ETH Eidgenössische
Technische Hochschule
Zürich

Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

Two Notes on the Implementation of Wavelet Galerkin Boundary Element Methods

C. Lage and C. Schwab

Note I: A Wavelet-Galerkin Boundary Element Method on Polyhedral Surfaces in $\mathbb{R}^3$

Note II: On the Implementation of a Fully Discrete Multiscale Galerkin BEM

Research Report No. 97-06
May 1997

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

Two Notes on the Implementation of Wavelet Galerkin Boundary Element Methods

Note I: A Wavelet-Galerkin Boundary Element Method on Polyhedral Surfaces in $\mathbb{R}^3$

Note II: On the Implementation of a Fully Discrete Multiscale Galerkin BEM

C. Lage and C. Schwab

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

Research Report No. 97-06 May 1997

Abstract

We report, in two notes, recent progress in the implementation of wavelet-based Galerkin BEM on polyhedra and study the performance.

A WAVELET-GALERKIN BOUNDARY ELEMENT METHOD ON POLYHEDRAL SURFACES IN $\mathbb{R}^3$ [1]

Christian Lage and Christoph Schwab
Seminar für Angewandte Mathematik
ETH-Zürich
Rämistrasse 101
CH-8092 Zürich, Switzerland

SUMMARY

The implementation of a wavelet-based Galerkin discretization of the double layer potential operator on polyhedral surfaces $\Gamma \subset \mathbb{R}^3$ is described. The algorithm generates an approximate stiffness matrix with $O(N(\log N)^2)$ entries in $O(N(\log N)^3)$ operations where N is the number of degrees of freedom on the boundary. The condition number of the compressed stiffness matrix is bounded uniformly with respect to N. A C++ realization of the data structure containing the compressed stiffness matrix is described. It can be set up in $O(N(\log N)^2)$ operations and requires $O(N(\log N)^2)$ memory. Numerical experiments show the asymptotic complexity estimates and convergence rates to be accurate already for moderate N. Problems with $N > 10^5$ were computed in core on a workstation.

PROBLEM FORMULATION

Let $\Omega \subset \mathbb{R}^3$ be a bounded polyhedron with Lipschitz boundary $\Gamma = \partial\Omega$ and let $n(y)$ denote the exterior unit normal vector at $y \in \Gamma$. We consider the Dirichlet Problem

$$\Delta U = 0 \text{ in } \Omega, \quad U|_\Gamma = f. \tag{1}$$

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

We equip $L^2(\Gamma)$ with the inner product

$$\langle u, v\rangle = \int_\Gamma uv ds_x. \tag{2}$$

[1] Presented at the 13th GAMM-Seminar on *Numerical Treatment of Multi-Scale Problems*, Christian-Albrechts-Universität Kiel, January 24th to 26th, 1997 (in press in *Notes on Numerical Fluid Mechanics*, Vieweg, 1997).

The double layer ansatz

$$U(x) = \langle K(x, \cdot), u \rangle \tag{3}$$

where the double layer kernel K is given by

$$K(x, y) = \frac{n(y) \cdot (x - y)}{4\pi \, |x - y|^3}$$

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

$$u \in L^2(\Gamma) : \quad \langle v, Au \rangle = \langle v, f \rangle \quad \forall v \in L^2(\Gamma) \tag{4}$$

for the unknown density $u : \Gamma \to \mathbb{R}$. Here A is the classical double layer potential operator which is defined almost everywhere on Γ by

$$(Au)(x) = -\frac{1}{2} u(x) + \int_\Gamma K(x, y) u(y) ds_y, \quad x \in \Gamma. \tag{5}$$

It is well-known that under our assumptions the operator $A : L^2(\Gamma) \to L^2(\Gamma)$ is bounded, thus the Galerkin equations (4) make sense. Moreover, in the pointwise definition (5), the fraction $1/2$ is to be modified when x is on an edge or a vertex; this, however, does not affect the weak formulation (4).

MULTIWAVELET DISCRETIZATION

We present now a fully discrete wavelet Galerkin discretization of (4) together with its properties. Full proofs for all assertions can be found in [PSS],[PS],[LS]. [R] analyzes also a fully discrete wavelet algorithm for boundary integral equations of the type $Au = f$; there, however, collocation and tensor product wavelets are discussed.

GALERKIN DISCRETIZATION

The boundary Γ is partitioned into N_0 planar, open pieces Γ_j which are affine images of the triangle $\mathcal{U}^0 = \{\, (x_1, x_2) \mid 0 < x_1 < 1, 0 < x_2 < x_1 \,\}$ in $\mathbb{R}^2$, i.e. there exist bijective, linear maps χ_j such that $\Gamma_j = \chi_j(\mathcal{U}^0)$. The partition $\{\Gamma_j\}_{j=1}^{N_0}$ is assumed to be regular, i.e. for $i \neq j$ the set $\overline{\Gamma}_i \cap \overline{\Gamma}_j$ is either empty, a vertex, or an entire edge.

An inner product $(\,\cdot\,,\,\cdot\,)$ equivalent to $\langle\,\cdot\,,\,\cdot\,\rangle$ in $L^2(\Gamma)$ can be defined by

$$(u, v) = \sum_{j=1}^{M} \int_{\mathcal{U}^0} \left(\overline{u}|_{\Gamma_j} \circ \chi_j \right) \left(v|_{\Gamma_j} \circ \chi_j \right) \, dx_2 \, dx_1 \tag{6}$$

For $s > 0$, we consider also the Sobolev spaces $H^s(\Gamma_j)$ of functions with pullback in $H^s(\mathcal{U}^0)$ endowed with the norm $|| \cdot ||_s$. The space of functions $u \in L^2(\Gamma)$ with $u|_{\Gamma_j} \in H^s(\Gamma_j)$ for

$s > 0$ is $\prod_{j=1}^{N_0} H^s(\Gamma_j)$, equipped with the norm

$$\|u\|_s := \left(\sum_{j=1}^{N_0} \|u\|_{H^s(\Gamma_j)}^2 \right)^{1/2} . \tag{7}$$

We define a dense subspace sequence $V^l \subset L^2(\Gamma)$, $l = 0, 1, 2, \dots$ as follows: divide $\mathcal{U}^0$ into 4^l congruent subdomains $\{\mathcal{U}_k^l\}$ by successively halving the sides l times. Then define spaces

$$V^l = \{ \, u \in L^2(\Gamma) \mid \left(u|_{\Gamma_j} \circ \chi_j \right)\Big|_{\mathcal{U}_k^l} = \text{const}, \quad j = 1, \dots, N_0, \; k = 1, \dots, 4^l \, \}.$$

of piecewise constant (with respect to χ_j), discontinuous functions. Obviously, the spaces V^l form a hierarchy, i.e.,

$$V^0 \subset V^1 \subset \cdots \subset V^l \subset V^{l+1} \subset \cdots, \tag{8}$$

$N_l = \dim(V^l) = N_0 4^l$ and the V^l are dense in $L^2(\Gamma)$.

The Galerkin approximation u^L of u is given by

$$u^L \in V^L \qquad \left\langle v, Au^L \right\rangle = \langle v, f \rangle \qquad \forall v \in V^L. \tag{9}$$

We make the fundamental *assumption* that the Galerkin scheme (9) is *stable* in the following sense: For L_0 sufficiently large, there exists $C > 0$ such that

$$\forall L \geq L_0 : \qquad \inf_{0 \neq u^L \in V^L} \sup_{0 \neq v^L \in V^L} \frac{\left\langle v^L, Au^L \right\rangle}{\left\| u^L \right\|_0 \left\| v^L \right\|_0} \geq c > 0 \tag{10}$$

where P_L denotes the L^2-projection

$$P_L \colon L^2(\Gamma) \to V^L, \qquad ((v - P_L v), \varphi) = 0 \quad \forall \varphi \in V^L. \tag{11}$$

The stability (10) implies that for sufficiently large L, the Galerkin solutions u^L of (9) exist and are quasioptimal, i.e.

$$\left\| u - u^L \right\|_0 \leq C \inf_{v \in V^L} \|u - v\|_0 . \tag{12}$$

Remark 1 The stability (10) of the Galerkin discretization (9) follows, for example, from a Gårding inequality in $L^2(\Gamma)$ of the operator A. However, on polyhedral boundaries Γ, such an inequality generally is not valid. Nevertheless, stability of (9) has been shown, at least for certain polyhedra, in $[El]$ provided that V^L is constrained to be zero in an $O(h)$ vicinity of the edges of Γ. It has been observed in numerous numerical experiments, however, that on polyhedra (10) apparently holds even without the zero constraint, although a proof does not seem to be available. We assume (10) in our analysis.

We denote by $\Gamma_I := \chi_j(\mathcal{U}_k^l),\ I \in \mathcal{I}_l,\ l = 0, \ldots, L$ the images of $\mathcal{U}_k^l$ under respective coordinate transformations where the index sets $\mathcal{I}_l$ are defined by

$$\mathcal{I}_l := \left\{(j, l, k) : 1 \leq j \leq N_0, 1 \leq k \leq 4^l\right\}.$$

A basis for V^L is given by $\{\varphi_I : I \in \mathcal{I}_L\}$ with

$$\varphi_I\big|_{\Gamma_{I'}} := \begin{cases} 2^l & \text{if } I = I', \\ 0 & \text{otherwise,} \end{cases} \qquad I, I' \in \mathcal{I}_l,\ l = 0, \ldots, L.$$

Then

$$(\varphi_I, \varphi_{I'}) = \delta_{II'} \quad I, I' \in \mathcal{I}_L$$

and (4) is equivalent to a linear system of equations with a dense stiffness matrix, i.e. it requires $O(N^2)$ memory.

