Technische Universität Graz

An efficient algebraic multigrid preconditioner for a fast multipole boundary element method

G. Of

Berichte aus dem Institut für Numerische Mathematik

Bericht 2007/8

Technische Universität Graz

An efficient algebraic multigrid preconditioner for a fast multipole boundary element method

G. Of

Berichte aus dem
Institut für Numerische Mathematik

Bericht 2007/8

Technische Universität Graz
Institut für Numerische Mathematik
Steyrergasse 30
A 8010 Graz

WWW: `http://www.numerik.math.tu-graz.at`

An efficient algebraic multigrid preconditioner for a fast multipole boundary element method

G. Of

Institute of Computational Mathematics
Graz University of Technology
Steyrergasse 30,
A-8010 Graz, Austria
of@tugraz.at

Abstract

Fast boundary element methods still need good preconditioning techniques for an almost optimal complexity. An algebraic multigrid method is presented for the single layer potential using the fast multipole method. The coarsening is based on the cluster structure of the fast multipole method. The effort for the construction of the nearfield part of the coarse grid matrices and for an application of the multigrid preconditioner is of the same almost optimal order as the fast multipole method itself.

1 Introduction

The boundary element method is a numerical scheme for the computer simulation of problems described by partial differential equations. The variational formulation of the partial differential equation is transferred into a boundary integral equation by Green's theorem. The discretization of the boundary integral operators results in fully populated matrices in the systems of linear equations. Therefore, the memory requirements and the computational costs grow at least quadratically in the number of boundary elements or degrees of freedom. Hence, standard Galerkin boundary element methods are restricted to problems with a rather small number of degrees of freedom.

There exist several fast boundary element methods reducing the memory requirements and the computational costs of a matrix times vector multiplication to almost linear complexity. Most of these methods rely on a clustering of the boundary elements. This leads to a clustering of the considered matrix, too. Then low rank approximations are used for an appropriate choice of such kind of matrix blocks. The methods mainly differ in the construction and the realization of the low rank approximations. Such methods are the fast multipole method [8], the panel clustering method [11], the adaptive cross approximation (ACA) method [2] and the $\mathcal{H}$–matrix arithmetic [9]. The wavelet approximation methods [5] construct special nested trial spaces which enable a sparse approximation of the matrix due to the rapid decay of the kernel.

In addition to fast discretization techniques, good preconditioners are needed for the efficient iterative solution. Some of the available methods are additive or multiplicative Schwarz methods [12, 13, 17, 28], BPX–like preconditioners [4, 6, 25], multigrid methods [14, 16, 29], boundary integral operators of opposite orders [27] and the application of an approximation of the inverse by means of the $\mathcal{H}$–matrix arithmetics [1, 9].

Coarsening strategies for boundary element spaces and their application in a multigrid preconditioner of the hypersingular operator are presented in [10]. A convergence analysis for a geometrical multigrid method using a data-sparse approximation of the single layer potential by the ACA method is presented in [15]. Weighted distance functions of the midpoints of neighbored boundary elements are used in [14, 16] for the algebraic coarsening and the construction of the coarse grid matrices, while the system matrix itself can be used in the case of algebraic multigrid methods for finite element methods. The coarse grid elements are selected based on the matrices of weighted distance functions and suitable criteria. The corresponding weighted prolongation and restriction operators are applied to the system matrix to compute the coarse grid matrices. In [14, 16], the adaptive cross approximation method is used as fast boundary element method. This method generates a data-sparse approximation of the fully populated boundary element matrices by low rank approximations of admissible blocks of a clustering of the matrices. The coarse grid matrices are computed by the application of the prolongations and restrictions to each block of the cluster matrices. Thereby, the dimensions of these blocks are reduced and sparse coarse grid matrices are computed.

This approach can be applied to the fast multipole method, too. It results in a rather involved implementation and the final procedure will not be very fast, as the number of blocks in the data-sparse approximation is not reduced for the coarse grid matrices. Each admissible block corresponds to a conversion of a fast multipole expansion. As the conversions amount a main part of the total effort of a matrix times vector multiplication of the stiffness matrix, the computational costs for one multigrid step cannot be optimal.

In this paper, the cluster tree of the fast multipole method is used to construct the coarse grid matrices of the multigrid method algebraically. The boundary elements of a cluster form the "coarse grid" elements. Thus, the operations of the fast multipole method can be used to construct the nearfield part of the coarse grid matrices and to realize the corresponding matrix times vector multiplications efficiently. The total memory requirements and the effort of an application of the multigrid preconditioner have the same asymptotic complexity as a matrix times vector product by the fast multipole method. This coarsening strategy can be applied for other fast boundary element methods, too.

In Sect. 2, the boundary element method and the fast multipole method are sketched. An algebraic multigrid method for the fast multipole method is described and an complexity analysis is given in Sect. 3. Finally, the numerical examples of Sect. 4 are in good agreement with the theory.

2 Fast boundary element method

First, the Galerkin discretization of the weakly singular boundary integral equation of the Laplace equation is sketched; for more details see, e.g., [24, 26]. Then some details on the fast multipole boundary element method are given, which are needed for the description of the algebraic multigrid method for the fast multipole method.

