Multiscale methods for the solution of the Helmholtz and Laplace equations

Wolfgang Dahmen* Bernd Kleemann† Siegfried Prößdorf‡
Reinhold Schneider§

February 6, 1996

1991 Mathematics Subject Classification. 65N12, 65N22, 65N35, 65N38, 65R20.

Keywords. Multiscale methods, compression, fast solution, Helmholtz equation, Laplace equation, collocation, preconditioning, scattering.

*RWTH Aachen, Institut für Geometrie und Praktische Mathematik, Templergraben 55, D–52062 Aachen, Germany

†Berliner Institut für Optik GmbH (BIFO), Rudower Chaussee 5, D–12484 Berlin, Germany

‡Weierstraß–Institute für Angewandte Analysis und Stochastik, Mohrenstraße 39, D–10117 Berlin, Germany

§TH Darmstadt, Fachbereich Mathematik, Schloßgartenstraße 7, D–64289 Darmstadt, Germany

1 Introduction

This paper presents some numerical results about applications of multiscale techniques to boundary integral equations. The numerical schemes developed here are to some extent based on the results of the papers [6]–[10]. Section 2 deals with a short description of the theory of generalized Petrov–Galerkin methods for elliptic periodic pseudodifferential equations in $\mathbb{R}^n$ covering classical Galerkin schemes, collocation, and other methods. A general setting of multiresolution analysis generated by periodized scaling functions as well as a general stability and convergence theory for such a framework is outlined. The key to the stability analysis is a local principle due to one of the authors. Its applicability relies here on a sufficiently general version of a so-called discrete commutator property of wavelet bases (see [6]). These results establish important prerequisites for developing and analysing methods for the fast solution of the resulting linear systems (Section 2.4). The crucial fact which is exploited by these methods is that the stiffness matrices relative to an appropriate wavelet basis can be approximated well by a sparse matrix while the solution to the perturbed problem still exhibits the same asymptotic accuracy as the solution to the full discrete problem. It can be shown (see [7]) that the amount of the overall computational work which is needed to realize a required accuracy is of the order $O(N(\log N)^b)$, where N is the number of unknowns and $b \geq 0$ is some real number.

We focus here on two problems which are solved by *fully discrete collocation* wavelet methods. Section 3 is devoted to various numerical experiments for the exterior Dirichlet problem for the Helmholtz equation. In this case the theoretical results for periodic problems apply and are confirmed by the numerical tests. In Section 4 we present some new results concerning a Dirichlet problem for the Laplace equation over three dimensional polyhedral domains. Linear systems with $\approx$ 100,000 unknowns are solved which corresponds to fully populated matrices of the same order. In this case our goal is not yet to present a fully developed scheme for such a complex three dimensional problem. In fact, one should note first that the general theory described in Section 2 is not directly applicable to this problem because it is not periodic and the boundary is not smooth. Therefore our main goal here is to explore to which extent the theoretical predictions from the model problem can still be confirmed under these less ideal circumstances. In particular, we focus on compression properties and the convergence behavior of the corresponding solutions of the perturbed problem. Therefore we content ourselves here still with a rather expensive way of computing the compressed stiffness matrices. Speeding up this portion of the solution process is less dependent on the topology of the domain and is meanwhile understood conceptually. Essentially following the analysis in [10], the practical realizations and numerical tests presented here do confirm a similar behavior as predicted by the theory for the periodic model problem.

2 Multiscale methods

2.1 Periodic pseudodifferential equations

At this stage we focus on the model case of periodic pseudodifferential equations to exploit the full advantages of Fourier transform techniques in connection with appropriate representations for the class of operators under consideration. However, we do consider

variable symbols and it should be mentioned that this class covers all the classical examples such as Hörmander's class, in particular, those operators arising in connection with boundary element methods. Moreover, most of the methods used here are of local nature and thus apply in essence to the case of non–periodic equations, as well (see [8]).

Consider the discrete Fourier transform by $\tilde{u}(k) := \int_{[0,1]} u(x)\mathrm{e}^{-2\pi\mathrm{i}x\cdot k}\mathrm{d}x$, $k \in \mathbb{Z}^n$, where $x \cdot k = \sum_{j=1}^n x_j k_j$. Let $\mathcal{T}^n$ denote the n–dimensional torus. Then a periodic pseudodifferential operator is defined by

$$\sigma(x,D)u(x) := \sum_{k\in\mathbb{Z}^n} \sigma(x,k)\tilde{u}(k)\mathrm{e}^{2\pi\mathrm{i}k\cdot x}\,.$$

The function $\sigma \in C^\infty(\mathcal{T}^n \times \mathbb{Z}^n)$, which is called the *symbol* of the operator $\sigma(x,D)$, is assumed to belong to a certain class $\sum^\mu$ for some $\mu \in \mathbb{C}$. Here $\sum^\mu$ is comprised of all symbols σ of the form $\sigma = \sigma_0 + \sigma_1$, where $\sigma_0 \in C^\infty(\mathcal{T}^n \times (\mathbb{R}^n/\{0\}))$ is homogeneous of degree $\mu \in \mathbb{C}$, i.e. $\sigma_0(x,0) = 1$, $\sigma_0(x,\lambda k) = \lambda^\mu \sigma_0(x,k)$, for $\lambda > 0$, $k \neq 0$, and $|\partial_{(x)}^\beta \tau_{(k)}^\alpha \sigma_1(x,k)| \leq c_{\alpha,\beta}(1+|k|)^{r_1-|\alpha|}$ for $x \in \mathcal{T}^n$, $k \in \mathbb{Z}^n$, and for some $r_1 < \mathrm{Re}\,\mu = r$. Here ∂ stands for the partial differential operator and τ for the partial forward difference operator. By $\mathcal{A}^r$ we denote the class of operators of the form $A = \sigma(\cdot,D) + K$ where $\sigma \in \sum^\mu$, $r = \mathrm{Re}\,\mu$ is the *order* of A, and $(Ku)(x) = \int_{\mathcal{T}^n} k(x,y)u(y)\mathrm{d}y$ with $k \in C^\infty(\mathcal{T}^n \times \mathbb{R}^n)$ is a smoothing operator. The operator $\sigma(x,D) \in \mathcal{A}^r$ is called *elliptic* if, for sufficiently large $|k|$, $|\sigma_0(x,k)| \geq c|k|^r$, $x \in \mathcal{T}^n$.

Any $A \in \mathcal{A}^r$ is a bounded linear operator $A : H^t \to H^{t-r}$, $t \in \mathbb{R}$. This operator is Fredholm if and only if it is elliptic. Here H^t denotes the classical periodic Sobolev space of order t, equipped with the norm $\|u\|_t := \left(\sum_{k\in\mathbb{Z}^n}(1+|k|)^{2t}|\tilde{u}(k)|^2\right)^{1/2}$.

2.2 Multiresolution and wavelets

Multiresolution or Multiscale Analysis (MSA) is by now a well–studied notion [16]. Here we focus only on those variants which are useful for our purpose. MSA of $L^2(\mathbb{R})$ is a sequence of nested closed subspaces $\cdots \subset V_{-1} \subset V_0 \subset V_1 \subset \cdots \subset L^2(\mathbb{R})$ with

1. $\overline{\cup_j V_j} = L^2(\mathbb{R})$,

2. $\cap_j V_j = \{0\}$,

3. $f(x) \in V_j \Leftrightarrow f(2x) \in V_{j+1}$,

4. There is a function $\varphi \in L^2(\mathbb{R})$ such that the translates $\{\varphi(x-k)\}$ $(k \in \mathbb{Z})$ form an orthonormal basis of V_0.

The function φ is called *scaling function*. Obviously, the functions $\varphi_{j,k}(x) := 2^{j/2}\varphi(2^j x - k)$ form an orthonormal basis of V_j $(j \in \mathbb{Z})$ (see [11] for examples). Since $\varphi \in V_0 \subset V_1$ there exists a sequence $\{h_k\}_{k\in\mathbb{Z}}$ (which is called the mask or the filter of φ) such that φ satisfies the *scaling equation*

$$\varphi(x) = \sqrt{2}\sum_{k\in\mathbb{Z}} h_k \varphi(2x-k)\,,\ x \in \mathbb{R}\,. \tag{2.1}$$

Moreover, the mask $\{h_k\}$ is finite if and only if φ is compactly supported. (2.1) is the key to the constructions of orthogonal wavelet bases and of fast wavelet algorithms. The wavelet space W_j ($j \in \mathbb{Z}$) is defined as the orthogonal complement of V_j with respect to $V_{j+1} = V_j \oplus W_j$. Thus

$$L^2(\mathbb{R}) = \oplus_{j\in\mathbb{Z}} W_j = V_m \oplus \oplus_{j\geq m} W_j \quad (\text{for each} \quad m \in \mathbb{Z}). \tag{2.2}$$

One of the main results of MSA reads as follows. Given φ, then there exists $\psi \in L^2(\mathbb{R})$ such that the functions $\psi_{j,k}(x) := 2^{j/2}\psi(2^j x - k)$, $k \in \mathbb{Z}$, form an orthonormal basis of W_j ($j \in \mathbb{Z}$). The function ψ is called the *mother wavelet* of MSA and is defined by

$$\psi(x) = \sqrt{2} \sum_{k\in\mathbb{Z}} g_k \varphi(2x - k), \quad g_k = (-1)^k h_{1-k}. \tag{2.3}$$

An important property of wavelets is that certain moments vanish, i.e.,

$$\int_{\mathbb{R}} x^l \psi(x) dx = 0, \quad 0 \leq l < d^*, \tag{2.4}$$

where in the case of orthogonal wavelets d^* is the order of polynomials which can be written as linear combinations of the translates $\varphi(\cdot - k), k \in \mathbb{Z}$.

More flexibility is offered by the concept of biorthogonal wavelets [11] which permits the employment of B–splines as scaling functions. In particular, the possibility of raising the order of moment conditions turns out to be essential for balancing compression and convergence rates.

In the following we need a *periodic* version of MSA. To this end, let $[\cdot]$ denote the periodization operator defined by $[f](x) := \sum_{k\in\mathbb{Z}} f(x + k)$ for any compactly supported function f. Given a compactly supported MSA $\varphi \in L^2(\mathbb{R})$, then the functions $[\varphi_{m,k}]$ and $[\psi_{m,k}]$ are 1–periodic. MSA of $L^2(\mathbb{R}^n)$ can be defined in a completely analogous manner. However, one needs a finite number of mother wavelets depending on the type of scaling.

2.3 Generalized Petrov–Galerkin schemes

The spaces V_j will be used as trial spaces for the approximate solution of the equation

$$Au = f\,, \tag{2.5}$$

where $A \in \mathcal{A}^r$ and $f \in H^{t-r}$ is given. In the following we will fix one such t and assume that $\eta \in H^{r-t}(\mathbb{R}^n)$ is a fixed linear functional with support in some compact set $\Gamma \subset \mathbb{R}^n$. Defining the functionals $\eta_{j,k}$ by $\eta_{j,k}(f) := 2^{-nj/2}\eta(f(2^{-j}(\cdot + k)))$, $k \in \mathbb{Z}^{n,j}$, we seek for an element $u_j \in V_j$ satisfying

$$\eta_{j,k}(Au_j) = \eta_{j,k}(f)\,,\ k \in \mathbb{Z}^{n,j} := \mathbb{Z}^n/2^j\mathbb{Z}^n\,. \tag{2.6}$$

Clearly, $\eta = \varphi$ corresponds to a classical Galerkin scheme, while $\eta = \delta(\cdot - x_0)$ give rises to collocation at the points $2^{-j}(k + x_0)$, $k \in \mathbb{Z}^{n,j}$, $j \in \mathbb{N}_0$.

Our first objective is to study the solvability of (2.6), and if this is the case, the convergence of the solution u_j as $j \to \infty$. The key to this problem is a suitable stability concept. To this end, it is convenient to consider projectors of the form

$$Q_j f = \sum_{k \in \mathbb{Z}^{n,j}} \eta_{j,k}(f) \zeta_{j,k},$$

where the $\zeta_{j,k}$ are suitable basis functions satisfying $\eta_{j,k}(\zeta_{j,l}) = \delta_{l,k}$ (possibly spanning spaces different from V_j, see [6] Sect. 4). Then (2.6) can be rewritten as operator equation $Q_j A u_j = Q_j f$. The scheme (2.6) is called (t,r)–*stable* if

$$\|Q_j A v\|_{t-r} \geq c \|v\|_t \tag{2.7}$$

for all $v \in V_j$ uniformly in $j \in \mathbb{N}_0$. Clearly, (2.7) means that the finite dimensional operators $A_j := Q_j A P_j : H^t \to H^{t-r}$ have uniformly bounded inverses $A_j^{-1} : \text{im } Q_j \to \text{im } P_j$.