The multiscale Galerkin method is also given by (4), but a multiwavelet basis is used for the representation of the solution and the stiffness matrix of (4). To describe this wavelet discretization, we define a sequence W^l of subspaces as orthogonal complement with respect to $(\,\cdot\,,\,\cdot\,)$ of V^{l-1} in V^l:

$$W^l := \{\, \psi \in V^l \mid (\varphi, \psi) = 0 \quad \forall \varphi \in V^{l-1}\,\},\ l \geq 1. \tag{13}$$

Then $V^{l+1} = V^l \oplus W^{l+1}$ and we obtain the multilevel splitting

$$V^L := W^0 \oplus W^1 \oplus \cdots \oplus W^L \tag{14}$$

where $W^0 := V^0$. Hence every function $u^L \in V^L$ admits a unique decomposition

$$u^L = w^0 + w^1 + \cdots + w^L, \qquad w^l \in W^L, l = 0, \ldots, L. \tag{15}$$

Let $P_{-1} = 0$. Then $w^l = (P_l - P_{l-1})u^L$ in (15).

To obtain an orthonormal basis for W^l we proceed as follows: consider the space $\tilde{W}^1$ of piecewise constant functions on the four subdomains $\mathcal{U}_k^1 \subset \mathcal{U}^0$, $k = 1, \ldots, 4$ which have vanishing mean. We denote by $\tilde{\psi}_1, \tilde{\psi}_2, \tilde{\psi}_3$ an orthogonal basis of $\tilde{W}^1$ with $\left\|\tilde{\psi}_i\right\|_{L^2(\mathcal{U}^0)} = 1$. These multiwavelets are the analogs of the Haar-wavelet in one dimension. For $l \geq 1$ we define the basis functions $\psi_J : \Gamma \to \mathbb{R}$ of W^l as usual by means of the mother wavelets $\tilde{\psi}_\nu$:

$$\psi_{(I,\nu)}\Big|_{\Gamma_{I'}} := \begin{cases} 2^{l-1}\tilde{\psi}_\nu \circ \chi_j^{-1} & \text{if } I = I' \\ 0 & \text{otherwise,} \end{cases} \qquad I, I' \in \mathcal{I}_{l-1},\ \nu = 1, \ldots, 3. \tag{16}$$

For $l = 0$ we use the basis functions

$$\psi_{(I,0)} := \varphi_I,\ I \in \mathcal{I}_0 \tag{17}$$

Then for $l \in \mathbb{N}_0$ an orthonormal basis of W^l is given by the functions $\{\, \psi_J \mid J \in \mathcal{J}_l \,\}$ with the index sets $\mathcal{J}_l$ defined by

$$\mathcal{J}_l := \begin{cases} \{\, (I, \nu) \mid I \in \mathcal{I}_{l-1}, 1 \leq \nu \leq 3 \,\} & \text{for } l \geq 1, \\ \{\, (I, 0) \mid I \in \mathcal{I}_0, \} & \text{for } l = 0. \end{cases} \tag{18}$$

By (14) an orthonormal basis of V^l for $l \in \mathbb{N}_0$ is then given by

$$\{\, \psi_J \mid J \in \mathcal{J}_0 \cup \ldots \cup \mathcal{J}_l \,\}.$$

Accordingly, $\|u\|_{L^2(\Gamma)}$ can be characterized by the multiwavelet expansion coefficients.

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

$$\|u\|^2_{L^2(\Gamma)} \sim \sum_{l=0}^{\infty} \sum_{J\in\mathcal{J}_l} |(u,\psi_J)|^2 \tag{19}$$

where $\sim$ denotes the equivalence of norms.

We use the multiwavelet basis (17), (16) in the Galerkin equations (4). To this end, we write u^L in its wavelet representation

$$u^L = \sum_{l=0}^{L} \sum_{J\in\mathcal{J}_l} u^L_J \psi_J, \qquad u^L_J = \left(u^L, \psi_J\right) \tag{20}$$

and denote by $\vec{u} = (u^L_J)_{J\in\mathcal{J}_0\cup\ldots\cup\mathcal{J}_L}$ the vector of wavelet coefficients of u^L. It is determined by the linear system

$$\sum_{l'=0}^{L} \sum_{J'\in\mathcal{J}_{l'}} \langle\psi_J, A\psi_{J'}\rangle\, u^L_{J'} = \langle\psi_J, f\rangle\,, \quad J \in \mathcal{J}_l, ; l = 0,\ldots,L. \tag{21}$$

We denote the $N_L \times N_L$ stiffness matrix in the wavelet basis by A^L, i.e.,

$$A^L_{JJ'} := \langle\psi_J, A\psi_{J'}\rangle\,, \quad J \in \mathcal{J}_l,\ J' \in \mathcal{J}_{l'},\ l,l' = 0,\ldots,L. \tag{22}$$

Then we can write (21) as

$$A^L\vec{u} = \vec{f} \tag{23}$$

where $\vec{f} = (\langle\psi_J, f\rangle)_{J\in\mathcal{J}_0\cup\ldots\cup\mathcal{J}_L}$. Note that A^L is not symmetric in general. It follows from the stability (10) and the norm equivalence (19) that the condition numbers of the sequence $\{A^L\}$ of matrices are bounded: There exists $\kappa^* \in \mathbb{R}$ such that for all L

$$\mathrm{cond}_2(A^L) \le \kappa^*. \tag{24}$$

COMPRESSION

The wavelet basis $\{\psi_J\}$ defined in (17), (16) has vanishing mean in local coordinates which implies that many of the entries $A^L_{JJ'}$ are smaller than in the standard Galerkin basis (see Lemma 6.1 of $[PSS]$ or $[Sch]$ for details). This allows to neglect most elements $A^L_{JJ'}$ without essentially compromising the asymptotic convergence rate of the scheme.

Theorem 3 *Let $s, \tilde{s} \in [0,1]$ and assume that*

$$1 \ge \alpha \ge \frac{s+1}{2} \quad , \quad 1 \ge \tilde{\alpha} \ge \frac{\tilde{s}+1}{2} \quad . \tag{25}$$

With $a \ge 1$ a parameter to be selected define truncation parameters

$$\delta_{ll'} = \max\{a2^{-L}2^{\alpha(L-l)}2^{\tilde{\alpha}(L-l')}, 2^{-l}, 2^{-l'}\} \tag{26}$$

and the corresponding compressed stiffness matrix $\tilde{A}^L$ by

$$\tilde{A}_{JJ'} := \begin{cases} A_{JJ'} & \text{if } d_{II'} \leq \delta_{ll'}, \\ 0 & \text{otherwise} \end{cases} \tag{27}$$

where $J = (I, \nu)$, $J' = (I', \nu')$ and $d_{II'} := \operatorname{dist}(\Gamma_I, \Gamma_{I'})$ the Euclidean distance of Γ_I and $\Gamma_{I'}$. Solve

$$\tilde{A}^L \vec{\tilde{u}} = \vec{f}. \tag{28}$$

and denote by

$$\tilde{u}^L = \sum_{l=0}^{L} \sum_{J \in \mathcal{J}_l} \tilde{u}_J^L \psi_J \in V^L$$

the corresponding approximate solution. Then the following holds:

1. Denote by $\mathcal{N}(\tilde{A}^L)$ the number of nonzero entries in $\tilde{A}^L$. Then

$$\mathcal{N}(\tilde{A}^L) = \begin{cases} O(N_L (\log N_L)^2) & \text{if } \alpha = \tilde{\alpha} = 1, \\ O(N_L \log N_L) & \text{otherwise.} \end{cases} \tag{29}$$

2. There exists a level L_0 such that for a sufficiently large in (26) the compressed Galerkin scheme is stable, i.e. for $L \geq L_0$ the $\tilde{A}^L$ are nonsingular and uniformly bounded, i.e.

$$\forall L \geq L_0 : \quad \operatorname{cond}(\tilde{A}^L) \leq \kappa. \tag{30}$$

3. Assume that we have for some $0 \leq s \leq 1$ the regularity

$$f \in H^s(\Gamma) \quad \Longrightarrow \quad u \in H^s(\Gamma) \tag{31}$$

Then, for sufficiently large a and L_0 in (26) and (30), respectively, we have

$$\left\| u - \tilde{u}^L \right\|_0 \leq C (\log N_L)^\nu N_L^{-s/2} \left\| u \right\|_s = C h^s \left| \log h \right|^\nu \left\| u \right\|_s \tag{32}$$

where $\nu = 0$ if $0 \leq s < 1$ and $\nu = 3/2$ if $s = 1$.

4. Let $x \in \Omega$ and denote by φ the solution of the adjoint problem $A^* \varphi = K(x, \cdot)$. Assume the regularity $\varphi \in H^{\tilde{s}}(\Gamma)$ for some $0 \leq \tilde{s} \leq 1$ and, as before, that $u \in H^s(\Gamma)$. Compute an approximate solution $\tilde{U}^L(x)$ to the Dirichlet problem (1) by inserting the approximate density $\tilde{u}^L$ into (3).

   Then, for sufficiently large a and L_0 and $\alpha, \tilde{\alpha}$ as in (25) there holds the error estimate

$$\begin{aligned} \left| U(x) - \tilde{U}^L(x) \right| &\leq C_x a^{-2} (\log N_L)^3 N_L^{-(s+\tilde{s})/2} \left\| u \right\|_s \left\| \varphi \right\|_{\tilde{s}} \\ &= C_x' a^{-2} h^{s+\tilde{s}} \left| \log h \right|^3 \left\| u \right\|_s \left\| \varphi \right\|_{\tilde{s}} . \end{aligned} \tag{33}$$

This theorem is a special case of $[PS]$ (see also $[DPS]$,$[Sch]$ and $[PSS]$). Even with the space V^L of piecewise constants, we have essentially the maximal asymptotic convergence rate $O(h^2)$ (this rate will usually not be achieved, however, due to edge- and vertex singularities of u).