2.1 Boundary element formulation

Let $\Omega \subset \mathbb{R}^3$ be a bounded, simply connected domain with piecewise smooth Lipschitz boundary $\Gamma = \partial\Omega$. The Dirichlet boundary value problem

$$\begin{aligned} -\Delta u(x) &= 0 &&\text{for } x \in \Omega, \\ u(x) &= g(x) &&\text{for } x \in \Gamma \end{aligned} \tag{1}$$

of the Laplace equation is considered as model problem. The solution u of the boundary value problem (1) is given by the representation formula

$$u(x) = \int_\Gamma \gamma_{0,y} U^*(x,y) \gamma_1 u(y) ds_y - \int_\Gamma \gamma_{1,y} U^*(x,y) g(y) ds_y \qquad \text{for } x \in \Omega, \tag{2}$$

where

$$U^*(x,y) = \frac{1}{4\pi} \frac{1}{|x-y|} \tag{3}$$

is the fundamental solution of the Laplace operator. γ_0 and γ_1 denote the trace operators

$$\begin{aligned} \gamma_0 u(x) &:= \lim_{\Omega \ni \widetilde{x} \to x \in \Gamma} u(\widetilde{x}) &&\text{for all } x \in \Gamma, \\ \gamma_1 u(x) &:= \lim_{\Omega \ni \widetilde{x} \to x \in \Gamma} [n(x) \cdot \nabla u(\widetilde{x})] &&\text{for almost all } x \in \Gamma, \end{aligned}$$

where $n(x)$ is the exterior normal vector. The unknown flux $t := \gamma_1 u$ can be determined from the weakly singular boundary integral equation

$$\gamma_0 u(x) = (Vt)(x) + \frac{1}{2}\gamma_0 u(x) - (K\gamma_0 u)(x) \tag{4}$$

for almost all $x \in \Gamma$. V denotes the single layer potential

$$(Vt)(x) = \int_\Gamma U^*(x,y) t(y) ds_y,$$

and K is the double layer potential

$$(Ku)(x) = \int_{\Gamma \setminus \{x\}} \gamma_{1,y} U^*(x,y) u(y) ds_y.$$

For the approximation of the unknown Cauchy data t a Galerkin discretization of the weakly singular boundary integral equation (4) is used with a surface representation by

N plane triangles τ_i and piecewise constant test and trial functions φ_i. The variational formulation reads: Find

$$t_h = \sum_{i=1}^{N} t_i \varphi_i \in S_h^0(\Gamma) = \text{span } \{\varphi_i\}_{i=1}^N$$

such that

$$\langle V t_h, w_h \rangle_\Gamma = \langle (\frac{1}{2} I + K) g, w_h \rangle_\Gamma$$

holds for all test functions $w_h \in S_h^0(\Gamma)$. The variational formulation is uniquely solvable due to the Lemma of Lax–Milgram and Cea's Lemma, respectively. The equivalent system of linear equations is

$$V_h \underline{t} = \underline{f} \qquad \text{where} \qquad V_h[i,j] = \langle V \varphi_j, \varphi_i \rangle_\Gamma \quad \text{and} \quad f_i = \langle (\frac{1}{2} I + K) g, \varphi_i \rangle_\Gamma$$

for $i, j = 1, \ldots, N$. The matrix V_h is symmetric and positive definite. Thus the system of linear equations can be solved by a conjugate gradient method.

2.2 Fast multipole boundary element method

The use of the fast multipole method [7, 8] as a fast boundary element method is described only very briefly. An overview on many papers on the fast multipole method is given in [18]. Here, we will describe only the main ideas and not all developed improvements.

As the Galerkin matrix V_h of the single layer potential is fully populated, there is a need to overcome the limits of the corresponding quadratic complexity. The fast multipole method offers a fast matrix times vector multiplication $\underline{w} = V_h \underline{t}$ of order $\mathcal{O}(N \log^2 N)$ in time and memory requirements. This matrix times vector product can be written component-wise by

$$w_\ell = \sum_{k=1}^{N} V_h[\ell, k] t_k = \sum_{k=1}^{N} \frac{t_k}{4\pi} \int_{\tau_\ell} \int_{\tau_k} \frac{1}{|x-y|} ds_y ds_x \tag{5}$$

for $\ell = 1, \ldots, N$. The first idea is to separate the variables in the kernel by a series expansion. In the case of the Laplacian, an expansion based on spherical harmonics is suitable. An approximation of the kernel $k(x,y) = |x-y|^{-1}$ is defined by

$$k_p(x,y) = \sum_{n=0}^{p} \sum_{m=-n}^{n} \overline{S_n^m}(y) R_n^m(x) \qquad \text{for } |y| > |x|, \tag{6}$$

using reformulated spherical harmonics [21, 30, 31]

$$\begin{aligned} R_n^{\pm m}(x) &= \frac{1}{(n+m)!} \frac{d^m}{du^m} P_n(u)\big|_{u=\widehat{x}_3} (\widehat{x}_1 \pm i \widehat{x}_2)^m \, |x|^n \,, \\ S_n^{\pm m}(y) &= (n-m)! \frac{d^m}{du^m} P_n(u)\big|_{u=\widehat{y}_3} (\widehat{y}_1 \pm i \widehat{y}_2)^m \frac{1}{|y|^{n+1}} \end{aligned}$$

and $\widehat{y} = y/|y|$. P_n are the Legendre polynomials

$$P_n(u) = \frac{1}{2^n n!} \frac{d^n}{du^n} (u^2 - 1)^n \qquad \text{for } |u| \leq 1.$$

As the expansion (6) converges only for $|y| > |x|$, the matrix V_h has to be split into a nearfield and farfield part. The expansion (6) can be applied in the farfield $\mathrm{FF}(\ell)$ of a boundary element τ_ℓ, while the matrix entries have to be computed in the nearfield $\mathrm{NF}(\ell)$ classically. The approximation of the matrix times vector product now reads as

$$\widetilde{w}_\ell = \sum_{k \in \mathrm{NF}(\ell)} V_h[\ell, k]\, t_k + \sum_{k \in \mathrm{FF}(\ell)} \frac{t_k}{4\pi} \int_{\tau_\ell} \int_{\tau_k} \sum_{n=0}^{p} \sum_{m=-n}^{n} \overline{S_n^m}(y) R_n^m(x) ds_y ds_x.$$

Using the coefficients

$$M_n^m(O, \ell) = \int_{\tau_\ell} R_n^m(x) ds_x \qquad \text{and} \qquad L_n^m(O, k) = \int_{\tau_k} \overline{S_n^m}(y) ds_y \tag{7}$$

with respect to a local origin O gives the representation

$$\widetilde{w}_\ell = \sum_{k \in \mathrm{NF}(\ell)} V_{L,h}[\ell, k] t_k + \frac{1}{4\pi} \sum_{n=0}^{p} \sum_{m=-n}^{n} M_n^m(O, \ell) \widetilde{L}_n^m(\mathrm{FF}(\ell)). \tag{8}$$

If the coefficients

$$\widetilde{L}_n^m(\mathrm{FF}(\ell)) = \sum_{k \in \mathrm{FF}(\ell)} t_k L_n^m(O, k) \qquad \text{for } n = 0, \ldots, p, m = -n, \ldots, n \tag{9}$$

are computed efficiently, the realization of the matrix times vector multiplication will be fast by the splitting (8). But the coefficients $\widetilde{L}_n^m(\mathrm{FF}(\ell))$ depend on the farfield $\mathrm{FF}(\ell)$ of each boundary element and differ consequently.