It turns out that the stability of (2.6) is equivalent to the ellipticity of a certain function λ which shares many features with the principal symbol σ_0 of the operator A and which is called the *numerical symbol* (more precisely, the "symbol of the Petrov–Galerkin scheme (2.6)"). The numerical symbol is defined by

$$\lambda(x,y) := \sum_{k \in \mathbb{Z}^n} \sigma_0(x, y+k) \hat{\varphi}(y+k) \overline{\hat{\eta}(y+k)} \tag{2.8}$$

for all $x, y \in \mathcal{T}^n$ (provided that the series on the right hand side of (2.8) converges absolutely for all $x, y \in \mathcal{T}^n$), where $\hat{}$ stands for the Fourier transform.

Theorem: *[[6]] The scheme* (2.6) *is* (t,r)*–stable if and only if the numerical symbol* λ *is elliptic, i.e.* $|\lambda(x,y)| \geq c|\theta(y)|^r$ *holds uniformly in* $x \in \mathcal{T}^n$*, where* $\theta = (\theta_1, ..., \theta_n)^T$, $\theta_j(y) = e^{2\pi i y_j} - 1$.

Example: Consider the knot collocation (i.e. $x_0 = 0$) with tensor product splines of degree d. Then $\hat{\eta} = 1$ and $\hat{\varphi}(\xi) = \prod_{l=1}^n (\sin \pi \xi_l)^{d+1} / (\pi \xi_l)^{d+1}$. Clearly, if d is odd, then λ is elliptic if and only if A is strongly elliptic, i.e. Re $\sigma_0 \geq \text{const} > 0$. Hence, in that case, Theorem 1 provides stability of the collocation with odd degree splines for strongly elliptic operators (cf. [1] for the case $n = 1$ and [6] for the multidimensional case). Notice that, in the case of the classical Galerkin scheme, the ellipticity of the numerical symbol for strongly elliptic operators is a consequence of the stability property of φ.

Applying Theorem 1 in combination with well known Galerkin techniques and approximation properties of the functions $[\varphi_{j,k}]$ one obtains optimal estimates for the error $\|u - u_j\|_t$ for a certain range of Sobolev norms (see [6]).

2.4 Matrix compression and fast solution

This technique is essentially based on estimates for the asymptotic behavior of the entries of A_j $(j \to \infty)$ in the wavelet representation. We will present such estimates only for the case of classical Petrov–Galerkin schemes, i.e., when η is actually an L^2–function.

Theorem: *[[7, 17]] Let $2d^* + n + r > 0$ where d^* denotes the number of vanishing moments of the wavelets. Then for $A \in \mathcal{A}^r$ the estimate*

$$|(A[\psi_{l,k}], [\psi_{l',k'}])| \leq c \, \frac{2^{-(l+l')(n/2+d^*)}}{\varrho^{n+r+2d^*+2}} \tag{2.9}$$

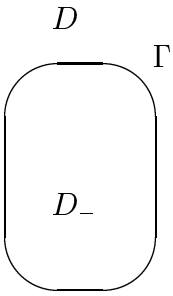
holds, where $\varrho = \text{dist}\,(\text{supp}\,[\psi_{l,k}], \text{supp}\,[\psi_{l',k'}])$ and the constant c depends only on r, n and d^*.

(2.9) are the crucial estimates which compression criteria rely on. Such criteria allow us to avoid the computation of the full stiffness matrix in the wavelet representation and tell us which entries must be computed in order to guarantee a required accuracy. Moreover, by realizing sufficiently high accuracy on lower scales, the asymptotical convergence rates of the solutions to the uncompressed systems can be preserved for those of the compressed system. Such a compression strategy reduces the computational work to the order $O(N(\log N)^b)$, where $N = 2^{jn}$ and b is a positive number ([7]).

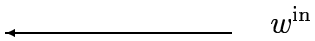
3 Exterior Dirichlet problem for the Helmholtz equation in 2D

3.1 Statement of the problem

The treatment of scattering of time–harmonic acoustic and electromagnetic waves on infinitely long cylindrical obstacles in $\mathbb{R}^3$ with simply connected cross section $D_- \subset \mathbb{R}^2$ and smooth boundary Γ leads to an exterior boundary value problem for the Helmholtz equation in $\mathbb{R}^2$.



Given the boundary data g and
the wave number $\bar{k} > 0$ in D, real.



We seek the solution w of the following problem

$$\Delta w(x) + \bar{k}^2 w(x) = 0 \quad \text{in} \quad D = \mathbb{R}^2 \backslash \overline{D_-}\,, \tag{3.1}$$

$$w(x) = g(x) \quad \text{on} \quad \Gamma\,, \tag{3.2}$$

$$\frac{\partial w(x)}{\partial r} - \mathrm{i}\bar{k} w(x) = o(r^{-1/2})\,,\ r = |x| \to \infty \tag{3.3}$$

uniformly in all directions (*Sommerfeld's radiation condition*). This problem is known to have a unique solution (cf. [5]).

Using a *single–layer approach* (indirect method), one seeks w in the form

$$w(x) = \int_\Gamma \phi(x,y)\xi(y)\mathrm{d}s(y)\,,\ x \in D\,, \tag{3.4}$$

where s is the arc length parametrization, and the fundamental solution corresponding to the Helmholtz equation is given by

$$\phi(x,y) := \frac{1}{2\mathrm{i}} H_0^{(1)}(\bar{k}|x-y|)\,, x \neq y\,.$$

Here $H_0^{(1)}$ means the Hankel function of order zero and of the first kind. Substituting (3.4) into (3.2) gives the boundary integral equation for the unknown density ξ:

$$\int_\Gamma \phi(x,y)\xi(y)\mathrm{d}s(y) = g(x)\,,\ x \in \Gamma\,. \tag{3.5}$$

Equation (3.5) is uniquely solvable in $H^s(\Gamma)$ provided that the homogeneous Dirichlet problem for the interior of Γ admits only the trivial solution.

Let $x(t) = (x_1(t), x_2(t))\,,\ 0 \le t \le 2\pi$, be a 2π–periodic regular parametrization of Γ satisfying $[x_1'(t)]^2 + [x_2'(t)]^2 > 0$, $t \in \Gamma$. Setting in (3.5) $u(t) := \xi(x(t))\left\{[x_1'(t)]^2 + [x_2'(t)]^2\right\}^{1/2}$ and $f(t) := g(x(t))$, the exterior Dirichlet problem for the Helmholtz equation leads to the following *logarithmic singular* integral equation of the first kind:

$$\begin{aligned} (Au)(t) &:= \frac{1}{2\pi}\int_0^{2\pi} K(t,\tau)u(\tau)\mathrm{d}\tau = f(t)\,,\ 0 \le t \le 2\pi, \\ K(t,\tau) &:= \frac{\pi}{\mathrm{i}} H_0^{(1)}(\bar{k}r(t,\tau)) \end{aligned} \tag{3.6}$$

where $r(t,\tau) := \left\{[x_1(t)-x_1(\tau)]^2 + [x_2(t)-x_2(\tau)]^2\right\}^{1/2}$.
In principle it is possible to separate the logarithmic part of the Hankel function and treat it separately. There exist several fast methods for the solution of the logarithmic kernel equation (also called Symm's equation). In particular, one can use the fast Fourier transform, apply the fast method [20] or the exponentially convergent method [14]. Here we do not separate the logarithmic part because we only consider it as a model problem for other cases where a separation is usually not applicable. A second related reason is to simplify the implementation by keeping it as independent as possible of further analytic investigations which have to be tuned to the particular application at hand.

3.2 Wavelet discretization method

For the numerical solution of (3.6) consider the knot collocation method on the following nested grids of $[0, 2\pi]$:
$\nabla^l = \{t_k^l | t_k^l := kh_l,\ k = 0, ..., N_l - 1,\ h_l = 2^{-l}\cdot 2\pi,\ N_l = 2^l\}$, $l = 0, 1, ..., j$. The set of additional knots for passing from level $l-1$ to level l is denoted by $\Delta^{l-1} := \nabla^l \backslash \nabla^{l-1}$. For a corresponding Galerkin wavelet method the reader is referred to [17]. As trial functions we use here continuous piecewise linear functions spanned by the canonical periodized hat functions $[\varphi_{j,k}]$, i.e.,

$$\varphi(x) := \begin{cases} 1+x & \text{if } -1 \le x \le 0, \\ 1-x & \text{if } 0 \le x \le 1, \\ 0 & \text{else.} \end{cases}$$

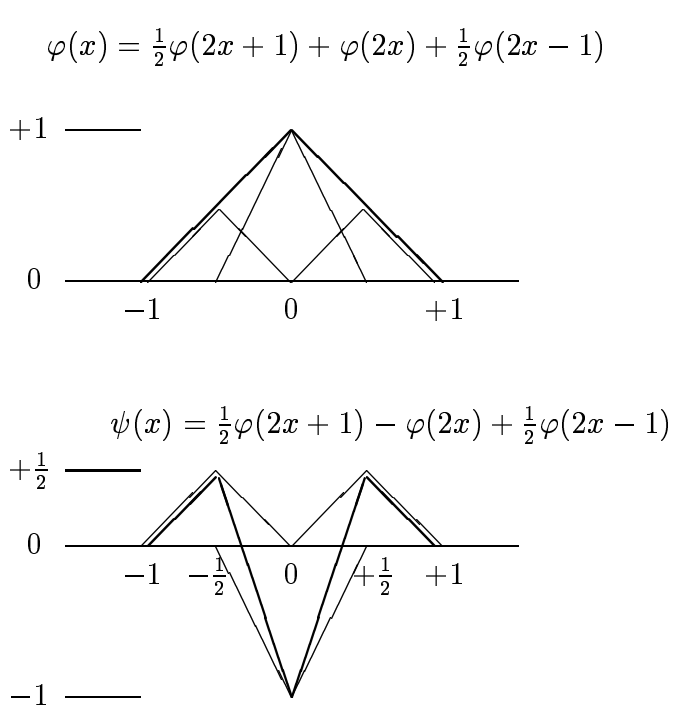


Figure 1: Two-scale relations for our choice of generating function and mother wavelet.

In this case the mask in the two-scale relation (2.1) is

$$\{h_k\}_{k=-1}^{1} = \{\frac{1}{2}, 1, \frac{1}{2}\}. \tag{3.7}$$

Since the translates $\varphi(\cdot - k)$ are not pairwise orthogonal the simple recipe from (2.3) for forming orthogonal wavelets does no longer work. Nevertheless, one can resort to the literature for a long list of candidates $\psi(x)$ whose integer translates are stable and span the orthogonal or other complements of V_0 in V_1. In particular, for any desired order d^* of vanishing moments biorthogonal wavelets have been constructed in [4] which all give rise to Riesz bases. Specifically, the wavelet for $d^* = 2$ from this family has support $[-1, 2]$ and the mask has the form

$$\{g_k\}_{k=-1}^{3} = \{\frac{1}{8}, \frac{1}{4}, -\frac{3}{4}, \frac{1}{4}, \frac{1}{8}\}. \tag{3.8}$$

To keep the support of ψ as small as possible we will focus here though on the following choice with also vanishing moments of order $d^* = 2$:

$$\{g_k\}_{k=-1}^{1} = \{\frac{1}{2}, -1, \frac{1}{2}\}, \tag{3.9}$$

which is depicted in Figure 1. These wavelets are likewise biorthogonal in the sense of [4] and therefore give rise to a Riesz basis as is shown in [18]. Since we are employing a collocation method the *Brandt/Lubrecht functionals* [2] are adequate "test wavelets" (because of pointwise evaluation) spanned by δ–distributions:

$$\eta_{l,k}(f) = f(t_{2k+1}^{l+1}) - \frac{1}{2}\Big[f(t_k^l) + f(t_{k+1}^l)\Big], \tag{3.10}$$

$$k = 0, ..., 2^{l-1} - 1, l = j - 1, ..., 1;\ \eta_{0,k}(f) = f(t_k^1), k = 0, 1.$$

One easily verifies that the $\eta_{l,k}$ have also vanishing moments of order $d^* = 2$. Figure 2 illustrates for three levels l how the point values are transformed into the data $\eta_{l,k}(f)$. Obviously, the complexity of this transformation is of the order $\mathcal{O}(N_j)$ when j is as above the finest level. Given (3.10), the collocation stiffness matrix A_j on level j relative to the

