ETH zürich

ETH Library

Advanced boundary element algorithms

Report

Author(s):
Lage, Christian; Schwab, Christoph

Publication date:
1999-06

Permanent link:
https://doi.org/10.3929/ethz-a-004288533

Rights / license:
In Copyright - Non-Commercial Use Permitted

Originally published in:
SAM Research Report 1999-11

This page was generated automatically upon download from the ETH Zurich Research Collection.
For more information, please consult the Terms of use.

Advanced boundary element algorithms*

C. Lage† and C. Schwab

Research Report No. 99-11
June 1999

Seminar für Angewandte Mathematik
Eidgenössische Technische Hochschule
CH-8092 Zürich
Switzerland

*Invited paper at the MAFELAP X Conference, Uxbridge, June 22-25, 1999
†Coyote Systems, 2740 Van Ness Ave #210, San Francisco, CA 94109, USA

Advanced boundary element algorithms**

C. Lage† and C. Schwab

Seminar für Angewandte Mathematik
Eidgenössische Technische Hochschule
CH-8092 Zürich
Switzerland

Research Report No. 99-11 June 1999

Abstract

We review recent algorithmic developments in the boundary element method (BEM) for large scale engineering calculations. Two classes of algorithms, the clustering and the wavelet-based schemes are compared. Both have $O(N(\log N)^a)$ complexity with some small $a \geq 0$ and allow in-core simulations with up to $N = O(10^6)$ DOF on the boundary on serial workstations. Clustering appears more robust for complex surfaces.

**This work was supported in part under the TMR network "Multiscale Methods in Numerical Analysis" of the EC by the Swiss National Fund

†Coyote Systems, 2740 Van Ness Ave #210, San Francisco, CA 94109, USA

1 Boundary Element Method

Let $\Omega \subset \mathbb{R}^3$ be a bounded polyhedron with N_0 straight faces Γ_k and boundary $\Gamma = \partial\Omega$. To present our algorithms, we consider the model problem

$$\Delta U = 0 \ \text{ in } \ \Omega, \quad U = f \ \text{ on } \ \Gamma \ . \tag{1.1}$$

Using the fundamental solution $e(x, y)$, (1.1) can be reduced in various ways to a boundary integral equation (BIE) for an unknown density u on Γ [3, 19, 21, 28, 43]:

$$Au = \lambda u + Ku = g \ \text{ on } \ \Gamma \ . \tag{1.2}$$

Here g is related to f, $\lambda = 0$ in formulations leading to equations of the first kind, $\lambda \neq 0$ in indirect methods, based on layer potentials. The integral operator K is given by

$$(Ku)(x) = \int_{y\in\Gamma} k(x, y)\, u(y) ds_y, \quad x \in \Gamma \ , \tag{1.3}$$

where the kernel $k(x, y)$ is a normal/ tangential derivative of $e(x, y)$ and satisfies for some $s \geq 0$

$$|D_x^\alpha\, D_y^\beta\ k(x, y)| \leq \frac{C(\alpha, \beta)}{|x - y|^{s+|\alpha|+|\beta|}}, \quad \forall \alpha,\, \beta \in \mathbb{N}_0^3. \tag{1.4}$$

Typically $k(x, y)$ is analytic if $x \neq y$ and singular at $x = y$. In most applications, $s = 1, 2$ or 3 and the "integral" in (1.3) has to be understood as finite part integral if $s \geq 2$ (see [41], [17] for more on such integrals). Many other, more general problems admit a BIE reformulation (1.2); for a reference, see e.g. [3, 21, 43] (in the case of vector-valued U in (1.1), the BIE (1.2) will consist of a matrix of integral operators with entries satisfying (1.3), (1.4)).

Boundary Element Methods (BEM) are discretizations which reduce (1.2) to the linear system

$$\boldsymbol{Au} = \lambda \boldsymbol{M} + \boldsymbol{Ku} = \boldsymbol{g} \tag{1.5}$$

for the unknown solution vector $\boldsymbol{u}$. Here $\boldsymbol{M} \in \mathbb{R}^{N\times N}$ is a "mass" matrix, and $\boldsymbol{K} \in \mathbb{R}^{N\times N}$ is the stiffness-matrix corresponding to K in (1.3). In BEM, it is obtained by

$$\boldsymbol{K} = \{K_{\lambda\lambda'}\}_{\lambda,\lambda'\in\Lambda},\ K_{\lambda\lambda'} = (\mathcal{X}_\lambda(\mathcal{Y}_{\lambda'}\, k))_{\lambda,\lambda'\in\Lambda} \tag{1.6}$$

where Λ is some index set of cardinality $N = |\Lambda|$ and $\mathcal{X}_\lambda$, $\mathcal{Y}_\lambda$ are linear functionals on a suitable function space V on Γ in which (1.2) is well-posed and uniquely solvable. For example, let $\{x_\lambda\}_{\lambda\in\Lambda}$ be a set of collocation points on Γ and let $V^N \subset V$ with $V^N = \text{span}\{\varphi_\lambda : \lambda \in \Lambda\}$ be a finite element space on Γ. One may take

$$\mathcal{X}_\lambda\varphi := \varphi(x_\lambda), \quad \mathcal{Y}_{\lambda'}\varphi; = \varphi(x_{\lambda'}) \quad \text{Nyström Method,} \tag{1.7a}$$

$$\mathcal{X}_\lambda\varphi := \varphi(x_\lambda), \quad \mathcal{Y}_{\lambda'}\varphi := \langle\varphi, \varphi_{\lambda'}\rangle \quad \text{Collocation Method,} \tag{1.7b}$$

$$\mathcal{X}_\lambda\varphi := \langle\varphi, \varphi_\lambda\rangle, \quad \mathcal{Y}_{\lambda'}\varphi := \langle\varphi, \varphi_{\lambda'}\rangle \quad \text{Galerkin Method} \tag{1.7c}$$

where $\langle \cdot, \cdot \rangle$ is a suitable duality pairing on Γ. (1.7a) is not possible in general if $k(x,y)$ is singular for $x = y$, (1.7b), (1.7c) require special numerical integrations which are by now completely understood ([17, 38] for (1.7b) and [1, 30, 27, 37] for (1.7c), see also [15] for the farfield integrals).

Most of the classical BEM algorithms in engineering require manipulation of the full matrix $\boldsymbol{A}$ leading to operations of $O(N^2)$ complexity. The performance of such BEM algorithms is hence inferior to e.g. FE-multigrid methods applied directly to the PDE (1.1). Nevertheless, in the last years a large body of literature on the *formulation* of BIEs has appeared, in particular the derivation of well-posed variational first-kind BIEs ([21, 43, 28]), inclusive mathematical convergence analysis of discretizations.

We conclude that *for the viability of the BEM in industrial problems it is mandatory to reduce the complexity of* $\boldsymbol{u} \longmapsto \boldsymbol{A}\boldsymbol{u}$ from $O(N^2)$ to (essentially) $O(N)$. This reduction should a) apply for general surfaces Γ of possibly complicated shape, b) be noticeable for N in the practical range of unknowns and c) be possible for all kernels $k(x,y)$ arising in applications.

We present in the following two classes of BE-Algorithms, referred to as *clustering-*, resp. *wavelet algorithms*, to achieve these goals. Both replace $\boldsymbol{A}$ in (1.5) by approximations $\widetilde{\boldsymbol{A}}$ which can be manipulated in $O(N)$ complexity, while the error $\|\boldsymbol{A} - \widetilde{\boldsymbol{A}}\|$ is small.

Unless stated otherwise, we assume in the following that $\{V^N\}$ is a finite element space of piecewise polynomials on a quasiuniform mesh $\mathcal{M}$ on Γ of shape regular triangles of diameter $h = O(N^{-\frac{1}{2}})$.

2 Cluster Methods

Cluster BEM contains the panel-clustering [18] as well as the multipole techniques [16] as special cases. The main idea is the approximation of $\boldsymbol{K}$ by

$$\widetilde{\boldsymbol{K}} = \boldsymbol{N} + \sum_{\sigma,\tau} \boldsymbol{X}_\sigma^T \, \boldsymbol{F}_{\sigma\tau} \, \boldsymbol{Y}_\tau \tag{2.1}$$

with a sparse *near-field matrix* $\boldsymbol{N} \in \mathbb{R}^{N\times N}$ and *far-field matrices* $\boldsymbol{X}_\sigma \in \mathbb{R}^{M\times N}$, $\boldsymbol{Y}_\tau \in \mathbb{R}^{M\times N}$ and $\boldsymbol{F}_{\sigma\tau} \in \mathbb{R}^{M\times M}$. The matrices $\boldsymbol{X}_\sigma$, $\boldsymbol{F}_{\sigma\tau}$, $\boldsymbol{Y}_\tau$ are never formed explicitly and $\boldsymbol{u} \longmapsto \widetilde{\boldsymbol{K}}\boldsymbol{u}$ is realized as

$$\widetilde{\boldsymbol{K}}\boldsymbol{u} = \boldsymbol{N}\boldsymbol{u} + \sum_{\sigma,\tau} \boldsymbol{X}_\sigma \left(\boldsymbol{F}_{\sigma\tau}(\boldsymbol{Y}_\tau \, \boldsymbol{u})\right) , \tag{2.2}$$

without direct access to entries of $\boldsymbol{K}$. The amount of storage used to keep the necessary information of the far field is of order $O(M'N)$ with $M' \ll N$ which reduces the $O(N^2)$ complexity. The basic assumption on the kernel $k(x,y)$ underlying cluster methods is its local approximability by a degenerate kernel. Let $D \subset \mathbb{R}^3$ be any domain containing Γ.

