Efficient Preconditioners for Boundary Element Matrices based on Algebraic Multigrid Methods *

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

9th January 2002

[1]Institute of Computational Mathematics
Johannes Kepler University Linz
{ulanger,pusch}@numa.uni-linz.ac.at
[2]Special Research Programme SFB F013
'Numerical and Symbolic Scientific Computing'
Johannes Kepler University Linz
reitz@sfb013.uni-linz.ac.at

Abstract

This paper is concerned with the iterative solution of the boundary element equations arising from standard Galerkin boundary element discretizations of first kind boundary integral operators of positive and negative order. We construct efficient preconditioners on the basis of algebraic multigrid methods that are well adapted to the treatment of boundary element matrices. In particular, the coarsening is based on an auxiliary matrix that represents the underlying topology in a certain sense. This auxiliary matrix is additionally used for the construction of the smoothers and the transfer operators. Finally, we present the results of some numerical studies that show the efficiency of the proposed algebraic multigrid preconditioners.

Keywords boundary element method, algebraic multigrid method, preconditioned conjugate gradient method

1 Introduction

Discretization of elliptic boundary value problems for second order (self-adjoint) partial differential equations is mostly done by the finite element method (FEM). However, in many situations it can be advantageous to use the boundary element method (BEM) as discretization technique. Exterior boundary value

*This work has been supported by the Austrian Science Fund 'Fonds zur Förderung der wissenschaftlichen Forschung (FWF)' under grant SFB F013 'Numerical and Symbolic Scientific Computing' and grant P14953.

problems are certainly one class of problems where BEM is clearly superior to FEM. The accurate computation of singularities and the handling of moving parts in the computational domain are other fields of the typical applications of BEM. BEM reduces the computational domain $\Omega \subset \mathbb{R}^d$ $(d = 2, 3)$ to the boundary $\partial\Omega \subset \mathbb{R}^{d-1}$ of Ω. This reduction of the dimension simplifies the mesh generation considerably, especially, for complicated geometries. Furthermore, the FEM and BEM can be combined which results in the coupled FE-BE-method that exploits the advantages of both methods (see [8, 9, 19, 28] and references therein).

In contrast to the FEM, where the discretization yields a large, sparse system matrix, the classical Galerkin BEM produces a (much) smaller, but dense system matrix, which is assumed to be symmetric and positive (semi) definite. The solution of such linear systems is the bottleneck of the most numerical simulation procedures. While the efficient iterative solution of finite element equations is well understood for several problem classes (see e.g. [13, 22]), this is not true for BE-matrices. In many boundary element codes, especially in the commercial ones, direct solvers are preferred. The goal of this paper consists in the construction of preconditioners on the basis of algebraic multigrid (AMG) techniques that are used as preconditioners in a conjugate gradient (CG) solver. Similar to their geometrical counterparts, one can expect that the corresponding AMG preconditioned CG methods should be optimal in the sense that the memory demand and arithmetical complexity are proportional to the memory required for the storage of the system matrix and the arithmetical cost for the matrix-by-vector product, respectively.

For simplifying the further discussion, we consider the hypersingular and the single-layer potential integral operators arising from the Laplace equation as typical model problems. While the first one can be handled and analyzed very easily (see [14, 34]), there are essential difficulties for the second one. These difficulties are due to the fact that the single-layer potential operator is a pseudo-differential operator of negative order, namely of the order -1. In this case, the small eigenvalues correspond to high frequency eigenfunctions and the large eigenvalues correspond to low frequency eigenfunctions. This behavior of the spectrum must be taken into account in order to build a efficient multigrid solver. S. Rjasanow considered and analyzed some twogrid method in [25]. One of the first papers related to a fast multilevel solution is by A. Brandt and A. A. Lubrecht [7], but without a rigorous proof. T. von Petersdorff and E.P. Stephan proposed to use modified ansatz functions according to the well-known partial integration link between the single-layer potential operator and the hypersingular integral operator [35]. J. Bramble, Z. Leyk, and J. Pasicak suggested an appropriate smoother (in the subsequent abbreviated by BLP-smoother) in [5]. Moreover, they proved that the geometric multigrid method with the proposed BLP-smoother is asymptotically optimal in case of the single-layer potential operator. For the general case, they showed at least level-dependent convergence rates. However, geometric multigrid versions need inherently a hierarchical grid and the coarsest grid must be small enough in order to be solved efficiently by a classical iterative or direct solver. Especially the hierarchical grid is often not available and an additional implementation is difficult.

The AMG method is an iterative solver which tries to preserve the properties of geometric multigrid methods as far as possible, but only with single grid information. Thus, the goal for an AMG method consists in the construction of a matrix hierarchy, appropriate transfer operators and the correct smoothing procedure. For several finite element equations the AMG method is well understood (see [4, 6, 15, 16, 18, 32, 33, 36] for different AMG realizations). These methods have in common that they take advantage of the sparsity structure of the system matrix. Since the classical BE-matrices are dense we present a more general technique. Actually we use the approach proposed in [10, 24].