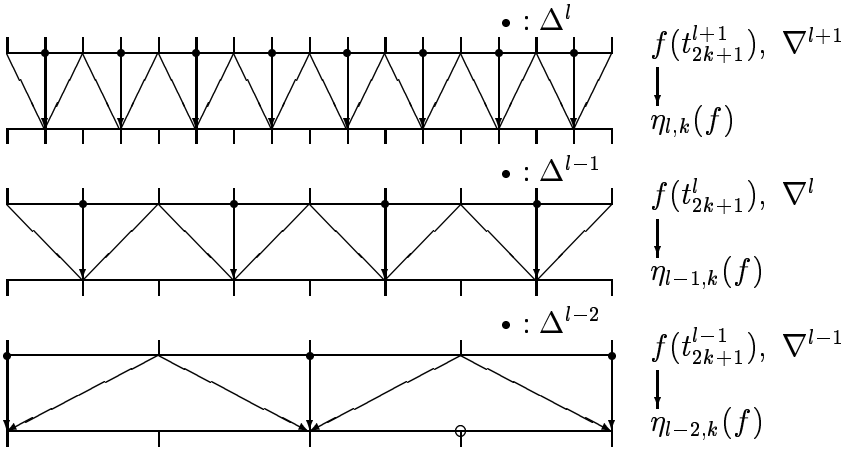


Figure 2: Generation of the functionals of Brandt/Lubrecht. Open circles indicate those grid points whose values are to be calculated in the next step.

wavelet basis has entries of the form

$$(A_j)_{(l,k),(l',k')} := \eta_{l',k'}(A[\psi_{l,k}]), \tag{3.11}$$

$$k = 0, \ldots, 2^{l-1} - 1,\ k' = 0, \ldots, 2^{l'-1} - 1,\ l, l' < j\,.$$

It is easily seen that the integral operator A defined by (3.6) belongs to the class $\mathcal{A}^r$ (see Sect. 2) with $r = -1$ and $r_1 = -3$. Moreover, it is strongly elliptic and, hence, the considered collocation method is stable (see Theorem 2). Furthermore, the error estimate

$$\|u - u_j\|_s \leq c 2^{j(s-2)} \|u\|_2 \tag{3.12}$$

holds for all s, $-1 \leq s < 3/2$, provided $f \in H^3$.

Using an `a-priori` *compression criterion* established in [7] and improved in [21] we only calculate and store approximations to those matrix entries of (3.11) obtained through quadrature for which

$$\operatorname{dist}(\Omega_{l,k}, \tilde{\Omega}_{l',k'}) \leq \max\left\{a 2^{-l}, a 2^{-l'}, a j^{5/6} \cdot 2^{(2/3)j - (4/3)l - (2/3)l'}\right\}, \tag{3.13}$$

where $\Omega_{l,k}$ is the support of $[\psi_{l,k}]$, $\tilde{\Omega}_{l',k'}$ the support of $\eta_{l',k'}$, and a a constant that has to be chosen appropriately. Note that, according to the general theory, to realize an (asymptotically) almost optimal compression rate for the choice of piecewise linear wavelets with two vanishing moments it would be necessary to use Brandt/Lubrecht functionals with three vanishing moments. In this case also criterion (3.13) changes somewhat. Nevertheless,

the present method turns out to be more efficient for the present scope of experiments because of the smaller support of the test functionals. Also our convergence studies confirm a sufficiently good nearly optimal performance of the present version in all our test problems (see Figs. 3,4,5).

In order to get a fully discrete method we have to employ a suitable quadrature method to approximate the integrals. Because of the logarithmic singularity of the kernel function it is necessary to use an adapted rule for the quadrature to guarantee that the order of convergence is better than one. Therefore we choose the following graded quadrature points τ_q of the interval $[0, 2\pi]$ with grading parameter $\alpha \geq 1$. More precisely, the points are graded near the singular point $t \in (0, 2\pi)$ as follows:

$$\tau'_q = \pi \Big|\frac{q}{\tilde{m}}\Big|^{\alpha} \operatorname{sign} q + t, \quad q = -\tilde{m}, ..., \tilde{m} - 1.$$

The integer $\tilde{m}$ has to be chosen proportional to N_j. To make the quadrature exact for the ansatz functions these graded points are united with equidistant ones: $\{\tau^{\mathrm{e}}\}_0^{N_j-1} = \{t_k^j\}$. Let $\tau_q, q = 0, \ldots, m-1,\ m = 2\tilde{m} + N_j$ be the quadrature points obtained by periodizing $\{\tau'_q\} \cup \{\tau^{\mathrm{e}}\}$. In our numerical examples we choose a grading parameter $\alpha = 2.5$ together with the trapezoidal rule. The latter is exact for linear polynomials, and hence sufficient to obtain second order convergence in L^2 (cf. (3.12)).

3.3 Assembling the compressed stiffness matrix

Let

$$(\tilde{A}_j[\psi_{l,k}])(t_i) = \sum_{\substack{q=0 \\ \tau_q \neq t}}^{m-1} K(t_i, \tau_q)\omega_q \psi_{l,k}(\tau_q), \quad k = 0, \ldots, 2^l - 1,\ l < j,$$

be the approximation of $(A\psi_{l,k})(t_i)$ obtained by quadrature. Here ω_q are quadrature weights.

Keeping in mind that

$$\eta_{l',k'}(A[\psi_{l,k}]) = (A[\psi_{l,k}])(t_{2k'+1}^{l'+1}) - 1/2\{(A[\psi_{l,k}])(t_{k'}^{l'}) + (A[\psi_{l,k}])(t_{k'+1}^{l'})\}$$

and that the vanishing moments of the test wavelets $\eta_{l,k}$ imply that these quantities decay, their approximations obtained through quadrature are expected to exhibit the same behavior. Thus we have to compute the following expressions:

$$\begin{aligned} \eta_{l',i}(\tilde{A}_j[\psi_{l,k}]) \ = \ & \sum_{\substack{q=0 \\ \tau_q \neq t_{2i+1}^{l'+1}}}^{m-1} K(t_{2i+1}^{l'+1}, \tau_q)\omega_q[\psi_{l,k}](\tau_q) - \\ & \frac{1}{2} \sum_{\substack{q=0 \\ \tau_q \neq t_i^{l'}}}^{m-1} K(t_i^{l'}, \tau_q)\omega_q[\psi_{l,k}](\tau_q) - \\ & \frac{1}{2} \sum_{\substack{q=0 \\ \tau_q \neq t_{i+1}^{l'}}}^{m-1} K(t_{i+1}^{l'}, \tau_q)\omega_q[\psi_{l,k}](\tau_q)\,, \end{aligned} \tag{3.14}$$

$$i = 0, \ldots, N_{l'} - 1, \quad k = 0, \ldots, N_l - 1, \quad l, l' = 1, \ldots, j-1,$$

$$\begin{aligned} \eta_{l',i}(\tilde{A}_j[\varphi_{1,k}]) \;=\; & \sum_{\substack{q=0 \\ \tau_q \neq t^{l'+1}_{2i+1}}}^{m-1} K(t^{l'+1}_{2i+1}, \tau_q)\omega_q[\varphi_{1,k}](\tau_q) - \\ & \frac{1}{2} \sum_{\substack{q=0 \\ \tau_q \neq t^{l'}_{i}}}^{m-1} K(t^{l'}_{i}, \tau_q)\omega_q[\varphi_{1,k}](\tau_q) - \\ & \frac{1}{2} \sum_{\substack{q=0 \\ \tau_q \neq t^{l'}_{i+1}}}^{m-1} K(t^{l'}_{i+1}, \tau_q)\omega_q[\varphi_{1,k}](\tau_q), \end{aligned} \tag{3.15}$$

$$i = 0, \ldots, N_{l'} - 1, \quad k = 0, 1, \quad l' = 1, \ldots, j-1,$$

$$\eta_{0,i}(\tilde{A}_j[\psi_{l,k}]) = \sum_{\substack{q=0 \\ \tau_q \neq t^1_i}}^{m-1} K(t^1_i, \tau_q)\omega_q[\psi_{l,k}](\tau_q), \tag{3.16}$$

$$i = 0, 1, \quad k = 0, \ldots, N_l - 1, \quad l = 1, \ldots, j-1,$$

$$\eta_{0,i}(\tilde{A}_j[\varphi_{1,k}]) = \sum_{\substack{q=0 \\ \tau_q \neq t^1_i}}^{m-1} K(t^1_i, \tau_q)\omega_q[\varphi_{1,k}](\tau_q), \tag{3.17}$$

$$i = 0, 1, \quad k = 0, 1.$$

Here it is important to note that, according to our above remarks, these calculations are only performed for those pairs $(l', k'), (l, k)$ which satisfy condition (3.13). Thus storage is only required, and hence is directly reduced, for the significant entries of the matrix A^ε_j defined by

$$(A^\varepsilon_j)_{(l',k'),(l,k)} := \begin{cases} 0 & \text{if (3.13) does not hold,} \\ \eta_{l',k'}(\tilde{A}_j[\psi_{l,k}]) & \text{if (3.13) holds,} \end{cases}$$

where $k' = 0, \ldots N_{l'} - 1$, $k = 0, \ldots, N_l - 1$, $l, l' = 0, \ldots, j-1$ and where we have set $\psi_{0,k} := \varphi_{1,k}$.

Next let us denote by B_j the transformation (3.10) which takes the array of data $f(t^j_k), k = 0, \ldots, N_j - 1$, into the array $\eta_{l,k}(f)$. One can check that A^ε_j is the compressed version of the matrix $B_j\tilde{A}_j$. Hence we end up with the following sparse matrix equation

$$A^\varepsilon_j u_j = B_j f_j, \tag{3.18}$$

where $f_j := (f(t^j_0), f(t^j_1), \ldots, f(t^j_{N_j-1}))^T$.

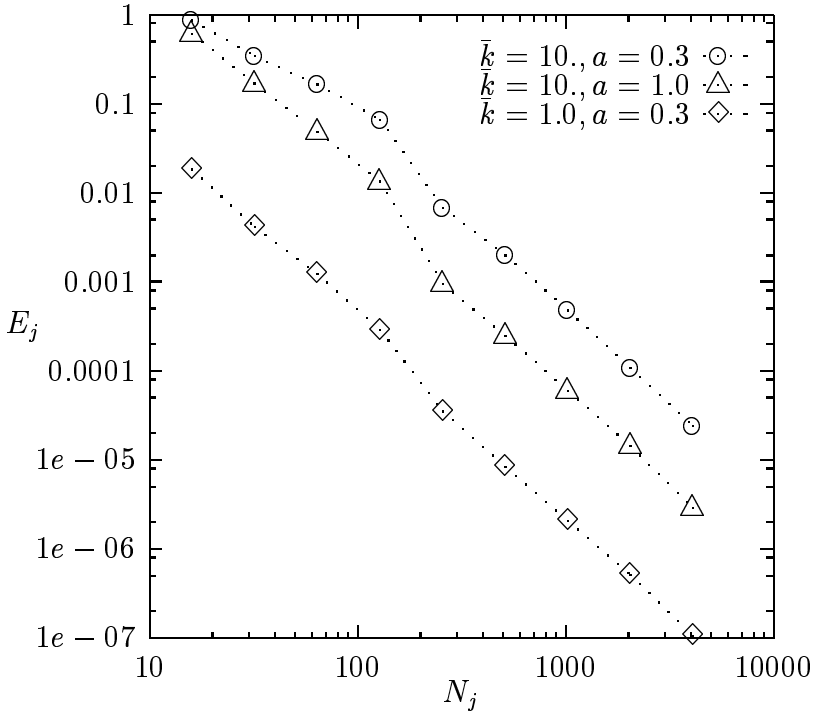


Figure 3: Error E_j in logarithmic scale for various wave numbers $\bar{k}$ and constants a for the `a-priori` compression strategy. Γ = circle.

Note that it is possible to use the same quadrature formula as for a standard collocation method not based on wavelets for all expressions in (3.14)–(3.17). It is only necessary to shift the grading for the singularity to the actual singular point.