Assumption 2.1 *Let $0 \le \eta < 1, k\colon D \times D \to \mathbb{C}$ a kernel function and $\mathcal{I}$ an index set. Then for all $x_0, y_0 \in D$, $x_0 \neq y_0$, and $m \in \mathbb{N}_0$ there exists an approximation $\tilde{k}$ of the form*

$$k(x,y) \sim \tilde{k}(x,y;x_0,y_0,m) := \sum_{(\mu,\nu)\in\mathcal{I}_m} \kappa^m_{(\mu,\nu)}(x_0,y_0)\, X_\mu(x;x_0)\, Y_\nu(y;y_0), \; \mathcal{I}_m \subset \mathcal{I}\times\mathcal{I} \tag{2.3}$$

such that for all $x, y \in D$ satisfying

$$|y - y_0| + |x - x_0| \leq \eta |y_0 - x_0| \tag{2.4}$$

the error is bounded by

$$|k(x,y) - \tilde{k}(x,y;x_0,y_0,m)| \leq C\, e^{-C(\eta)m} |y-x|^{-s} \tag{2.5}$$

with $C(\eta) > 0$ a decreasing function and C a constant both independent of m.

Examples of kernel approximations $\tilde{k}$ follow:

2.1 Taylor expansion ([18, 22, 35])

Let the kernel function k only depend on the difference of its arguments:

$$k(x,y) = k(y-x)\ . \tag{2.6}$$

We expand $k(y-x)$ formally into a Taylor series centered at $y_0 - x_0$ with $x_0, y_0 \in \mathbb{R}^d$:

$$\begin{aligned} k(y-x) &= \sum_{\mu \in \mathbb{N}_0^d} \frac{1}{\mu!} (D^\mu k)(y_0 - x_0)(y - x - y_0 + x_0)^\mu \\ &= \sum_{(\nu,\mu) \in \mathbb{N}_0^d \times \mathbb{N}_0^d} (D^{\mu+\nu} k)(y_0 - x_0) \frac{(x_0 - x)^\mu}{\mu!} \frac{(y - y_0)^\nu}{\nu!} \end{aligned}$$

This motivates an approximation (2.3) described by

$$\begin{gathered} \mathcal{I} := \mathbb{N}_0^d,\ \ \mathcal{I}_m := \{(\mu,\nu) \in \mathcal{I} \times \mathcal{I} :\ |\mu + \nu| < m\}\,, \\ \kappa^m_{(\mu,\nu)}(x_0,y_0) := (D^{\mu+\nu} k)(y_0 - x_0)\,, \\ X_\mu(x,x_0) := (x_0 - x)^\mu/\mu!\,,\ \ Y_\nu(y,y_0) := (y - y_0)^\nu/\nu!\ . \end{gathered} \tag{2.7}$$

In [18, 19, 20, 22] the error bound (2.5) has been verified for several kernel functions. For example, (2.5) holds for the fundamental solution of Laplace's equation in $\mathbb{R}^3$,

$$k(x,y) = \frac{1}{4\pi}\, |y-x|^{-1},\ \ x, y \in \mathbb{R}^3,\ x \neq y \tag{2.8}$$

with $C = 1$ and $C(\eta) = \log \frac{1}{\eta}$.

2.2 Multipole expansion [16, 34]

The multipole expansions are a special case of (2.3). They are kernel specific in that the expansion coefficients must be evaluated analytically (but once and for all) for the kernel of interest. For the ubiquitous Coulomb-potential $k(x,y) = |x-y|^{-1}$, we have (2.3) with

$$\mathcal{J}_m := \{\mu \in \mathbb{N}_0 \times \mathbb{Z} : |\mu_2| \leq \mu_1,\ \mu_1 < m\},\ \mathcal{I}_m := \{(\mu,\nu) \in (\mathcal{J}_m)^2 : \mu_1 + \nu_1 < m\} \tag{2.9}$$

$$\kappa^m_{(\mu,\nu)}(x_0,y_0) := \kappa_{\mu+\nu}(x_0,y_0) := \frac{1}{C^{\mu_2+\nu_2}_{\mu_1+\nu_1}\,|y_0-x_0|^{\mu_1+\nu_1+1}}\; Y^{\mu_2+\nu_2}_{\mu_1+\nu_1}\left(\frac{y_0-x_0}{|y_0-x_0|}\right) \tag{2.10}$$

$$X_\mu(x;x_0) := C^{\mu_2}_{\mu_1}\,|x-x_0|^{\mu_1}\; Y^{-\mu_2}_{\mu_1}\left(\frac{x-x_0}{|x-x_0|}\right), Y_\nu(y;y_0) := X_\nu(-y;-y_0) \tag{2.11}$$

with

$$C^p_\ell := \frac{i^{|p|}}{\sqrt{(\ell-p)!(\ell+p)!}}\,,\quad Y^p_\ell(x) := P^{|p|}_\ell(\cos\theta)\,e^{ip\phi} \tag{2.12}$$

for $x = (\cos\phi\sin\theta,\ \sin\theta,\ \cos\theta)^T \in \boldsymbol{S}_2$. The functions X_μ and Y_ν are solid spherical harmonics of positive degree whereas the expansion coefficients $\kappa^m_{(\mu,\nu)}$ are homogeneous harmonic polynomials of negative degree. *Note that the multipole expansion is nothing else but an efficient representation of the Taylor expansion* of $|y-x|^{-1}$. While for arbitrary kernel functions k, the index set of a truncated Taylor expansion contains $O(m^3)$ indices, only $O(m^2)$ coefficients must be stored to evaluate the Taylor expansion of $|y-x|^{-1}$ using the multipole ansatz according to (2.9) - (2.11). For the Helmholtz Equation, X_μ, Y_ν should depend on the wavenumber [14, 4], for elasticity [20] give a multipole expansion and [13] show how to obtain FMM for isotropic elasticity from FMM for the Laplacean.

2.3 Clustering Error

To employ a kernel approximation (2.3) for the construction of $\widetilde{\boldsymbol{K}}$ in (2.1), we partition the set $D\times D$. Let $\mathcal{P}(D)$ denote the set of all subsets of D, let $\check{r}_A$, $\check{c}_A$ be Chebyšev radius and center, respectively, and 1_A the characteristic function of a set $A\subset \mathbb{R}^3$.

Definition 2.1 *Subsets σ,τ of D are called clusters. A finite set $\mathcal{C}\subset\mathcal{P}(D)\times\mathcal{P}(D)$ of related clusters is called a clustering of $D\times D$, iff*

$$\mathcal{X}_\lambda(\mathcal{Y}_{\lambda'}(k)) = \sum_{(\sigma,\tau)\in\mathcal{C}} \mathcal{X}_\lambda(\mathcal{Y}_{\lambda'}(k1_{\sigma\times\tau})).$$

For each $c\in\mathcal{C}$ let the relative size η_c be defined by

$$\eta_{(\sigma,\tau)} := \begin{cases} \dfrac{\check{r}_\sigma+\check{r}_\tau}{\|\check{c}_\sigma-\check{c}_\tau\|}, & \text{if } \check{c}_\sigma\neq\check{c}_\tau,\\ \infty & \text{otherwise.}\end{cases}$$

A far field $\mathcal{F} := \mathcal{F}_\mathcal{C}$ of $\mathcal{C}$ is a subset of $\{c\in\mathcal{C} : \eta_c<1\}$ and the corresponding near field $\mathcal{N} := \mathcal{N}_\mathcal{C}$ the complement $\mathcal{C}\backslash\mathcal{F}_\mathcal{C}$. Far and near field imply a bipartition of $D\times D$:

$$D_\mathcal{F} := \bigcup_{(\sigma,\tau)\in\mathcal{F}} \sigma\times\tau,\quad D_\mathcal{N} := (D\times D)\backslash D_\mathcal{F}\ . \tag{2.13}$$

The grain of a far field is defined by $\eta_\mathcal{F} := \max_{c\in\mathcal{F}}\ \eta_c$.