As for the geometric counterpart the main idea for an AMG method consists in the efficient interplay of coarse grid correction and smoothing. Therefore, the key ingredients for an AMG method, i.e. the coarsening technique, the transfer operators, the coarse grid operators and the smoothing procedure, have to be defined in such a way that their interplay results in a rapidly convergent AMG method, or at least in an efficient AMG preconditioner. Since we use the approach of [10, 24], we have to define an appropriate auxiliary matrix which represents the underlying BE-mesh and the main features of the problem in some sense. This auxiliary matrix reflects the local topology in such a way that it allows us to define the correct sparse transfer operators. Moreover, the auxiliary matrix will be used for deriving BLP-kind-smoothers in a pure algebraic setting. The coarse grid operator is usually constructed by the Galerkin method. With these ingredients, we try to mimic the geometric multigrid version as well as possible. A recursive application immediately leads to a matrix hierarchy with transfer and smoothing operators. Therewith, a complete multigrid cycle can be assembled (see, e.g. [11]).

The paper is organized as follows: In Section 2, we give a brief overview on the Galerkin BE-method. Furthermore, we present the main properties of the considered discretized operators. Section 3 is devoted to the construction of the preconditioner for the conjugate gradient method based on AMG methods. In Section 4, first numerical studies are presented. Finally, in Section 5, we give further remarks and draw some conclusions.

2 Problem Formulation and Properties

Let $\Omega \subset \mathbb{R}^d$ (d=2,3) be a bounded, single connected domain with a sufficiently smooth boundary $\Gamma = \partial\Omega$, and let us consider a continuous, linear, self-adjoint pseudo-differential operator

$$\mathcal{A} : H^{\alpha}(\Gamma) \mapsto H^{-\alpha}(\Gamma), \tag{1}$$

where $H^s(\Gamma)$ denotes the usual Sobolev space for some real $s \in \mathbb{R}$, see e.g. [1]. For the rest of this paper, we are concerned with the single-layer potential operator

$$(V\psi)(y) = \int_\Gamma E(x,y)\psi(x)ds_x \tag{2}$$

and the hypersingular integral operator

$$(D\psi)(y) = -\partial_{n_y}\int_\Gamma \partial_{n_x}E(x,y)\psi(x)ds_x \tag{3}$$

which are special mappings $\mathcal{A} \in \mathcal{L}(H^\alpha(\Gamma), H^{-\alpha}(\Gamma))$ with $\alpha = -1/2$ and $\alpha = +1/2$ for the single-layer potential operator and the hypersingular integral operator, respectively. In addition, $E(x, y)$ denotes a fundamental solution of the boundary value problem under consideration, and n_z the unit outward normal vector to Γ at some point $z \in \Gamma$.

Example 2.1. *Let us consider the interior Dirichlet problem for Laplace's equation, i.e.,*

$$\begin{aligned} -\Delta u &= 0 \qquad \text{in } \Omega \\ u &= g_0 \qquad \text{on } \Gamma \end{aligned}$$

which can be reduced to the single-layer potential integral equation in order to get the Neumann data $\frac{\partial u}{\partial n_x}$ *on the boundary* Γ*. Similarly, for the interior Neumann problem of Laplace's equation, we get the Dirichlet data on the boundary* Γ *via the hypersingular integral equation. Then, by using the representation formula*

$$\sigma(y)u(y) = -\int_\Gamma u(x)\frac{\partial E(x,y)}{\partial n_x} ds_x + \int_\Gamma \frac{\partial u}{\partial n_x}(x) E(x,y) ds_x$$

with $\sigma(y) = 0, 1/2, 1$ *for* $y \notin \overline{\Omega}, y \in \Gamma, y \in \Omega$*, respectively, the solution of the boundary value problem* u *considered above can be represented on* Ω*. Let us mention that the value of* $\sigma(y)$ *must be modified at those points* $y \in \Gamma$ *where the boundary is not smooth (see [8, 28] and references therein).*

Although the suggested method can be extended to other second order self-adjoint elliptic equations we consider only the Laplace equation for the sake of simplicity.