For the efficient computation of the coefficients $\widetilde{L}_n^m(\mathrm{FF}(\ell))$ a hierarchical structure build upon the boundary elements is used. All boundary elements $\{\tau_k\}_{k=1}^N$ are located in a cube containing the domain Ω and form the cluster ω_1^0 of level 0. The hierarchical cluster structure is built recursively. The cube identified with a cluster ω_i^λ is subdivided into eight smaller cubes and the triangles of the cluster are assigned to these eight cubes and form up to eight new clusters $\omega_j^{\lambda+1}$ of level $\lambda + 1$ as sons of the cluster ω_i^λ. In the case of a quasi-uniform boundary discretization this construction is executed up to a maximal level L such that the clusters of this level contain only some boundary elements. A cluster ω_i^λ is in the nearfield of a cluster ω_j^λ of the same level λ, if there holds

$$\mathrm{dist}\,\{C_i^\lambda, C_j^\lambda\} \leq (d + 1) \max\,\{r_i^\lambda, r_j^\lambda\}. \tag{10}$$

C_i^λ is the center of the cube associated with the cluster ω_i^λ and r_i^λ is the cluster radius, i.e., $r_i^\lambda = \sup_{x \in \omega_i^\lambda} \left| x - C_i^\lambda \right|$. This definition is transferred to the boundary elements by the leafs of the cluster tree:

$$\begin{aligned}\mathrm{NF}(\ell) &:= \big\{k, 1 \le k \le N \text{ und (10) holds for the cluster } \omega_i^L \text{ of } \tau_k \\ &\quad \text{and } \omega_j^L \text{ is the cluster containing } \tau_\ell.\big\}\,, \\ \mathrm{FF}(\ell) &:= \{1,\ldots,N\}\setminus \mathrm{NF}(\ell).\end{aligned}$$

The cluster hierarchy is now used to compute the coefficients $\widetilde{L}_n^m(\mathrm{FF}(\ell))$. First, the multipole coefficients

$$\widetilde{M}_n^m(C_j^L, \mathrm{P}(\omega_j^L)) = \sum_{k\in \mathrm{P}(\omega_j^L)} t_k M_n^m(C_j^\lambda, k) \tag{11}$$

are computed for all clusters ω_j^L of the finest level L. $\mathrm{P}(\omega_j^\lambda) := \{k, \tau_k \in \omega_j^\lambda\}$ denotes the set of indices of all boundary elements τ_k of the cluster ω_j^λ. The coefficients $\widetilde{M}_n^m$ of the coarser levels $\lambda < L$ are computed by the translation

$$\widetilde{M}_n^m(C_j^\lambda, \mathrm{P}(\omega_j^\lambda)) = \sum_{\omega_i^{\lambda+1}\in \mathrm{sons}(\omega_j^\lambda)} \sum_{s=0}^{n}\sum_{t=-s}^{s} R_s^t(\overrightarrow{C_j^\lambda C_i^{\lambda+1}})\widetilde{M}_{n-s}^{m-t}(C_i^{\lambda+1}, \mathrm{P}(\omega_i^{\lambda+1})). \tag{12}$$

The multipole coefficients of a cluster ω_j^λ are transferred to the center of a cluster ω_i^λ in the farfield by the conversion

$$\widetilde{L}_n^m(C_i^\lambda, \mathrm{P}(\omega_j^\lambda)) = \sum_{s=0}^{\infty}\sum_{t=-s}^{s}(-1)^n \overline{S_{n+s}^{m+t}}(\overrightarrow{C_j^\lambda C_i^\lambda})\widetilde{M}_s^t(C_j^\lambda, \mathrm{P}(\omega_j^\lambda)). \tag{13}$$

These conversions are executed on the coarsest level for which the admissibility condition is satisfied, i.e., for two clusters in their mutual farfield but their father clusters are in their mutual nearfield. The sum of all these transferred coefficients form the local coefficients of each cluster. Further, these coefficients are transferred from each cluster ω_i^λ to its sons $\omega_j^{\lambda+1}$ by the translation

$$\widetilde{L}_n^m(C_j^{\lambda+1}, \mathrm{FF}(\omega_i^\lambda)) = \sum_{s=n}^{p}\sum_{t=-s}^{s} R_{s-n}^{t-m}(\overrightarrow{C_i^\lambda C_j^{\lambda+1}})\widetilde{L}_s^t(C_i^\lambda, \mathrm{FF}(\omega_i^\lambda)). \tag{14}$$

The sums of all these coefficients $\widetilde{L}_n^m(C_j^L, \cdot)$ build the coefficients $\widetilde{L}_n^m(\mathrm{FF}(\ell))$ needed in the matrix times vector product (8), where ω_j^L is the cluster containing τ_ℓ.

An error analysis [20] with a fixed nearfield parameter d shows that the expansion degree p has to be chosen as $p \sim \log N$, if the same expansion degree is used for all cluster levels. This analysis shows that the effort of a matrix times vector product is of order $\mathcal{O}(N \log^2 N)$. The memory requirements are of the same order.

3 Algebraic multigrid method

First, a summary of the used multigrid scheme is given. Then the algebraic version for the fast multipole method is described in detail and a complexity analysis is given.

