Comparison of Geometrical and Algebraic Multigrid Preconditioners for Data-Sparse Boundary Element Matrices *

U. Langer[1,2] D. Pusch[1]

28th September 2004

[1]Institute of Computational Mathematics
Johannes Kepler University Linz
{ulanger,pusch}@numa.uni-linz.ac.at
[2]Special Research Program SFB F013
'Numerical and Symbolic Scientific Computing'
Johannes Kepler University Linz

Abstract

This paper presents geometrical (GMG) and algebraic multigrid (AMG) preconditioners for data-sparse boundary element matrices arising from the adaptive cross approximation (ACA) to dense boundary element matrices. As model problem, we consider the single layer potential integral equation resulting from the direct boundary integral formulation of the interior Dirichlet problem for the Laplace equation in 3D. The standard implementation of collocation, or Galerkin boundary element discretizations lead to fully populated system matrices which require $O(N_h^2)$ of storage units and causes the same complexity for a single matrix-by-vector multiplication, where N_h denotes the number of boundary unknowns. Data-sparse matrix approximations schemes such as ACA reduce this complexity to an almost linear behavior in N_h. Since the single layer potential operator is a pseudo-differential operator of the order minus one, the resulting boundary element matrices are ill-conditioned. Iterative solvers dramatically suffer from this property for growing N_h. Our multigrid preconditioners avoid the increase of the iteration numbers and result in almost optimal solvers with respect to the total complexity for storage and arithmetical operations required. Numerical experiments are concentrated on the comparison of GMG and AMG preconditioners.

Keywords integral equations of the first kind, boundary element method, adaptive cross approximation, algebraic multigrid, geometrical multigrid, preconditioners, iterative solvers

*This work has been supported by the Austrian Science Fund 'Fonds zur Förderung der wissenschaftlichen Forschung (FWF)' under grant P14953 'Robust Algebraic Multigrid Methods and their Parallelization'.

1 Introduction

In this paper we are concerned with the fast solution of data-sparse boundary element equations by geometrical and algebraic multigrid methods. The most common technique for discretizing elliptic boundary value problems for second order partial differential equations is the finite element method (FEM). Nevertheless, in many applications it will be advantageous to use alternative approaches. The boundary element method (BEM) is certainly a preferable discretization technique for some specific problem classes. One typical application area for BEM is, for instance, the treatment of unbounded domains.

Since only the boundary $\Gamma = \partial\Omega$ of the computational domain $\Omega \subset \mathbb{R}^d$ ($d = 2, 3$) has to be discretized, the dimension of the arising matrices will be reduced essentially. Once the complete Cauchy data (Dirichlet data u and Neumann data $v = \partial u/\partial n$) are available on Γ, the solution in the total computational domain Ω can easily be computed by the representation formulae.

Using standard BEM has one essential drawback. The system matrices are dense, i.e. for a growing number of (boundary) unknowns N_h every iterative solving algorithm will result in a complexity of $O(N_h^2)$ with respect to the arithmetical cost and memory demand. We refer the reader to [7] or [19] for a detailed treatment of boundary element methods.

Therefore, the application of iterative solvers only will be reasonable, if the cost for a single matrix-by-vector multiplication can be reduced essentially (especially in 3D). In the last years different sparse approximation techniques for boundary element matrices have been developed. The multipole method [18, 6], the panel clustering method [10], the $\mathcal{H}$-matrix approach [9] and wavelet techniques [12] are certainly now the most popular ones. In our paper we will consider the adaptive cross approximation (ACA) method recently suggested by M. Bebendorf and S. Rjasanow [3, 2, 4]. The basic idea is to decompose the system matrix into its near-field and far-field contributions. Finding an appropriate low-rank approximation for the far-field matrix yields a data-sparse BEM matrix approximating the original dense matrix in such a way that the discretization error is not affected. In conclusion, the application of a sparse representation algorithm allows us to realize the matrix-by-vector multiplication in almost $O(N_h)$ operations.