In order to define precisely a cluster approximation of $\boldsymbol{K}$, we introduce the following notation for the restriction of subsets $\mathcal{T}$ of product sets $\mathcal{S} \times \mathcal{S}'$ to one of its components:

$$(\mathcal{T})_1 := \{s \in \mathcal{S} : \exists s' \in \mathcal{S}'(s, s') \in \mathcal{T}\}, \quad (\mathcal{T})_2 := \{s \in \mathcal{S}' : \exists s \in \mathcal{S}(s, s') \in \mathcal{T}\} . \tag{2.14}$$

Definition 2.2 *Let $m \in \mathbb{N}_0$ and $\mathcal{F}$ a far field of $D \times D$. Let, in addition, $\tilde{k}$ be an approximation according to Assumption 2.1. Then, a cluster approximation $\widetilde{\boldsymbol{K}} := \widetilde{\boldsymbol{K}}(\mathcal{F}, m, \tilde{k})$ of $\boldsymbol{K}$ is defined by*

$$\widetilde{\boldsymbol{K}} := \boldsymbol{N} + \sum_{(\sigma,\tau)\in\mathcal{F}} \boldsymbol{X}_\sigma^T \boldsymbol{F}_{\sigma\tau} \boldsymbol{Y}_\tau \tag{2.15}$$

with $\boldsymbol{N} \in \mathbb{R}^{N\times N}$, $\boldsymbol{X}_\sigma \in \mathbb{R}^{(\mathcal{I}_m)_1 \times N}$, $\boldsymbol{Y}_\tau \in \mathbb{R}^{(\mathcal{I}_m)_2 \times N}$ and $\boldsymbol{F}_{\sigma\tau} \in \mathbb{R}^{(\mathcal{I}_m)_1 \times (\mathcal{I}_m)_2}$ given by *

$$(\boldsymbol{N})_{\lambda\lambda'} := \mathcal{X}_\lambda\left(\mathcal{Y}_{\lambda'}(k 1_{\mathcal{D}_\mathcal{N}})\right), \ (\boldsymbol{X}_\sigma)_{\mu,\lambda} := \mathcal{X}_\lambda(X_\mu(\cdot, \check{c}_\sigma) 1_\sigma), \ \ (\boldsymbol{Y}_\tau)_{\nu,\lambda'} := \mathcal{Y}_{\lambda'}(Y_\nu(\cdot, \check{c}_\tau) 1_\tau) ,$$

$$(\boldsymbol{F}_{\sigma\tau})_{\mu,\nu} := \begin{cases} \kappa^m_{(\mu,\nu)}(\check{c}_\sigma, \check{c}_\tau) & \text{if } (\mu,\nu) \in \mathcal{I}_m, \\ 0 & \text{otherwise} . \end{cases}$$

Then we have the following error bounds [26, 19, 18].

Theorem 2.1 *Suppose the linear functionals $\mathcal{X}_\lambda$ and $\mathcal{Y}_\lambda$ satisfy $\mathcal{X}_\lambda \phi \le \mathcal{X}_\lambda \phi'$ and $\mathcal{Y}_\lambda \phi \le \mathcal{Y}_\lambda \phi'$, respectively, for all $\lambda \in \Lambda$ and $0 \le \phi \le \phi'$ and consider a cluster approximation $\widetilde{\boldsymbol{K}}(\mathcal{F}, m, \tilde{k})$ of $\boldsymbol{K}$ according to Definition 2.2. Let*

$$\rho := C(\eta_\mathcal{F})\, m > 0 \tag{2.16}$$

with $\eta_\mathcal{F}$ the grain of $\mathcal{F}$ and $C(\cdot)$ as in (2.5). Then, the following error bounds are satisfied:

$$\begin{gathered} \|\widetilde{\boldsymbol{K}} - \boldsymbol{K}\|_\infty \le C\, e^{-\rho}\, M_\mathcal{X}, \quad \|\widetilde{\boldsymbol{K}} - \boldsymbol{K}\|_1 \le C\, e^{-\rho}\, M_\mathcal{Y} , \\ \|\widetilde{\boldsymbol{K}} - \boldsymbol{K}\|_2 \le C\, e^{-\rho} \sqrt{M_\mathcal{X} M_\mathcal{Y}} \end{gathered} \tag{2.17}$$

where C is independent of m and Γ, and

$$M_\mathcal{X} := \max_{\lambda\in\Lambda} \sum_{\lambda'\in\Lambda} \mathcal{X}_\lambda(\mathcal{Y}_{\lambda'}(\tfrac{1_{D_\mathcal{F}}}{|y-x|^s})), \quad M_\mathcal{Y} := \max_{\lambda'\in\Lambda} \sum_{\lambda\in\Lambda} \mathcal{X}_\lambda(\mathcal{Y}_{\lambda'}(\tfrac{1_{D_\mathcal{F}}}{|y-x|^s})) . \tag{2.18}$$

One verifies that for (2.17) to be bounded by ε, m in (2.18) must be of the order $|\log \varepsilon|$.

Remark 2.1 If $k(x, y) = e(x, y)$, the error in the kernel approximation (2.3) is completely independent of the surface Γ and its parametrization as well as on the distribution of the integration points on Γ. If $k(x, y)$ is a derivative of $e(x, y)$, as e.g. $k(x, y) = \partial_{n(y)} e(x, y)$ in the double layer potential, (2.3) can be derived in two ways: i) by applying $\partial_{n(y)}$ to an approximation $\tilde{e}(x, y)$ of e †, and ii) by expanding $\partial_{n(y)} e(x, y)$ directly. *We emphasize that with option i) the constant C in the clustering errors* (2.17) *is independent of the complexity of Γ, resp. its parametric representation.* This is not so for option ii) and for the wavelet methods below. *Moreover, having a cluster approximation of the single layer potential only the $\boldsymbol{Y}_\tau$ matrices have to be recalculated to obtain a cluster approximation of the double layer potential when using option i).*

*Note that identical formulations arise in the presence of numerical integration. In this case, the functionals $\mathcal{X}_\lambda, \mathcal{Y}_{\lambda'}$ are suitable quadrature formulas.

†The application of $\partial_{n(y)}$ can then be considered part $\mathcal{Y}_{\lambda'}$

2.4 Clustering Algorithms

Essential for the efficiency of the algorithm is (i) the construction of a partition $\mathcal{C}$ such that the near field matrix $\boldsymbol{N}$ is a sparse matrix, i.e., contains only $O(N)$ entries, and (ii) the fast evaluation of the approximate far field contribution, in particular the fast evaluation of the matrix vector product

$$\boldsymbol{v} = \sum_{(\sigma,\tau)\in\mathcal{F}} \boldsymbol{X}_\sigma^T \boldsymbol{F}_{\sigma\tau} \boldsymbol{Y}_\tau \boldsymbol{u}. \tag{2.19}$$

The key is a hierarchical organization of clusters. Let $\mathcal{P}$ denote the given panelization of Γ. We subdivide $\mathcal{P}$ into two about equally large sets recursively until the subsets contain $O(1)$ panels. This defines a binary tree with root $\mathcal{P}$. Each node of the tree represents a subset of $\mathcal{P}$ which in turn implies a subset of Γ, i.e. the binary tree defines a hierarchical decomposition of Γ into clusters.

Let $0 < \eta < 1$. By traversing the tree a clustering $\mathcal{C} = \mathcal{F} \cup \mathcal{N}$ can be constructed:

```
partition (σ, τ, F, N) {
  if (η_(σ,τ) < η) then
    F ← {(σ, τ)} ∪ F
  else if (σ is a leaf) or (τ is a leaf) then
    N ← {(σ, τ)} ∪ N
  else if (ř_σ < ř_τ) then
    for all children τ' of τ   partition(σ, τ', F, N)
  else
    for all children σ' of σ   partition(σ', τ, F, N)
}
```

The grain of the far field $\mathcal{F}$ will be bounded by η.

The matrix vector product (2.19) is evaluated in three steps:

1. evaluate for all τ: $\boldsymbol{u}_\tau := \boldsymbol{Y}_\tau \boldsymbol{u}$,
2. evaluate for all σ: $\boldsymbol{v}_\sigma := \begin{cases} \boldsymbol{F}_{\sigma\tau} u_\tau & \text{for } (\sigma,\tau) \in \mathcal{F}, \\ 0 & \text{otherwise,} \end{cases}$
3. evaluate $\boldsymbol{v} = \sum_\sigma \boldsymbol{X}_\sigma^T \boldsymbol{v}_\sigma$.