3.1 The multigrid algorithm

The description of the multigrid preconditioner of the Galerkin matrix V_h of the single layer potential follows the techniques used in [14, 16]. First, the existence of a set of single layer potential matrices V_ℓ for $\ell = \ell_0, \ldots, L+1$, which are related to a nested sequence of trial spaces, is assumed. The algebraic construction of these matrices will be described in Sect. 3.2. A multigrid method based on a V-cycle is used for the preconditioning of the system matrix $V_{L+1} = V_h$. This method is given by Algorithm 1. The index of a vector $\underline{u}_\ell$ denotes the related level ℓ.

Algorithm 1 multigrid method based on a V-cycle

> **MultiGrid** ($\underline{u}_\ell$, $\underline{f}_\ell$, ℓ)
> **if** $\ell = \ell_0$ **then**
>   Compute $\underline{u}_{\ell_0} = (V_{\ell_0})^{-1}\underline{f}_{\ell_0}$ by a coarse grid solver.
> **else**
>   Apply ν_F smoothing steps for $V_\ell \underline{u}_\ell = \underline{f}_\ell$.
>   Compute the defect $\underline{d}_\ell = \underline{f}_\ell - V_\ell \underline{u}_\ell$.
>   Restrict the defect to the next coarser level $\ell - 1$: $\underline{d}_{\ell-1} = P_\ell^\top \underline{d}_\ell$.
>   Set $\underline{u}_{\ell-1} = 0$.
>   Call **MultiGrid** ($\underline{u}_{\ell-1}$, $\underline{d}_{\ell-1}$, $\ell - 1$).
>   Prolongate the correction $\underline{s}_\ell = P_\ell \underline{u}_{\ell-1}$.
>   Update of the solution $\underline{u}_\ell = \underline{u}_\ell - \underline{s}_\ell$.
>   Apply ν_B smoothing steps for $V_\ell \underline{u}_\ell = \underline{f}_\ell$.
> **end if**

A difficulty of the multigrid method arises from the mapping properties of the single layer potential $V : H^{-1/2}(\Gamma) \to H^{1/2}(\Gamma)$. Due to these mapping properties, the high-frequency eigenfunctions belong to the small eigenvalues and not to the large ones like in finite element methods. Therefore a discretization $A_\ell \in \mathbb{R}^{N_\ell \times N_\ell}$ of the Laplace–Beltrami operator is used as smoother [3]. The corresponding smoothing step of Algorithm 1 reads as

$$\underline{u}_\ell = \underline{u}_\ell + \mu_\ell A_\ell(\underline{f}_\ell - V_\ell \underline{u}_\ell)$$

with a damping parameter $0 < \mu_\ell \leq 1/\lambda_{\ell,\max}$. $\lambda_{\ell,\max}$ denotes the largest eigenvalue of the generalized eigenvalue problem

$$V_\ell \underline{w}_\ell = \lambda A_\ell^{-1} \underline{w}_\ell.$$

These maximal eigenvalues can be computed efficiently by a few steps of a modified gradient method [23].

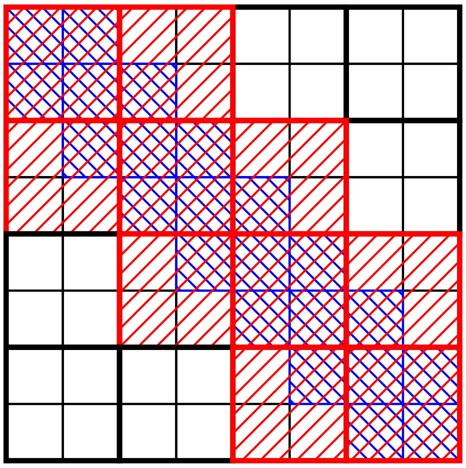
3.2 An AMG method for the fast multipole method

Here, the cluster hierarchy of the fast multipole method is used to define the coarsening. In the case of complicated and not simply connected domains, a suitable clustering resulting in an admissible coarsening might be necessary. The clusters of the finest level merge the contained boundary elements to coarse grid elements. In all further coarsening steps, the clusters are merged in coarse grid elements by their father son relations. The big advantage of this approach is that the operations of the fast multipole method can be used for an efficient construction of the coarse grid nearfield matrices and for efficient realization of the matrix times vector products of the coarse grid matrices.

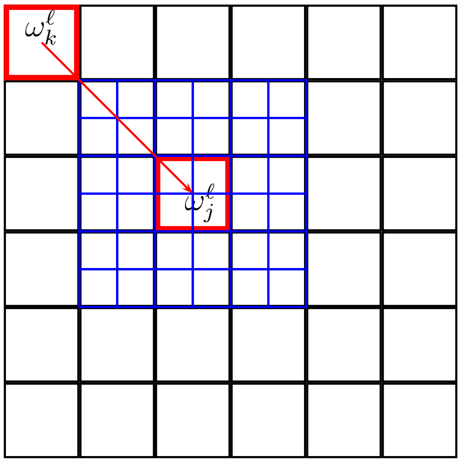
Thus, a coarse grid element τ_j^ℓ of the "discretization" levels $\ell = \ell_0, \ldots, L$ is given by the union of the triangles of the cluster ω_j^ℓ of the cluster tree. On the finest level $L+1$, the given boundary elements $\tau_i^{L+1} := \tau_i$ are used. These boundary elements will be called sons of the clusters ω_j^L for an easier description of the method. The finest cluster level with less than $1000 - 1500$ clusters is used as coarsest level ℓ_0. The corresponding matrix can be set up and inverted by a direct method for this size efficiently. First the nearfield parts of the matrices V_ℓ of the coarser levels $\ell = \ell_0, \ldots, L$ have to be computed. The nearfield is always given by the sons of the nearfield of the father. The matrix entries of the coarser levels can be computed recursively by

$$V_\ell[i,j] = \sum_{m \in \mathrm{sons}(i)} \sum_{n \in \mathrm{sons}(j)} V_{\ell+1}[m,n] \tag{15}$$

for the used piecewise constant test and trial functions. The problem about this computation is that not all entries of the matrix $V_{\ell+1}$ of the finer level, which are needed for the computation of the nearfield entries of the matrix V_ℓ of the coarser level, are known. This is shown in Figure 1a schematically.