The task of finding the solution $u \in H^\alpha(\Gamma)$ of the integral equation

$$\mathcal{A}u = f \tag{4}$$

for given $\mathcal{A} \in \mathcal{L}(H^\alpha(\Gamma), H^{-\alpha}(\Gamma))$ and $f \in H^{-\alpha}(\Gamma)$ is equivalent to the solution of its variational formulation

$$\text{find } u \in H^\alpha(\Gamma) : \langle \mathcal{A}u, v\rangle = \langle f, v\rangle \qquad \forall v \in H^\alpha(\Gamma) \tag{5}$$

that is the starting point for the Galerkin discretization (see [2, 8, 28, 30]). Here $\langle\cdot,\cdot\rangle : H^{-\alpha}(\Gamma) \times H^\alpha(\Gamma) \to \mathbb{R}$ denotes the duality product. We refer to [8] for an overview over other discretization schemes. As usual for the Galerkin discretizations, we define a finite-dimensional subspace

$$\mathbb{V}_h \subset H^\alpha(\Gamma) \quad \text{and} \quad \mathbb{V}_h \sim \mathbb{R}^{N_h}$$

that is related to $\mathbb{R}^{N_h}$ by the Galerkin isomorphism, where h denotes the typical mesh-size. Consequently, the Galerkin scheme results in the symmetric, (semi) positive definite system of algebraic equations

$$K_h \underline{u}_h = \underline{f}_h\,. \tag{6}$$

The convergence analysis and error estimates can be found in [37] and references therein.

The properties of the system matrix K_h are very important for the construction of an optimal preconditioner. It can be shown that the spectral condition number $\kappa(K_h) = O(h^{-1})$ as h tends to zero for both boundary integral operators V and D. Further, let the eigenvalues of K_h be arranged in a rising order

$$0 \leq \lambda_1, \leq \ldots, \leq \lambda_{N_h}$$

with corresponding eigenvectors $\underline{\phi}_i$, where $i = 1, \ldots, N_h$. Then, for the hypersingular integral operator, the high frequency eigenfunctions correspond to the large eigenvalues and the low frequency eigenfunctions correspond to the small eigenvalues. For the single layer potential, the relationship of eigenfunctions and eigenvalues is vice versa. This observation is due to the positive and the negative order of the hypersingular and the single-layer potential operators, respectively, see e.g. [26].

3 Algebraic Multigrid Method

In order to solve (6) by means of some AMG method or some AMG preconditioned conjugate gradient method, the multigrid constituents have to be defined properly. For further discussion, it is enough to discuss the construction of these components for the twogrid algorithm, for which the indices h and H are related to the fine and coarse grid quantities, respectively.

3.1 General Concept

Similar to geometric multigrid methods, the efficient interplay of the smoothing process and the coarse grid correction is the crucial point for an efficient AMG method too. The main difference to geometric multigrid methods is the lacking grid hierarchy. In order to come along with that deficiency a coarsening strategy is introduced which decreases the number of unknowns. Most coarsening techniques are based on the matrix graph, see e.g. [4, 6, 18, 27, 33]. Since these approaches are designed for sparse matrices we can not directly apply them to the dense system matrix K_h. Instead we use the general approach proposed in [10, 24]. Therefore, we have to specify an auxiliary matrix $B_h \in \mathbb{R}^{N_h \times N_h}$ which represents the BE-mesh. As we will see later we can use the same matrix that is used for shifting the smoothing procedure in the case of the single-layer potential operator. Meanwhile let us assume that B_h is a sparse M-matrix and that each diagonal entry of B_h can be related to some unknown and therefore to some node or element. There are only local connections to the neighbors. Since this auxiliary matrix is defined locally we are able to define the following index sets on a pure algebraic level (with ω_h the set of grid points or elements

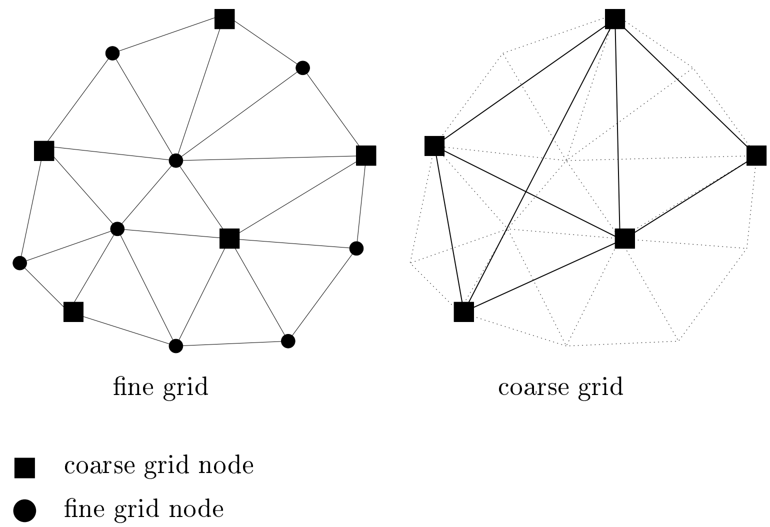

Figure 1: Detail view of a fine and coarse 'virtual' BE-mesh.

on level h):

$$\begin{aligned} N_h^i &= \{j \in \omega_h \,:\, |(B_h)_{ij}| \neq 0, i \neq j\}\,, \\ S_h^i &= \{j \in N_h^i \,:\, |(B_h)_{ij}| > \text{coarse}(B_h, i, j),\ i \neq j\}\,, \\ S_h^{i,T} &= \{j \in N_h^i \,:\, i \in S_h^j\}\,, \end{aligned}$$

which are related to the set of neighbors around a node $i \in \omega_h$, the set of strong connections, and the set of nodes with a strong connection to node i, respectively. Moreover the function 'coarse(B_h, i, j)' is an appropriate cut-off or coarsening function (see [24]). In an analogous way the above relations are defined on the coarse grid H provided the matrix B_H is known.