Theorem 3 still assumed that the entries

$$\tilde{A}_{JJ'} = -\frac{1}{2}\int_\Gamma \psi_J(x)\psi_{J'}(x)ds_x + \int_\Gamma\int_\Gamma K(x,y)\psi_J(x)\psi_{J'}(y)ds_y ds_x, \tag{34}$$

for $J, J' \in \mathcal{J}_0 \cup \ldots \cup \mathcal{J}_L$ of the compressed stiffness matrix $\tilde{A}^L$ were evaluated exactly. This is not possible, in general, and approximations $\hat{A}_{JJ'} = Q\, K\psi_J\psi_{J'}$ must be used, where $Q_{JJ'}$ is a family of quadrature rules to be specified. In our implementation we used the quadrature scheme that was proposed and analyzed in $[PS]$. We now describe it briefly and present its principal properties (for the sake of brevity, we discuss only the quadruple integral in (34)).

Since each multiwavelet ψ_J is piecewise constant in local coordinates, the integral (34) with $J = (I,\nu)$, $J' = (I',\nu')$ may be assembled from 16 "elementary" integrals over pairs $\Gamma_{I_c} \times \Gamma_{I'_c}$, $(I_c, I'_c) \in \text{child}(I) \times \text{child}(I')$ where the set

$$\text{child}(I) := \{I_c \in \mathcal{I}_{l+1}\colon\ \Gamma_{I_c} \subset \Gamma_I\}, \quad I \in \mathcal{I}_l \tag{35}$$

specifies the indices of the four subdomains of Γ_I. We consider therefore now only the fourfold integrals over such pairs. We distinguish two types of integrals to be evaluated: a) non-singular case $d_{II'} > 0$, b) singular case $d_{II'} = 0$. Since $K(x,y)$ is analytic for $x \neq y$ and the multiwavelets piecewise constant in local coordinates, in case a) the integrands are analytic on $\Gamma_I \times \Gamma_{I'}$. For analytic integrands Gaussian quadrature formulas converge, as it is well-known, exponentially with the rate depending on the size of the integrand's domain of analyticity. The only difficulty that arises is that for $d_{II'} < \max\{\text{diam}\,(\Gamma_I)\,, \text{diam}\,(\Gamma_{I'})\}$ the integrand's domain of analyticity and consequently the rate of exponential convergence of Gaussian quadrature may be very small. This can be prevented by subdivision of the larger panel (see $[S]$,$[PS]$).

Lemma 4 *Let $\gamma > 0$ be a parameter that depends only on the boundary Γ and its parametrization. Assume* $\text{diam}(\Gamma_{I'}) \geq \text{diam}(\Gamma_I)$ *and $d_{II'} > 0$. Then there exists a binary partition $\{\Gamma_{\tilde{I}} : \tilde{I} \in \Lambda(I,I')\}$ of $\Gamma_{I'}$ such that $d_{I\tilde{I}} \geq \gamma^{-1}2^{-\tilde{l}-1}$ for all $\tilde{I} \in \Lambda(I,I')$. Moreover, the number of subelements $\Gamma_{\tilde{I}} \subset \Gamma_{I'}$ is bounded by*

$$|\Lambda(I,I')| \leq C(\gamma)(l - l' + 1). \tag{36}$$

A recursive partition with a stronger subdivison ratio would yield better errors $[S]$, but the binary partition in Lemma 4 is natural in our multiscale context (the data structures for discretization and quadrature coincide).

Next, we apply Gaussian quadrature to each pair $(\Gamma_I, \Gamma_{\tilde{I}})$. For that, the triangular elements are transformed to $(-1,1)^2$ via the degenerate mapping

$$\Phi(\xi) = \begin{pmatrix} \xi_1 \\ \xi_1\xi_2 \end{pmatrix} \tag{37}$$

yielding the transformed kernel

$$K_{II'}(\xi,\xi') := 4\,|\Gamma_I|\,|\Gamma_{I'}|\,\xi_1\,\xi'_1\,K\Big(\chi_j \circ (\kappa^l_k)^{-1} \circ \Phi(\xi), \chi_{j'} \circ (\kappa^{l'}_{k'})^{-1} \circ \Phi(\xi')\Big)\,. \tag{38}$$

In summary, we formulate a variable order, composite quadrature rule (see $[S]$) for the approximation of $A_{JJ'}$:

Lemma 5 *Let $J = (I,\nu) \in \mathcal{J}_l$ and $J' = (I',\nu') \in \mathcal{J}_{l'}$ with $l \geq l'$ and $d_{II'} > 0$.Compute approximations $\hat{A}_{JJ'}$ to the corresponding nonzero entries $\tilde{A}_{JJ'}$ of $\tilde{A}^L$ by the variable order, composite quadrature rule*

$$\begin{aligned} \hat{A}_{JJ'} &= \sum_{\tilde{I}\in\Lambda(I,I')} \sum_{I_c\in\mathrm{child}(I)} \sum_{\tilde{I}_c\in\mathrm{child}(\tilde{I})} \psi_J|_{I_c}\, \psi_{J'}|_{\tilde{I}_c}\; G_\xi^{n_1} G_{\xi'}^{n_2}\; K_{I_c\tilde{I}_c}(\xi,\xi') \\ &=: Q_{II'}\, K\psi_J\psi_{J'} \end{aligned} \tag{39}$$

where the orders of the Gaussian quadrature rules $G_\xi^{n_1}$ and $G_{\xi'}^{n_2}$ satisfy

$$n_i(I,\tilde{I},I') = \frac{1}{2} + \frac{2\log_2\gamma + \log_2(l-l'+1) + 2(2L-l-l') + l' - l + 2}{\log_2\rho_i} \tag{40}$$

with

$$\rho_1 := 1 + \gamma\, 2^{l+1}\, d_{I\tilde{I}}, \qquad \rho_2 := 1 + \gamma\, 2^{\tilde{l}+1}\, d_{I\tilde{I}}. \tag{41}$$

Let the singular integrals be computed exactly. Then the total quadrature work is bounded by $O(N_L(\log N_L)^3)$ kernel evaluations and all assertions of Theorem 3 still hold for the numerically integrated, compressed stiffness matrix.

The singular case $d_{II'} = 0$ is, after possible subdivision of the larger element, reduced to the above case plus a singular integral over two panels of equal size. Here special variable transformations are applied ($[HSa]$,$[PS]$) which render the integrand analytic, yielding also here a consistent quadrature approximation in $O(N_L(\log N_L)^3)$ work (see $[PS]$ and note that integrals over panel pairs on the same side of Γ do not contribute). In summary, we have

Theorem 6 *For the quadrature approximation $\hat{A}^L$ of the compressed stiffness matrix according to Lemma 5, all results of Theorem 3 still hold. The total cost of the fully discrete method is not greater than $O(N_L(\log N_L)^3)$ kernel evaluations.*

IMPLEMENTATION

For the implementation of our method we used the object oriented framework for boundary element methods described in $[L1]$ and $[L2]$. This framework provides the mathematical concepts of boundary elements such as subspaces, dualforms, operators or functions, in terms of classes. Two major extensions of the framework are necessary to cover the demands of our wavelet discretization: a specialization of the base class DualForm endowed with the variable order, composite quadrature rule of the previous Section and a specialization of the base class Operator to assemble and store the compressed stiffness matrix.

For the sake of brevity, we only discuss the algorithmic aspects of the underlying classes and omit their precise definition.

QUADRATURE ALGORITHM

The assembly of the stiffness matrix imposes the evaluation of the element matrices

```
integrate($I$, $\tilde I$, $I'$) {
  if  dist$(\Gamma_I, \Gamma_{\tilde I}) \geq \gamma^{-1}\, 2^{-\tilde l - 1}$  {
    for $(I_c, \tilde I_c) \in$ child$(I) \times$ child$(\tilde I)$ {
      calculate $(E'_{I\tilde I})_{I_c\tilde I_c} := 2^{l+\tilde l+2}\; G_\xi^{n_1(I,\tilde I,I')} G_{\xi'}^{n_2(I,\tilde I,I')}\; K_{I_c\tilde I_c}(\xi,\xi')$
      calculate $(E'_{\tilde I I})_{\tilde I_c I_c} := 2^{l+\tilde l+2}\; G_\xi^{n_2(I,\tilde I,I')} G_{\xi'}^{n_1(I,\tilde I,I')}\; K_{\tilde I_c I_c}(\xi,\xi')$
    }
  } else {
    for  $\tilde I_c \in$ child$(\tilde I)$  $(E'_{I\tilde I_c}, E'_{\tilde I_c I})$ $\leftarrow$ integrate($I$, $\tilde I_c$, $I'$)
    evaluate $E'_{I\tilde I}$, $E'_{\tilde I I}$ using $E'_{I\tilde I_c}$, $E'_{\tilde I_c I}$ with $\tilde I_c \in$ child$(\tilde I)$
  }
  return $(E'_{I\tilde I}, E'_{\tilde I I})$
}
```

Algorithm 1: Composite quadrature rule, $l \geq \tilde l \geq l'$.

$$(E_{II'})_{\nu\nu'} := (\hat A_{(I,\nu)(I',\nu')})_{\nu\nu'}, \quad \nu,\nu' = \begin{cases} 0,\ldots,3 & \text{if } I, I' \in \mathcal{I}_0, \\ 1,\ldots,3 & \text{otherwise} \end{cases}$$

for $I, I' \in \mathcal{I}_0 \cup \ldots \cup \mathcal{I}_L$. We manage this task by calculating the corresponding element matrices related to the standard basis functions φ_I using the variable order, composite quadrature rule (39):

$$(E'_{II'})_{I_c I'_c} := Q_{II'}\, K\varphi_{I_c}\varphi_{I'_c}, \quad I_c \in \text{child}(I),\; I'_c \in \text{child}(I').$$

Afterwards, a transformation $E'_{II'} \mapsto E_{II'}$ is applied. Algorithm 1 illustrates the recursive implementation of the subdivision. The initial call *integrate*(I, I', I') evaluates the matrix $E'_{II'}$ as well as its symmetric counterpart $E'_{I'I}$ simultaneously. In this way the construction of the subdivision which is identical in both cases is fully exploited. In the interests of efficiency temporary matrices generated during the subdivision on higher levels are cached, such that subsequent calls can profit. It turns out that a cache size of $O(N_L)$ together with special calling sequence is sufficient to guarantee an optimal usage of data. Moreover, the size of the cache could be reduced to $O(\log N_L)$ still yielding the same performance as before. But, from the object oriented point of view, this approach weakens the important encapsulation between assembly and quadrature.