The steps 1 and 3 could be accelerated by using so-called shift operations:

$$\boldsymbol{Y}_\tau = \sum_{\tau' \text{child of } \tau} \boldsymbol{D}_{\tau\tau'} \boldsymbol{Y}_{\tau'}, \tag{2.20}$$

with matrices $\boldsymbol{D}_{\tau\tau'}$, i.e.,

$$\boldsymbol{u}_\tau = \begin{cases} \boldsymbol{Y}_\tau \boldsymbol{u} & \text{for } \tau \text{ a leaf,} \\ \sum_{\tau' \text{ child of } \tau} \boldsymbol{D}_{\tau\tau'} \boldsymbol{u}_{\tau'} & \text{otherwise.} \end{cases} \tag{2.21}$$

Hence, to evaluate $\boldsymbol{u}_\tau$ for all τ we only have to assemble matrices $\boldsymbol{Y}_\tau$ if τ is a leaf. These matrices are sparse. In the case of the multipole ansatz, for example, they contain only $O(|\mathcal{J}_m|) = O(m^2)$ entries. The products $\boldsymbol{D}_{\tau\tau'}\boldsymbol{u}_{\tau'}$ are handled by efficient algorithms without assembling $\boldsymbol{D}_{\tau\tau'}$ explicitly [16, 34]. The same holds for step 3. With matrices $C_{\sigma\sigma^*}$ defined by

$$\boldsymbol{X}_{\sigma^*}^T = \sum_{\sigma \text{ child of } \sigma^*} \boldsymbol{X}_\sigma^T \boldsymbol{C}_{\sigma\sigma^*}, \tag{2.22}$$

and vectors $\bar{\boldsymbol{v}}_\sigma := \boldsymbol{v}_\sigma + \boldsymbol{C}_{\sigma\sigma^*}\bar{\boldsymbol{v}}_{\sigma^*}$, σ child of σ^*, it follows that

$$\boldsymbol{v} = \sum_\sigma \boldsymbol{X}_\sigma^T \boldsymbol{v}_\sigma = \sum_{\sigma \text{ a leaf}} \boldsymbol{X}_\sigma^T \bar{\boldsymbol{v}}_\sigma. \tag{2.23}$$

Again, only matrices $\boldsymbol{X}_\sigma$ for leaves σ must be assembled.

An analysis of the complexity (cf. [26], [18]) shows that using the multipole evaluation the number of operations necessary to perform the matrix vector product (2.19) is of order $O(m^4 N)$, with N the number of unknowns.[‡] The memory requirements are of order $O(m^2 N)$. To ensure that the error of the far field approximation is asymptotically equal to the order of the discretization error, we have to choose $m = O(\log N)$.

2.5 Numerical experiment

Here we present some numerical experiments in order to show the performance of the method. The goals of the experiments are to investigate the error dependence on the order m of the cluster expansion, and validation of the $O(N(\log N)^4)$ complexity of the algorithm.

We therefore consider the problem (1.1) in the unit shere Ω where the "true" potential $U(x)$ is given by

$$U(x) = |x|^{-1} + x_1 x_2 \, |x|^{-5} \tag{2.24}$$

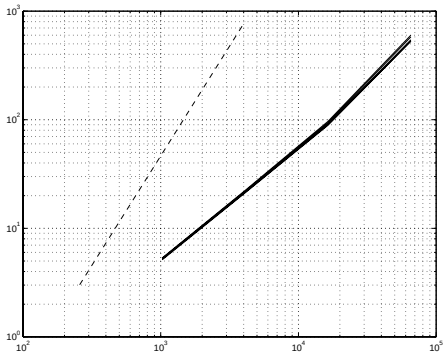
and the boundary condition is $\partial_n U = f$ with $n(x)$ the exterior unit normal vector to Γ.

We approximated the unit sphere by planar triangles. Continuous, piecewise linear polynomials have been used as trial and test functions. The numerical quadrature for the near field integrals has been done using special quadrature techniques [22, 27].

The results were obtained on a *SUN* Ultra-Enterprise 4000/5000 on a single processor (*UltraSPARC*, 248MHz), 2 GB RAM using the *SUN* C++ 4.2 Compiler and the class library *Concepts-1.3* for boundary elements.

The linear system of equations was solved using a GMRES solver without any preconditioning. About 30 iterations were necessary to keep the error lower than the discretization error, independent of the number of unknowns. For our cluster algorithm the

[‡]With a more sophisticated approach to evaluate the products $\boldsymbol{F}_{\sigma\tau}\boldsymbol{u}_\tau$ using exponential expansions this could be reduced to $O(m^3 N)$ [16].



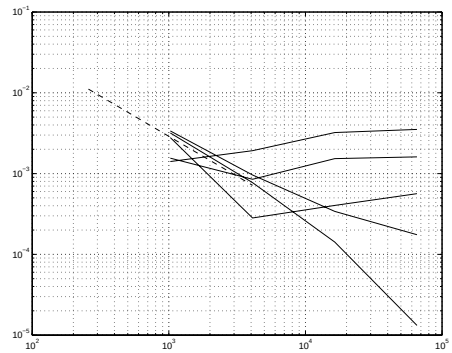


Figure 1: (left) CPU-time for matrix assembly ($m = 3, 4, 5, 6, 7$)(in seconds) versus number N of panels: standard BEM (dashed line) versus fast algorithm (solid lines)

Figure 2: (right) Relative mean absolute error in a set of points with distance 0.5 from the surface of the unit sphere: standard BEM (dashed line) versus fast algorithm for $m = 3, 4, 5, 6, 7$ (solid lines)

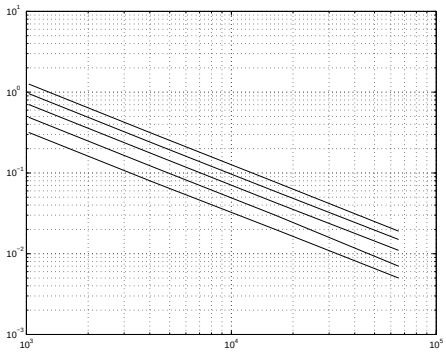
matrix-vector operations for the calculation of the far field contribution have been done in every iteration step. The necessary information about the $\boldsymbol{X}_\sigma$, $\boldsymbol{Y}_\tau$ and $\boldsymbol{F}_{\sigma\tau}$ matrices have been stored in core on the workstation. The quality of the solution has been checked at a grid of points with distance 0.5 to the surface of the unit sphere.

Figure 1 shows the CPU-time for the matrix assembly for the standard BEM (dashed line) and our fast algorithm (solid lines). The latter depends on the order m of the multipole expansion. The computations have been done for $m = 3 \ldots 7$. The results are shown as function of the number of unknowns, i.e., of the resolution. The finest resolution contains 65538 unknowns, i.e., 131072 panels. The dependence of the CPU-time on the expansion order m is minor, because N dominates. Compared with the standard method a speed-up of up to 3 orders of magnitude is realized for the finest resolution.

Figure 2 shows the relative mean absolute error in the potential in exterior points located at a distance of 0.5 from the surface $\partial\Omega$ for various m. The solid lines represent the cluster-BEM solution for $m = 3 \ldots 7$, the dashed line represents the standard-BEM solution. Only for $m = 6, 7$ we observe an almost monotone decreasing error with increasing number of unknowns. This indicates that small values of m corresponding to low expansion orders produce approximation errors that dominate the total error budget if the discretization becomes finer. At a certain discretization level $m = 5$ gives a better accuracy than $m = 7$. This can be explained by the influence of the discretization error which dominates at this discretization level the total error budget.

Figure 3 shows the compression rate as a function of the number of unknowns. A compression factor of 0.01 means that the total of entries to store the necessary information of the $\boldsymbol{X}_\sigma$, $\boldsymbol{F}_{\sigma\tau}$, $\boldsymbol{Y}_\tau$ matrices is equal to 1% of the entries of the dense stiffness matrix $\boldsymbol{A}$.

In Figure 4 we show the number of necessary matrix entries for the cluster-BEM and the standard-BEM as a function of the potential error in exterior points. It clearly shows that the higher the accuracy requirements are the more storage could be saved with the cluster-BEM.



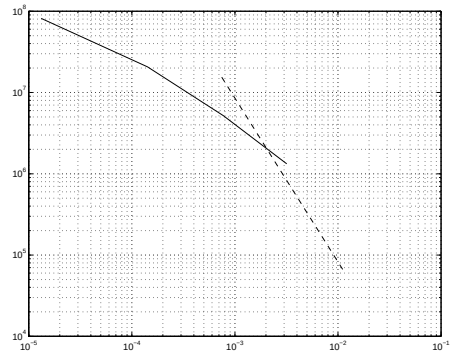


Figure 3: (left) Compression of the stiffness matrix for $m = 3, 4, 5, 6, 7$.

Figure 4: (right) Number of necessary matrix entries as function of the potential error in exterior points: standard BEM (dashed line) versus fast algorithm ($m = 7$) (solid line).

3 Wavelet BEM

Wavelet-BEM are collocation or Galerkin BEM (1.7b), (1.7c) where the FE basis $\{\varphi_\lambda : \lambda \in \Lambda\}$ of V^N has been changed into a wavelet basis $\{\psi_\lambda : \lambda \in \Lambda\}$[§] The basic idea is simple: the wavelets have *vanishing moments* in local surface coordinates and this causes most entries in the stiffness matrix $\boldsymbol{A}^\psi$ with respect to the wavelet basis ψ to be negligible. Let us illustrate this by the simplest surface wavelet.