The next step consists in a standard coarsening on B_h. Motivated by some BE-grid (see Figure 1), a 'virtual' BE-grid can be split into coarse grid nodes ω_C and fine grid nodes ω_F, i.e.,

$$\omega_h = \omega_C \cup \omega_F\,, \qquad \omega_C \cap \omega_F = \emptyset\,,$$

such that there are (almost) no direct connections between any two coarse grid nodes. Furthermore, the resulting number of coarse grid nodes should be as large as possible. Then, the coarse grid is simply defined by

$$\omega_H = \omega_C\,.$$

Remark 3.1. *The coarse grid selection can be done by several different coarsening strategies (see [4, 18, 27, 33]). On the one hand side, a pure matrix graph based method can be used [4, 18], or on the other hand side a coarsening method depending on the matrix entries can be introduced [27, 33]. The latter case has the chance to detect anisotropies.*

In order to construct a coarse auxiliary matrix we use the well-known Galerkin projection method, and, therefore, we need an appropriate prolongation operator $P_h^B : \mathbb{R}^{N_H} \mapsto \mathbb{R}^{N_h}$, where $N_h = |\omega_h|$ and $N_H = |\omega_H|$ $(N_H < N_h)$

denote the number of unknowns on the fine and coarse level, respectively. Once the prolongation operator P_h^B is defined, the auxiliary coarse grid matrix becomes

$$B_H = (P_h^B)^T B_h P_h^B \,.$$

The efficient interplay of the coarse grid correction and the smoothing procedure is the key ingredient for every multigrid method. This fact is due to the decomposition into the direct sum

$$\mathbb{R}^{N_h} = \mathrm{im}(P_h^K) \oplus \ker((P_h^K)^T)\,,$$

with the full rank prolongation operator P_h^K for the system matrix. This means that the smooth error components (elements in $\mathrm{im}(P_h^K)$) can be well approximated on the coarse grid, while the high frequency components (elements in $\ker((P_h^K)^T)$) must be efficiently reduced by the smoother (see [21, 24]). It is worth to mention that one possibility to define the prolongation operators P_h^K and P_h^B consists in choosing them equal, i.e., $P_h^K \equiv P_h^B \equiv P_h$. The exact definition of both multigrid components will be made for our specific field of application in the next subsection. The coarse system matrix K_H is also constructed via Galerkin's method. A recursive application of that process immediately leads to a matrix hierarchy with the corresponding transfer operators. If an appropriate smoothing process is defined, then a complete multigrid V-cycle can be assembled as is shown in Algorithm 1.

Algorithm 1 V(ν_F,ν_B)-cycle: MG($\underline{u}_\ell$, $\underline{f}_\ell$, ℓ)

```
if ℓ = COARSELEVEL then
  Define u_ℓ = (K_ℓ)^{-1} f_ℓ by some direct solver
else
  Smooth ν_F times on K_ℓ u_ℓ = f_ℓ
  Calculate the defect d_ℓ = f_ℓ − K_ℓ u_ℓ
  Restrict the defect to the next coarser level ℓ + 1:  d_{ℓ+1} = P_ℓ^T d_ℓ
  Set u_{ℓ+1} ≡ 0
  Call MG(u_{ℓ+1}, d_{ℓ+1}, ℓ + 1)
  Prolongate the correction s_ℓ = P_ℓ u_{ℓ+1}
  Update the solution u_ℓ = u_ℓ + s_ℓ
  Smooth ν_B times on K_ℓ u_ℓ = f_ℓ
end if
```

In the following we define the required multigrid components in a more specific way. Because of the above motivation for a multigrid cycle we have to distinguish the underlying boundary integral operators due to the total different behavior of their spectrum.

3.2 The Hypersingular Integral Operator

The hypersingular integral operator is assumed to be discretized by piecewise linear ansatz-functions. The set ω_h is therefore related to the nodes of the BE-mesh (see also Figures 2 and 3 below). We define the auxiliary matrix B_h by the so-called nodal distance matrix

$$(B_h)_{ij} = \begin{cases} -\frac{1}{\|e_{ij}\|_0} & \text{if } j \in N_h^i \\ \sum_{k\in N_h^i} \frac{1}{\|e_{ij}\|_0} & \text{if } i = j \\ 0 & \text{else} \end{cases}$$

that is obviously an M-matrix. Here e_{ij} is the vector connecting the node x_i and x_j, i.e., $e_{ij} = x_j - x_i$, and $\|\cdot\|_0$ denotes the Euclidean norm $\mathbb{R}^d$. In the hypersingular case, the auxiliary matrix B_h is only used for the coarsening process. Since the high frequency eigenfunctions correspond to a large eigenvalue, the standard Gauss-Seidel smoother is efficient enough for this problem class (see e.g. [11, 34]).