ASSEMBLY AND COMPRESSION

According to Theorem 3, the truncation criterion (27) yields a compressed stiffness matrix with $O(N_L(\log N_L)^2)$ nonzero entries. For a practical implementation it is essential,

```
assemble($I,\ I'$) {
  if dist($\Gamma_I, \Gamma_{\tilde{I}}$) $\le \delta_{ll'}$ {
    if $l = l'$ and $(j,k) < (j',k')$ return
    $(E_{II'}, E_{I'I}) \leftarrow (E'_{II'}, E'_{I'I}) \leftarrow$ integrate($I,\ I',\ I'$)
    store $E_{II'}$, $E_{I'I}$ in $\hat{A}^L$ in compressed form
    if $l > l'$
      for $I'_c \in$ child($I'$) assemble($I,\ I'_c$)
  }
  return
}
```

Algorithm 2: Assembly and compression.

however, that the nonzero entries in the compressed stiffness matrix can be localized also in essentially $O(N_L)$ operations, i.e. without scanning the whole N_L^2 entries of the full matrix. This objective is easily achieved if one exploits the tree structure of the index set $\mathcal{I}_0 \cup \ldots \cup \mathcal{I}_L$ implied by the sets child(I) as it is done in the function *assemble()* sketched in Algorithm 2. With this function the compressed stiffness matrix is generated by:

```
for $I \in \mathcal{I}_0 \cup \ldots \cup \mathcal{I}_L$
  for $I' \in \mathcal{I}_0$ assemble($I,\ I'$)
```

Except the generation of element matrices, each task of the recursive function *assemble()* is of order $O(1)$. Hence, the complexity of determining the nonzero entries is proportional to the number of calls of the function which again is proportional to the number of nonzero entries, i.e. $O(N_L(\log N_L)^2)$. This is borne out by our numerical experiments below.

For our compression technique we borrow from the well know encoding algorithm of Ziv, Lempel and Welch described for example in $[Wo]$ which encodes repeated substrings more efficiently. In our case we only encode sequences of zero entries in the stiffness matrix by their length. In order to achieve high compression rates long sequences of zero and nonzero entries, respectively, are desired. Therefore, we arrange the wavelets ψ_J according to the preorder depth first traversal of the domains Γ_I (cf. Algorithm 2) which takes the local relations of the domains into account. Note that the symmetry $\delta_{ll'} = \delta_{l'l}$ of the threshold values is transmitted to the pattern of zero and nonzero entries of the stiffness matrix, such that the simultaneous evaluation of the element matrix $E_{II'}$ and its dual in Algorithm 1 is justified. Moreover, the same pattern of compression can be applied to rows and columns of the stiffness matrix. For a detailed discussion of this topic see $[LS]$.

NUMERICAL EXPERIMENTS

Here we present some preliminary numerical experiments obtained with the described

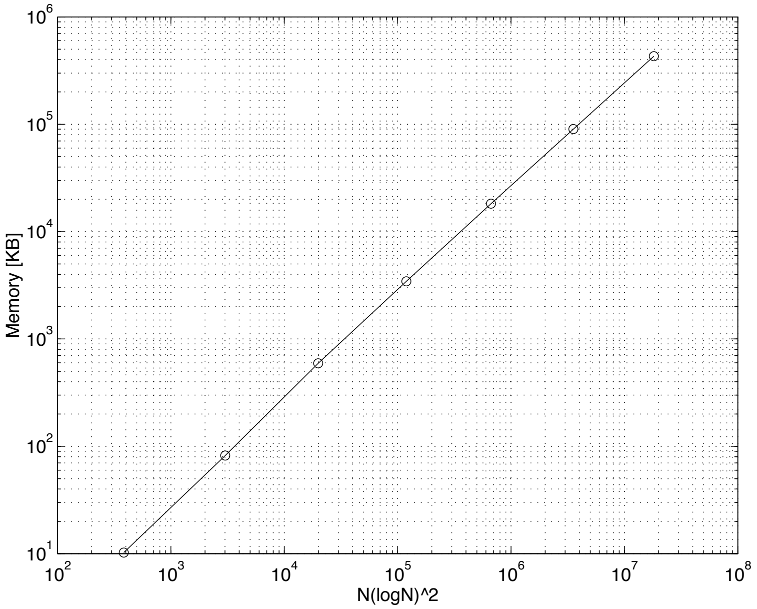

Figure 2: Total memory used for storing the compressed stiffness matrix.

We point out that the CPU-time for the solution accounts only for about 15 per cent of the total CPU-time. 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 matrixassembly. In Figure 5 the error at interior points is depicted. The dashed lines shown illustrate the bounds $O(N^{-1})$ and $O(N^{-1}(\log N)^3)$, respectively (cf. (33)). It can be observed that the expected convergence of essentially $O(h^2)$ is preserved by the compression. The same is true for the convergence of the L^2-norm of the discrete density u^l (Figure 6) showing the behaviour $||u^l||_0 - ||u||_0 \leq Ch^\kappa$ with $\kappa \sim 1.3$.

CONCLUSION

We close the present paper with some remarks. We have realized an $O(N(\log N)^3)$ implementation of a fully discrete Galerkin boundary element method for potential problems in general polyhedral domains. For the problems considered the method as well as the implementation converge at interior points of the domain with essentially $O(h^2)$. The implementation is sufficiently general so that other partial differential equations of mathematical physics, such as Stokes, elasticity or Helmholtz (for moderate wavenumber) can be easily accomodated $[LS]$. Problems with $N > 10^5$ DOF can be computed in core on a serial workstation in less than half an hour (see Figure 3).

For a simplistic complexity comparison with a FEM in Ω, we assume uniform mesh as for the BEM, and linear continuous elements. The number of unknowns in the FEM would then grow like $O(h^{-3})$, but the rate of convergence, using the H^2-regularity in a

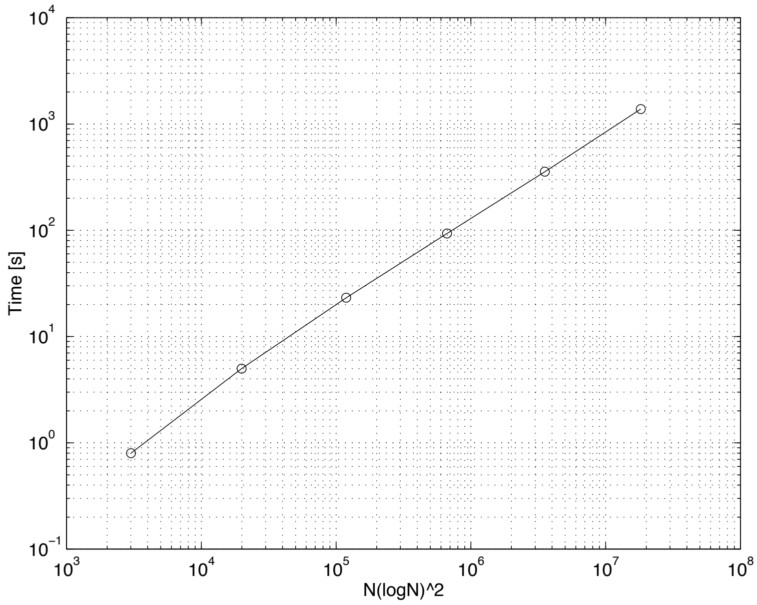

Figure 3: CPU-time for the assembly of the compressed stiffness matrix.

convex polyhedron, would also be essentially $O(h^2)$. Using an optimal $O(N) = O(h^{-3})$ multigrid solver, we would therefore achieve convergence at an interior point of order $O(W^{-2/3})$ in terms of the work measure W whereas our wavelet BEM gives interior point convergence of order $O(W^{-1})$, i.e. the lower dimension of the domain to be discretized pays off if the solution is only desired in a few points.

We finally emphasize that the quadrature- and compression strategies described here apply verbatim also to curved surfaces with analytic parametrizations χ_j (see $[PS]$). Apart from earlier work on collocation schemes (see $[Sch]$, $[R]$), the present work appears to be the first realization of a wavelet-Galerkin boundary element method for general surfaces and second kind equations.