3.1 Haar Multiwavelets on Γ

We assume each $\Gamma_k \subset \Gamma$ to be triangular and generate a dyadic sequence of meshes $\{\mathcal{M}_k^\ell\}_{0 \le \ell \le L}$ on Γ_k by halving the sides of each triangle $T \in \mathcal{M}_k^\ell$. The corresponding mesh on Γ is then $\mathcal{M}^\ell = \cup_{k=1}^{N_0} \mathcal{M}_k^\ell$ with $N_\ell = N_0\, 4^\ell$ triangles T_j^ℓ, $j = 0, \ldots, N_\ell - 1$. Let φ_j^ℓ denote the characteristic function of T_j^ℓ, then we may write any piecewise constant function u^L on M_L as

$$u^L = \sum_{j=0}^{N_L - 1} \{\boldsymbol{u}_\varphi^L\}_j\, \varphi_j^L \ . \tag{3.1}$$

A wavelet basis for $V^L = \mathrm{span}\{\varphi_j^L\}$ is obtained by adding only "new", incremental unknowns when refining from $\mathcal{M}^\ell \to \mathcal{M}^{\ell+1}$. This is illustrated in Fig. 3.1. This concept can also be described in terms of function spaces on Γ. Let V^ℓ be the space of piecewise constants on $\mathcal{M}^\ell$. Clearly,

$$V^0 \subset V^1 \subset \cdots \subset V^\ell \subset \ldots \tag{3.2}$$

[§]Since V^N was assumed to be a finite element space of piecewise, continuous or discontinuous, polynomials on Γ, $V^N = \mathrm{span}\{\psi_\lambda : \lambda \in \Lambda\}$ means that we consider here only *spline-wavelets*; the fractal, fully orthogonal Daubecies wavelets (e.g. [12]), could in principle also be used, but do not seem suitable to us for BEM since they cause difficulties in numerical integration.

and the sequence $\{V^\ell\}_\ell$ is dense in $V = L^2(\Gamma)$. Define $W^0 = V^0$ and

$$W^\ell = \{\psi \in V^\ell : \langle \psi, \varphi \rangle = 0 \quad \forall \varphi \in V^{\ell-1}\}, \;\; \ell = 1, 2, 3, \ldots \tag{3.3}$$

where $\langle \cdot, \cdot \rangle$ denotes the L^2-innerproduct on Γ. Then

$$\dim W^\ell = \dim V^\ell - \dim V^{\ell-1} = N_\ell - N_{\ell-1} = N_0(4^\ell - 4^{\ell-1}) = 3N_0\, 4^{\ell-1} \; .$$

For $\ell = 1$ there are 3 shapefunctions ψ_j^1 on each Γ_k, the so-called "motherwavelets", which are piecewise constant on $\mathcal{M}^1$ and have vanishing mean. A basis for W^ℓ is obtained by replication, i.e. translation and scaling, from the $\{\psi_j^1\}$. Clearly,

$$V^L = W^L \oplus W^{L-1} \otimes \cdots \otimes W^0 \tag{3.4}$$

and $u^L \in V^L$ can equivalently be expressed in the basis $\{\psi_j^\ell\}$:

$$u^L = \sum_{j=0}^{N_L-1} \{\boldsymbol{u}_\varphi^L\}_j\, \varphi_j^L = \sum_{j=0}^{N_0-1} \{\boldsymbol{u}_\psi^0\}_j\, \psi_j^0 + \sum_{\ell=0}^{L-1} \sum_{j=0}^{3N_\ell-1} \{\boldsymbol{u}_\psi^{\ell+1}\}_j\, \psi_j^{\ell+1} \; . \tag{3.5}$$

Translation between the coefficient vectors $\{\boldsymbol{u}_\varphi^L\}$ and $\{\boldsymbol{u}_\psi^\ell\}_{\ell \le L}$ is achieved in $O(N_L)$ operations using the pyramid scheme:

$$\{\boldsymbol{u}_\varphi^{\ell+1}\} = \boldsymbol{H}_\ell \{\boldsymbol{u}_\varphi^\ell\} + \boldsymbol{G}_\ell \{\boldsymbol{u}_\psi^{\ell+1}\} \tag{3.6}$$

where

$$\boldsymbol{H}_\ell = \text{blockdiag} \left\{ \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \right\} \in \mathbb{R}^{N_{\ell+1} \times N_\ell}, \; \boldsymbol{G}_\ell = \text{blockdiag} \left\{ \begin{pmatrix} 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \right\} \in \mathbb{R}^{N_\ell \times 3N_\ell} \; .$$

It is clear that higher order, discontinuous wavelets can be generated completely analogously (see [31] for degree 1 multiwavelets on triangles and [7], Section 9.3, for the general construction valid also on quads). Such wavelets satisfy an analog of Parseval's equality in $L^2(\Gamma)$: let $u \in L^2(\Gamma)$ be expanded in wavelets:

$$u = \sum_\lambda \langle u, \psi_\lambda \rangle\, \psi_\lambda \; , \tag{3.7}$$

then

$$\|u\|_{L^2(\Gamma)}^2 = \sum_\lambda |\langle u, \psi_\lambda \rangle|^2 \; . \tag{3.8}$$

In particular, the Haar multiwavelets are fully orthogonal, i.e.

$$\langle \psi_\lambda, \psi_{\lambda'} \rangle = \delta_{\lambda\lambda'} \quad \forall \lambda\lambda' \; , \tag{3.9}$$

and the wavelets of (piecewise) degree p have vanishing moments of order $d = p+1$, i.e.[¶])

$$\langle p, \psi_\lambda \rangle = 0 \quad \forall p \in P_d(\Gamma), \; |\lambda| > 0 \; , \tag{3.10}$$

where P_d is the set of piecewise polynomials on Γ of total degree $< d$.

[¶]for curved surfaces Γ, (3.5) must hold for the pullback of ψ_λ in local coordinates.

3.2 Other wavelet families

The Haar-type multiwavelets are only suitable for second kind BIEs. For hypersingular BIEs (1.2) posed in $V = H^{1/2}(\Gamma)$, evidently discontinuous ψ_λ are not admissible. Several continuous, biorthogonal wavelets have been proposed. As a rule, to achieve the analog of (3.8) in $H^{1/2}(\Gamma)$ is difficult, since the $H^{1/2}(\Gamma)$ norm is, unlike the $L^2(\Gamma)$ norm, not additive on the Γ_k. The same applies to $H^{-1/2}(\Gamma)$ - here discontinuous wavelets are admissible, but one has to have a higher number of vanishing moments than for the Haar wavelets. Let in (1.4) $r = s - 2$ the order of A, and assume the vanishing moment condition

$$\langle p, \psi_\lambda \rangle = 0 \quad \forall p \in P_{\widetilde{d}}(\Gamma),\ |\lambda| > 0 \tag{3.11}$$

where ψ_λ is a spline wavelet of degree $< d$. In order to achieve $O(N)$-algorithms $\widetilde{d}$ must satisfy ([39])

$$d < \widetilde{d} + r \Longleftrightarrow \widetilde{d} > d - r \tag{3.12}$$

If equality holds in (3.12), as in the case of Haar-wavelets for second kind BIEs ($d = \widetilde{d} = p + 1, r = 0$), only $O(N(\log N)^a)$, $0 < a \leq 2$ complexity is achievable.

Biorthogonal spline wavelets satisfying (3.12) for $r = -1, 0, 1$ are available on $\mathbb{R}^2$ (see [11], [7], [6]). On polyhedral surfaces, some delicate compatibility conditions at the edges must be satisfied; this can be done (see [9], [10] for these constructions) at the expense of a larger support. We also mention [32], [33] for C^0-wavelets and collocation of second kind BIEs. There, in particular, the first fully discrete schemes with nonanalytic surfaces Γ were obtained. The computational performance of these wavelets on surfaces is yet open.