Clearly, the solution u_j of (3.18) consists of the coefficients of the approximate solution relative to the wavelet basis. If one wishes to represent the solution in terms of the nodal basis, a further transformation

$$u_j^\varphi = T_j u_j$$

has to be performed. This transformation also has the following familiar pyramid structure

$$\begin{array}{ccccccccccc} w_{1,\cdot} & & w_{2,\cdot} & & w_{3,\cdot} & & \cdots & & w_{j-1,\cdot} & & \\ & \searrow & C_1 & \searrow & C_2 & & & & \searrow & & C_{j-1} \\ u_{1,\cdot} & \rightarrow & u_{2,\cdot} & \rightarrow & u_{3,\cdot} & \cdots & u_{j-1,\cdot} & \rightarrow & u_j^\varphi & , & \\ & M_1 & & M_2 & & & & M_{j-1} & & & \end{array} \tag{3.19}$$

where the matrices $M_i, C_i,\ i = 1, \ldots, j-1$ are sparse and contain the masks from (3.7) and (3.9). For a description of the algorithm in pseudocode see [10]. Thus T_j takes coefficients $u_j := \{u_{1,k}, w_{l,k}\}$ of the wavelet representation $\sum_{k=0,1} u_{1,k}\varphi_{1,k} + \sum_{l=1}^{j-1}\sum_{k\in\Delta^l} w_{l,k}\psi_{l,k}$ into the coefficients $u_j^\varphi := \{u_{j,k}\}$ of the corresponding nodal representation $\sum_{k\in\nabla^j} u_{j,k}\varphi_{j,k}$. Also the application of the transformation T_j is easily seen to require only $\mathcal{O}(N_j)$ operations.

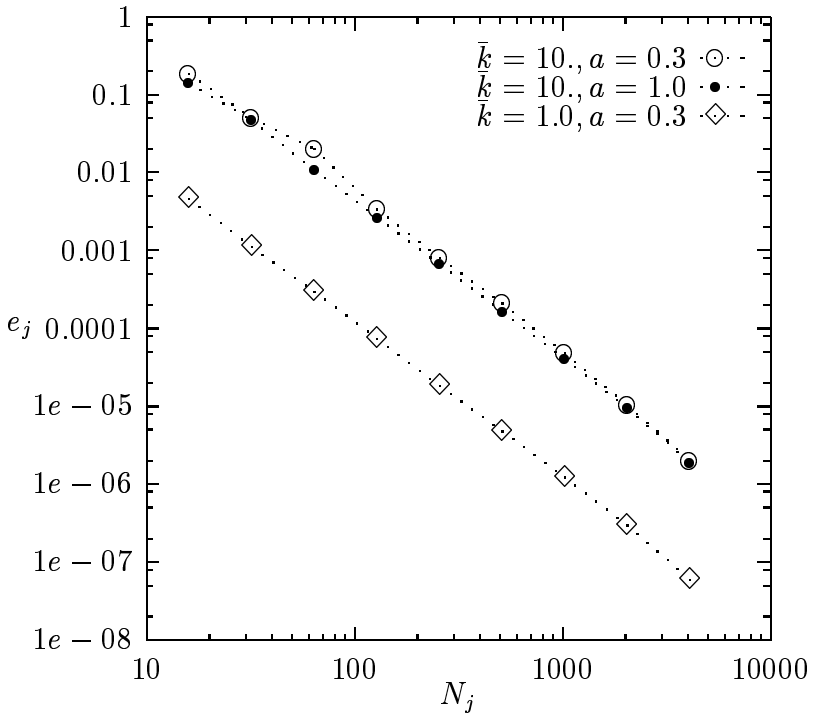


Figure 4: Error e_j in logarithmic scale for various wave numbers $\bar{k}$ and constants a for the `a-priori` compression strategy. Γ = circle.

3.4 Preconditioning and solution

The linear system (3.18) is solved with the aid of the iterative Krylov subspace method GMRES (cf. [19],[24]). This method has been proved to be efficient for this kind of systems (cf. e.g. [23],[13],[10]). Since $r = -1$, the condition numbers $\kappa(A_j^\epsilon)$ grow like N_j so that the number of iterations which is necessary to accomplish a desired accuracy increases when N_j becomes large. The theory in [21] suggests multiplying A_j^ϵ with the $N_j \times N_j$ diagonal matrix D from the right where $D_{(l,k),(l',k')} = (\bar{k}^2 + 2^{2l})^{\frac{1}{2}} \cdot \delta_{(l,k),(l',k')}$. After solving then the system $A_j^\epsilon D v_j = B_j f_j$ one has to transform the resulting vector according to $u_j = D v_j$ to obtain the wavelet coefficients of the solution.

We have also tested the following alternative preconditioning strategies based on diagonal scaling. Let $\bar{D}_{(l,k),(l',k')} = 2^l \cdot \delta_{(l,k),(l',k')}$.

1. Solve $\bar{D} A_j^\epsilon u_j = \bar{D} B_j f_j$ or

2. solve $\bar{D}^{\frac{1}{2}} A_j^\epsilon \bar{D}^{\frac{1}{2}} v_j = \bar{D}^{\frac{1}{2}} B_j f_j$ followed by corresponding diagonal back transformations.

The results of the tests for a fixed example are given in Table 1. The best preconditioner is the one suggested by the theory which gives a constant iteration count. May be that with higher order Brandt/Lubrecht functionals the iteration count can further be reduced. The corresponding tools are available and, in principle, easily implemented.

The main steps of the solution procedure may be summarized now as follows:

1. Compute the matrix A_j^ϵ directly in the wavelet representation as described by (3.14) – (3.17) and apply the `a-priori` compression criterion (3.13) yielding a sparse matrix;

2. apply the transformation B_j (3.10) to the right-hand side:
$B_j : \{f(t_k^j)\} \mapsto \{\eta_{j,k}(f)\}, \quad k = 0, \ldots, N_j - 1 : B_j f_j$, which requires only $\mathcal{O}(N_j)$ operations;

3. solve the sparse matrix equation $A_j^\varepsilon D v_j = B_j f_j$ by an iterative solver (e.g. GMRES). The complexity of this part is nearly $\mathcal{O}(N_j(\log N_j)^b)$;

4. back transform the solution of this linear system to $u_j = D v_j$ ($= N_j$ operations), which provides the coefficient vector u_j relative to the wavelet basis and

5. (optionally) apply the transformation T_j which yields the coefficients $u_j^\varphi = T_j u_j$ for the nodal basis representation. Again this only requires $\mathcal{O}(N_j)$ operations.

Clearly, the multiplications with the diagonal matrices are applied to the iteration vector rather than to the matrix A_j^ε: $A_j^\varepsilon(D u_j^{(i)})$, with $u_j^{(i)}$ the actual iteration vector of u_j during the GMRES-iteration. This only requires $2N_j$ additional operations for the diagonal matrix times vector products in each iteration step.

Summing up the operation count in steps 2.-5. the overall complexity is of order $\mathcal{O}(N_j(\log N_j)^b)$ which gives rise to a fast solution of the linear system. Unfortunately, step 1., the matrix assemblation, is still of the order $\mathcal{O}(N_j^2)$ although only (nearly) $\mathcal{O}(N_j(\log N_j)^b)$ matrix entries are calculated. This is due to the poor quadrature scheme which does not take the distance between the supports $\Omega_{l,k}$ and $\Omega_{l',k'}$ into account. Meanwhile it has been shown in [21] that such a refined quadrature strategy reduces the computational work for assembling the matrix to the order of nonvanishing matrix entries.

After the evaluation of the solution, the relative L^2 error E_j of the solution and the convergence rate β_1 are calculated:

$$E_j = \frac{\sum_{i=0}^{N_j-1} |u_j(t_i^j) - u(t_i^j)|^2}{\sum_{i=0}^{N_j-1} |u(t_i^j)|^2}, \quad \beta_1 = -\frac{\log(E_j/E_{j-1})}{\log 2}.$$

Preconditioning											
j	2	3	4	5	6	7	8	9	10	11	12
N_j	4	8	16	32	64	128	256	512	1024	2048	4096
without	3	5	9	14	18	21	29	33	37	40	44
$\bar{D}A_j^\varepsilon$	3	5	9	15	18	20	25	27	28	28	29
$\bar{D}^{\frac{1}{2}} A_j^\varepsilon \bar{D}^{\frac{1}{2}}$	3	5	9	15	17	19	24	27	26	26	27
$A_j^\varepsilon D$	3	5	9	14	**15**	**16**	**21**	**23**	**23**	**22**	**22**

Table 1: Number of GMRES iterations without preconditioning and with different preconditioners for the following example: $\Gamma =$ ellipse, $\bar{k} = 10$, $a = 1.0$.

3.5 Numerical examples for scattering

We consider the scattering of an E-polarized electromagnetic plane wave u^{in} by a perfectly conducting cylinder. In acoustics this problem is equivalent to the scattering of a plane wave by an impenetrable, sound-soft cylinder with smooth cross section. Let the incident

wave be given by $u^{\mathrm{in}}(x) = \exp(\mathrm{i}\bar{k}\tilde{e}_{\mathrm{in}}x)$, with $\tilde{e}_{\mathrm{in}}$ a unit vector describing the direction of propagation of the incident wave, and let u^{s} be the scattered wave. Then the exterior wave (total field) $u^{\mathrm{e}} = u^{\mathrm{in}} + u^{\mathrm{s}}$ is also a solution of the Helmholtz equation $\Delta u^{\mathrm{e}} + \bar{k}^2 u^{\mathrm{e}} = 0$ in D and satisfies the boundary condition $u^{\mathrm{e}} = 0$ on Γ. So we have $u^{\mathrm{s}} = -u^{\mathrm{in}}$ on Γ and u^{s} also satisfies the conditions (3.1) and (3.3). Thus in terms of our initial unknowns we have:

- $w := u^{\mathrm{s}}$,
- $g := u^{\mathrm{s}}|_{\Gamma} = -\exp(\mathrm{i}\bar{k}\tilde{e}_{\mathrm{in}}x)|_{\Gamma}$ and if we set for the incident direction $\tilde{e}_{\mathrm{in}} = (1,0)$, we obtain

$$f(x(t)) := -(\cos(\bar{k}x_1(t)) + \mathrm{i}\sin(\bar{k}x_1(t))).$$

With these data one can then solve the integral equation (3.6).

The solution u^{s}, the backscattered electric field, exhibits the following asymptotic behavior which follows from the asymptotics for the Hankel function for large argument:

$$u^{\mathrm{s}}(x) = \frac{\exp(\mathrm{i}\bar{k}|x|)}{\sqrt{|x|}} \left\{ u^{\infty}(\tilde{x}) + \mathcal{O}\left\{ \frac{1}{|x|} \right\} \right\}, \ |x| \longrightarrow \infty,$$

$$u^{\infty}(\tilde{x}) = \frac{\exp(\mathrm{i}\frac{\pi}{4})}{\sqrt{8\pi\bar{k}}} \int_{\Gamma} \xi(y) \exp(-\mathrm{i}\bar{k}\tilde{x}y) \mathrm{d}s(y).$$

uniformly for all directions $\tilde{x} := x/|x|$. The function u^{∞} is known as *far field pattern* or scattering amplitude of u^{s}. Substituting for $\xi(y)$ the solution $u(t)$ of our integral equation, we obtain

$$u^{\infty}(\tilde{x}) = \frac{\exp(\mathrm{i}\frac{\pi}{4})}{\sqrt{8\pi\bar{k}}} \int_{\Gamma} u(t) \exp(-\mathrm{i}\bar{k}\tilde{e}_{\mathrm{ob}}x(t)) \mathrm{d}t. \tag{3.20}$$

A more interesting quantity is the *radar cross section* (RCS). It is defined by $\sigma^{\mathrm{c}}(\theta') = 2\pi \lim_{|x|\to\infty} x \frac{|u^{\mathrm{s}}|^2}{|u^{\mathrm{in}}|^2}$, with $0^{\circ} \leq \theta' \leq 360^{\circ}$, the backscatter angle. With the aid of the asymptotics of u^{s} we obtain the following approximation $\sigma^{\mathrm{c}}(\theta') = 2\pi|u^{\infty}(\tilde{x})|^2$ for the RCS where $\tilde{x}$ is the direction given by θ'. When plotting the RCS as a function of the observation angle, the quantity

$$\sigma = 10\log_{10}(\sigma^{\mathrm{c}}/\lambda), \ \lambda = \text{ wavelength} \tag{3.21}$$

are shown in the figures rather than σ^{c} which is common in the engineering literature.

After the solution of the integral equation by the wavelet method, the integral (3.20) is calculated which is a functional of the solution. This is realized by the rectangular rule which in the periodic case is equivalent to the trapezoidal rule. Let the approximation of (3.20) be u_j^{∞}. For the determination of u^{∞} and u_j^{∞} we fix the direction of observation.