REFERENCES

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

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

$[HSa]$ W. Hackbusch, S.A. Sauter: *On the efficient use of the Galerkin method to solve Fredholm integral equations*, Applications of Mathematics (formerly: aplikace matematiky) **38** 301–322 (1993).

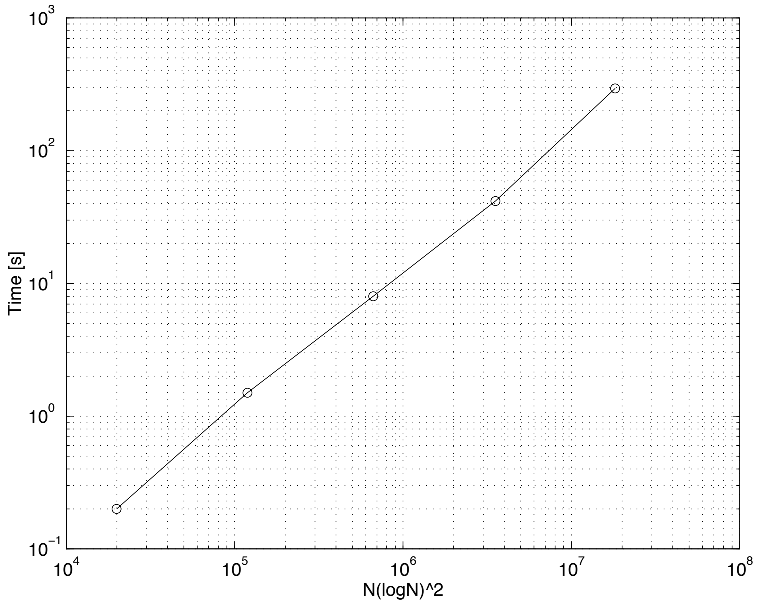

Figure 4: CPU-time for the solution of the linear system.

[$L1$] Ch. Lage: Object-oriented design aspects for boundary element methods, in W. Hackbusch and G. Wittum, editors, *Boundary Elements: Implementation and Analysis of Advanced Algorithms*, volume 54 of *Notes on Numerical Fluid Mechanics*, Vieweg Verlag, Braunschweig, Germany (1996).

[$L2$] Ch. Lage: *The Application of Object Oriented Methods to Boundary Elements*, To appear in Computer Methods in Applied Mechanics and Engineering.

[LS] Ch. Lage and Ch. Schwab: A Galerkin Wavelet Algorithm for Boundary Integral Equations, Preprint, Seminar für Angewandte Mathematik, ETH Zürich (1997).

[PS] T. von Petersdorff and Ch. Schwab: Fully Discrete Multiscale Galerkin BEM, Report 95-08 (September 1995), Seminar für Angewandte Mathematik, ETH Zürich, Switzerland. To appear in the volume "Multiresolution Analysis and Partial Differential Equations", W. Dahmen, P. Kurdila and P. Oswald (Eds.), in the series "Wavelet Analysis and its applications", Academic Press (1997).

[PSS] T. von Petersdorff, R. Schneider and Ch. Schwab: Multiwavelets for Second Kind Integral Equations, Technical Note BN-1153, IPST, Univ. Maryland College Park, August 1994 (in press in SINUM).

[R] A. Rathsfeld: *A wavelet algorithm for the solution of a singular integral equation over a smooth two-dimensional manifold*, Preprint WIAS, Berlin (1996).

[Sch] R. Schneider: *Multiskalen- und Wavelet-Matrixkompression: Analysisbasierte Methoden zur effizienten Lösung grosser vollbesetzter Gleichungssysteme*, Preprint 96-12, Technische Universität Chemnitz-Zwickau (in German).

[S] Ch. Schwab: *Variable Order Composite Quadrature of Singular and Nearly Singular Integrals*, Computing **53**, 173-194 (1994).

[Wo] D. Wood: *Data Structures, Algorithms, and Performance*, Addison-Wesley, Reading/ Massachusetts (1993)

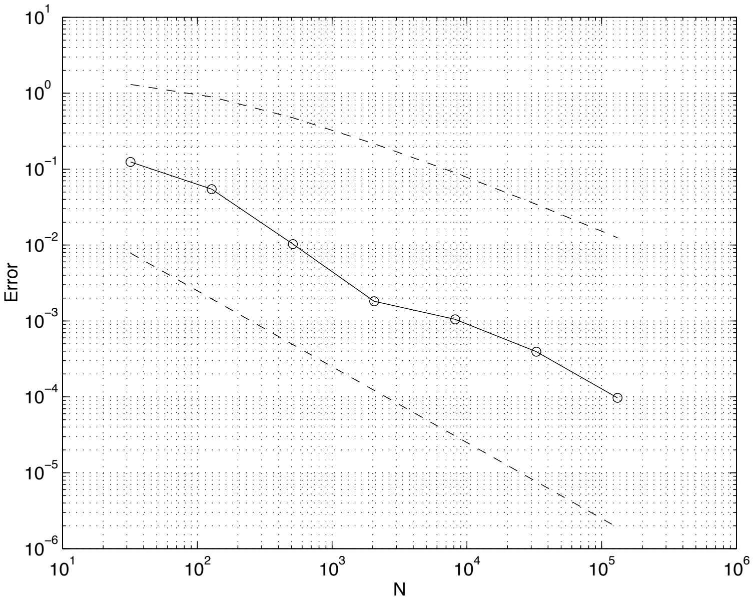

Figure 5: Error at interior points versus N with bounds $O(N^{-1})$ and $O(N^{-1}(\log N)^3)$.

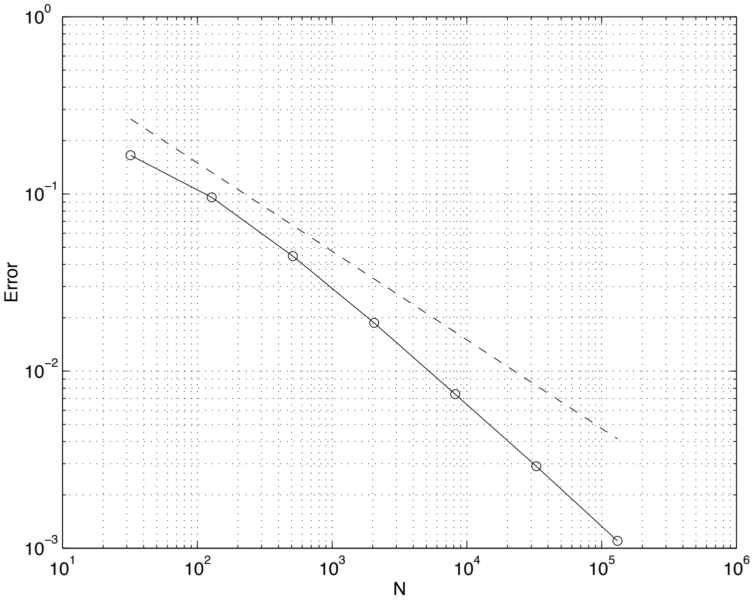

Figure 6: $\|u^l\|_0 - \|u\|_0$ versus N with bound $O(N^{-\frac{1}{2}})$.

implementation of the multiscale scheme. We solved (4) on several polyhedral domains with quite similar performance. For all considered domains one can verify that Theorem 3 holds with $s = \tilde{s} = 1$ [LS]. We report here only those from the prism $\Omega = T \times (0,1)$ where T is an equilateral triangle with sides of length 1. The initial triangulation was the coarsest possible one to cover Γ and contained $N_0 = 8$ triangles. For the right hand side f of (4) we chose the harmonic function $\frac{1}{2}\sin(\frac{\pi}{2}x)\cos(\pi y)\sinh(\frac{\sqrt{5}}{2}\pi z)$.

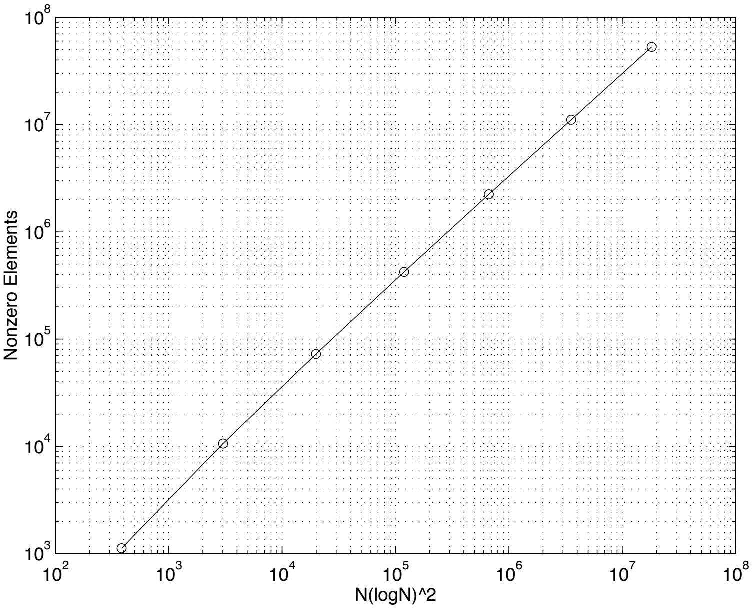

Figure 1: Number of nonzero elements in the compressed stiffness matrix.