3.3 Single-Layer Potential

For this case it is possible to use piecewise constant ansatz-functions for the discretisation (see also Figures 2 and 3). Thus, the set ω_h is now related to the elements of the BE-mesh. Because of the properties of the system matrix, a Gauss-Seidel smoother (GS-smoother) is not the correct choice, since the GS-smoother works on the frequencies corresponding to the large eigenvalues that are smooth in the single-layer potential case. Therefore, we need an other smoother that acts on the high frequency components. Such a smoother was proposed by J. Bramble, Z. Leyk, and J.E. Pasciak in [5] where the reader also finds a rigorous analysis for the geometrical version of the corresponding multigrid algorithm.

The BLP smoother is defined as follows. Let $A_h \in \mathbb{R}^{N_h \times N_h}$ be some discretization corresponding to the Laplace-Beltrami operator living on the boundary Γ. The spectral properties of such kind of matrices A_h are well known (see, e.g. [12]). It was shown in [5] that smoothing sweeps of the form

$$\underline{u}_h \leftarrow \underline{u}_h + \tau_h \cdot A_h(\underline{f}_h - K_h \underline{u}_h)$$

actually reduce the high frequency components of the error provided that the damping parameter τ_h satisfies the inequality

$$0 < \tau_h \leq \frac{1}{\mu_{N_h}}$$

where μ_{N_h} denotes the largest eigenvalue of the generalized eigenvalue problem

$$K_h \underline{\varphi} = \mu\, A_h^{-1} \underline{\varphi}. \tag{7}$$

Example 3.2. *Let us consider a boundary Γ which is subdivided into N_h elements with corresponding mesh-size h. Further, for every element $i \in \omega_h$, we define the bary-center x_i^s (see Figures 2 and 3 below). Since we use piecewise constant ansatz-functions for the single-layer potential, we define the matrix A_h by the relation*

$$(A_h)_{ij} = \begin{cases} -\frac{1}{\|e_{ij}\|_0} & \text{if } j \in N_h^i, \\ \sum_{k\in N_h^i} \frac{1}{\|e_{ij}\|_0} & \text{if } i = j, \\ 0 & \text{else.} \end{cases}$$

In the last formula, $e_{ij} = x_i^s - x_j^s$, and N_h^i contains all neighboring elements with respect to element number i. Thus the auxiliary matrix can be seen as a distance matrix that connects neighboring elements. In the 2d case of a circle, the matrix A_h corresponds to the well-known three-point finite difference approximation to the Laplace equation in 1D provided that the discretization is uniform.

It is worth to mention that the shifting matrix A_h from the smoothing procedure can be used as our auxiliary matrix B_h at the same time, i.e.,

$$B_h = A_h$$

at every level.

In order to get an approximation to the required largest eigenvalue μ_{N_h} of the generalized eigenvalue problem (7) we use the following technique at every level (see [20]).

Algorithm 2 Preconditioned gradient method: PGM(K_h, A_h)

$\underline{v}_h \leftarrow$ initial guess
$\underline{r}_h \leftarrow 0$
$\underline{r}_h \leftarrow -K_h \underline{v}_h$
for all i=0,k **do**
  $\underline{s}_h \leftarrow A_h \underline{r}_h$
  $\zeta_i = \frac{(\underline{s}_h, \underline{r}_h)}{(K_h \underline{s}_h, \underline{s}_h)}$
  $\underline{v}_h \leftarrow \underline{v}_h + \zeta_i \cdot \underline{s}_h$
  $\underline{r}_h \leftarrow \underline{r}_h - \zeta_i \cdot \underline{s}_h$
end for

We apply k ($k \geq 3$) A_h^{-1}-preconditioned gradient iterations (see Algorithm 2) to the homogeneous system

$$K_h \underline{v}_h = 0$$

with a non-trivial initial guess $\underline{v}_h^0$. The roots of the quadratic polynomial

$$p_k(\lambda) = (1 - \zeta_k \cdot \lambda) \cdot (1 - \zeta_{k-1} \cdot \lambda) - \delta$$

approximate the largest and the smallest eigenvalues of the generalized eigenvalue problem (7), where $\delta^2 = (K_h \underline{v}_h^k, \underline{v}_h^k)_0 / (K_h \underline{v}_h^{k-2}, \underline{v}_h^{k-2})_0$ is built from the iterates v_h^k and v_h^{k-2}. This procedure is very fast. Usually less than 10 iteration steps produce very accurate approximations to the smallest and largest eigenvalues.