(a) Nearfield of the coarse grid matrices



(b) nearfield and interaction

Figure 1: Construction of the nearfield of the coarse grid matrices.

The nearfield entries of the 8×8 matrix of the finer level are hatched from top left to bottom right, while the nearfield entries of the 4×4 matrix of the coarser level are hatched from bottom left to top right. An entry of the coarse grid is given by the sum (15) of the corresponding entries of the finer level. Only some of the nearfield entries of the coarser grid can be computed directly. For the others, some of the fine grid entries are unknown. Each of the unknown matrix entries of the finer level is related to a conversion of a multipole expansion as illustrated in Figure 1b.

Figure 1b represents the nearfield of an element τ_j^ℓ which is identified with the cluster ω_j^ℓ in a two-dimensional clustering. The nearfield of τ_j^ℓ is given by the elements τ_i^ℓ associated with the sons ω_i^ℓ of the clusters which are in the nearfield of the father cluster of ω_j^ℓ. The nearfield entries $V_\ell[j,i]$ of the element τ_j^ℓ can be computed due to (15). But the nearfield entries of $V_{\ell+1}$ are just known for the part which is indicated by the finer clustering. The part of the matrix times vector multiplication of $V_{\ell+1}$ related to ω_j^ℓ and the clusters outside the region marked by the finer clustering is realized by the fast multipole method and not by the nearfield part. The conversion (13) related to two clusters ω_j^ℓ and ω_k^ℓ has to be executed to determine the nearfield entry $V_\ell[j,k]$ of the elements τ_j^ℓ and τ_k^ℓ. One of these conversions is indicated by the plotted arrow.

Due to the recursive computation (15) of the matrix entries, the coefficients $\widehat{M}_n^m(\omega_j^\ell)$ of the multipole expansion of a cluster ω_j^ℓ, which has to be converted, can be computed recursively as the sum of translations (12) of the multipole coefficients of the sons by

$$\widehat{M}_n^m(C_j^\ell, \mathrm{P}(\omega_j^\ell)) = \sum_{\omega_i^{\ell+1} \in \mathrm{sons}(\omega_j^\ell)} \sum_{s=0}^{n} \sum_{t=-s}^{s} R_s^t(\overrightarrow{C_j^\ell C_i^{\ell+1}}) \widehat{M}_{n-s}^{m-t}(C_i^{\ell+1}, \mathrm{P}(\omega_i^{\ell+1})). \tag{16}$$

The multipole coefficients of the clusters ω_i^L of the finest levels are given by

$$\widehat{M}_n^m(C_i^L, \mathrm{P}(\omega_i^L)) = \sum_{k \in \mathrm{P}(\omega_i^L)} M_n^m(C_i^L, k) \tag{17}$$

where $M_n^m(C_i^L, k)$ are the coefficients related to a single panel τ_k. These coefficients $\widehat{M}_n^m$ are also used for the evaluation of the converted expansions. The entry of two elements τ_j^ℓ and τ_k^ℓ is finally computed by

$$V_\ell[j,k] = \sum_{n=0}^{p} \sum_{m=-n}^{n} \widehat{M}_n^m(C_j^\ell, \mathrm{P}(\omega_j^\ell)) \left(\sum_{s=0}^{p} \sum_{t=-s}^{s} (-1)^n \overline{S_{n+s}^{m+t}}(\overrightarrow{C_k^\ell C_j^\ell}) \widehat{M}_s^t(C_k^\ell, \mathrm{P}(\omega_k^\ell)) \right). \tag{18}$$

This formula realizes the computation (15) by the operations of the fast multipole method. The farfield part of the application of a coarse grid matrix V_ℓ is realized by the fast multipole method. There, only the operations down to the cluster level $\ell - 1$ are applied. The coefficients $\widehat{M}_n^m$ are used for setting up the multipole coefficients and for the evaluation of the local expansion, too.

3.3 Complexity estimates

Here, a detailed analysis of the memory requirements and the effort for a single application of the described multigrid preconditioner is presented. The application of the matrices V_ℓ

for $\ell = \ell_0, \dots, L+1$ is decisive for the overall asymptotic effort of the method. As in the complexity analysis in [20], the clustering is built in such a way that the number of boundary elements in the clusters of the finest level is proportional to $\log^2 N$. Therefore, $M_T = \mathcal{O}(N/\log^2 N)$ clusters exist on the finest level. The total number of clusters can be estimated by $2M_T$ due to the tree structure. For a fixed nearfield parameter d the expansion degree p of the multipole approximation has to be chosen as $p \sim \log N$ corresponding to an error analysis [20]. The memory requirements and the complexity of a matrix times vector product are of order $\mathcal{O}(N \log^2 N)$ for V_h on the finest level $L+1$.

The number of nearfield entries of the discretization level $L+1$ is $\mathcal{O}(N \log^2 N)$. The elements of the coarser levels $\ell < L+1$ are defined by the clusters. The number of nearfield elements for a single element τ_i^ℓ on the coarser levels is limited by a constant c_{NF}, as these nearfield entries are determined by the sons of the nearfield of the father cluster $\omega_j^{\ell-1}$. Therefore, the coarsening creates $\mathcal{O}(2M_T\, c_{\mathrm{NF}})$ new nearfield entries at most. Extra memory requirements are caused by the storage of the $\mathcal{O}(p^2)$ coefficients $\widehat{M}_n^m$ for all $2M_T$ clusters. Thus the total memory requirements are of order $\mathcal{O}(N \log^2 N)$.

Operation and comment	costs
Computation (17) of the $\mathcal{O}(p^2)$ coefficients $\widehat{M}_n^m$ for each of the N boundary elements and translation (16) with $\mathcal{O}(p^4)$ to all $2M_T$ father clusters.	$N\mathcal{O}(p^2)$ $+2M_T\mathcal{O}(p^4)$
Each nearfield entry is used for the computation of the nearfield entries of the next coarser level by (15) only once.	$\mathcal{O}(N \log^2 N)$ $+\mathcal{O}(2M_T\, c_{\mathrm{NF}})$
Construction of the remaining nearfield entries by (18) with an $\mathcal{O}(p^4)$ effort. Each of the maximal $2M_T c_I$ conversions is applied only once.	$2M_T c_I \mathcal{O}(p^4)$

Table 1: Complexity analysis for the computation of the coarse grid nearfield matrices.