3.3 Wavelet BEM

For ease of notation, we combine the indices (ℓ, j) of ψ_j^ℓ into a multiindex $\lambda = (\ell, j)$, $|\lambda| = \ell$; then $V^L = \text{span}\{\psi_\lambda : \lambda \in \Lambda\}$. Assume that A in (1.2) is a Fredholm operator of the second kind, i.e. $s \leq 2$ in (1.4). A wavelet version of the classical panel method can be obtained by letting x_λ be the barycenter of the $T^\lambda = T_j^\ell \in \mathcal{M}_\ell$, and $\mathcal{X}_\lambda \varphi = \varphi(x_\lambda)$, $\mathcal{Y}_{\lambda'} \varphi = \langle \varphi, \psi_{\lambda'} \rangle$. In the Galerkin variant also $\mathcal{X}_\lambda \varphi = (\varphi, \psi_\lambda)$, and we denote the resulting stiffness matrix by $\boldsymbol{A}^L$ to emphasize dependence on the level L. The problem (1.5) reads:

$$\boldsymbol{A}^L \boldsymbol{u}^L = \boldsymbol{g}^L \ . \tag{3.13}$$

3.4 Compression

The main effect of using a wavelet basis is the smallness of farfield entries of $\boldsymbol{A}^L$. Let

$$U_\lambda = \text{supp}\ \mathcal{X}_\lambda, \quad U_{\lambda'} = \text{supp}\ \mathcal{Y}_{\lambda'}, \quad \lambda, \lambda' \in \Lambda \ ,$$

and assume that the $\mathcal{X}_\lambda$, $\mathcal{Y}_{\lambda'}$ have *vanishing moments:*

$$\mathcal{X}_\lambda p = 0 \quad \forall p \in \mathcal{P}_{\widetilde{d}},\ \ \mathcal{Y}_\lambda p = 0 \quad \forall p \in \mathcal{P}_{\widetilde{d}'} \ , \tag{3.14}$$

where $\mathcal{P}_d$ denotes the polynomials of total degree $< d$ in local coordinates. Then

$$|\boldsymbol{A}^L_{\mathcal{X}\mathcal{Y}}| \leq C\,(d_{\lambda\lambda'})^{-s-\widetilde{d}-\widetilde{d}'} \tag{3.15}$$

whenever $d_{\lambda\lambda'} := \mathrm{dist}(U_\lambda, U_{\lambda'}) \geq c\,2^{-\min(|\lambda|,|\lambda'|)}$, i.e. *each vanishing moment of either $\mathcal{X}_\lambda$ or $\mathcal{Y}_{\lambda'}$ gives one extra order of decay in the farfield.* The constant C in (3.15) depends, in general, on the normalization of the ψ_λ. The decay estimate (3.15) is then a direct consequence of (1.4) and (3.14) ([7, 30, 39, 25]). The *matrix-compression* of $\boldsymbol{A}^L$ is:

$$\widetilde{\boldsymbol{A}}^L_{\lambda,\lambda'} := \begin{cases} \boldsymbol{A}^L_{\lambda,\lambda'} & \text{if } d_{\lambda\lambda'} < \tau_{\lambda\lambda'} \\ 0 & \text{else} \end{cases} \tag{3.16}$$

where $\tau_{\lambda\lambda'}$ are judiciously chosen truncation parameters [‖]). For second kind BIEs we have $s = 2$ in (1.4), and, for Galerkin BEM with Haar-multiwavelets of degree p, (3.14) holds with $\widetilde{d} = \widetilde{d}' = p + 1$, and we find [25, 30] the rule

$$\tau_{\lambda\lambda'} = \max\{a2^{-|\lambda|}, a2^{-|\lambda'|}, a2^{L-|\lambda|-|\lambda'|}\} \tag{3.17}$$

which was used in our numerical simulations below. Note that for collocation, or operators of nonzero order, other truncation parameters have to be selected (see [7, 32, 33, 39] for more).

The estimate (3.15) indicates that vanishing moments of both, $\mathcal{X}_\lambda$ and $\mathcal{Y}_{\lambda'}$, contribute to the compression. Galerkin BEM (1.7c) are preferable from this point of view, since for the collocation case $\mathcal{X}_\lambda(\varphi) = \varphi(x_\lambda)$ has no vanishing moments at all. This can, however, be compensated by judicious linear combinations of Dirac's (see [32], Sect. 3.3 for an example).

3.5 Compression error estimate

We restrict ourselves to operators A of order zero and to Galerkin-BEM using the discontinuous Haar multiwavelets ψ_λ of degree $p \geq 0$ on triangulations. We have the

Theorem 3.1 *[30, 25]: Under the above assumptions, if a in (3.17) is sufficiently large, the compressed stiffness matrix $\widetilde{\boldsymbol{A}}^L$ in (3.16) is stable, i.e. cond$(\widetilde{\boldsymbol{A}}^L) \leq C < \infty$ for all L. Moreover, the error in the BEM solution $\widetilde{u}^L$ obtained from the compressed matrix $\tilde{\boldsymbol{A}}^L$ can be estimated by*

$$\|u - \widetilde{u}^L\|_{L^2(\Gamma)} \leq C\,N_L^{-s/2}\,L^\nu \|u\|_{H^s(\Gamma)} = C\,h^s\,|\log h|^\nu\,\|u\|_{H^s(\Gamma)} \tag{3.18}$$

for $0 \leq s \leq p+1$ with $\nu = 0$ if $s < p+1$, $\nu = 1$ if $s = p+1$. Moreover, for the approximate solution $\widetilde{U}^L$ of (1.1) holds: $\forall x \in \Omega$ ex. $C(x) > 0$ such that

$$|U(x) - \widetilde{U}^L(x)| \leq C(x)\,h^{-(s+\widetilde{s})}|\log h|^{\nu(s)+\nu(\widetilde{s})}\,\|u\|_{H^s(\Gamma)}\|g\|_{H^{\widetilde{s}}(\Gamma)} \tag{3.19}$$

[‖] In addition, also certain elements in the near field can be neglected, if the distance between U_λ and $U^{\mathrm{sing}}_{\lambda'} = \mathrm{sing\,supp}(\mathcal{Y}_{\lambda'})$ gets large, see [7], (9.28) and [39].

where $0 \le s,\ \widetilde{s} \le p+1$ *and* $g \in H^{\widetilde{s}}(\Gamma)$, $A^*\varphi = g \Longrightarrow \varphi \in H^{\widetilde{s}}(\Gamma)$ *and* $\nu(t) = 0$ *for* $0 \le t < p+1$, $\nu(p+1) = 3/2$. *The compressed stiffness matrix* $\widetilde{\boldsymbol{A}}^L$ *can be obtained using* $O(N_L(\log N_L)^4)$ *work and* $O(N_L(\log N_L)^2)$ *memory.*

There are analogs of the present theorem also for first kind, weakly and hypersingular BIEs as well as for collocation schemes; see [7, 39]. The estmate (3.19) shows that the superconvergence of potentials at interior points in Galerkin-BEM is preserved under compression, *provided* $Au = f$ *and* $A^*\varphi = g$ *have sufficiently regular solutions for smooth* f *and* g. This is unrealistic on polyhedra due to edge- and vertex singularities. In many cases, one has (3.18), (3.19) only with $0 \le s, \widetilde{s} < 1$, which is why the Haar-Galerkin wavelet with $p = 0$ is close to optimal *if uniform mesh refinement is used.* To exploit higher order wavelets on polyhedra, *adaptivity* must come into play. Recent, very technical results indicate that adaptive wavelet algorithms can be designed (see [5]) which realize optimal convergence rates at $O(N)$ complexity, but numerical experience has yet to be gained with these algorithms.

3.6 Numerical Experiments

In this section, we present the results of three numerical experiments obtained with the described implementation of the multiscale scheme. In a polyhedron $\Omega \subset \mathbb{R}^3$ we considered (1.1). The double layer ansatz $U(x) = \langle k(x,\cdot), u\rangle$ where the double layer kernel is given by

$$k(x,y) = -\frac{1}{4\pi}\,\frac{\langle n(y), y-x\rangle}{|y-x|^3} \tag{3.20}$$

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

$$u \in L^2(\Gamma): \quad \langle v, Au\rangle = \langle v, f\rangle \quad \forall v \in L^2(\Gamma) \tag{3.21}$$

with

$$(Au)(x) = -\frac{1}{2}\,u(x) + \int_\Gamma k(x,y)u(y)\,ds_y \tag{3.22}$$

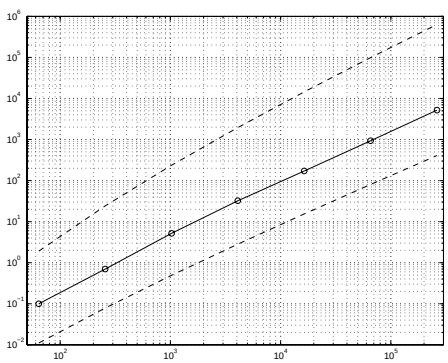
defined almost everywhere on Γ. We solved (3.21) on several polyhedral domains with quite similar performance. Here, we only report the results obtained with a tetrahedron defined by four equilateral triangles with vertices on the unit sphere ($\gamma \sim 1$) and the right hand side

$$f(x) = |x - x_0|^{-1}, \qquad x_0 := (1,1,1)^T \text{ in the exterior of } \Omega. \tag{3.23}$$

For the discretization constant test and trial functions ($d = 0$) were used. We did not make use of the fact that entries in the stiffness matrix corresponding to panels located in the same face of the polyhedron Ω vanish. All results were obtained on a *SUN Ultra-Enterprise 4000/5000* on a single processor (*UltraSPARC*, 248 MHz) and 2 GB RAM using the *SUN C++ 4.2* Compiler.