In Figure 1, we show the number of nonzero entries in the compressed stiffness matrix – we clearly see the predicted $O(N(\log N)^2)$ behaviour. This, however, is not worth much if the memory overhead used to administer the sparse data structure is excessive. Therefore, in Figure 2 we show total memory used to store the compressed stiffness matrix $\tilde{A}^L$ (i.e. memory used for the nonzero entries of $\tilde{A}^L$ as well as for the supporting data structure). We see also here the predicted $O(N(\log N)^2)$ behaviour. Next, in Figure 3, we show the CPU time used to generate the sparse stiffness matrix $\tilde{A}^L$; this includes the time to locate the nonzero entries as well as the set-up of the sparse data structure. We see here a $O(N(\log N)^2)$ behaviour rendering the estimates of Theorem 6 (numerical quadrature) overly pessimistic. We also note that the total memory required to store the compressed matrix at $L = 7$ and $N_L = 131072$ unknowns amounts to 421MB which just fits in the core of our workstation. At this level, the compression rate is, inclusive the overhead for administering the sparse data structure, $3 \cdot 10^{-3}$. The CPU-time for the iterative solution (GMRes) is shown in Figure 4. Its behaviour consonant to the number of iterations

level	1	2	3	4	5	6	7
iterations	15	19	20	21	22	22	22

validates the bounded condition numbers of the compressed stiffness matrices $\hat{A}^l$ (see item 2 of Theorem 3).

On the Implementation of a Fully Discrete Multiscale Galerkin BEM [1]

Ch. Lage, Ch. Schwab
Seminar für Angewandte Mathematik, ETH-Zürich
CH-8092 Zürich, Switzerland

Abstract

We present a boundary element method on general polyhedral surfaces in space. The shape functions are piecewise constant multiwavelets. The $N \times N$ stiffness matrix is numerically sparse, i.e. only $O(N(\log N)^2)$ entries with a-priori known locations need to be computed. Numerical results for $> 10^5$ unknowns are presented.

1 Model Problem and BIE Formulation

Let $\Omega \subset \mathbb{R}^3$ be a bounded polyhedron with N_0 straight faces Γ_k and boundary $\Gamma = \partial\Omega$. In Ω, consider the problem

$$\Delta U = 0 \text{ in } \Omega, \quad U = f \text{ on } \Gamma. \tag{1}$$

We look for $U(x)$ in the form of a double layer potential, i.e.

$$U(x) = \int\limits_{\Gamma} \frac{n(y)\cdot(x-y)}{4\pi|x-y|^3}\, u(y)\, dS(y) =: \int\limits_{\Gamma} K(x,y)u(y)\, dS(y) \tag{2}$$

with $n(y)$ the exterior unit normal at $y \in \Gamma$, and obtain the BIE

$$-\frac{1}{2}u(x) + \int_{\Gamma} K(x,y)u(y)dS(y) = f(x) \tag{3}$$

for any point $x \in \Gamma$ that is not on an edge or a vertex.

[1] Presented at the BEM 19 conference, Rome, September 9th to 12th, 1997.

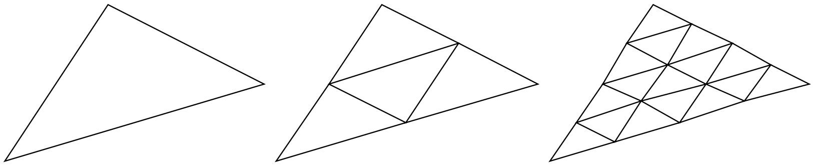

Figure 1: Sequence $\left\{\mathcal{M}_k^l\right\}_{l=0}^2$ of meshes on Γ_k.

2 Galerkin BEM

We assume each side Γ_k of Γ to be triangular (this can always be achieved). We generate a sequence of meshes $\left\{\mathcal{M}_k^l\right\}_{l\le L}$ on Γ_k by successively halving the sides of each triangle $T \in \mathcal{M}_k^l$ (Figure 1). The corresponding mesh on Γ is $\mathcal{M}_l = \bigcup_{k=1}^{N_0} \mathcal{M}_k^l$ with $N_l = N_0 4^l$ triangles T_j^l, $j = 0, \ldots, N_l - 1$, on the surface.

Similar to the classical panel method, we look for an approximate solution

$$u^L = \sum_{j=0}^{N_L-1} \left\{u_\varphi^L\right\}_j \varphi_j^L \tag{4}$$

where φ_j^l is one on triangle $T_j^l \in \mathcal{M}_l$, and zero elsewhere. Inserting (4) into (3) and collocating at the barycenters leads to the panel method. Here we use instead the Galerkin method where we multiply the BIE (3) by test functions φ_i^L and integrate over Γ:

$$\begin{aligned}\int\limits_\Gamma \varphi_i^L(x) \left(-\frac{1}{2} u^L(x) + \int\limits_\Gamma K(x,y) u^L(y)\, dS(y)\right) dS(x) & \\ = \int\limits_\Gamma \varphi_i^L(x) f(x)\, dS(x) &\end{aligned} \tag{5}$$

or, in matrix formulation,

$$\left(-\frac{1}{2}\left[M_\varphi^L\right] + \left[K_\varphi^L\right]\right)\left\{u_\varphi^L\right\} = \left\{f_\varphi^L\right\} \tag{6}$$

where mass- and stiffness matrix are given by

$$\left[M_\varphi^L\right] = \operatorname*{diag}_{0\le i<N_L} \left\{\int\limits_\Gamma \left(\varphi_i^L(x)\right)^2 dS(x)\right\}, \tag{7}$$

$$\left[K_\varphi^L\right] = \left\{\int\limits_\Gamma \varphi_i^L(x) \int\limits_\Gamma K(x,y)\varphi_j^L(y)\, dS(y)\, dS(x)\right\}_{0\le i,j<N_L} \tag{8}$$

and $\left\{u_\varphi^L\right\}$, $\left\{f_\varphi^L\right\}$ denote the solution- and load vector, respectively.

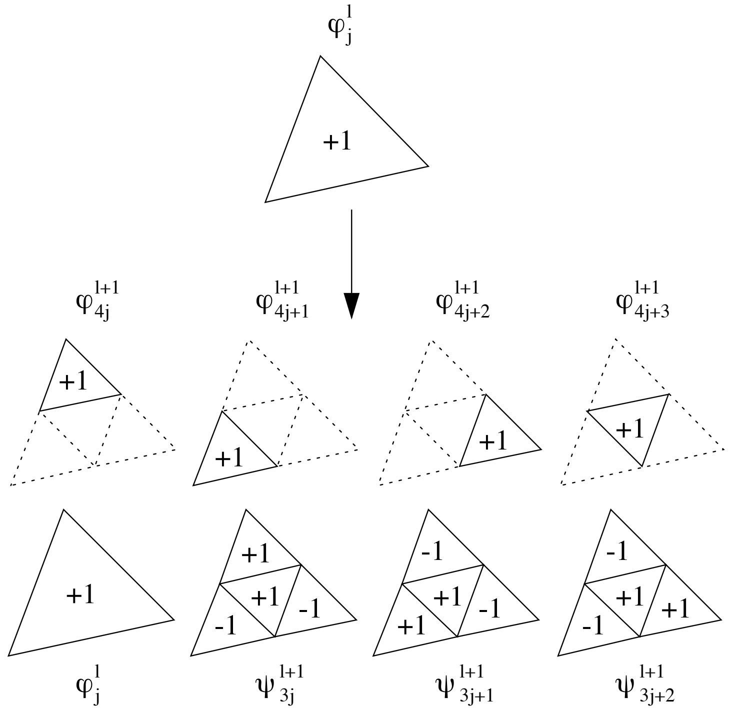

Figure 2: Old and new refinement.

3 Wavelet Basis

The idea is now to pick a set special of shape functions $\left\{\psi_j^l\right\}_{l\leq L}$ for which most entries in the corresponding stiffness matrix can be neglected. Rather than taking all degrees of freedom (DOF) on the finest mesh $\mathcal{M}_L$, we now keep some DOF on the coarser meshes and only add the DOF which give new information (Figure 2). We observe that the new scheme leads to three new shape functions which are discontinuous, piecewise constant and have *vanishing average*. Clearly the new functions are equivalent to the old ones, i.e. using the basis $\left\{\psi_j^l\right\}_{l\leq L}$ instead of $\left\{\varphi_j^L\right\}$ does not change the BEM solution:

$$u^L = \sum_{j=0}^{N_L-1} \left\{u_\varphi^L\right\}_j \varphi_j^L = \sum_{j=0}^{N_0-1} \left\{u_\psi^0\right\}_j \psi_j^0 + \sum_{l=0}^{L-1} \sum_{j=0}^{3N_l-1} \left\{u_\psi^{l+1}\right\}_j \psi_j^{l+1}. \qquad (9)$$

We may easily pass from the coefficient vector $\left\{u_\psi^l\right\}_{l\leq L}$ to $\left\{u_\varphi^L\right\}$ by the pyramid scheme:

$$\begin{tikzcd}
\{u_\varphi^0\} \arrow[r, "{[H_0]}"] & \{u_\varphi^1\} \arrow[r, "{[H_1]}"] & {} & \cdots & {} \arrow[r] & \{u_\varphi^{L-1}\} \arrow[r, "{[H_{L-1}]}"] & \{u_\varphi^L\} \\
\{u_\psi^1\} \arrow[ur, "{[G_0]}"'] & \{u_\psi^2\} \arrow[ur, "{[G_1]}"'] & & & {} \arrow[ur] & \{u_\psi^L\} \arrow[ur, "{[G_{L-1}]}"'] &
\end{tikzcd},$$

i.e. $\left\{u_\varphi^{l+1}\right\} = [H_l]\left\{u_\varphi^l\right\} + [G_l]\left\{u_\psi^{l+1}\right\}$ with block diagonal matrices of the form

$$\begin{aligned}
[H_l] &= \text{blockdiag}\left\{\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}\right\} \in \mathbb{R}^{N_{l+1}\times N_l}, \\
[G_l] &= \text{blockdiag}\left\{\begin{pmatrix} 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \\ 1 & 1 & 1 \end{pmatrix}\right\} \in \mathbb{R}^{N_{l+1}\times 3N_l}.
\end{aligned}$$