It is well-known that iterative solvers heavily suffer from the behavior of the condition number $\kappa(K_h)$ of the system matrices K_h. FE-discretization of boundary value problems for self-adjoint second-order elliptic partial differential equations (PDEs) leads to $\kappa(K_h)$ proportional to $O(h^{-2})$, where h is the typical mesh size. Boundary element matrices originating from the discretization of the single layer potential or the hypersingular operator also lead to ill-conditioned system matrices K_h with a condition number $\kappa(K_h)$ of order $O(h^{-1})$. Thus, it is obvious that we need appropriate preconditioning techniques in order to avoid the steady rise of the number of iterations for finer and finer discretization. In [14, 15, 13] we introduced algebraic multigrid preconditioners for dense BEM matrices as well as for large-scaled data-sparse BEM matrices arising from standard collocation or Galerkin discretizations of the single layer potential and the hypersingular operators. In this paper we focus on the comparison between the

GMG and AMG approach. More precisely, we give some key numerical features concerning the quality of the multigrid preconditioners.

The paper is organized as follows: Section 2 gives a brief overview on the considered single layer potential operator and its properties. In addition, the ACA-method is briefly described. In Section 3, we introduce the multigrid components designed for ACA-matrices. Some results of our numerical studies are presented in Section 4. Finally, we end with some conclusions and discuss further investigations in Section 5.

2 Problem Formulation and the ACA-Method

2.1 Single Layer Potential Equation

Let $\Omega \subset \mathbb{R}^d$ (d=2,3) be a bounded, simply connected domain with one closed boundary piece $\Gamma = \partial\Omega$ that is supposed to be sufficiently smooth. We consider the boundary element technique by means of the interior/exterior Dirichlet problem for Laplace's equation:

$$\begin{aligned} &\textit{Interior Dirichlet}: \\ &\qquad\begin{array}{rcll} -\Delta u(x) &=& 0 & x \in \Omega \\ u(x) &=& g(x) & x \in \Gamma \end{array} \end{aligned} \tag{1}$$

$$\begin{aligned} &\textit{Exterior Dirichlet}: \\ &\qquad\begin{array}{rcll} -\Delta u(x) &=& 0 & x \in \Omega^c = \mathbb{R}^d \backslash \overline{\Omega} \\ u(x) &=& g(x) & x \in \Gamma \\ |u(x)| &=& \mathcal{O}(1/|x|) & |x| \to \infty \end{array} \end{aligned} \tag{2}$$

Once the Neumann and Dirichlet data are available, it is possible to formulate the solution of the interior Dirichlet equation by the representation formula

$$\sigma(y)u(y) = -\int\limits_\Gamma u(x)\frac{\partial E}{\partial n_x}(x,y)ds_x + \int\limits_\Gamma \frac{\partial u}{\partial n_x}(x)E(x,y)ds_x, \quad y \in \Omega \tag{3}$$

where n_x denotes the unit outward normal vector and $E(x,y)$ is the fundamental solution for the Laplace equation, i.e.

$$E(x,y) = \begin{cases} -\frac{1}{2\pi}\log|x-y| & x,y \in \mathbb{R}^2 \\ \frac{1}{4\pi}\frac{1}{|x-y|} & x,y \in \mathbb{R}^3. \end{cases} \tag{4}$$

For $y \in \Omega$ (as mentioned above) we have $\sigma(y) = 1$, for $y \notin \bar{\Omega}$ it changes to $\sigma(y) = 0$. In the case of $y \in \Gamma$ we have $\sigma(y) = \Theta/2\pi$ where Θ denotes the angle enclosed by the domain Ω at the point y. For smooth Γ we will obtain $\sigma = 1/2$.

In order to get a convenient formulation of (3) we introduce the following boundary integral operators. Let us define $v = \partial u/\partial n$ and let $H^{-1/2}(\Gamma)$, $H^{1/2}(\Gamma)$ be the usual Sobolev spaces with fractional indices, see [1]. Then we consider the single layer potential operator $V : H^{-1/2}(\Gamma) \to H^{1/2}(\Gamma)$

$$(Vv)(y) = \int_\Gamma E(x,y)v(x)ds_x \tag{5}$$

and in addition the double layer potential operator $K : H^{1/2}(\Gamma) \to H^{1/2}(\Gamma)$

$$(Ku)(y) = \int_\Gamma \frac{\partial E}{\partial n_x}(x,y)u(x)ds_x. \tag{6}$$

Consequently, on the boundary representation formula (3) gives the equation for calculating the missing Neumann data

$$Vv = f := (\frac{1}{2}I + K)g. \tag{7}$$

Similarly, we obtain the operator equation for the exterior Dirichlet problem for the Laplace equation which reads as

$$Vv = (-\frac{1}{2}I + K)g. \tag{8}$$

Let us now assume that the boundary $\Gamma \in C^{0,1}$ is Lipschitz. Then several properties for the single layer potential operator can be observed (see, e.g., [19]):