In both cases, the prolongation operator P_h is defined by the relation

$$(P_h)_{ij} = \begin{cases} 1 & i = j \in \omega_C, \\ \frac{1}{|\omega_C \cap S_h^{i,T}|} & i \in \omega_F,\, j \in \omega_C, \\ 0 & else\,. \end{cases}$$

Such prolongation operator preserves constant vectors. In the last formula we assume the coarse grid unknowns to be ordered first.

3.4 Complexity Analysis

The complexity of an iterative solver for linear systems of algebraic equations is usually measured by the complexity that is needed for one matrix-times-vector operation. For our dense matrices this is equivalent to $O(N_h^2)$ arithmetic operations as h tends to zero. The aim of this subsection is to show that the proposed AMG preconditioner is also of order $O(N_h^2)$:

1. *Smoother:* The application of one smoothing step is essentially one multiplication by K_h and one multiplication by B_h, where the latter one is of order $O(N_h)$ arithmetical operations due to the sparsity of A_h.

2. *Prolongation (Restriction):* Since we constructed a local prolongation (restriction) operator the application of them is of order $O(N_h)$.

3. *Galerkin:* This is the most critical operation, but because of the sparse prolongation operator the computation of the coarse grid matrix can be done by $O(N_H \cdot N_h)$ operations which is clearly less than $O(N_h^2)$.

Consequently, the proposed AMG cycle is of order $O(N_h^2)$ as h tends to zero. The above estimates holds also for coarser levels, since the average number of non-zero entries per row of the auxiliary matrix is bounded from above for a lot of practical examples.

For the memory demand, the same arguments hold as for the arithmetic costs as long as the coarsening process reduces sufficiently the degrees of freedom from one grid to the next coarser grid.

Remark 3.3. *A further reduction in the arithmetic costs can only be achieved if the matrix-times-vector operation is reduced for the overall preconditioner. One possible technique is the adaptive cross approximation of the system matrix proposed in [2, 3]. In this case the memory requirement is considerably reduced too. Other sparse matrix techniques proposed for the boundary element method during the last years will also lead to the reduction of the arithmetical complexity and the memory demand.*

4 Numerical Studies

In order to show the efficiency of the proposed approach we present some first numerical results. The Galerkin BE-matrices are generated by the BE-software package OSTBEM [31]. The AMG preconditioner is realized within the package PEBBLES [23]. The proposed AMG technique is used as a preconditioner C_h in the preconditioned conjugate gradient (PCG) method (see [17]). In every example presented below we choose the symmetric $V(1,1)$-cycle with 1 presmoothing and 1 postsmoothing step per cycle. The iteration error is measured in the $K_h C_h^{-1} K_h$ energy norm. All calculations were done on an SGI Octane workstation with 300 MHz RISC processor.

First we consider $\Omega \subset \mathbb{R}^2$ to be a disk with radius 0.5, Thus, $\Gamma = \partial\Omega$ is the circle with the same radius. We choose a uniform discretization such that we arrive at $N_h = 2^n$ nodes (or equivalently edges), with $n \geq 2$ (see

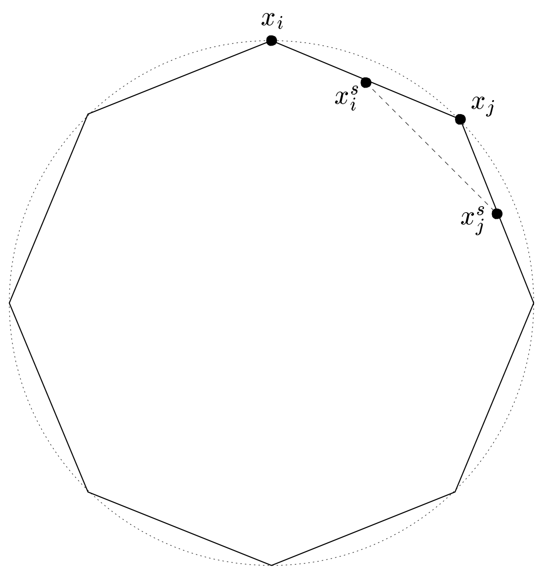

Figure 2: Circle with an uniform discretization of the boundary Γ.

Figure 2). In Table 1 the CPU-times and the number of constructed levels are given. For this special example it can be shown that the grid hierarchy of a geometric multigrid is reproduced. We clearly observe the quadratic increase in the complexity needed for the construction of the matrix hierarchy (*setup*) and for one multigrid application (*cycle*). Table 2 shows the condition number

N_h	setup (sec)	cycle (sec)	level
1024	0.41	0.16	5
2048	1.59	0.69	6
4096	6.39	2.85	7
8192	25.77	11.51	8

Table 1: CPU-times and number of constructed levels for the circle.

for different problem settings on the circle, namely,

1. the hypersingular integral operator with a Gauss-Seidel smoother: *hyper*,
2. the single-layer potential with a Gauss-Seidel smoother: *single GS*, and
3. the single-layer potential with the BLP-smoother: *single BLP*.