For a sequence of discretizations we determine the error e_j and the convergence rate β_2 for the functionals u_j^∞:

$$e_j = \frac{|u_j^\infty - u^\infty|}{|u^\infty|}, \quad \beta_2 = -\frac{\log(e_j/e_{j-1})}{\log 2}.$$

For all numerical examples we have selected $\alpha = 2.5$ as an optimal grading parameter for the quadrature points and employing the `a-priori` compression criterion, we have examined the convergence and compression behavior for various wave numbers $\bar{k}$ and constants a. In the first example we consider a disc of radius $r = 0.6$ as cross section of a scattering obstacle (cf. e.g. [3]) and we set $\tilde{e}_{\text{ob}} := (1, 0)$ while the second considered obstacle is assumed to have an elliptical cross section with $a_x = 2b_y = 0.6$.

Figures 3 and 4 present the behavior of the errors E_j, e_j for the disc. Figure 5 shows results for the case of an ellipse. Figures 3,4,5 clearly show the predicted behavior for the convergence rate of E_j, e_j for the investigated boundaries if the constant a for the `a-priori` compression criterion is appropriately chosen. The convergence rates of ≈ 2 coincide with the expected convergence rate of the solutions to the full systems when using the same discretization method. Because of (3.12) for e_j a convergence rate of 3 is possible when using a quadrature of higher exactness.

The constant a from (3.13) in practical applications should be chosen so that the supports of $\psi_{j,k}, \eta_{j,k'}$ do not overlap on the finest level j. This leads to a fixed bandwidth for the diagonal bandmatrix on the finest levels j and determines all other bandwidths of the appearing block band matrices. For our special choice of $\psi_{l,k}, \eta_{l,k}$ we recommend a bandwidth $\bar{b}$ of $\bar{b} \geq 4$. For greater values of $\bar{k}$ or more complex domains D_- it may be necessary to increase this value by small amounts.

Obviously, the error e_j for the functional of the solution is not as sensitive to different compression constants a as the error E_j. This is due to the fact that a quadrature rule of higher exactness would yield a better compression rate for this error.

Figures 6 and 7 show the number nze of nonzero elements of the compressed matrix A_j^ϵ for some constants a and the time t_{GM} for the iterative solution of the linear system by a preconditioned GMRES. One clearly observes that the time depends on the wave number $\bar{k}$ as well as on the constant a for the compression criterion.

Finally, Figs. 8,9,10 present some results for the RCS for two domains and two different wave numbers $\bar{k}$. The results agree with those obtained with a conventional collocation method.

4 Dirichlet problem for the Laplace equation in 3D

In this section new results for the multiscale method proposed in [10] are presented for a large number of degrees of freedom N_j. Note that linear systems with nearly 100,000 unknowns are solved. Since the thresholding procedure is chosen so that the optimal asymptotic accuracy of the discretization error is preserved this corresponds to solving fully populated linear systems of the same size arising from non-wavelet approaches.

We start briefly recalling the main ingredients of the method and refer to [10] for further details. Note that the theory outlined in Section 2 does not apply directly because we are

dealing now with a non-periodic problem and the boundary of the considered polyhedral domain is not smooth. Nevertheless, the decay of the matrix entries in the wavelet representation on a face of the polyhedron [10] is of the same form as for the periodic case [7]. Accordingly, the results are also similar to those predicted in the model problem.

It is a well known fact that the Dirichlet problem for the Laplace equation in some domain $\mathcal{P} \subset I\!R^3$ can be transformed into a second kind integral equation over its boundary [15]. For a polyhedral domain $\mathcal{P}$ this equation over the boundary $\Omega = \partial\mathcal{P}$ reads

$$\begin{aligned} Au &:= (I + 2W)u = f, \\ Wu(x) &:= [1/2 - \theta_\Omega(x)]u(x) + \frac{1}{4\pi}\int_\Omega \frac{n_y \cdot (x-y)}{|y-x|^3} u(y) \mathrm{d}_y\Omega, \end{aligned} \tag{4.1}$$

where $\theta_\Omega(x)$ is the inner solid angle of Ω at $x \in \Omega$ and n_y the unit vector of the interior normal to $\mathcal{P}$ at y. W is the double layer potential operator. 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.

4.1 Triangulation and discretization

As mesh for the boundary Ω a regular triangulation is used. Given some initial partition (mesh Ω^0) of Ω into triangles, subsequent subdivision of each triangle of Ω^0 into four congruent subtriangles leads to Ω^1. This process generates a sequence of nested meshes Ω^j of *depth* or *level* $j \in I\!N_0$. Clearly, the *meshsize* h on the level j is proportional to 2^{-j}. The vertices x_K of the triangles are the *knot points* of the mesh. The indices K on a given level define a *grid* $\nabla^j = \{K : x_K \text{ is a knot in } \Omega^j\}$. Again, the set of additional knots added when passing from level $j-1$ to level j is $\Delta^{j-1} := \nabla^j \backslash \nabla^{j-1}$.

As trial functions, we employ properly normalized *Courant hat functions* $\varphi_{j,K}$ which are frequently used in finite element or boundary element methods. More precisely, each basis function $\varphi_{j,K}$ is a continuous function on Ω whose restriction to any triangle τ in Ω^j is affine and satisfies the nodal conditions $\varphi_{j,K}(x_{K'}) = 2^j \delta_{K,K'}$, for all $K, K' \in \nabla^j|_\tau$. The spaces $V_j := \text{ span } \{\varphi_{j,K} : K \in \nabla^j\}$ of piecewise linear continuous functions on Ω are by construction nested $V_0 \subset V_1 \subset \cdots \subset V_{j-1} \subset V_j$. By considering the restriction to the planar faces of $\mathcal{P}$ one readily confirms second order accuracy of the trial spaces.

Consider knot collocation on the finest grid ∇^j, i.e., we wish to determine a piecewise linear and continuous function $u_j^\varphi \in V_j$ such that

$$Au_j^\varphi(x_K) = f(x_K), \quad K \in \nabla^j \ . \tag{4.2}$$

Thus the entries of the collocation stiffness matrix are given by

$$(A\varphi_{j,K'})(x_K) = \frac{1}{2\pi}\int_{\text{supp}\,\varphi_{j,K'}} \frac{n_y \cdot (x_K - y)}{|y - x_K|^3} \varphi_{j,K'}(y) \mathrm{d}_y\Omega, \tag{4.3}$$

$$K, K' \in \nabla^j, K \neq K'.$$

$\text{supp}\varphi_{j,K'}$ consists of the six triangles τ, for which knot K' is a common vertex. In order to get a fully discrete method quadrature which is known to be exact for polynomials

p of degree at most two is used for approximating the integrals (e.g. [22]). 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 11). The quadrature formula for a function $v(x)$ then reads

$$\int_\tau v(x)\mathrm{d}x \simeq \frac{1}{3}\sum_{i=1}^{3} v(x_{K_i'})|\tau|. \tag{4.4}$$

Moreover, to treat the singularities in a proper way, usually a regularization technique is employed (cf. [13]) based on rewriting $Au = f$ as

$$2u(x) + \frac{1}{2\pi}\int_\Omega \frac{n_y \cdot (x-y)}{|y-x|^3}[u(y) - u(x)]\mathrm{d}y = f(x), \quad x \in \Omega. \tag{4.5}$$

Note that some calculations presented in Tables 2 and 3 are performed without singularity subtraction.

In summary one obtains the following linear system with the discrete collocation matrix

$$A_j^\varphi u_j^\varphi = f_j, \quad A_j^\varphi = (a_{K,K'})_{K,K' \in \nabla^j}. \tag{4.6}$$

4.2 Decomposition of function spaces

Our approach is based on a multiscale decomposition of the given trial spaces induced by the following two scale relations. The refinement equation (2.1) now takes the form

$$\varphi_{j,K} = \sum_{K' \in \nabla^{j+1}} m_{j,K',K}\varphi_{j+1,K'}, \ K \in \nabla^j. \tag{4.7}$$

The coefficients $m_{j,K',K}$ are called *mask* or *filter* coefficients. On those grid points located in the interior of the triangles $\tau \in \omega^0$ the entries of the sparse $N_{j+1} \times N_j$ matrix $(m_{j,K',K})_{K' \in \nabla^{j+1}, K \in \nabla^j}$ can be represented by the following 7 point stencil

$$\begin{pmatrix} 0 & 1 & 1 \\ 1 & \mathbf{2} & 1 \\ 1 & 1 & 0 \end{pmatrix}. \tag{4.8}$$

Here the bold value indicates a position K in the coarse grid ∇^j. Relation (2.3) which complements the basis in V_j to one for V_{j+1}, now reads

$$\psi_{j,K} := \sum_{K' \in \nabla^{j+1}} c_{j,K',K}\varphi_{j+1,K'}, \quad K \in \Delta^j = \nabla^{j+1} \setminus \nabla^j, \tag{4.9}$$

and in the interior of each $\tau \in \Omega^0$, $c_{j,K',K}$ corresponds to one of the following 3 point stencils

$$\begin{pmatrix} 0 & 0 & 0 \\ -1 & \mathbf{2} & -1 \\ 0 & 0 & 0 \end{pmatrix}\frac{1}{2}, \quad \begin{pmatrix} 0 & -1 & 0 \\ 0 & \mathbf{2} & 0 \\ 0 & -1 & 0 \end{pmatrix}\frac{1}{2}, \quad \begin{pmatrix} 0 & 0 & -1 \\ 0 & \mathbf{2} & 0 \\ -1 & 0 & 0 \end{pmatrix}\frac{1}{2}, \tag{4.10}$$

depending on the direction of the edge containing the knot x_K, $K \in \Delta^j$. Here the boldface values correspond to indices $K \in \Delta^j$.

Now let as before T_j denote the transformation that takes the coefficient vector $\tilde{u}_j$ of some element of V_j relative to the *multiscale basis* $\{\varphi_{0,K} : K \in \nabla^0\} \cup \{\psi_{l,K} : K \in \Delta^l,\ l = 0, \ldots, j-1\}$ into the coefficient vector u_j^φ relative to the *nodal basis* $\{\varphi_{j,K} : K \in \nabla^j\}$. The linear system (4.6) is equivalent to

$$A_j \tilde{u}_j = T_j^* f_j, \quad f_j = f(x_K),\ K \in \nabla^j,$$

where $A_j := T_j^* A_j^\varphi T_j$ is the collocation stiffness matrix relative to the multiscale basis.

The transformation T_j has the same pyramid structure as described in Section 3. Due to the finite supports of the masks its application requires the order of $\mathcal{O}(N_j)$ operations.

The essential steps of the present preliminary version of the algorithm may now be described as follows:

1. The discretized collocation method in the nodal basis gives rise to a linear system (4.6). The complexity of assembling the matrix is $\mathcal{O}(N_j^2)$.

2. The transformations T_j, T_j^* lead to the matrix $A_j := T_j^* A_j^\varphi T_j$, so that the system is equivalent to $A_j \tilde{u}_j = T_j^* f_j$, and $u_j^\varphi = T_j \tilde{u}_j$. Since A_j^φ is fully populated the complexity of this part is still $\mathcal{O}(N_j^2)$. In view of the order of our quadrature rule, the estimates in [10] predict a significantly faster decay of the entries in A_j than of those in A_j^φ. Replacing all those entries in A_j by zero whose modulus stays below a given threshold th yields the *compressed* system $A_j^\varepsilon u_j = T_j^* f_j$. To improve the scheme one has to compute A_j^ε directly without going through A_j^φ. The theoretical concepts needed to perform this step in an efficient way have now become available [21].

3. This latter system is solved with the aid of a sparse iterative solver (e.g. GMRES) without preconditioning since the order of the operator is zero in this case.

The decay estimates (see [10]) allow us to estimate the deviation of the solution of the compressed system from the solution to the full system and, in some cases, also from the exact solution of $Au = f$. As pointed out above at this stage this method has still an overall complexity of $\mathcal{O}(N_j^2)$. Only the last step, after the thresholding procedure, has a complexity which is nearly linear and therefore leads to a fast solution of the linear system. As one can see in Tables 2 and 3, quadrupling the number N_j of unknowns leads to quadrupled nze only. In contrast to the method of Section 3 it does not save yet memory because in the first two steps, N_j^2 matrix entries are handled. Nevertheless, so far the primary objective of these test has been to confirm the performance of such techniques with regard to compression and convergence behavior in situations where not all the assumptions of the model case are fulfilled. The results show that efficient compression is possible without deteriorating the resulting accuracy of the discrete solutions significantly.

4.3 Numerical examples

For our numerical experiments we consider a Dirichlet problem for Laplace's equation $\Delta U(x) = 0 \quad , \quad x \in \mathcal{P}$, and smooth Dirichlet data, $U(x)|_\Omega = f(x), \quad x = ({}_1x, {}_2x, {}_3x) \in \Omega$.