1. The operator V is self-adjoint in the $L_2(\Gamma)$ inner product, i.e.

$$\langle v, Vu\rangle_{L_2(\Gamma)} = \langle Vv, u\rangle_{L_2(\Gamma)} \quad \forall u, v \in H^{-1/2}(\Gamma).$$

2. In the case of $\Omega \subset \mathbb{R}^3$ the single layer potential is positive definite, i.e.

$$\exists \mu_V > 0 \quad \langle v, Vv\rangle_{L_2(\Gamma)} \geq \mu_V \|v\|^2_{H^{-1/2}(\Gamma)} \quad \forall v \in H^{-1/2}(\Gamma).$$

   If we consider $\Omega \subset \mathbb{R}^2$ then V will not be positive definite in general. Nevertheless, ellipticity can be obtained if the condition diam $\Omega < 1$ is satisfied. But this condition can easily be fulfilled by an appropriate scaling of the domain Ω.

2.2 Galerkin BEM and Adaptive Cross Approximation (ACA)

The Galerkin discretization of (7) with the use of piecewise constant trial functions $\{\phi_i\}$ leads to the Galerkin equation:
Find $v_h = \sum_{i=1}^{N_h} v_i\phi_i \in X_h := \text{span}\{\phi_1, \ldots, \phi_{N_h}\} \subset H^{-1/2}(\Gamma)$ such that

$$\langle Vv_h, u_h\rangle_{L_2(\Gamma)} = \langle(\frac{1}{2}I + K)g, u_h\rangle_{L_2(\Gamma)} \quad \forall u_h \in X_h \tag{9}$$

The Galerkin equations (9) are equivalent to the system of boundary element equations:
Find $v_h = (v_1, \ldots, v_{N_h})^\top \in \mathbb{R}^{N_h}$ such that

$$V_h \underline{v}_h = \underline{f}_h = (\frac{1}{2}I_h + K_h)\underline{g}_h \tag{10}$$

where $\underline{g}_h$ is the discrete Dirichlet data obtained by linear interpolation, $(V_h)_{ij} = \int_{\Gamma_j}\int_{\Gamma_i} E(x,y)ds_x ds_y$ and $(K_h)_{ij} = \int_{\Gamma_i}\int_\Gamma \frac{\partial E}{\partial n_x}(x,y)\psi_j(x)ds_x ds_y$ with the linear trial function ψ_j. Since the system matrix V_h is symmetric positive definite, we can use the common conjugate gradient (CG) algorithm for solving the problem. At this point we have to notice that V_h is still fully populated and the condition number is of order $O(h^{-1})$. Hence, it is a quite difficult task to solve (10) efficiently. To overcome the drawback of dense matrices we replace the system matrix with some approximation matrix provided from the ACA-algorithm discussed in the next paragraph.

The adaptive cross approximation is a very elegant method to approximate matrices originating from boundary element discretization. On the contrary to other matrix approximation techniques, an explicit description of the integral kernel is not necessary. More precisely, only a procedure for evaluating selected matrix entries has to be available, the rest are simple algebraic operations.

The basic idea is to decompose the computational domain into smaller clusters $\mathcal{D}_i$ and classify the interaction of two clusters into a near-field part and a far-field part of the generated matrix, respectively. Furthermore, the far-field submatrices can be approximated by a sum of dyadic products of the vectors u_i, v_i, i.e. one matrix block $A \in \mathbb{R}^n_m$ has the following representation of rank-1 corrections

$$A = \sum_{i=1}^{r} u_i v_i^\top \qquad u_i \in \mathbb{R}^n,\ v_i \in \mathbb{R}^m,\ r \ldots \text{rank of } A. \tag{11}$$

Thus, we take a closer look to the domain decomposition and the approximation of the resulting system matrix. The notations connected with the ACA are directly adopted from [3, 2].