The effort of the construction of the coarse grid nearfield matrices is listed in Table 1. c_I denotes the maximal number of conversions for a cluster and is bounded since the nearfield parameter d is chosen to be fixed. The total effort for the construction of the nearfield matrices of the coarser grids is of order $\mathcal{O}(N \log^2 N)$ and corresponds almost to a matrix times vector multiplication of the original matrix V_h. Only essentially more evaluations of local expansions are needed. If a direct inversion of the matrix on the coarsest level ℓ_0 is applied, extra conversions in-between the corresponding clusters are needed. Since the dimension N_{ℓ_0} of this matrix is chosen to be fixed with about 1000–1500, the extra memory requirements are constant and the effort for the computation of all these matrix entries is bounded by $N_{\ell_0}^2 \mathcal{O}(p^4)$.

Each of the matrices V_ℓ for $\ell = \ell_0 + 1, \dots, L+1$ is applied twice in the preconditioner of Algorithm 1. This has to be kept in mind for the complexity analysis of an application of the preconditioner in Table 2.

Operation and comment	cost	multigrid	matrix
Set up the $\mathcal{O}(p^2)$ multipole coefficients for each cluster and each boundary element.	$\mathcal{O}(p^2)$	$2 \cdot (2M_T + N)$	N
Translations (12) of the multipole coefficients	$\mathcal{O}(p^4)$	$2 \cdot 4M_T$	$2M_T$
Conversions (13) of the expansions	$\mathcal{O}(p^4)$	$2 \cdot 4M_T \cdot c_I$	$2M_T \cdot c_I$
Translations (14) of the local coefficients	$\mathcal{O}(p^4)$	$2 \cdot 4M_T$	$2M_T$
Evaluation of the local expansions for each cluster and all boundary elements.	$\mathcal{O}(p^2)$	$2 \cdot (2M_T + N)$	N
Nearfield matrix multiplication	$\mathcal{O}(1)$	$2 \cdot (2M_T \cdot c_{NF} + N \log^2 N)$	$N \log^2 N$

Table 2: Complexity analysis for an application of the multigrid preconditioner.

In the first column of Table 2, the operations of the fast multipole method are described. The single costs of these operations are given in the second column. The numbers of applications of these operations are given in the columns three and four for the multigrid preconditioner and the standard matrix times vector multiplication, respectively. The number of translations and conversions is determined by the frequency of the application depending on the cluster level. This is due to the fact that the fast multipole operations have to be executed only down to level $\ell - 1$ for the application of the matrix V_ℓ. The number of clusters of level $L - \ell$ can be estimated by $M_T/2^{L-k}$. Thus, the number of operations can be estimated by the use of the Cauchy product and the geometric series by

$$\sum_{\ell=\ell_0}^{L+1} \sum_{k=0}^{\ell-1} \frac{M_T}{2^{L-k}} \leq \sum_{\ell=1}^{L+1} \sum_{k=0}^{\ell-1} \frac{M_T}{2^{L-k}} = \sum_{i=0}^{L} \frac{M_T}{2^i}(i+1) \leq M_T \sum_{i=0}^{\infty} \frac{i+1}{2^i} = M_T \left(\sum_{i=0}^{\infty} \frac{1}{2^i} \right)^2 = 4M_T.$$

The total effort of an application of the multigrid preconditioner according to Algorithm 1 is of order $\mathcal{O}(N \log^2 N)$ and less expensive than four fast multipole realizations of the matrix V_h. Note that each matrix V_ℓ has to be applied twice, in particular even V_h.

The required accuracy for the approximation of the coarse grid matrices is warranted since the expansion degree p of the finest level which needs the most accuracy is used for all fast multipole operations. A suitable adaption of the expansion degree p for the single coarse grid matrices V_ℓ could reduce the computational times slightly. But even with a fixed expansion degree the overall complexity is optimal with respect to the complexity of a single matrix times vector multiplication.

4 Numerical results

The three-dimensional L-shaped domain shown in Figure 2 is used to compare three preconditioning techniques. The unrefined boundary element mesh consists of 14 squares which are described by 2 triangles each.

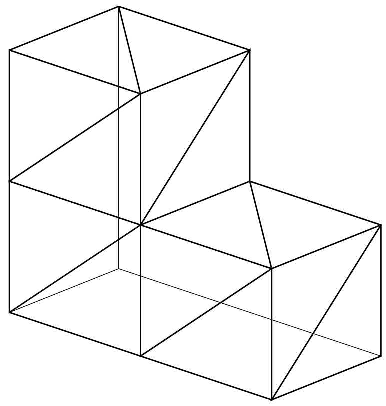


Figure 2: unrefined three-dimensional L-shaped domain.

A simple diagonal scaling, an artificial BPX-like multilevel preconditioner [25] and the described multigrid preconditioner are compared in Table 3 until the seventh refinement level with 458752 triangles. All computations were executed on a personal computer with an AMD Opteron processor 146 2.0 GHz and 4 GB RAM. A conjugate gradient method was used as solver with a relative accuracy of 10^{-8}. The BPX-like preconditioner shows a slightly increasing number It of iterations for a increasing number of boundary elements N as expected from theory. The numbers of iterations are less than those for the diagonal scaling. Therefore, the times t_2 for solving the systems of linear equations are reduced significantly. The algebraic multigrid preconditioner reduces the number of iterations once more. This reduces the computational times for solving the system only slightly, since the BPX-like preconditioner is very inexpensive while one preconditioning step is about four times as expensive as a single matrix times vector product. The extra effort for setting up the multigrid preconditioner can be read off the times t_1 for setting up the system of linear equations and the preconditioning. The extra ten percent effort for setting up the multigrid preconditioner is acceptable. Here, the determination of the damping parameter is more expensive than the construction of the nearfield matrices of the coarse grids. Overall both preconditioning techniques are recommendable.