As we would expect, we get constant condition numbers, that are close to 1, for the hypersingular integral operator as well as for the single-layer potential operator with the BLP-smoother. On the other hand, we observe that the GS-smoother is not optimal for the single-layer potential. In the next example, the domain $\Omega \subset \mathbb{R}^3$ is the unit sphere. The surface mesh is generated by NETGEN [29] (see Figure 3). In this case, we can not necessarily expect constant iteration numbers since it is hard to reproduce the grid hierarchy of a geometric multigrid

N_h	hyper	single GS	single BLP
1024	1.03	2.24	1.05
2048	1.03	2.71	1.05
4096	1.03	3.16	1.05
8192	1.03	3.55	1.05

Table 2: Condition number $\kappa(C_h^{-1}K_h)$ for the circle.

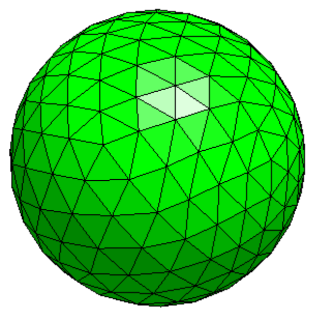

Figure 3: Sphere with a uniform discretization of the surface.

version. However, as we can see in Table 3 the condition number stays constant as the number of unknowns grow for the hypersingular integral operator with a GS-smoother. Additionally, the expected quadratic increase in the CPU-times can be observed. The single-layer potential operator behaves slightly different

N_h	setup (sec)	cycle (sec)	$\kappa(C_h^{-1}K_h)$	level
242	0.09	0.05	1.04	2
962	1.26	0.92	1.05	3
3842	19.25	14.49	1.06	4

Table 3: CPU-times, condition number, and number of constructed levels for the sphere: hypersingular integral operator.

(see Table 4). The BLP-smoother improves the condition number considerably in comparison to the GS-smoother (the condition number for the GS-smoother is given in brackets). However, the condition number can not be kept constant as N_h grows. Further numerical studies showed us that the condition number in Table 4 depends moderately on the number of unknowns. But this fact is no problem for practical applications.

N_h	setup (sec)	cycle (sec)	$\kappa(C_h^{-1}K_h)$	level
480	0.33	0.18	1.85 (3.87)	3
1920	5.10	3.52	2.21 (4.52)	4
7680	86.23	52.72	3.72 (6.15)	5

Table 4: CPU-times, condition number, and number of constructed levels for the sphere: single-layer potential.

5 Conclusions and Further Remarks

We have described the construction of efficient preconditioners which are essentially based on the AMG method for the single-layer potential and the hypersingular boundary integral operators. These first studies promise an efficient solver tool for real practical applications. However, the basic matrix-times-vector multiplication of the form $K_h * \underline{u}_h$ can be done much faster by different sparse matrix representation techniques (see [2] and the references therein). In order to assemble a preconditioner for the sparse matrix representation the coarsening scheme of AMG has to be adopted properly. This is one topic of current research and will be presented in a forthcoming paper. On the other hand, we are interested in the treatment of non-symmetric BE-matrices arising from collocation discretization techniques that are widely used in practice. Obviously, it is possible to apply the same techniques for constructing AMG preconditioners as in the Galerkin case. Of course, instead of the conjugate gradient method, some Krylov space method for solving linear systems with a non-symmetric system matrix must be used [22].

References

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

[2] M. Bebendorf, *Effiziente numerische Lösung von Randintegralgleichungen unter Verwendung von Niedrigrang-Matrizen*, Ph.D. thesis, Unviversität Saarbrücken, 2000.

[3] M. Bebendorf and S. Rjasanov, *Adaptive low-rank approximation of collocation matrices*, Preprint No. 39, Universität des Saarlandes, Fachrichtung 6.1 - Mathematik, 2001.

[4] D. Braess, *Towards algebraic multigrid for elliptic problems of second order*, Computing **55** (1995), 379–393.

[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** (1994), no. 208, 461 – 478.

[6] A. Brandt, *Generally highly accurate algebraic coarsening*, Elec. Trans. Num. Anal. **10** (2000), 1–20.

[7] A. Brandt and A. A. Lubrecht, *Multilevel multi-integration and fast solution of integral equations*, J. Comput. Phys. **90** (1990), 348 – 370.

[8] G. Chen and J. Zhou, *Boundary element methods*, Computational Mathematics and Applications, Academic Press, 1992.

[9] G. Haase, B. Heise, M. Kuhn, and U. Langer, *Adaptive domain decomposition methods for finite and boundary element equations*, Boundary Element Topics (W.L. Wendland, ed.), Springer, Berlin, Heidelberg, New York, 1997, pp. 121–147.

[10] G. Haase, U. Langer, S. Reitzinger, and J. Schöberl, *A general approach to algebraic multigrid*, SFB-Report 00-33, Johannes Kepler University Linz, SFB "Numerical and Symbolic Scientific Computing", 2000.