Based on geometrical information we split the index set $I = \{1, ..., N_h\}$ into index clusters $t_i \subset I$ which corresponds to the partitioning of the domain $\Omega = \bigcup_i \mathcal{D}_i$. In order to select the blocks which can be approximated by low-rank matrices, we give an admissibility condition that classifies clusters-pairs into a near-field part and a far-field part. Roughly speaking, if the distance of a cluster-pair $(\mathcal{D}_1, \mathcal{D}_2)$ is large enough, it will be counted to the far-field. Otherwise, the effect of the singularity in the kernel cannot be neglected and therefore, one has to calculate the matrix entries directly (near-field).

Definition 2.1. *Let $(\mathcal{D}_1, \mathcal{D}_2)$ be a cluster pair with $\mathcal{D}_1, \mathcal{D}_2 \subset \mathbb{R}^d$, then $(\mathcal{D}_1, \mathcal{D}_2)$ is called η - admissible if*

$$\operatorname{diam} \mathcal{D}_2 \leq \eta \operatorname{dist}(\mathcal{D}_1, \mathcal{D}_2). \tag{12}$$

In addition, we define η-admissibility for index cluster pairs (t_1, t_2)

$$\operatorname{diam} \mathcal{D}_h^{t_1} \leq \eta \operatorname{dist}(\mathcal{D}_h^{t_1}, \mathcal{D}_h^{t_2}). \tag{13}$$

As usually $\operatorname{dist}(X, Y) = \inf\{|x - y|, x \in X, y \in Y\}$.

Assuming that a block matrix $A = (a_{ij})_{i=1,\ldots,n,\ j=1,\ldots,m}$ satisfies condition (12) then only few matrix entries have to be calculated and thus the cost of storage and CPU-time for matrix-by-vector multiplication will decrease essentially. If a cluster pair does not fulfill η-admissibility the according matrix will be calculated directly. The following variant of the ACA algorithm is called partially pivoted ACA. It is obvious that only those entries of the vectors u_i and v_i have to be calculated, which are actually used for the construction of the approximated matrix. In [4] we find the following Algorithm 1 and more detailed information.

Algorithm 1 Partially Pivoted ACA

$\mathcal{I} = \emptyset$
for all $k = 1, 2, \ldots$ **do**
  let $i_k \in \{1, \ldots, m\}$, if possible $i_k = \text{argmax}_{i=1,\ldots,n} |(u_k)_i|$, $i_k \notin \mathcal{I}$
  with $a_{i_k j} - \sum_{l=1}^{k-1} (u_l)_{i_k} (v_l)_j \neq 0, \quad j = 1, \ldots, m$
  if i_k does not exist **then**
    stop the algorithm
  else
    $\mathcal{I} = \mathcal{I} \cup \{i_k\}$
    $(\tilde{v}_k)_j = a_{i_k j} - \sum_{l=1}^{k-1} (u_l)_{i_k} (v_l)_j, \quad j = 1, \ldots, m$
    set $v_k = (\tilde{v}_k)_{j_k}^{-1} \tilde{v}_k$ with $j_k = \text{argmax}_{j=1,\ldots,m} |(\tilde{v}_k)_j|$
    $(u_k)_i = a_{i j_k} - \sum_{l=1}^{k-1} (u_l)_i (v_l)_{j_k}, \quad i = 1, \ldots, n.$
  end if
end for

A proper stopping criterion in practice is $\|u_k\|_F \|v_k\|_F < \epsilon \|S_k\|_F$, where $\|.\|_F$ denotes the Frobenius norm, u_k, v_k the calculated ACA vectors and S_k the approximation after the k^{th} step, cf. (11).

Both, the clustering procedure and the approximation algorithm will cause a overall complexity of $O(\epsilon^{-\alpha} N_h^{1+\alpha})$ with an arbitrarily small positive α.

Based on the splitting of the system matrix V_h into a near-field matrix V_h^{near} and a far-field matrix V_h^{far} we are able to construct an approximated system matrix $\widetilde{V}_h$. Since the proposed adaptive cross approximation technique provides a low-rank approximation of V_h^{far} consisting of submatrices which are η-admissible we obtain the result

$$\widetilde{V}_h = V_h^{near} + \widetilde{V}_h^{far}. \tag{14}$$

Starting from this representation we are able to present an appropriate construction of multigrid methods in the next section. Finally, we refer to [3, 2, 4] for more detailed proofs and further remarks concerning the ACA-technique.