The same test problem is considered in [15] using a geometrical multigrid preconditioner. The numbers of iterations are comparable considering the differing relative accuracies in the conjugate gradient method of 10^{-6} and 10^{-8}, respectively. The computational times are not really comparable as other integration routines are used; further, the accuracy of the ACA method is not adapted to the discretization level in [15], while this is done in our computations. This adaption of the accuracy of the low rank approximations is needed for the reduction of the approximation error and has a big impact on the performance of the methods [22].

		diagonal scaling			ABPX			AMG		
L	N	t_1	t_2	It	t_1	t_2	It	t_1	t_2	It
0	28	0	0	18	0	0	18	0	0	1
1	112	1	0	26	0	0	25	0	0	9
2	448	4	1	36	5	0	28	5	0	10
3	1792	14	1	48	15	1	32	15	1	10
4	7168	56	6	65	56	3	36	61	3	9
5	28672	193	62	85	192	29	39	212	27	10
6	114688	677	363	113	674	140	43	736	140	11
7	458752	2684	2678	144	2691	902	48	3024	855	11

Table 3: Comparison of preconditioners for the Laplace equation.

The algebraic multigrid preconditioner can also be applied to the Galerkin matrix of the single layer potential in linear elastostatics; for details of the used fast multipole realization see [19]. Here, we apply the multigrid preconditioner of the single layer potential of the Laplacian component-wise for the single layer potential of linear elastostatics. This approach is easier to implement and the construction of the coarse grid matrices is a lot cheaper since the coarsening has to be applied for only one block. The component-wise preconditioning results in a larger condition number but the good numerical results justify this approach.

In Table 4, a diagonal scaling, the BPX preconditioner and the algebraic multigrid method are compared for a Dirichlet problem of linear elastostatics for a cuboid. All preconditioners are applied component-wise.

Six uniform refinement steps have been executed. The finest mesh consists of 163840 triangles and 491520 degrees of freedoms on the boundary and in the system of linear equations. The number of iterations increases strongly with the numbers of elements in the case of diagonal scaling. The number of iterations of the BPX-like preconditioner grows logarithmically as expected from theory. As the BPX-like preconditioner is very cheap, the reduced number of iterations leads to a much faster solving of the system of linear equations. Again, the algebraic multigrid preconditioner reduces the number of iterations significantly. A slight increase of the number of iterations is noticeable as in other publications see [14, 16]. The multigrid preconditioner is rather expensive. Nevertheless, the lower iteration numbers reduce the time for solving the linear system by a factor of five compared to the diagonal scaling and by a factor of about two compared to the BPX-like preconditioner. This justifies the extra effort for setting up the multigrid preconditioner. Here, the multigrid preconditioner is the best choice of the tested preconditioners.

L	N	DoF	diagonal scaling			ABPX			AMG		
			t_1	t_2	It	t_1	t_2	It	t_1	t_2	It
0	40	120	0	0	26	0	0	26	0	0	13
1	160	480	1	2	36	2	1	33	2	1	15
2	640	1920	5	13	51	5	10	40	5	4	16
3	2560	7680	18	93	70	19	58	44	21	25	16
4	10240	30720	75	680	92	76	370	50	88	160	17
5	40960	122880	365	6945	124	368	3080	55	457	1392	19
6	163840	491520	1749	55984	165	1750	20386	60	2325	9481	21

Table 4: Comparison of preconditioners for linear elastostatics

5 Conclusions

We have presented an algebraic multigrid method for the single layer potential using fast boundary element methods. The cluster hierarchy of the fast multipole method is used to define the coarse grid elements. An application of the presented multigrid preconditioner has the same almost optimal order of complexity as a matrix times vector product by the fast multipole method. The numerical examples show the efficiency of the presented preconditioning technique.

Acknowledgement

This work was supported by the German Research Foundation 'Deutsche Forschungsgemeinschaft (DFG)' under the grant SFB 404 'Multifield Problems in Continuum Mechanics', University of Stuttgart.

References

[1] M. Bebendorf. Hierarchical LU decomposition-based preconditioners for BEM. *Computing*, 74(3):225–247, 2005.

[2] M. Bebendorf and S. Rjasanow. Adaptive low-rank approximation of collocation matrices. *Computing*, 70(1):1–24, 2003.

[3] J. H. Bramble, Z. Leyk, and J. E. Pasciak. The analysis of multigrid algorithms for pseudodifferential operators of order minus one. *Math. Comput.*, 63(208):461–478, 1994.

[4] J. H. Bramble, J. E. Pasciak, and J. Xu. Parallel multilevel preconditioners. *Math. Comput.*, 55(191):1–22, 1990.

[5] W. Dahmen, S. Prößdorf, and R. Schneider. Wavelet approximation methods for pseudodifferential equations. II: Matrix compression and fast solution. *Adv. Comput. Math.*, 1(3-4):259–335, 1993.

[6] S. A. Funken and E. P. Stephan. The BPX preconditioner for the single layer potential operator. *Appl. Anal.*, 67(3-4):327–340, 1997.

[7] L. Greengard. *The Rapid Evaluation of Potential Fields in Particle Systems.* The MIT Press, 1987.

[8] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. Comput. Phys.*, 73:325–348, 1987.

[9] W. Hackbusch. A sparse matrix arithmetic based on $\mathcal{H}$-matrices. I: Introduction to $\mathcal{H}$-matrices. *Computing*, 62(2):89–108, 1999.

[10] W. Hackbusch, M. Löhndorf, and S. A. Sauter. Coarsening of boundary-element spaces. *Computing*, 77(3):253–273, 2006.

[11] W. Hackbusch and Z. P. Nowak. On the fast matrix multiplication in the boundary element method by panel clustering. *Numer. Math.*, 54(4):463–491, 1989.

[12] N. Heuer, E. P. Stephan, and T. Tran. Multilevel additive Schwarz method for the *h-p* version of the Galerkin boundary element method. *Math. Comput.*, 67(222):501–518, 1998.