[11] W. Hackbusch, *Multigrid methods and application*, Springer Verlag, Berlin, Heidelberg, New York, 1985.

[12] ———, *Theorie und Numerik elliptischer Differentialgleichungen*, BG Teubner, Stuttgart, 1986.

[13] ———, *Iterative Lösung großer schwachbesetzter Gleichungssysteme*, BG Teubner, Stuttgart, 1991.

[14] F.-K. Hebeker, *On multigrid methods of the first kind for symmetric boundary integral equations of nonnegative order*, Preprint 1120, TH Darmstadt, 1988.

[15] V.E. Henson and P.S. Vassilevski, *Element-free AMGe: General algorithms for computing interpolation weights in AMG*, SIAM J. Sci. Comput. **23** (2001), no. 2, 629–650.

[16] J.E. Jones and P.S. Vassilevski, *AMGe based on element agglomeration*, SIAM J. Sci. Comput. **23** (2001), no. 1, 109 – 133.

[17] M. Jung and U. Langer, *Applications of multilevel methods to practical problems*, Surveys Math. Indust. **1** (1991), 217–257.

[18] F. Kickinger, *Algebraic multigrid for discrete elliptic second-order problems*, Multigrid Methods V. Proceedings of the 5th European Multigrid conference (W. Hackbusch, ed.), Springer Lecture Notes in Computational Science and Engineering, vol. 3, 1998, pp. 157–172.

[19] M. Kuhn, *Efficient parallel numerical simulation of magnetic field problems*, Ph.D. thesis, University of Linz, 1998.

[20] U. Langer and W. Queck, *Preconditioned Uzawa-type iterative methods for solving mixed finite element equations*, 3/1987, Wissenschaftliche Schriftenreihe der TU Karl-Marx-Stadt, Karl-Marx-Stadt, 1987.

[21] S. F. McCormick, *An algebraic interpretation of multigrid methods*, SIAM J. Numer. Anal. **19** (1982), no. 3, 548–560.

[22] G. Meurant, *Computer solution of large linear systems*, Studies in Mathematics and its Applications, vol. 28, Elsevier, 1999.

[23] S. Reitzinger, *PEBBLES - User's Guide*, Johannes Kepler University Linz, SFB "Numerical and Symbolic Scientific Computing ", 1999, www.sfb013.uni-linz.ac.at/~reitz/pebbles.html.

[24] ______, *Algebraic Multigrid Methods for Large Scale Finite Element Equations*, Reihe C - Technik und Naturwissenschaften, Universitätsverlag Rudolf Trauner, Linz, 2001.

[25] S. Rjasanow, *Zweigittermethode für eine Modellaufgabe bei BEM Diskretisierung*, Wiss. Z. d. TU Karl-Marx-Stadt **29** (1987), 230–235.

[26] ______, *Vorkonditionierte iterative Auflösung von Randelementgleichungen für die Dirichlet-Aufgabe*, 7/1990, Wissenschaftliche Schriftenreihe der TU Chemnitz, Chemnitz, 1990.

[27] J. W. Ruge and K. Stüben, *Algebraic multigrid (AMG)*, Multigrid Methods (S. McCormick, ed.), Frontiers in Applied Mathematics, vol. 5, SIAM, Philadelphia, 1986, pp. 73–130.

[28] A. H. Schatz, V. Thomée, and W. L. Wendland, *Mathematical theorie of finite and boundary element methods*, DMV Seminar, no. 15, Birkhäuser, 1990.

[29] J. Schöberl, *NETGEN - An advancing front 2D/3D-mesh generator based on abstract rules*, Computing and Visualization in Science **1** (1997), 41–52.

[30] O. Steinbach, *Gebietszerlegungsmethoden mit Randintegralgleichungen und effiziente numerische Lösungsverfahren für gemischte Randwertprobleme*, Ph.D. thesis, University of Stuttgart, 1996.

[31] ______, *OSTBEM - A boundary element software package*, University of Stuttgart, 2000.

[32] K. Stüben, *Algebraic multigrid: An introduction with applications*, Report 53, GMD, 1999.

[33] P. Vanek, J. Mandel, and M. Brezina, *Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems*, Computing **56** (1996), 179–196.

[34] T. von Petersdorf and E.P. Stephan, *On the convergence of the multigrid method for a hypersingular integral equation of the first kind*, Numer. Math. **57** (1990), 379–391.

[35] ______, *Multigrid solvers and preconditioners for first kind integral equations*, Numer. Methods Partial Differential Equations **8** (1992), 443–450.

[36] C. Wagner, *On the algebraic construction of multilevel transfer operators*, Computing **65** (2000), 73–95.

[37] W. L. Wendland, *On some mathematical aspects of boundary element methods for elliptic problems*, The mathematics of finite elements and applications V, Academic Press, London, 1985, pp. 193 – 227.