level	N_L	a, α	time[s]	mem[MB]	it	cpr
2	64	0.3, 1.0	0.1	0.03	16	0.941
3	256	0.3, 1.0	0.8	0.21	18	0.424
4	1024	0.3, 1.0	5.9	1.17	19	0.146
5	4096	0.3, 1.0	36.7	5.80	19	0.045
6	16384	0.3, 1.0	193.2	27.44	18	0.013
7	65536	0.3, 1.0	1045.1	126.29	18	0.004
8	262144	0.3, 1.0	5744.9	570.43	19	0.001

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



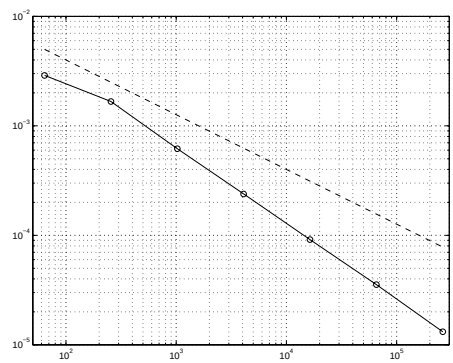


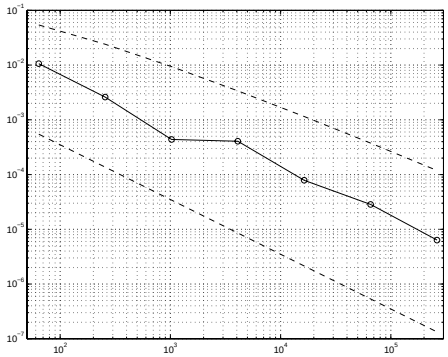
Figure 5: (left) Time for assembly and compression of the matrix.

Figure 6: (right) $|\,\|u^l\|_0 - \|u\|_0\,|$ versus N_L.

In the first experiment we kept the parameters a and $\alpha = \alpha'$ of the thresholds (3.16) controlling the compression fixed and solved the problem on various levels up to about 260000 unknowns (Table 1). On the finest mesh the compressed matrix consists of only 0.1% of the entries of the dense stiffness matrix. In addition, it can be observed that the number of iterations used by the solver (GMRes without restart) is almost constant validating the bounded condition numbers of the compressed matrices.

In Figure 5 the time of assembly and compression is depicted. Here, the upper dashed line corresponds to the bound $O(N_L(\log N_L)^4)$ in Theorem 3.1 [30, 25]. The plot indicates that the influence of the higher order logarithmic terms on the computing time seems to be negligible compared to the $O(N_L(\log N_L)^2)$ term illustrated by the lower dashed line. Roughly speaking, on an average nearly a constant number of operations is used to evaluate an entry of the stiffness matrix.

In all numerical experiments the time for GMRes accounts only for less than 10% of the total time shown in the tables. Therefore, with the present method the BEM-paradigm that most of the work is spent for quadrature is still valid and a speed up similar to the one for dense matrices can be achieved with the parallelization of the matrix assembly.



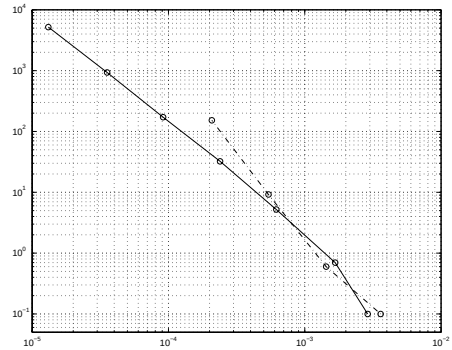


Figure 7: (left) Error at interior points versus N_L.

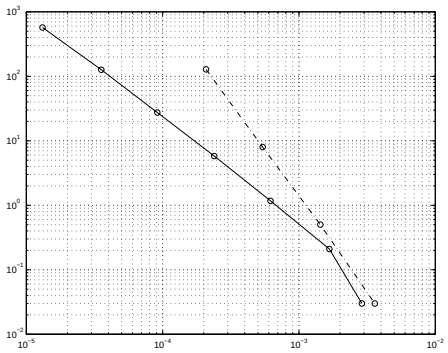
Figure 8: (right) CPU-Time in sec versus $|\,\|u^l\|_0 - \|u\|_0\,|$.

Figure 6 and Figure 7 show the behaviour of the L^2-error of the density u on the boundary and the average error in several interior points of the solution U, respectively. The L^2-error is approximated by the difference of the norm of the discrete density and the norm of the exact density. Since an exact solution is not available we computed an approximate value by higher order quadrature on a higher level with a parallelized implementation. According to Theorem 3.1, the expected rate of convergence is determined by regularity properties of A and its adjoint A^*. From the known edge and vertex singularities of the Laplacean in polyhedra it can be verified that in the example under consideration here both operators admit solutions belonging to $H^1(\Gamma)$ for smooth right hand sides. This means that we have Theorem 3.1 with $s = \tilde{s} = 1$ and expect essentially $O(N_L^{-1/2})$ convergence in the $L^2(\Gamma)$-norm and $O(N_L^{-1})$ convergence at an interior point (note that collocation or Nyström schemes do not display this kind of superconvergence at an interior point and would require H^2-regularity on Γ and $d = 1$ to achieve $O(N_L^{-1})$ convergence at an interior point).

Again, the dashed line in Figure 6 illustrates the expected behaviour of essentially $O(N_L^{-1/2})$. For the error in interior points, Figure 7, twice the convergence rate should be observed, hence essentially $O(N_L^{-1})$ (lower dashed line) or, according to Theorem 3.1, $O(N_L^{-1}(\log N_L)^2)$ (upper dashed line).

Finally, we compared our method with a standard boundary element implementation generating the fully populated stiffness matrix with an optimized quadrature rule. For both methods the time and memory used to generate a solution satisfying a given L^2-error are depicted in Figure 8 and Figure 9, respectively, where the dashed line corresponds to the standard approach. It turns out that already for moderate accuracy, "moderate" with respect to our model problem, the wavelet method beats the standard approach: assuming an error of about 10^{-4} the wavelet method is 10 times faster. Moreover, in this case it saves about 98% of the memory. In addition, it can be observed that the memory consumption is always less than in the standard case although additional information to recover the structure of the compressed matrix (tags) must be stored.

The second experiment investigates the behaviour of the method when the amount of compression driven by the parameter a changes (Table 2). The constant number of



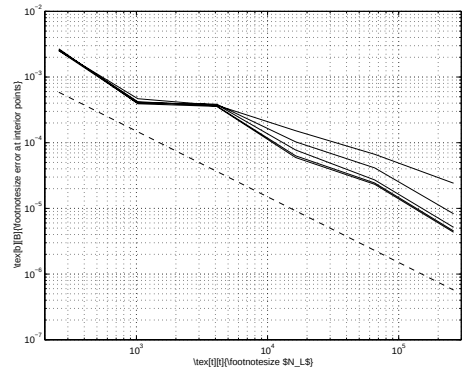


Figure 9: (left) Memory versus $|\,\|u^l\|_0 - \|u\|_0\,|$.

Figure 10: (right) Error at interior points versus N_L, $a = 0.1, 0.2, 0.4, 0.7, 1.0$, $\alpha = 1.0$.

level	N_L	a, α	time[s]	mem[MB]	it	cpr
8	262144	0.1, 1.0	5606.5	424.57	19	0.0008
8	262144	0.2, 1.0	5669.8	490.47	19	0.0009
8	262144	0.4, 1.0	5842.8	672.14	18	0.0013
8	262144	0.7, 1.0	6224.1	1009.25	18	0.0019
8	262144	1.0, 1.0	6778.2	1450.70	18	0.0028

Table 2: Second experiment

iterations shows that even for a high compression the algorithm remains stable. The convergence rates, in addition, are in all cases preserved as indicated by the error in interior points shown in Figure 10: when the influence of the coarser meshes, where practically no compression is possible, vanishes, the lines corresponding to different values of a fan out. Nevertheless, they finally take the same slope. However, if the amount of compression is reduced by means of parameter α instead of a, this is not the case as predicted by Theorem 3.1 and observed in the last experiment (Table 3).

We point out that the influence of the amount of compression on the computing time, in particular the time of assembly, is small compared to the influence on the memory consumption (Tables 2,3). The reason for this is that the time to evaluate an entry of the stiffness matrix depends, via the quadrature order, on the distance of the supports of the

level	N_L	a, α	time[s]	mem[MB]	it	cpr
8	262144	1.0, 0.2	5702.2	663.75	19	0.0013
8	262144	1.0, 0.4	5774.8	753.93	18	0.0014
8	262144	1.0, 0.6	5944.6	920.13	18	0.0018
8	262144	1.0, 0.8	6256.1	1137.70	18	0.0022
8	262144	1.0, 1.0	6778.2	1450.70	18	0.0028

Table 3: Third experiment.

level	N_L	a, α	sequential time[s]	4 processors time[s] (speed up)
4	1024	0.3, 1.0	5.2	1.3 (4.0)
5	4096	0.3, 1.0	32.3	8.1 (4.0)
6	16384	0.3, 1.0	171.5	42.9 (4.0)
7	65536	0.3, 1.0	933.3	232.6 (4.0)
8	262144	0.3, 1.0	5200.3	1296.1 (4.0)
9	1048576	0.3, 1.0		4102.9

Table 4: Assembly of the system matrix using threads.

related wavelets whereas the amount of memory to store the value is always the same. Increasing the thresholds means adding entries to the matrix with more or less distant support, which can be computed very fast compared to the entries near the diagonal. The time of solution, however, increases as fast as the memory.