The main point in using the wavelet basis is that the stiffness matrix $\left[K_\psi^L\right]$ can be "sparsified": Let $S_j^l = \{x \in \Gamma : \psi_j^l(x) \neq 0\}$ be the support of wavelet ψ_j^l. Then define the sparse matrix $\left[\tilde{K}_\psi^L\right]$ by

$$\left[\tilde{K}_\psi^L\right]_{(l,j)(l',j')} = \begin{cases} \left[K_\psi^L\right]_{(l,j)(l',j')} & \text{if } \operatorname{dist}(S_j^l, S_{j'}^{l'}) \leq a\, 2^{L-l-l'}, \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

and determine the solution $\left\{\tilde{u}_\psi^L\right\}$ of the system

$$\left(-\frac{1}{2}\left[M_\psi^L\right] + \left[\tilde{K}_\psi^L\right]\right)\left\{\tilde{u}_\psi^L\right\} = \left\{f_\psi^L\right\}. \tag{11}$$

Note that $\left[M_\psi^L\right]$ is diagonal due to the $L^2(\Gamma)$ orthogonality of the ψ_j^l. We scale the basis $\left\{\psi_j^l\right\}_{l\leq L}$ such that $\left[M_\psi^L\right]$ is the identity matrix. It is known [5][4][1] that the accuracy of the solution $\left\{\tilde{u}_\psi^L\right\}$ is comparable to that of $\left\{u_\varphi^L\right\}$ in (6), but that the *number of nonzeros* in the compressed matrix $\left[\tilde{K}_\psi^L\right]$ is $\leq CN_L(\log N_L)^2$.

4 Numerical Integration

In general, it is not possible to evaluate the entries of the stiffness matrix exactly such that approximations must be used. For example Gaussian quadrature formulas converge for analytic integrands, as it is well-known, exponentially with a rate depending on the size of the integrand's domain of analyticity. The only difficulty that arises is that for given ψ_j^l, $\psi_{j'}^{l'}$ the domain of analyticity of $K(x,y)\psi_j^l(x)\psi_{j'}^{l'}(y)$ and consequently the rate of convergence of Gaussian quadrature applied to (8) may be very small if $\operatorname{dist}(S_j^l, S_{j'}^{l'}) < \max\left\{\operatorname{diam}(S_j^l), \operatorname{diam}(S_{j'}^{l'})\right\}$ such that the quadrature cannot be performed in logarithmic complexity. This can be prevented by subdivision of the larger panel (Figure 3) using the given sequence of meshes as proposed in [5]. Now the integrals over each combination of the smaller panel with a panel of the subdivision can be handled by means of Gaussian quadrature of variable order depending on the level and the distance

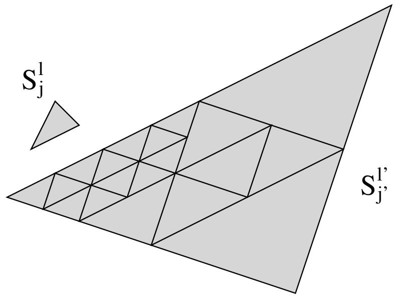

Figure 3: Subdivision of the larger panel.

of the panels. Moreover, due to the subdivision the singular cases, i.e. $\operatorname{dist}(S_j^l, S_{j'}^{l'}) = 0$, are reduced to three basic cases: identical panels, panels sharing a common edge or panels sharing a vertex. For these cases efficient quadrature rules are available [6][3].

It turns out that the total cost of the assembly of the compressed stiffness matrix $\left[\tilde{K}_\psi^L\right]$ is not greater than $O(N_L(\log N_L)^3)$ kernel evaluations [3].

5 Implementation

For the implementation of our method we used the object oriented framework for boundary element methods described in [2]. This framework provides the mathematical concepts of boundary elements such as subspaces, dualforms, operators or functions, in terms of classes. Two major extensions of the framework are necessary to cover the demands of our wavelet discretization: a specialization of the base class DualForm endowed with the variable order, composite quadrature rule of the previous Section and a specialization of the base class Operator to assemble and store the compressed stiffness matrix.

For the sake of brevity, we only discuss the algorithmic aspects of the underlying classes and omit their precise definition.

5.1 Quadrature Algorithm

The assembly of the stiffness matrix imposes the evaluation of the element matrices $E_{TT'}^\psi$, $T, T' \in \bigcup_{l=0}^{L-1} \mathcal{M}_l$. We manage this task by calculating the corresponding 4×4 element matrices related to the standard basis functions $E_{TT'}^\varphi$ using the variable order, composite quadrature rule of the previous Section and applying a transformation $E_{TT'}^\varphi \mapsto E_{TT'}^\psi$ afterwards. Algorithm 1 illustrates the recursive implementation of the subdivision. The initial call *integrate*(T, T') evaluates the matrix $E_{TT'}^\varphi$ as well as its symmetric counter-

```
integrate($T, T'$) {
  let $l, l' \in \mathbb{N}_0$ such that $T \in \mathcal{M}_l$ and $T' \in \mathcal{M}_{l'}$
  if dist($T, T'$) $\geq 2^{-l'}$ or $l = l'$
    evaluate the element matrices $E^{\varphi}_{TT'}$ and $E^{\varphi}_{T'T}$
    using Gaussian quadrature
  else
    for every $\tilde{T} \subset T'$ with $\tilde{T} \in \mathcal{M}_{l'+1}$
      update $E^{\varphi}_{TT'}$, $E^{\varphi}_{T'T}$ with integrate($T, \tilde{T}$)
  return $E^{\varphi}_{TT'}$ and $E^{\varphi}_{T'T}$
}
```

Algorithm 1: Composite quadrature rule, $l \geq l'$.

part $E^{\varphi}_{T'T}$ simultaneously. In this way the construction of the subdivision which is identical in both cases is fully exploited. In the interests of efficiency temporary matrices generated during the subdivision on higher levels are cached, such that subsequent calls can profit. It turns out that a cache size of $O(\log N_L)$ together with a special calling sequence is sufficient to guarantee an optimal usage of data.

5.2 Assembly and Compression

For a practical implementation it is essential, however, that the nonzero entries in the compressed stiffness matrix can be localized also in essentially $O(N_L)$ operations, i.e. without scanning the whole N_L^2 entries of the full matrix. This objective is easily achieved if one exploits the tree structure of the sequence of meshes as it is done in the function *assemble()* sketched in Algorithm 2. With this function the compressed stiffness matrix is generated by:

```
for $T \in \mathcal{M}_0 \cup \ldots \cup \mathcal{M}_{L-1}$
  for $T' \in \mathcal{M}_0$  assemble($T, T'$)
```

with a complexity proportional to the number of nonzero entries, i.e. $O(N_L (\log N_L)^2)$.

For our compression technique we borrow from the well know encoding algorithm of Ziv, Lempel and Welch which encodes repeated substrings. In our case we only encode sequences of zero entries in the stiffness matrix by their length. Note, that the symmetry with respect to l, l' of the threshold value $a\,2^{L-l-l'}$ of the compression implies a symmetric pattern of zero and nonzero entries of the stiffness matrix (Figure 4), such that the simultaneous evaluation of $E^{\psi}_{TT'}$ and $E^{\psi}_{T'T}$ in Algorithm 1 is justified. Moreover, the same

```
assemble($T, T'$) {
  let $l, l' \in \mathbb{N}_0$, $j, j' \in \mathbb{N}_0$ such that $T = T_j^l$ and $T' = T_{j'}^{l'}$
  if dist$(T, T') \le a\ 2^{L-l-l'}$ and ($l > l'$ or $j \ge j'$) {
    $(E_{TT'}^{\psi}, E_{T'T}^{\psi}) \leftarrow (E_{TT'}^{\varphi}, E_{T'T}^{\varphi}) \leftarrow$ integrate($T, T'$)
    store $E_{TT'}^{\psi}$, $E_{T'T}^{\psi}$ in compressed form
    if $l > l'$
      for every $\tilde{T} \subset T'$ with $\tilde{T} \in \mathcal{M}_{l'+1}$
        assemble($T, \tilde{T}$)
  }
  return
}
```

Algorithm 2: Assembly and Compression.

pattern of compression can be applied to rows and columns of the stiffness matrix. For a detailed discussion of this topic see [3].

6 Numerical Results

Here we present some preliminary numerical experiments obtained with the described implementation of the multiscale scheme on a SUN Ultra-Enterprise. We solved (1) on several polyhedral domains with quite similar performance. We report here only those from the prism $\Omega = T \times (0, 1)$ where T is an equilateral triangle with sides of length 1. The initial triangulation was the coarsest possible one to cover Γ and contained $N_0 = 8$ triangles. For the right hand side f of (1) we chose the harmonic function $\frac{1}{2}\sin(\frac{\pi}{2}x)\cos(\pi y)\sinh(\frac{\sqrt{5}}{2}\pi z)$.

In Figure 5, we show the CPU time used to generate the sparse stiffness matrix $\left[\tilde{K}_{\psi}^{L}\right]$; this includes the time to locate the nonzero entries as well as the set-up of the sparse data structure. We see here a $O(N(\log N)^2)$ behaviour showing that the contribution of $O(N(\log N)^3)$ kernel evaluations in the quadrature to the overall complexity is small. We also note that the total memory required to store the compressed matrix at $L = 7$ and $N_L = 131072$ unknowns amounts to 421MB which just fits in the core of our workstation. At this level, the compression rate is, inclusive the overhead for administering the sparse data structure, $3 \cdot 10^{-3}$.