In spite of the smoothness of the Dirichlet data the solution will generally not be smooth because the boundary is not smooth.

Here we mainly present numerical experiments for two different polyhedra, a cube, and a pyramid. For a nonconvex domain (bench) and further more detailed results, see [10].

Recall that the solution U of the above Dirichlet problem with boundary data $f(x),\ x \in \Omega$ has the representation

$$U(x) = \frac{1}{4\pi} \int_\Omega \frac{n_y \cdot (x-y)}{|x-y|^3} u(y) \mathrm{d}_y \Omega, \quad x \in \mathcal{P}, \tag{4.11}$$

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

In order to control our compression error, we have to monitor the error between the exact solution u, and the approximate solution u_j^φ obtained by the given (uncompressed) discretized collocation method on a grid ∇^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^φ 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. Fig.11):

$$U_j^\varphi(x) := \frac{1}{4\pi} \sum_{\tau \in \Omega^j} \frac{1}{3} \sum_{i=1}^{3} \frac{n_\tau \cdot (x - y^\tau_{K_i'})}{|x - y^\tau_{K_i'}|^3} u_j^\varphi(y^\tau_{K_i'}) |\tau|, \tag{4.12}$$

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 $u_j^\varphi(y^\tau_{K_i'}) := \frac{1}{2}(u_j^\varphi(y^\tau_{K_i}) + u_j^\varphi(y^\tau_{K_k}))$.

As a further control we replace the exact solution u of (4.5) by the numerical solution u_J on a very fine grid and compute $\mathrm{ERR}_j^\varphi = \frac{\|u_j^\varphi - u_J\|_0}{\|u_J\|_0}$ 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 (see [10]).

In the multiscale representation A_j of the operator we discard those elements which are below a given threshold th and end up with a compressed or sparsified matrix A_j^ϵ. 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 := $\frac{N^2}{\mathrm{nze}}$. By compression we obtain a perturbed system. The solution of this perturbed system yields the approximate solution u_j (which at this stage is given in its multiscale representation). Evaluating u_j at the points $y^\tau_{K_i'}$ and substituting these values into the discrete representation formula (4.12) for u_j^φ, provides $U_j(x)$. The evaluation is most efficiently performed by transforming the solution of the compressed system into nodal basis coefficients with the aid of T_j and exploiting then the localness of the basis functions $\varphi_{j,K}$.

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 u_j we again replace the exact solution u of (4.5) by the numerical solution u_J on a very fine level J and compute $\mathrm{ERR}_j = \frac{\|u_j - u_J\|_0}{\|u_J\|_0}$. In order to determine which threshold results in a suitable compression, we compare this error with the error ERR_j^φ of the discretization scheme. Similarly we compare ERR_{x^i} with the corresponding error $|U(x^i) - U_j(x^i)|$. To estimate this error we compute the maximum $\mathrm{MERR}_x := \max_{i=1,2,3}|U(x^i) - U_j(x^i)|$. This is the second quantity that should help determining which compression rate still has a negligable effect on the precision of the approximate solution. Finally, we compute the additional error arising from the compression $\mathrm{ERR} := \frac{\|u_j^\varphi - u_j\|_0}{\|u_j^\varphi\|_0}$, which is also displayed in most of the tables.

The main results for the geometries we have tested are presented in Tables 2, and 3. The bold quantities refer to the largest threshold th and the corresponding solution u_j for which $\mathrm{ERR}_j \approx \mathrm{ERR}_j^\varphi$ on the same level j. In this case, thresholding apparently has a *negligable* influence on the solution u_j. Furthermore for the bold quantities one has $\mathrm{MERR}_x \approx \mathrm{ERR}_{x^i}, i = 1, 2, 3$ and for $j > 4$ even $\mathrm{MERR}_x \leq \mathrm{ERR}_{x^i}, i = 1, 2, 3$. We observe that we can choose a larger threshold, and this results in a better compression for larger N, satisfying $\mathrm{MERR}_x \approx \inf_i \mathrm{ERR}_{x^i},\ i = 1, 2, 3$.

In Figs. 12, and 13 the number nze of nonzero elements of the compressed matrices A_j^ε, for which we observed acceptable precision (bold quantities in Tables 2, and 3) versus the number of knot points $N = \#\nabla^j$ are plotted and compared with the N^2 elements of the dense matrix. Figure 13 exhibits a nearly linear increase of nze for greater N. The compression reduces storage significantly to $1/\mathrm{cpr} \cdot N^2$. Likewise the CPU time for the matrix-vector multiplications during the iterative solution of the linear system decreases substantially (see Fig.14).

For the solution of the discrete and compressed scheme we again use the iterative method GMRES. For the estimation of the condition numbers $\kappa(A_j^\varphi)$ and $\kappa(A_j^\varepsilon)$ we apply a direct solver. This method is an expert driver of the well known LA-package [12]. The number of iteration steps, the estimated condition numbers and the CPU time t_{GM} for the GMRES solution on a DEC 3000 AXP 500 workstation are presented in Table 4. The termination bound for the iteration process is chosen to be about $\mathrm{ERR}_j^\varphi/100$. Note that since the operator has order zero in this case and since we have not employed 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 which is also confirmed by our experiments.

Our aim was to show the nearly linear asymptotic behavior of the necessary number of nonzero elements nze also in the case of a large number of degrees of freedom. So quadrupling the number N_j of unknowns leads to quadrupled nze only. In spite of a relatively strong compression one observes acceptable accuracy. For $N \approx 100,000$ such a compression rate is about 500 and we observed this rate to be more or less independent of the underlying geometry. Therefore in CPU–time solving the linear system by an iterative scheme the speed up factor was dramatic – solving a system corresponding to $100,000$ degrees of freedom, only takes 3 minutes on a sequential workstation and the whole matrix goes into a main memory of $\approx$ 200 MB. Our method damps the coefficients away from the singularity. Thus the double layer potential operator for the Laplacian can be well compressed. We expect the same to persist for the double layer potential for the Stokes system.

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.

5 Concluding remarks

We have outlined some theoretical foundations of multiscale methods and have carried out corresponding numerical experiments for two types of boundary integral equations. Galerkin methods being better understood we have concentrated here on collocation. The first case study concerning the Helmholtz equation is covered by the existing theory and shows that aside from the natural effects of large wave numbers also operators of order different from zero can be handled efficiently. The experiments are of significant help in clarifying the quantitative effects of the various ingredients of the scheme. The second example is a more complex three dimensional problem which is not fully covered by the analysis of the model problems. It could be shown though that nevertheless even for large problem sizes the basic compression and convergence properties predicted by the theory for the periodic case still persist to be valid.

Acknowledgement

We would like to thank A. Rathsfeld for helpful discussions and valuable suggestions.

References

[1] Arnold, D.N., Wendland, W.L. (1985): The convergence of spline collocation for strongly elliptic equations on curves. Numer. Math. **47**, 317–431.

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

[3] Bristeau, M.O., Glowinski, R., Périaux, J. (1993): Using exact controllability to solve the Helmholtz equation at high wave numbers. In: Mathematical and Numerical aspects of wave propagation, Ed. R. Kleinman et al., SIAM Philadelphia.

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

[5] Colton, D., Kress, R. (1983): Integral equation methods in scattering theory. Springer, Berlin - Heidelberg - New York.

[6] Dahmen, W., Prößdorf, S., Schneider, R. (1994): Wavelet approximation methods for pseudodifferential equations I: Stability and convergence. Math. Zeitschrift **215**, 583–620.

[7] Dahmen, W., Prößdorf, S., Schneider, R. (1993): Wavelet approximation methods for pseudodifferential equations II: Matrix compression and fast solution. Advances in Computational Mathematics, 2nd Issue, 259–335.

[8] Dahmen, W., Prößdorf, S., Schneider, R. (1994): Multiscale methods for pseudodifferential equations on smooth closed manifolds. Wavelets: Theory, Algorithms, and Applications (eds. C.K. Chui, L. Montefusco, and L. Puccio), Academic Press, 385–424.

[9] Dahmen, W., Prößdorf, S., Schneider, R. (1994): Multiscale methods for pseudodifferential equations. Recent Advances in Wavelet Analysis (eds. L.L. Schumaker and G. Webb), Academic Press, 191–235.

[10] Dahmen, W., Kleemann, B., Prößdorf, S., Schneider, R. (1994): A multiscale method for the double layer potential equation on a polyhedron. Advances in Computational Mathematics (eds. H.P. Dikshit and C.A. Micchelli), World Scientific, Singapore, New Jersey, London, Hong Kong, 15–57.

[11] Daubechies, I. (1992): Ten Lectures on Wavelets. SIAM, Philadelphia, PA.

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

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

[14] Kress, R., Sloan, I.H. (1993): On the numerical solution of a logarithmic integral equation of the first kind for the Helmholtz equation. Numer. Math. **66**, 199–214.

[15] Maz'ya, V.G. (1991): Boundary integral equations. in: V.G. Maz'ya and S.M. Nikol'skiĭ (eds.) Encyclopaedia of Math. Sciences, vol. 27, Analysis IV, Springer, Berlin - Heidelberg - New York.

[16] Meyer, Y. (1992): Wavelets and Operators. Cambridge University Press.

[17] Petersdorff, T. von, Schwab, C. (1994): Wavelet approximation for first kind boundary integral equations on polygons. Techn. Note BN–1157, Institute for Physical Science and Technology, University of Maryland at College Park.

[18] Rathsfeld, A. (1995): A wavelet algorithm for the solution of the double layer potential equation over polygonal boundaries. J. Integral Equations Appl. **7**, No. 1, 47–98.

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

[20] Saranen, J., Vainikko, G. (1994): Fast solvers of integral and pseudodifferential equations on closed curves. To appear in: Mathematics of Computation.

[21] Schneider, R. (1995): Multiskalen- und Wavelet-Matrixkompression: Analysisbasierte Methoden zur effizienten Lösung großer vollbesetzter Gleichungssysteme. Habilitationsschrift, TH Darmstadt.

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

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

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

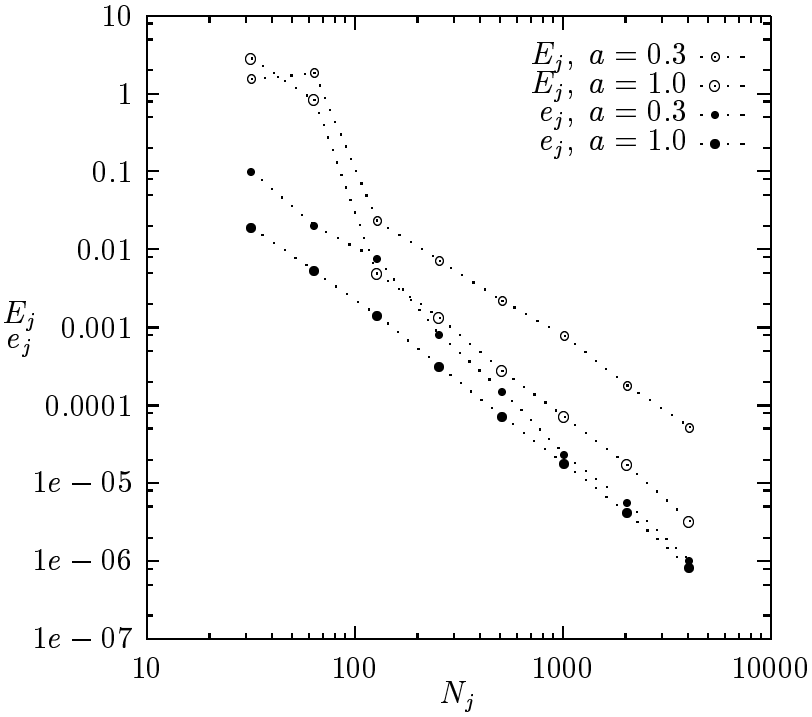


Figure 5: Errors E_j, e_j in logarithmic scale for an elliptical cross section and fixed wave number $\bar{k} = 10$ for two different constants a for the `a-priori` compression strategy.