To conclude, we report on an implementation of the matrix assembly using threads which were assigned to different processors on the *SUN Ultra-Enterprise 4000/5000*. Each thread processes rows and columns of the compressed matrix with a local cache. Rows and columns are assigned cyclically to ensure a static load balance. With this implementation the considered problem was solved up to level 9, i.e. with more than a million unknowns, on four processors. The CPU-times compared to the sequential version are shown in Table 4. A nearly perfect speed up is observed.

References

[1] Atkinson, K.: The numerical solution of integral equations of the second kind, Cambridge Univ. Press (1997).

[2] Beylkin, G., Coifman, R. and Rokhlin, V.: Fast wavelet transforms and numerical algorithms, *Comm. Pure Appl. Math.* **44** (1991), 141-183.

[3] Bonnet, M.: Boundary integral equation methods for solid and fluids, Wiley (1999).

[4] Chew, W.C., Jin, J.-M., Lu, C.-C., Michielssen, E. and Song, J.M.: Fast solution methods in eletromagnetics, *IEEE Trans. Ant. Propagation* **45** (1997), 533-543.

[5] Cohen, A., Dahmen, W. and DeVore, R.: Adaptive Wavelet Methods for elliptic Operator Equations, Convergence Rates, Preprint, IGPM, RWTH Aachen, (1998).

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

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

[8] Dahmen, W., Prössdorf, S. and Schneider, R.: Wavelet approximation methods for pseudodifferential equations on smooth manifolds, Proc. of the Int. Conf. on wavelets (C.K. Chui, L. Montefusco and L. Puccio, eds.), Academic Press, (1994), 385-424.

[9] Dahmen, W. and Schneider, R.: Composite wavelet bases for operator equations, Preprint SFB393/96-19, TK-Chemnitz, 1996.

[10] Dahmen, W. and Schneider, R.: Wavelets on manifolds I: Construction and Domain Decomposition, Preprint.

[11] Dahmen, W. and Stevenson, R.: Element-by-element construction of wavelets satisfying stability and moment conditions, to appear in *SIAM J. Numer. Anal.*

[12] Daubecies, I: 10 Lectures on wavelets, SIAM Publ. Philadelphia (1992).

[13] Y. Fu, K. J. Klimkowski, G. J. Rodin, E. Berger, J. C. Browne, J. K. Singer, R. A. van de Geijn, and K. S. Vemaganti "A fast solution method for three-dimensional many-particle problems of linear elasticity" Int. J. Numer. Meth. Engng 42, 1215-1229 (1998)

[14] Giebermann, K.: Schnelle Summationsverfahren zur numerischen Lösung von Integralgleichungen für Streuprobleme in $\mathbb{R}^3$, (in german), Ph.D. Dissertation, Dept. Mathematics, Univ. of Karlsruhe (1997).

[15] Graham, I.G., Hackbusch, W. and Sauter, S.A.: Fast integration techniques in 3-d BEM, Preprint 98-29, School of Math. Sciences, Univ. Bath, U.K.

[16] Greengard, L. and Rokhlin, V.: A new version of the fast multipole method for the Laplace equation in three dimensions, Acta Numerica **6** (1997), 229-269, Cambridge University Press.

[17] Guiggiani, M., Krishnasamy, G., Rudolphi, T.J., Rizzo, F.: A general algorithm for the numerical solution of hypersingular BIEs, *Trans. ASME J. Appl. Mech.* **59** (1992), 604-614.

[18] Hackbusch, W. and Nowak, Z.P.: On the fast matrix multiplication in the BEM by panel clustering, *Numer. Math.* **54** (1989), 463-491.

[19] Hackbusch, W.: Integral equations, Birkhäuser Publ. Basel (1995).

[20] Hayami, K. and Sauter, S.: Application of panel clustering to three-d elastostatics, pp. 625-634 in Proc. of the 19th Intl. Conf. on BEM, Rome, Sept. 1999, M. Marchetti, C.A. Brebbia and M.H. Aliabadi (Eds.), Comp. Mech. Publ., Southampton, U.K.

[21] Hsiao, G. and Wendland, W.: Variational methods for boundary integral equations, Springer Verlag (to appear).

[22] Lage, Ch.: Softwareentwicklung zur Randelementmethode: Analyse und Entwurf effizienter Techniken (in german), Dissertation Univ. Kiel, Germany (1995).

[23] Lage, Ch. and Schwab, C.: A wavelet-Galerkin boundary element method on polyhedral surfaces in $\mathbb{R}^3$, in: Lecture Notes in Numerical Fluid Mechanics, Vol. 55 (W. Hackbusch and G. Wittum, eds.), Vieweg Publ. Braunschweig (1996), 194-206.

[24] Lage, Ch. and Schwab, C.: On the implementation of a fully discrete multiscale Galerkin BEM, Proc. of the BEM XIX Conference, Rome, Sp. 9-12, 1997. Comp. Mechanics Publications, Southampton, UK (1998), 635-644.

[25] Lage, Ch. and Schwab, C.: Wavelet Galerkin algorithms for boundary integral equations, (in press in SIAM J. Scient. Stat. Computing), (1999).

[26] Lage, Ch.: Fast evaluation of singular kernel functions by cluster-methods, (in preparation).

[27] Lage, Ch. and Sauter, S.A.: General black box quadrature in BEM, Math. Comp. (1999), to appear.

[28] Nedelec, J.C.: Integral equations associated with elliptic boundary value problems in $\mathbb{R}^3$, pp. 114 ff, in: Mathematical Analysis and Numerical Methods in Science and Technology, Vol. 4, (J.L. Lions & R. Dautray, eds.), Springer Verlag, 1990.

[29] von Petersdorff, T. and Schwab, C.: Wavelet discretization of first kind boundary integral equations on polygons, *Numer. Math.* **74** (1996), 479-519.

[30] von Petersdorff, T. and Schwab, C.: Fully discrete multiscale Galerkin BEM, in: Multiresolution Analysis and Partial Differential Equations, W. Dahmen, P. Kurdila and P. Oswald (eds.), Wavelet Analysis and its applications, Vol. 6, Academic Press, New York (1997), 287-346.

[31] von Petersdorff, T., Schneider, R. and Schwab, C.: Multiwavelets for second kind integral equations, *SIAM J. Num. Anal.* **34** (6) (1997), 2212-2227.

[32] Rathsfeld, A.: A wavelet algorithm for the solution of a singular integral equation over a smooth 2-d manifold, *J. Int. Eq. Appl.* **10** (1998).

[33] Rathsfeld, A. and Ehrich, S.: Piecewise linear wavelet collocation on triangular grids, Preprint 434, WIAS, Berlin (1998).

[34] Rokhlin, V.: Diagonal forms of translation operators for the Helmholtz equation in three dimensions. *Appl. Comp. Harmonic Anal.* **1** (1993), 82-93.

[35] Boundary Elements: Implementation and Analysis of advanced algorithms, Notes on Num. Fluid Mech. 54 (W. Hackbusch and G. Wittum, eds.), Vieweg Publ. Braunschweig, Germany (1996).

[36] Sauter, S.A. and Krapp, A.: On the effect of numerical integration in the Galerkin BEM, *Numer. Math.* **74** (1996), 337-359.

[37] Sauter, S.A. and Schwab, C.: Quadrature for *hp*-Galerkin BEM in $\mathbb{R}^3$, *Numer. Math.* **78** (1997), 211-258.

[38] Sauter, S.A. and Schwab, C.: Semianalytic integration in BEM I: collocation methods, (in preparation).

[39] Schneider, R.: Multiskalen und Wavelet-Matrixkompression (in german), Teubner Publ. Stuttgart (1998).

[40] Schwab, C.: Variable Order Composite Quadrature for singular and nearly singular integrals, Computing **53** (1994) 173-194.

[41] Schwab, C. and Wendland, W.: Kernel properties and representations of boundary integral operators, *Math. Nachr.* **156** (1992), 187-218.

[42] Yamada, Y. and Hayami, K.: A multipole BEM for 2-d elastostatics, Report METR 95-07, Fac. of Engg., University of Tokyo, Bunkyo-ku, Japan.

[43] Wendland, W.L.: On boundary integral equations and applications, Acc. Naz. dei Lincei, Atti dei Convegni Lincei **147**, (1999).