3 Multigrid Methods

In the previous section we showed, that our system matrix coincides with the approximated discretized single layer potential operator $\widetilde{V}_h$, which is the most interesting case concerning our multigrid approach. Hence, we have to solve

$$\widetilde{V}_h \underline{v}_h = \underline{f}_h \quad \text{in } \mathbb{R}^{N_h} \tag{15}$$

with $\underline{v}_h$ are the unknown Neumann data and $\underline{f}_h$ the corresponding load vector. In order to make multigrid methods really efficient, it is necessary to adapt the multigrid components properly according to the underlying physical problem and variational formulation. In the following we are discussing the multigrid components by means of a twogrid algorithm. The indices h and H denote the fine grid and coarse grid quantities, respectively.

In fact, the efficiency of multigrid methods depends on a clever interaction of smoothing sweeps on the fine level and coarse grid correction on the coarse level. Once a grid hierarchy (GMG) or a matrix hierarchy (AMG) is available we can apply multigrid methods. An simple two-grid procedure consists of one smoothing step on the fine level. After restricting the defect on the coarse level and solving the defect equation with some coarse grid solver (often a direct solver), the update will be prolongated and added to the approximation of the solution. A recursive application results in the well-known V-cycle, which is presented in Algorithm 2. The coarsest level is denoted by the variable COARSELEVEL therein.

Algorithm 2 Multigrid V-Cycle

MG($\underline{u}_\ell, \underline{f}_\ell, \ell$)
if $\ell =$ COARSELEVEL **then**
  calculate $\underline{u}_\ell = (K_\ell)^{-1}\underline{f}_\ell$ by some coarse grid solver
else
  smooth ν_F times on $K_\ell \underline{u}_\ell = \underline{f}_\ell$
  calculate the defect $\underline{d}_\ell = \underline{f}_\ell - K_\ell \underline{u}_\ell$
  restrict the defect to the next coarser level $\ell + 1 : \underline{d}_{\ell+1} = P_\ell^\top \underline{d}_\ell$
  set $\underline{u}_{\ell+1} \equiv 0$
  call MG($\underline{u}_{\ell+1}, \underline{d}_{\ell+1}, \ell + 1$)
  prolongate the correction $\underline{s}_\ell = P_\ell \underline{u}_{\ell+1}$
  update the solution $\underline{u}_\ell = \underline{u}_\ell + \underline{s}_\ell$
  smooth ν_B times on $K_\ell \underline{u}_\ell = \underline{f}_\ell$
end if

Since the single layer operator represents a pseudo-differential operator of order minus one, the eigenvalues and eigenvectors act conversely compared to those of finite element matrices. Therefore, standard smoothing procedures like damped Jacobi or Gauß-Seidel does not provide a satisfying smoothing sweep. Bramble, Leyk and Pasciak [5] present an appropriate approach to this problem class of operators. In order to reduce the highly oscillating components of the error we introduce a matrix $A_h \in \mathbb{R}^{N_h \times N_h}$ being some discretization of the Laplace-Beltrami operator on the boundary Γ. Consequently, we obtain a smoothing iteration of the form

$$\underline{u}_h \leftarrow \underline{u}_h + \tau_h \cdot A_h(\underline{f}_h - \widetilde{V}_h \underline{u}_h) \tag{16}$$

with a well chosen damping parameter τ_h, see e.g. [15, 13].

In the case of algebraic multigrid we need a matrix hierarchy which represents a 'virtual' grid on each level. Therefore, we first construct prolongation operators $P_h : \mathbb{R}_H^N \mapsto \mathbb{R}_h^N$ by exploiting a sparse auxiliary matrix B_h which includes geometrical information [14]. Then, we are applying Galerkin's method

$$V_H = P_h^\top V_h P_h \tag{17}$$

to obtain the system matrix V_H on the coarse level. In addition, the restriction of a fine ACA matrix $\tilde{V}_h$ immediately leads to matrices on the coarse level

$$\begin{aligned} V_H^{near} &= P_h^\top V_h^{near} P_h, \\ \widetilde{V}_H^{far} &= \sum_{i=1}^{N_B} \sum_{j=1}^{r_i} P_h^\top u_j^i \; (P_h^\top v_j^i)^\top \end{aligned} \tag{18}$$

where N_B denotes the number of admissible blocks and r_i the rank of the i^{th} block. Thus, the approximated coarse system matrix $\widetilde{V}_H$ has also a near-field contribution V_H^{near} and an low-rank far-field matrix $\widetilde{V}_H^{far}$. Due to the exact preserving of representation (14) on the coarse grid, we are able to use the same ACA-datastructures in our numerical realization.