The number of iterations for the iterative solution (GMRes)

level	1	2	3	4	5	6	7
iterations	15	19	20	21	22	22	22

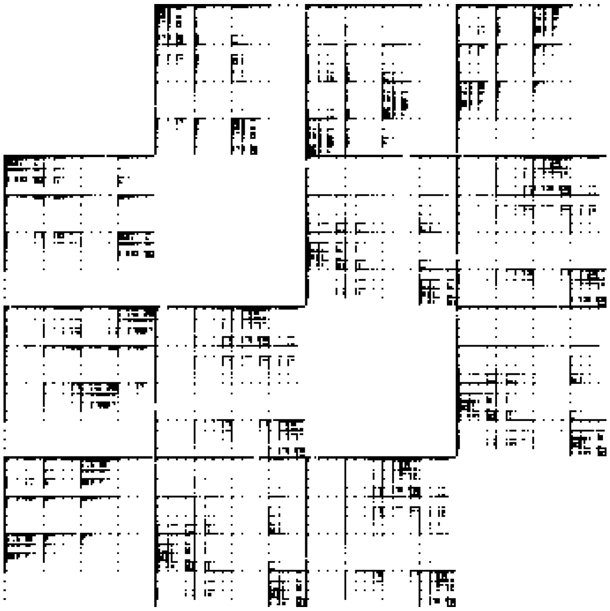

Figure 4: Pattern of zero and nonzero entries of the compressed stiffness matrix, Ω = tetrahedron, $L = 4$. The wavelets ψ_j^l are ordered according to the preorder depth first traversal of the triangles in Algorithm 2.

is independent of the number of levels validating the bounded condition numbers of the compressed stiffness matrices as predicted in [5]. We point out that the CPU-time for the solution accounts only for about 20 per cent of the total CPU-time. 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 matrixassembly.

In Figure 6 the error at interior points is depicted. The dashed line shown illustrates the bound $O(N^{-1})$. It can be observed that the expected convergence of essentially $O(h^2)$ is preserved by the compression. The same is true for the convergence of the L^2-norm of the discrete density u^l (Figure 7) showing the behaviour $||u^l||_0 - ||u||_0 \leq Ch^\kappa$ with $\kappa \sim 1.3$.

References

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

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

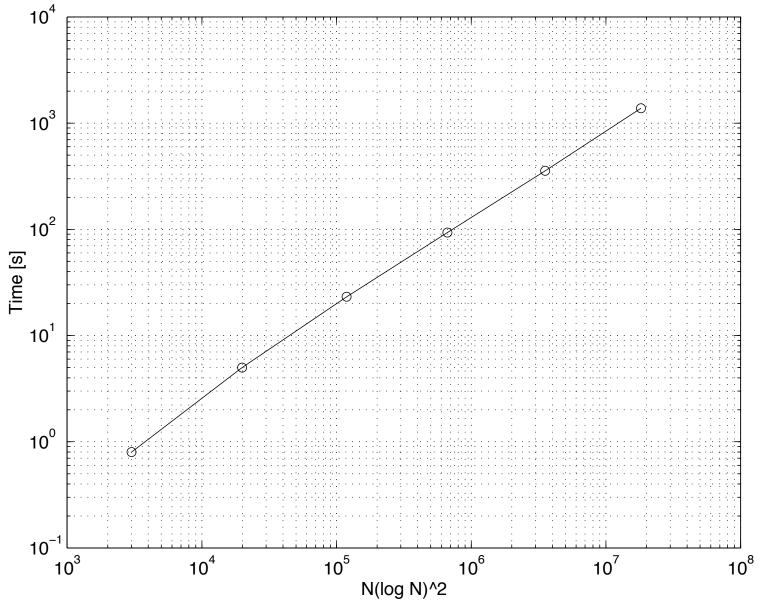

Figure 5: CPU-time for the assembly of the compressed stiffness matrix.

1997. To appear.

[3] Ch. Lage and Ch. Schwab. A galerkin wavelet algorithm for boundary integral equations. Technical report, Seminar für Angewandte Mathematik, ETH Zürich, CH-8092 Zürich, 1997. To appear.

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

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

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

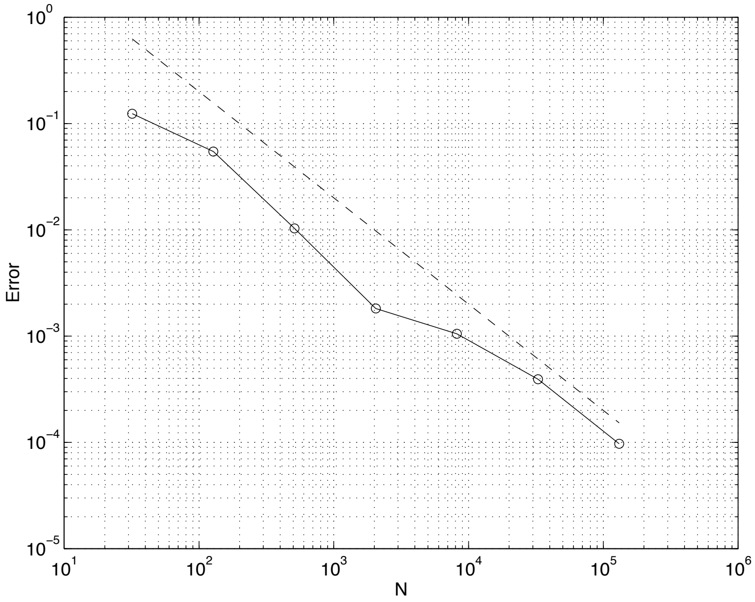

Figure 6: Error at interior points versus N with bound $O(N^{-1})$.

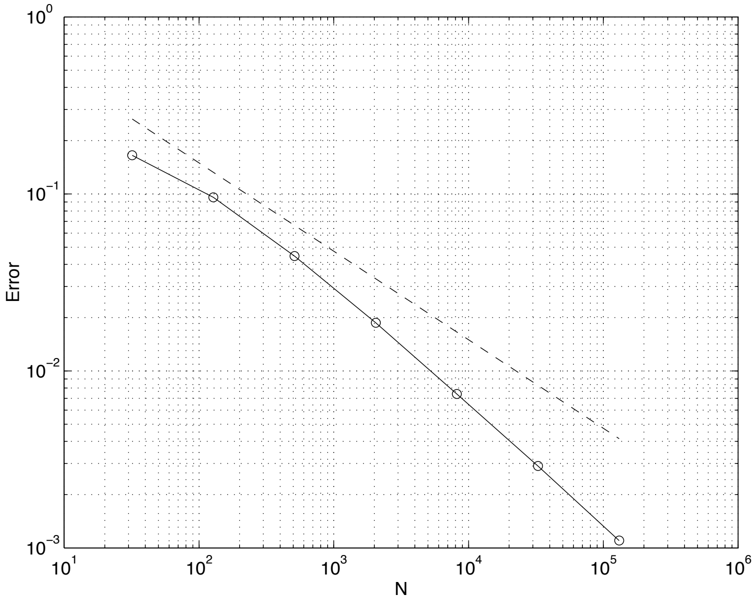

Figure 7: $\|u^l\|_0 - \|u\|_0$ versus N with bound $O(N^{-\frac{1}{2}})$.

Research Reports

No.	Authors	Title
97-06	C. Lage, C. Schwab	Two Notes on the Implementation of Wavelet Galerkin Boundary Element Methods
97-05	J.M. Melenk, C. Schwab	An *hp* Finite Element Method for convection-diffusion problems
97-04	J.M. Melenk, C. Schwab	*hp* FEM for Reaction-Diffusion Equations. II. Regularity Theory
97-03	J.M. Melenk, C. Schwab	*hp* FEM for Reaction-Diffusion Equations. I: Robust Exponentiel Convergence
97-02	D. Schötzau, C. Schwab	Mixed *hp*-FEM on anisotropic meshes
97-01	R. Sperb	Extension of two inequalities of Payne
96-22	R. Bodenmann, A.-T. Morel	Stability analysis for the method of transport
96-21	K. Gerdes	Solution of the $3D$-Helmholtz equation in exterior domains of arbitrary shape using HP-finite infinite elements
96-20	C. Schwab, M. Suri, C. Xenophontos	The *hp* finite element method for problems in mechanics with boundary layers
96-19	C. Lage	The Application of Object Oriented Methods to Boundary Elements
96-18	R. Sperb	An alternative to Ewald sums. Part I: Identities for sums
96-17	M.D. Buhmann, Ch.A. Micchelli, A. Ron	Asymptotically Optimal Approximation and Numerical Solutions of Differential Equations
96-16	M.D. Buhmann, R. Fletcher	M.J.D. Powell's work in univariate and multivariate approximation theory and his contribution to optimization
96-15	W. Gautschi, J. Waldvogel	Contour Plots of Analytic Functions
96-14	R. Resch, F. Stenger, J. Waldvogel	Functional Equations Related to the Iteration of Functions
96-13	H. Forrer	Second Order Accurate Boundary Treatment for Cartesian Grid Methods
96-12	K. Gerdes, C. Schwab	Hierarchic models of Helmholtz problems on thin domains
96-11	K. Gerdes	The conjugated vs. the unconjugated infinite element method for the Helmholtz equation in exterior domains
96-10	J. Waldvogel	Symplectic Integrators for Hill's Lunar Problem
96-09	A.-T. Morel, M. Fey, J. Maurer	Multidimensional High Order Method of Transport for the Shallow Water Equations