[13] B. N. Khoromskij and W. L. Wendland. Spectrally equivalent preconditioners for boundary equations in substructuring techniques. *East-West J. Numer. Math.*, 1(1):1–26, 1993.

[14] U. Langer and D. Pusch. Data-sparse algebraic multigrid methods for large scale boundary element equations. *Appl. Numer. Math.*, 54(3-4):406–424, 2005.

[15] U. Langer and D. Pusch. Convergence analysis of geometrical multigrid methods for solving data-sparse boundary element equations. *Comput. Vis. Sci.*, 2007. published online.

[16] U. Langer, D. Pusch, and S. Reitzinger. Efficient preconditioners for boundary element matrices based on grey-box algebraic multigrid methods. *Int. J. Numer. Methods Eng.*, 58(13):1937–1953, 2003.

[17] M. Maischak, E. P. Stephan, and T. Tran. Multiplicative Schwarz algorithms for the Galerkin boundary element method. *SIAM J. Numer. Anal.*, 38(4):1243–1268, 2000.

[18] N. Nishimura. Fast multipole accelerated boundary integral equation methods. *Appl. Mech. Rev.*, 55(4):299–324, 2002.

[19] G. Of, O. Steinbach, and W. L. Wendland. Applications of a fast multipole Galerkin boundary element method in linear elastostatics. *Comput. Visual. Sci.*, 8:201–209, 2005.

[20] G. Of, O. Steinbach, and W. L. Wendland. The fast multipole method for the symmetric boundary integral formulation. *IMA J. Numer. Anal.*, 26:272–296, 2006.

[21] J. M. Perez-Jorda and W. Yang. A concise redefinition of the solid spherical harmonics and its use in the fast multipole methods. *J. Chem. Phys.*, 104(20):8003–8006, 1996.

[22] S. Rjasanow and O. Steinbach. *The Fast Solution of Boundary Integral Equations.* Mathematical and Analytical Techniques with Applications to Engineering. Springer, New York, 2007.

[23] A. A. Samarskij and E. S. Nikolaev. *Numerical methods for grid equations. Volume I: Direct methods. Volume II: Iterative methods.* Basel etc.: Birkhäuser Verlag, 1989.

[24] S. Sauter and C. Schwab. *Randelementmethoden. Analyse, Numerik und Implementierung schneller Algorithmen.* Stuttgart: Teubner, 2004.

[25] O. Steinbach. Artificial multilevel boundary element preconditioners. *Proc. Appl. Math. Mech.*, 3:539–542, 2003.

[26] O. Steinbach. *Numerische Näherungsverfahren für elliptische Randwertprobleme. Finite Elemente und Randelemente.* B.G. Teubner, Stuttgart, Leipzig, Wiesbaden, 2003.

[27] O. Steinbach and W. L. Wendland. The construction of some efficient preconditioners in the boundary element method. *Adv. Comput. Math.*, 9(1-2):191–216, 1998.

[28] E. P. Stephan. Multilevel methods for the h-, p-, and hp-versions of the boundary element method. *J. Comput. Appl. Math.*, 125(1-2):503–519, 2000.

[29] T. von Petersdorff and E. Stephan. Multigrid solvers and preconditioners for first kind integral equations. *Numer. Methods Partial Differ. Equations*, 8(5):443–450, 1992.

[30] C. A. White and M. Head-Gordon. Derivation and efficient implementation of the fast multipole method. *J. Chem. Phys.*, 101(8):6593–6605, 1994.

[31] K. Yoshida, N. Nishimura, and S. Kobayashi. Application of fast multipole Galerkin boundary integral equation method to elastostatic crack problems in 3D. *Int. J. Numer. Methods Eng.*, 50(3):525–547, 2001.

Erschienene Preprints ab Nummer 2006/1

2006/1 S. Engleder, O. Steinbach: Modified Boundary Integral Formulations for the Helmholtz Equation.

2006/2 O. Steinbach (ed.): 2nd Austrian Numerical Analysis Day. Book of Abstracts.

2006/3 B. Muth, G. Of, P. Eberhard, O. Steinbach: Collision Detection for Complicated Polyhedra Using the Fast Multipole Method of Ray Crossing.

2006/4 G. Of, B. Schneider: Numerical Tests for the Recovery of the Gravity Field by Fast Boundary Element Methods.

2006/5 U. Langer, O. Steinbach, W. L. Wendland (eds.): 4th Workshop on Fast Boundary Element Methods in Industrial Applications. Book of Abstracts.

2006/6 O. Steinbach (ed.): Jahresbericht 2005/2006.

2006/7 G. Of: The All–floating BETI Method: Numerical Results.

2006/8 P. Urthaler, G. Of, O. Steinbach: Automatische Positionierung von FEM–Netzen.

2006/9 O. Steinbach: Challenges and Applications of Boundary Element Domain Decomposition Methods.

2006/10 S. Engleder: Stabilisierte Randintegralgleichungen für äussere Randwertprobleme der Helmholtz–Gleichung.

2007/1 M. Windisch: Modifizierte Randintegralgleichungen für elektromagnetische Streuprobleme.

2007/2 M. Kaltenbacher, G. Of, O. Steinbach: Fast Multipole Boundary Element Method for Electrostatic Field Computations.

2007/3 G. Of, A. Schwaigkofler, O. Steinbach: Boundary integral equation methods for inverse problems in electrical engineering.

2007/4 S. Engleder, O. Steinbach: Stabilized Boundary Element Methods for Exterior Helmholtz Problems.

2007/5 O. Steinbach, G. Unger: A Boundary Element Method for the Dirichlet Eigenvalue Problem of the Laplace Operator.

2007/6 O. Steinbach, M. Windisch: Modified combined field integral equations for electromagnetic scattering.

2007/7 S. Gemmrich, N. Nigam, O. Steinbach: Boundary Integral Equations for the Laplace–Beltrami Operator.

2007/8 G. Of: An efficient algebraic multigrid preconditioner for a fast multipole boundary element method.