^j$, respectively, with GMRES; CPU-times t_{GM} in seconds. Condition numbers κ of the corresponding matrices. For comparison: CPU-times t_{GE} for Gaussian elimination.

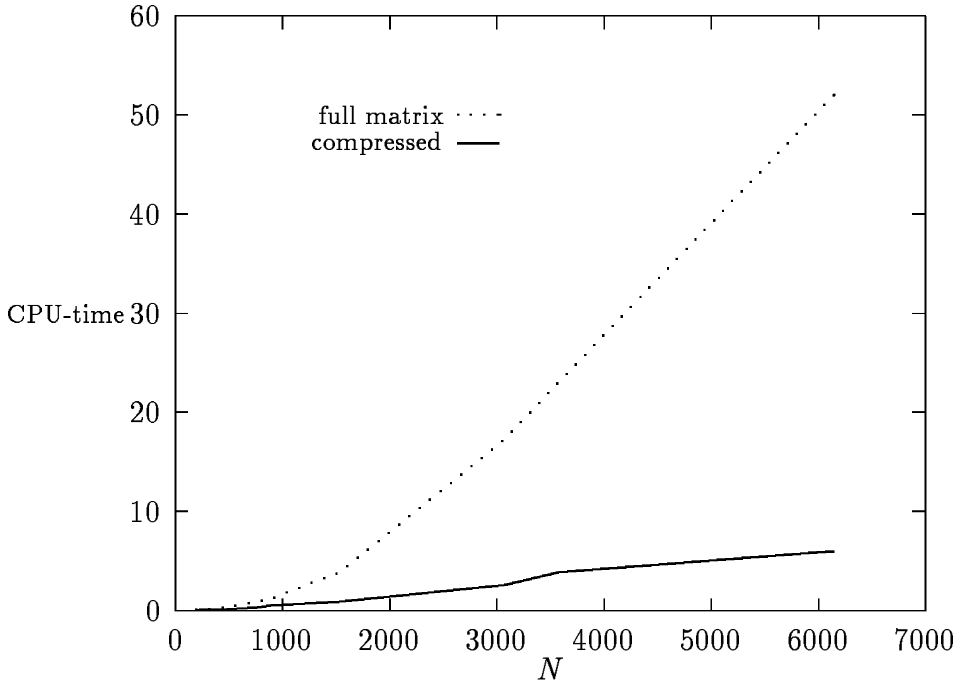

Figure 6: CPU-time in seconds for iterative solution of linear system with uncompressed (dotted line) and compressed matrix, respectively.

Cube; original stiffness matrix								
j	N_j	N_j^2	th	nze	cpr	ERR_e^{ψ}	ERR	$MERR_x$
3	386	148.996	$3 \cdot 10^{-4}$	123.636	1.2	$1.0 \cdot 10^{-2}$	$1.4 \cdot 10^{-3}$	$5.7 \cdot 10^{-4}$
			$1 \cdot 10^{-3}$	94.832	1.6	$4.7 \cdot 10^{-2}$	$5.2 \cdot 10^{-2}$	$1.6 \cdot 10^{-2}$
			$3 \cdot 10^{-3}$	24.374	6.1	$2.4 \cdot 10^{-1}$	$2.4 \cdot 10^{-1}$	$2.5 \cdot 10^{-1}$
4	1538	$2.4 \cdot 10^{6}$	$3 \cdot 10^{-5}$	1.969.924	1.2	$3.5 \cdot 10^{-3}$	$1.9 \cdot 10^{-4}$	$1.6 \cdot 10^{-4}$
			$3 \cdot 10^{-4}$	1.334.912	1.8	$5.2 \cdot 10^{-2}$	$5.3 \cdot 10^{-2}$	$3.4 \cdot 10^{-2}$
			$1 \cdot 10^{-3}$	259.658	9.1	$2.9 \cdot 10^{-1}$	$2.9 \cdot 10^{-1}$	$3.3 \cdot 10^{-1}$

Table 12: Number nze of nonzero elements, compression rate cpr, errors ERR_e^{ψ}, ERR and $MERR_x$ for thresholding the original stiffness matrix $\mathbf{A}^j$ of the cube for some levels.