On the other hand in the geometrical version of our multigrid approach a nested mesh-hierarchy is available. In this case we are calculating the discretized single layer potential on each grid separately. Strictly speaking, we apply the ACA-algorithm level by level to obtain the approximated single layer potential operators. Again we provide a set of data-sparse system matrices, which are used within the V-cycle.

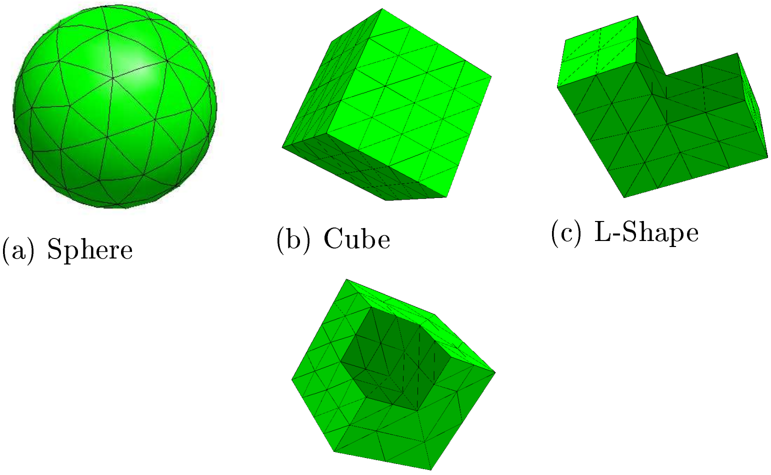
Finally let us remark, that for the coarse grid correction a direct solver is used. That implies to eliminate the data-sparse representation of the coarse system matrix. Therefore, the low-rank approximation will be evaluated on each position of the far-field matrix entries.

4 Numerical Studies

In order to show the efficiency of the suggested multigrid approach we present some results in 3D for the interior Dirichlet boundary value problem for the Laplace equation. The Galerkin boundary element matrices are generated by the software package OSTBEM developed by O. Steinbach, cf. [20], the AMG-preconditioner is realized within the software package PEBBLES [11]. Let us mention, that the multigrid algorithm is often used as a preconditioner in an iterative solver (e.g. PCG). This variant turned out as the most efficient one. In particular, we use the symmetric $V(1,1)$-cycle, i.e. one pre-smoothing and one post-smoothing step with the BLP-smoother per CG-cycle. The iteration error is measured in the $V_h C_h^{-1} V_h$ energy norm for the CG-solver. Finally, the algorithm will stop, if the iteration error is reduced by the factor 10E-6. All calculations are done on a PC with 2000 MHz AMD Athlon(tm) processor.

For our numerical comparison of the geometrical multigrid preconditioner and the algebraic multigrid preconditioner, we choose a few rather simple 3D geometries, see Figure 1. Nevertheless, these domains include a wide spectrum of problem classes, for e.g. smooth boundaries, edges, corners, convex and non-convex domains.

First of all, we compare the times for constructing the AMG matrix hierarchy by Galerkin projection and building up the ACA matrices for GMG on the coarser grids. These CPU-times are almost of the same order, see Table 1. It is obvious, that most of the assembling time is needed for construction the system matrix $\widetilde{V}_h$.



(a) Sphere (b) Cube (c) L-Shape

(d) Fichera-Corner

Figure 1: 3D Geometries

L-Shape	AMG (sec)		GMG (sec)	
Number of Unknowns	Assembling $\widetilde{V}_h$	Galerkin Projection	Assembling $\widetilde{V}_h$	Matrix-hierarchy
1792	6.0	2.0	6.0	0.9
7168	32.5	7.5	32.5	6.9
28672	158	30	158	40

Table 1: Assembling $\widetilde{V}_h$ and Setup Times for L-Shape

Secondly, we compare the numbers of iterations, that are needed within the preconditioned conjugate gradient (PCG) method. Moreover, the CPU-time of one single PCG-iteration for different numbers of unknowns are listed in Table 2. The efficiency of our multigrid preconditioners is impressively reflected, for our GMG-preconditioners as well as for the AMG-preconditioners. One can clearly observe the expected almost linear increase of the CPU-times for one iteration with respect to the number of unknowns. Considering the time for one PCG-iteration, we notice, that the GMG variant is faster for larger problems.