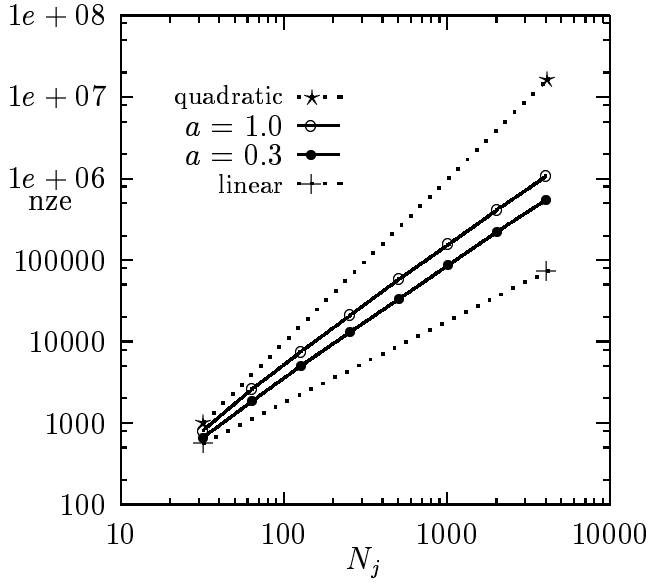


Figure 6: Number nze of nonzero elements of the compressed matrix for two constants a for different wave numbers $\bar{k}$.

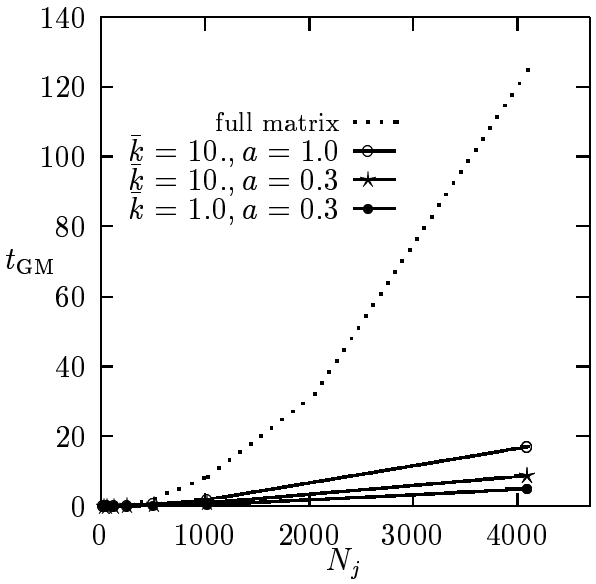


Figure 7: CPU-time t_{GM} in seconds on a DEC 3000 AXP 500 α-processor workstation for the iterative solution of the arising linear system by GMRES for different wave numbers $\bar{k}$ and constants a.

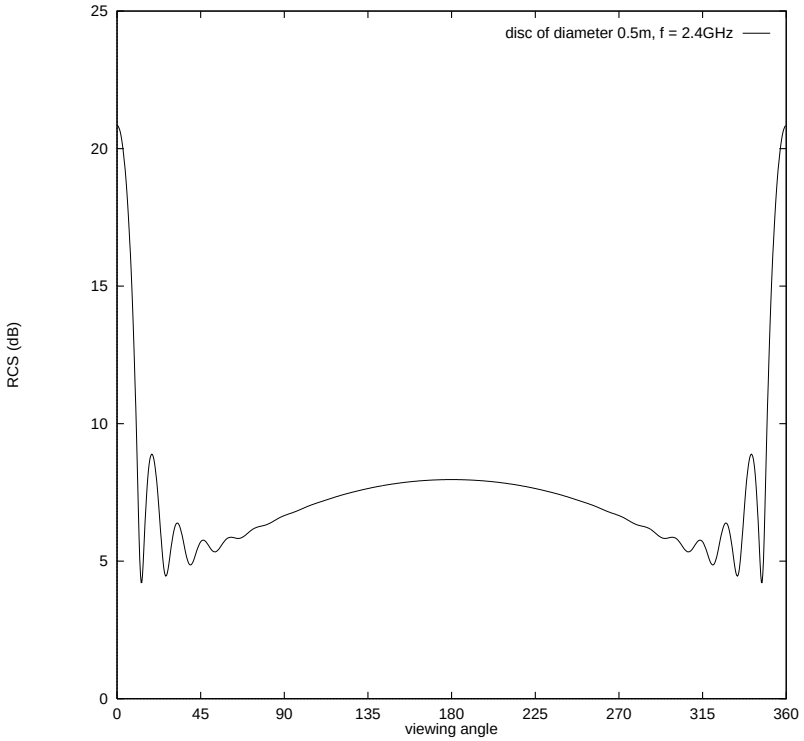


Figure 8: Radar cross section (RCS) σ versus viewing angle for the scattering of a frequency of 2.4 GHz implying a wavelength of $\lambda = 0.125$ m at a disc with radius $r = 0.25$ m. The quantity $\bar{k} \cdot 2r$ determining the Hankel function values is equal to 25.13.

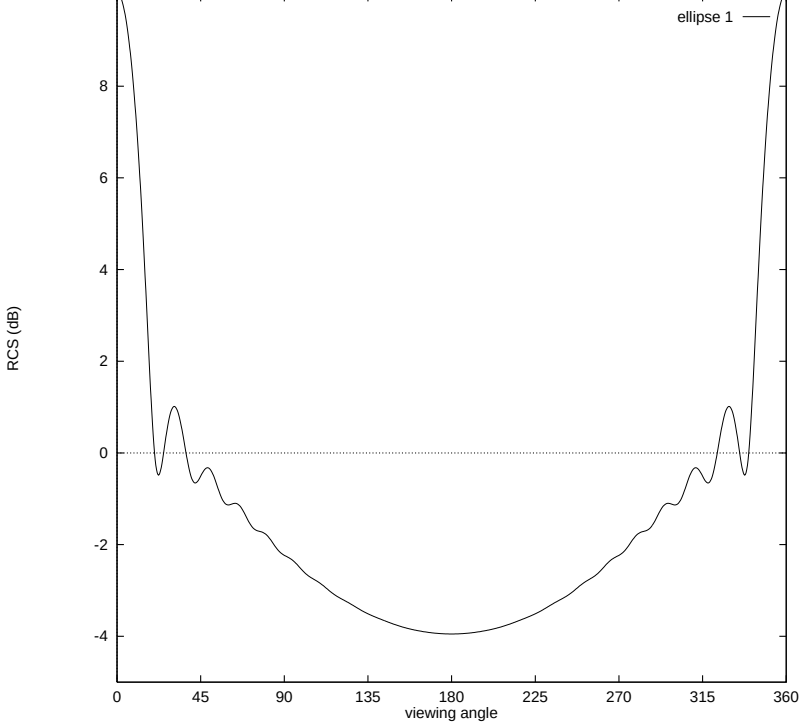


Figure 9: Radar cross section (RCS) σ versus viewing angle for the scattering of a frequency of 10 GHz implying a wavelength of $\lambda = 30$ mm at an ellipse with $a_x = 60$ mm, $b_y = 30$ mm. The quantity $\bar{k} \cdot 2a$ determining the Hankel function values is equal to 25.13.

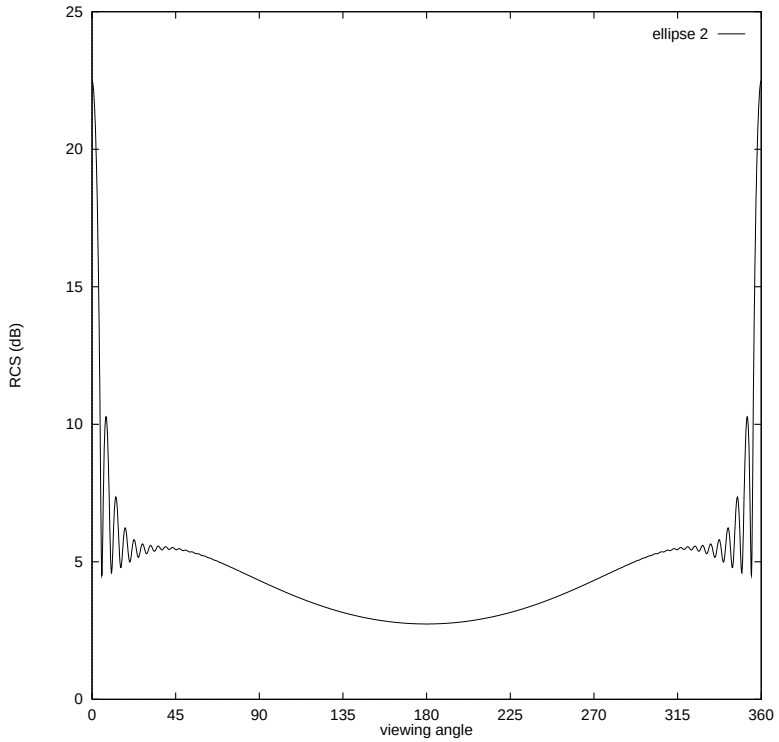


Figure 10: Radar cross section (RCS) σ versus viewing angle for the scattering of a frequency of 10 GHz implying a wavelength of $\lambda = 30$ mm at an ellipse with $a_x = 300$ mm, $b_y = 150$ mm. The quantity $\bar{k} \cdot 2a$ determining the Hankel function values is equal to 125.66.

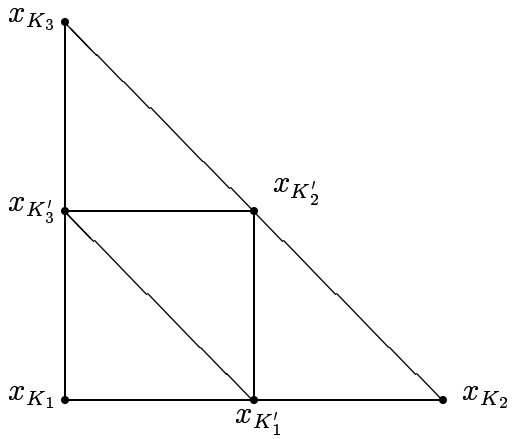


Figure 11: Triangle $[x_{K_1}, x_{K_2}, x_{K_3}]$ of a face of a polyhedral boundary and its refinement. $\{K_1, K_2, K_3\} \subset \nabla^j \Rightarrow \{K_1, K_2, K_3, K_1', K_2', K_3'\} \subset \nabla^{j+1}, \quad \{K_1', K_2', K_3'\} \subset \Delta^j$.

Cube						
j	N_j	th	nze	cpr	ERR	MERR_x
1	26	$3 \cdot 10^{-3}$	568	1.2	$2.0 \cdot 10^{-4}$	$2.9 \cdot 10^{-2}$
		$1 \cdot 10^{-2}$	**466**	**1.4**	$4.3 \cdot 10^{-3}$	$2.9 \cdot 10^{-2}$
		$3 \cdot 10^{-2}$	274	2.5	$8.5 \cdot 10^{-2}$	$3.0 \cdot 10^{-2}$
2	98	$1 \cdot 10^{-3}$	5176	1.8	$2.3 \cdot 10^{-3}$	$4.6 \cdot 10^{-3}$
		$3 \cdot 10^{-3}$	**3346**	**2.9**	$5.5 \cdot 10^{-3}$	$4.4 \cdot 10^{-3}$
		$1 \cdot 10^{-2}$	1990	4.8	$2.5 \cdot 10^{-3}$	$2.9 \cdot 10^{-3}$
3	386	$3 \cdot 10^{-4}$	36,484	4.1	$1.0 \cdot 10^{-3}$	$6.7 \cdot 10^{-4}$
		$1 \cdot 10^{-3}$	**21,124**	**7.0**	$4.4 \cdot 10^{-3}$	$5.2 \cdot 10^{-4}$
		$3 \cdot 10^{-3}$	12,646	11.8	$2.3 \cdot 10^{-2}$	$1.9 \cdot 10^{-3}$
4	1,538	$1 \cdot 10^{-4}$	401,272	5.9	$4.7 \cdot 10^{-4}$	$1.3 \cdot 10^{-4}$
		$3 \cdot 10^{-4}$	**235,588**	**10.3**	$1.3 \cdot 10^{-3}$	$1.2 \cdot 10^{-4}$
		$1 \cdot 10^{-3}$	125,524	18.8	$3.8 \cdot 10^{-3}$	$1.8 \cdot 10^{-4}$
5	6,146	$3 \cdot 10^{-5}$	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	$2.1 \cdot 10^{-3}$	$2.4 \cdot 10^{-5}$
6[a]	24,578	$3 \cdot 10^{-6}$	6,755,512	89.4	-	$5.0 \cdot 10^{-6}$
		$1 \cdot 10^{-5}$	**3,953,114**	**152.8**	-	$5.5 \cdot 10^{-6}$
		$3 \cdot 10^{-5}$	2,478,484	243.7	-	$2.1 \cdot 10^{-5}$
7[b]	98,306	$3 \cdot 10^{-6}$	**17,857,936**	**541.0**	-	$2.0 \cdot 10^{-6}$
		$3 \cdot 10^{-5}$	7,097,044	1361.0	-	$2.2 \cdot 10^{-5}$

[a] Calculations without singularity subtraction. Coarsest level is set to $l = 2$.
[b] Calculations without singularity subtraction. Coarsest level is set to $l = 3$.