Furthermore, we obtain constant iteration numbers for a wide range of problem sizes. That implies that our data-sparse multigrid preconditioner for the single layer potential operator is of high quality. In the case of AMG preconditioning we also have small iteration numbers, nevertheless they are slightly increasing. Because the coarser matrix levels are produced in a purely algebraic way, it is hardly possible to preserve corresponding 'virtual' coarse grids of the original geometry.

A closer look to the matrix or grid hierarchy is given in Table 3. For each level we present the number of unknowns of the underlying system of equations. Geometrical multigrid yields the finer grids by dividing the coarse triangles into four smaller one. In order to guarantee a regular refinement this is done by

Number of Unknowns	AMG		GMG	
	PCG-Cycle (sec)	Iterations	PCG-Cycle (sec)	Iterations
Sphere				
1920	0.1	8	0.1	8
7680	0.8	8	0.5	9
30720	4.1	10	2.9	9
L-Shape				
1792	0.1	6	0.1	7
7168	0.8	6	0.6	7
28672	4.2	9	2.9	7
114688	-	-	33*)	7
Cube			*) memory exceeded - cache effects	
3072	0.2	5	0.2	6
12288	1.7	8	1.1	7
49152	9.0	11	5.7	7
Fichera-Corner				
1920	0.1	14	0.1	15
7680	0.8	15	0.6	15
30720	5.0	17	3.2	15

Table 2: Key data for AMG/GMG Preconditioner

connecting the midpoints of their edges. Considering the AMG column, the matrices are larger on comparable levels. Therefore, one preconditioning step involves more effort with respect to the CPU-time. According dates in Table 2 confirm this assertion.

5 Conclusions and Further Remarks

In this paper we presented a geometrical multigrid and algebraic multigrid approach for the solution of large-scale boundary element equations. For that purpose an approximation of the boundary element matrices is absolutely essential. Our numerical experiments have been realized by the adaptive cross approximation technique which guarantees that the effort for storing the matrices and for a single matrix-by-vector multiplication can be reduced to almost $O(N_h)$. The discretized single layer potential operator yields symmetric positive definite matrices in the original dense version as well as in the ACA representation. Therefore, the system of boundary element equations can be solved by means of multigrid preconditioned CG-algorithms.

Due to the sparse representation of our matrices, we had to adapt each component of our AMG-algorithm properly. In order to set up the matrix hierarchy and the corresponding transfer operators an auxiliary matrix was constructed for the AMG method. On the other hand the matrices were built accordingly to the grid hierarchy in the GMG method. The smoothing procedure was realized by the proposed BLP-smoother for pseudo-differential operators of order minus

Level	AMG	GMG
Sphere		
fine 1	30720	30720
2	14963	7680
3	4660	1920
coarse 4	1233	480
L-Shape		
fine 1	28672	28672
2	14011	7168
3	4369	1792
coarse 4	1189	448

Level	AMG	GMG
Cube		
fine 1	49152	49152
2	23828	12288
3	7311	3072
4	2025	768
coarse 5	625	-
Fichera-Corner		
fine 1	30720	30720
2	14995	7680
3	4677	1920
coarse 4	1282	480

Table 3: Number of Unknowns for Matrix/Grid Hierarchy

one.

The overall algorithm provides interesting numerical results. One can notice small constant iteration numbers for the GMG method and also small (but slightly increasing) iteration numbers for the AMG approach. That confirms the high quality of our multigrid preconditioners. In addition, the CPU time for a single iterative step almost grows like $O(N_h)$. As expected, the GMG variant is faster than the AMG version.

The presented multigrid techniques, both the algebraic and the geometrical, induce an almost optimal solver for boundary element equations arising from 3D boundary value problems. In the case of the single layer potential operator, only an appropriate realization of the Laplace-Beltrami operator on the corresponding surface of a 3D domain and a sufficiently accurate coarse equation system will ensure an optimal precondition step which is necessary for a fast convergence of the PCG. Finally, we mention that efficient multigrid preconditioner for the discrete single layer potential operator are very important as building blocks in primal and dual domain decomposition preconditioners [8, 16, 17].

References

[1] R. A. Adams. *Sobolev Spaces.* Pure and Applied Mathematics. Academic Press, New York, 1975.

[2] M. Bebendorf. Approximation of Boundary Element Matrices. *Numerische Mathematik*, 86:565–589, 2000.

[3] M. Bebendorf. *Effiziente numerische Lösung von Randintegralgleichungen unter Verwendung von Niedrigrang-Matrizen.* PhD thesis, Universität Saarbrücken, 2000.

[4] M. Bebendorf and S. Rjasanov. Adaptive Low-Rank Approximation of Collocation Matrices. *Computing*, 70(1):1–24, 2003.

[5] J. H. Bramble, Z. Leyk, and J. E. Pasciak. The Analysis of Multigrid Algorithms for Pseudo-Differential Operators of Order Minus One. *Math. Comp.*, 63(208):461–478, 1994.

[6] J. Carrier, L. Greengard, and V. Rokhlin. A fast adaptive multipole algorithm for particle simulations. *SIAM J. Sci. Statist. Comput.*, 9(4):669–686, 1988.

[7] G. Chen and J. Zhou. *Boundary Element Methods.* Computational Mathematics and Applications. Academic Press, 1992.

[8] G. Haase, B. Heise, M. Kuhn, and U. Langer. Adaptive domain decomposition methods for finite and boundary element equations. In *Boundary Element Topics (ed. by W. Wendland)*, pages 121–147, Berlin, 1998. Springer-Verlag.

[9] W. Hackbusch. A Sparse Matrix Arithmetic based on $\mathcal{H}$-Matrices. *Computing*, 62(2):89–108, 1999.

[10] 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.

[11] Johannes Kepler University Linz, Institute of Computational Mathematics. *PEBBLES - User's Guide*, 1999. http://www.numa.uni-linz.ac.at/Research/Projects/pebbles.html.

[12] C. Lage and C. Schwab. Wavelet Galerkin Algorithms for Boundary Integral Equations. *SIAM J. Sci. Comput.*, 20:2195–2222, 1999.

[13] U. Langer and D. Pusch. Data-Sparse Algebraic Multigrid Methods for Large Scale Boundary Element Equations. *Applied Numerical Mathematics*, 2004. (to appear).

[14] U. Langer, D. Pusch, and S. Reitzinger. Efficient Preconditioners for Boundary Element Matrices Based on Grey-Box Algebraic Multigrid Methods. *International Journal for Numerical Methods in Engineering*, 58(13):1937–1953, 2003.

[15] U. Langer, D. Pusch, and S. Reitzinger. Sparse Algebraic Multigrid Methods for Large Scale Boundary Element Equations. SFB-Report 2003-47, University of Linz, December 2003. http://www.sfb013.uni-linz.ac.at/∼sfb/reports/2003/ps-files.

[16] U. Langer and O. Steinbach. Boundary element tearing and interconnecting methods. *Computing*, 71(3):205–228, 2003.

[17] U. Langer and O. Steinbach. Coupled boundary and finite element tearing and interconnecting methods. In *Proceedings of the 15th Int. Conference on Domain Decomposition (ed. by R. Kornhuber, R. Hoppe, J. Periaux, O. Pironneau, O. Widlund and J. Xu), Lecture Notes in Computational Sciences and Engineering, vol. 40*, pages 83–97, Heidelberg, 2004. Springer.

[18] V. Rokhlin. Rapid solution of integral equations of classical potential theory. *J. Comput. Phys.*, 60(2):187–207, 1985.

[19] O. Steinbach. *Numerische Näherungsverfahren für elliptische Randwertprobleme: Finite Elemente und Randelemente*. Teubner-Verlag, Stuttgart, Leipzig, Wiesbaden, 2003.

[20] O. Steinbach. *Stability Estimates for Hybrid Coupled Domain Decomposition Methods*, volume 1809 of *Lectures Notes in Mathematics*. Springer-Verlag, Berlin, 2003.