Table 2: Number nze of nonzero elements, compression rate cpr, errors ERR and MERR_x for the multiscale algorithm for the Laplace equation on the cube for several thresholds th and levels j.

Pyramid						
j	N_j	th	nze	cpr	ERR	MERR_x
1	14	$3 \cdot 10^{-3}$	188	1.0	$1.1 \cdot 10^{-4}$	$2.2 \cdot 10^{-2}$
		$1 \cdot 10^{-2}$	**172**	**1.1**	$1.8 \cdot 10^{-4}$	$2.2 \cdot 10^{-2}$
		$3 \cdot 10^{-2}$	122	1.6	$5.6 \cdot 10^{-2}$	$3.7 \cdot 10^{-2}$
2	50	$1 \cdot 10^{-3}$	1948	1.3	$7.5 \cdot 10^{-4}$	$5.2 \cdot 10^{-3}$
		$3 \cdot 10^{-3}$	**1494**	**1.7**	$1.5 \cdot 10^{-3}$	$5.3 \cdot 10^{-3}$
		$1 \cdot 10^{-2}$	1016	2.5	$1.5 \cdot 10^{-2}$	$6.3 \cdot 10^{-3}$
3	194	$3 \cdot 10^{-4}$	22,546	1.7	$2.2 \cdot 10^{-4}$	$2.2 \cdot 10^{-4}$
		$1 \cdot 10^{-3}$	**15,108**	**2.5**	$1.6 \cdot 10^{-3}$	$1.8 \cdot 10^{-4}$
		$3 \cdot 10^{-3}$	9,736	3.9	$5.8 \cdot 10^{-3}$	$1.5 \cdot 10^{-4}$
4	770	$1 \cdot 10^{-4}$	159,022	3.7	$3.1 \cdot 10^{-4}$	$1.2 \cdot 10^{-4}$
		$3 \cdot 10^{-4}$	**96,244**	**6.2**	$1.1 \cdot 10^{-3}$	$1.2 \cdot 10^{-4}$
		$1 \cdot 10^{-3}$	56,668	10.5	$3.0 \cdot 10^{-3}$	$2.3 \cdot 10^{-4}$
5	3,074	$3 \cdot 10^{-5}$	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	$1.4 \cdot 10^{-3}$	$2.5 \cdot 10^{-5}$
6[a]	12,290	$3 \cdot 10^{-6}$	4,071,856	37.1	-	$9.0 \cdot 10^{-6}$
		$1 \cdot 10^{-5}$	**2,401,236**	**62.9**	-	$7.3 \cdot 10^{-6}$
		$3 \cdot 10^{-5}$	1,486,256	101.6	-	$9.2 \cdot 10^{-6}$
7[b]	49,154	$3 \cdot 10^{-6}$	**10,776,400**	**224.4**	-	$1.6 \cdot 10^{-6}$

[a] Calculations without singularity subtraction. Coarsest level is set to $l = 2$.
[b] Calculations without singularity subtraction. Coarsest level is set to $l = 3$.

Table 3: Number nze of nonzero elements, compression rate cpr, errors ERR and MERR_x for the multiscale algorithm for the Laplace equation on the pyramid for several thresholds th and levels j.

j	N_j	$\kappa(A_j^\varphi)$	$\mathrm{it}(A_j^\varphi)$	$t(A_j^\varphi)$	cpr	$\kappa(A_j^\varepsilon)$	$\mathrm{it}(A_j^\varepsilon)$	$t(A_j^\varepsilon)$
Cube:								
3	386	2.4	6	0.2	7.0	38.9	14	**0.1**
4	1,538	2.8	8	3.9	10.3	133.0	16	**0.9**
5	6,146	3.2	11	52.0	33.8	215.7	23	**6.0**
6	24,578	-	-	-	152.8[a]	-	29	**33.8**
7	98,306	-	-	-	541.0[b]	-	32	**178.0**
Pyramid:								
3	194	2.3	7	0.06	3.9	31.1	13	**0.03**
4	770	5.3	8	0.9	6.2	63.4	15	**0.3**
5	3,074	8.6	10	17.3	14.3	118.2	19	**2.6**
6	12,290	-	-	-	62.9[a]	-	28	**19.9**
7	49,154	-	-	-	224.4[b]	-	31	**74.2**

[a] Calculations without singularity subtraction. Coarsest level is set to $l = 2$.
[b] Calculations without singularity subtraction. Coarsest level is set to $l = 3$.

Table 4: Laplace equation: Number of iterations it for the solution of the linear systems $A_j^\varphi u_j^\varphi = f_j$ and $A_j^\varepsilon u_j = (T_j)^* f_j$, respectively, with GMRES; CPU-times t in seconds on a DEC 3000 AXP 500 α–processor workstation. Condition numbers κ of the corresponding matrices.

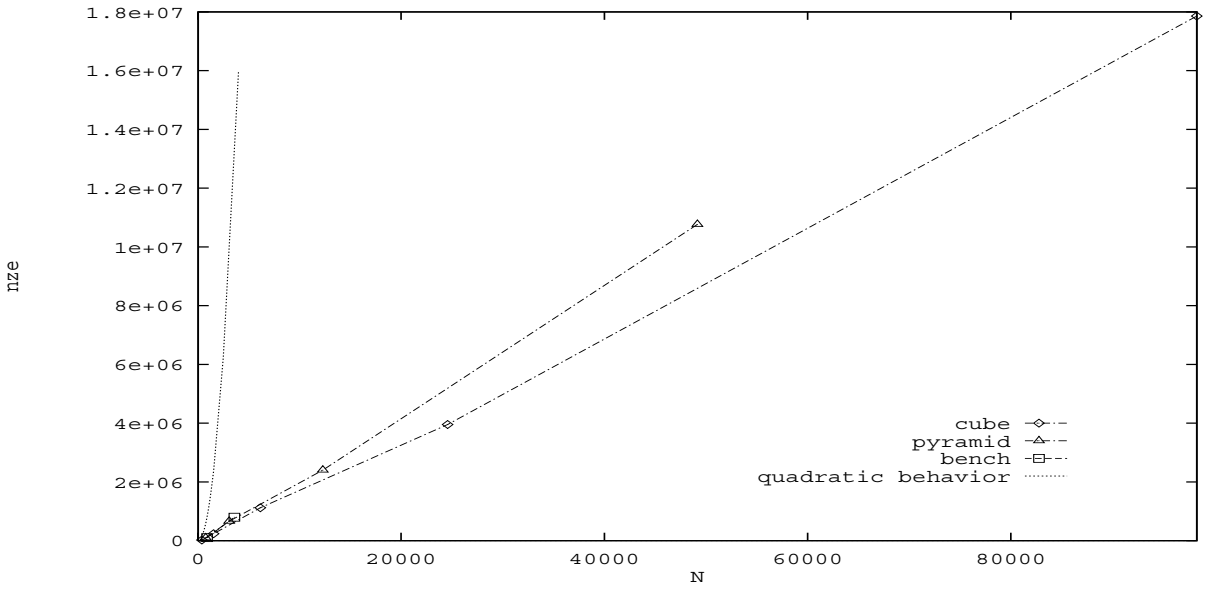


Figure 12: Number nze of nonzero elements of the transformed matrix A_j^ϵ for the Laplace equation after thresholding in dependence of N_j. Dotted line for comparison: quadratic behavior of the number of elements of the original stiffness matrix.

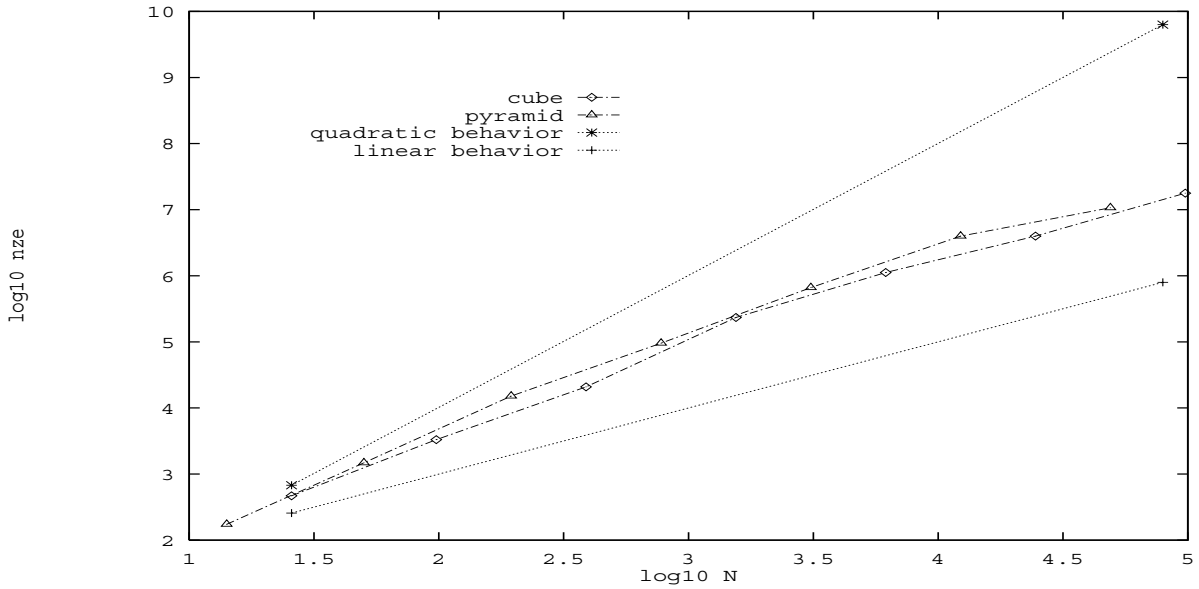


Figure 13: Number nze of nonzero elements of the transformed matrix A_j^ϵ for the Laplace equation after thresholding in dependence of N_j in logarithmic scale. Dotted lines for comparison: quadratic behavior and linear behavior.

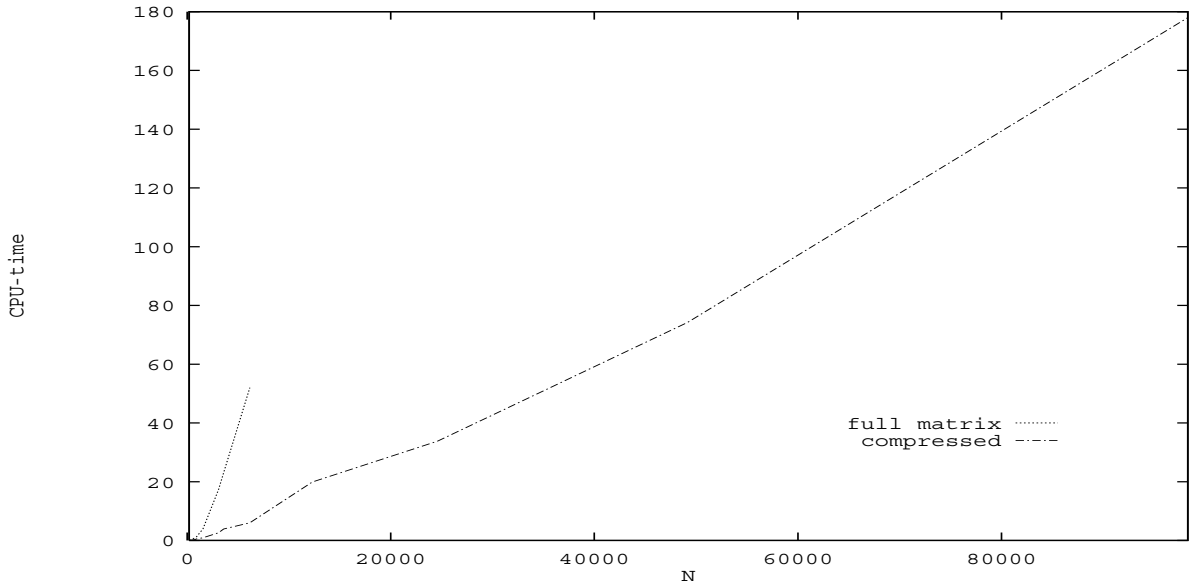


Figure 14: CPU-time in seconds on a DEC 3000 AXP 500 α–processor workstation for the iterative solution of the linear systems for the Laplace equation with uncompressed (dotted line) and compressed matrix, respectively, with GMRES.