Block-adaptive Cross Approximation of Discrete Integral Operators*

M. Bauer and M. Bebendorf†

June 3, 2019

In this article we extend the adaptive cross approximation (ACA) method known for the efficient approximation of discretisations of integral operators to a block-adaptive version. While ACA is usually employed to assemble hierarchical matrix approximations having the same prescribed accuracy on all blocks of the partition, for the solution of linear systems it may be more efficient to adapt the accuracy of each block to the actual error of the solution as some blocks may be more important for the solution error than others. To this end, error estimation techniques known from adaptive mesh refinement are applied to automatically improve the block-wise matrix approximation. This allows to interlace the assembling of the coefficient matrix with the iterative solution.

Keywords: ACA, error estimators, BEM, non-local operators, hierarchical matrices, fast solvers

1 Introduction

Methods for the solution of boundary value problems of elliptic partial differential equations typically involve non-local operators. Examples are fractional diffusion problems (see [1, 14, 15]) and boundary integral methods (see [25, 26]), in which amenable boundary value problems are reformulated as integral equations over the boundary of the computational domain. The discretisation of such non-local operators usually leads to fully populated matrices. The corresponding systems of linear equations are quite expensive to solve numerically in the typical case of large numbers of unknowns. Additionally, assembling and storing the discrete operators may be quite costly. Even if the formulation of the boundary value problem is local and its discretisation (e.g. by the finite element method) leads to a sparse matrix, it may be useful to construct non-local approximations (e.g. for preconditioning) to its inverse or to triangular factorisations. In both cases, suitable techniques have to be employed if the boundary value problem is to be solved with linear or almost linear complexity.

There are several ways how fully populated discretisations of elliptic operators can be treated efficiently. In addition to *fast multipole methods* [20] or *mosaic decomposition methods* [27], *hierarchical*

*This work was supported by the DFG project BE2626/4-1.

†Mathematisches Institut, Universität Bayreuth, 95447 Bayreuth, Germany

matrices ($\mathcal{H}$-matrices) [21, 22] have been introduced for the asymptotic optimal numerical treatment. This kind of methods can be regarded to be based on a hierarchical partition of the discretised operator $A \in \mathbb{R}^{N \times N}$ into suitable blocks combined with a local low-rank approximation. Hence, fully populated matrices can be approximated by data-sparse representations which require only a logarithmic-linear amount of storage. Furthermore, these representations can be applied to vectors with logarithmic-linear complexity. In addition, $\mathcal{H}$-matrices provide approximate substitutes for the usual matrix operations including the LU factorisation with logarithmic-linear complexity; see [8]. These properties strongly accelerate iterative solvers like the *conjugate gradient* (CG) method or GMRES (see [24]) for the solution of the arising linear systems of equations.

A crucial step for the efficient treatment of fully populated matrices A is their assembling. While explicit kernel approximations were used in the early stages of fast methods for the treatment of integral operators, in recent years the *adaptive cross approximation* (ACA) (see [4, 13]) has become quite popular. The latter method employs only few of the original matrix entries of A for its data-sparse approximation by hierarchical matrices. Since one is usually interested in approximations $\tilde{A}$ of A leading to a sufficiently accurate result when $\tilde{A}$ or its inverse is applied to arbitrary vectors, it is common to approximate all blocks of the matrix partition uniformly with a prescribed accuracy. The universality of the approximation to the discrete operator A comes at a high price due to the generation of information that may be redundant for the solution of the linear system. When the quantity of interest is the error of the solution x of the linear system $Ax = b$, $\tilde{A}$ obtained from usual ACA may not be the ideal approximation to A. In such a situation, certain parts of the discrete operator A may be more important than others. Conversely, depending on the right-hand side b some blocks may not be that important for the error of x.

In this article we propose to construct the hierarchical matrix approximation in a block-adaptive way, i.e., in addition to the adaptivity of ACA on each block we add another level of adaptivity to the construction of the hierarchical matrix. This variant of ACA will be referred to as BACA. In order to find an $\mathcal{H}$-matrix approximation that is better suited for the solution error, we employ techniques known from adaptivity together with suitable error estimators. The strategy of error estimation is well known in the context of numerical methods for partial differential or integral equations. Adaptive methods usually focus on mesh refinement. Here, these strategies will be used to successively improve block-wise low-rank approximations, whereas the underlying grid (and the hierarchical block structure) does not change. Although the ideas of this approach are presented in connection with ACA, they can also be applied to other low-rank approximation techniques.

In contrast to constructing the approximation $\tilde{A}$ of A the usual way and solving a single linear system $\tilde{A}x = b$, we generate a sequence of $\mathcal{H}$-matrix approximations A_k to A and solve each linear system $A_k x_k = b$ for x_k. At first glance this seems to be more costly than solving $\tilde{A}x = b$ only once. However, the construction of the next approximation A_{k+1} can be steered by the approximate solution x_k, and x_{k+1} can be efficiently computed as an update of x_k. Furthermore, the accuracy of x_k in the iterative solver can be adapted to the error estimator. This allows to interlace the assembling process of the matrix with the iterative solution of the linear system. In addition to reducing the numerical effort for assembling and storing the matrix, this approach has the practical advantage that the block-wise accuracy of ACA is not a required parameter any more. Choosing this parameter in usual ACA approximations is not obvious as the relation between the block-wise accuracy and the error of the solution usually depends on the respective problem. With the new approach presented in this article the accuracy of the approximation is automatically adapted.

The article is organised as follows. At first the model problem is formulated for which the new method will be explained. After that we give a brief overview of hierarchical matrices and ACA as these methods are the foundations of our new method. Suitable error estimators will be proposed and investigated with respect to efficiency and reliability in section four. Using these, a marking and refinement strategy will be presented and analysed. Numerical examples will be presented at the end of the article which show that the new method is more efficient than ACA with respect to both computational time and required storage in situations where the solution in some sense contains structural differences or situations that result from locally over-refined meshes.

2 Model Problem

As a model problem we consider the Laplace equation on a Lipschitz domain $\Omega \subset \mathbb{R}^d$ (or its complement $\Omega^c := \mathbb{R}^d \setminus \overline{\Omega}$), i.e.

$$\begin{aligned} -\Delta u &= 0 \quad \text{in } \Omega \text{ (or } \Omega^c), \\ u &= g \quad \text{on } \partial\Omega, \end{aligned} \tag{1}$$

where the Dirichlet data g is a given function in the trace space $H^{1/2}(\partial\Omega)$ of the Sobolev space $H^1(\Omega)$. This kind of problem (in particular the exterior boundary value problem with suitable conditions at infinity) can be treated via boundary integral formulation using the fundamental solution

$$S(x) = \begin{cases} -\frac{1}{2\pi} \ln |x|, & d = 2, \\ \frac{1}{(d-2)\omega_d} |x|^{2-d}, & d > 2, \end{cases}$$

for $x \in \mathbb{R}^d \setminus \{0\}$, where ω_d denotes the area of the d-dimensional unit sphere. Defining the operators

$$\begin{aligned} \mathcal{V}\psi(x) &:= \int_{\partial\Omega} \psi(y)\, S(x-y)\, \mathrm{d}s_y \quad \text{(single layer potential)}, \\ \mathcal{K}\phi(x) &:= \int_{\partial\Omega} \phi(y)\, \partial_{n(y)} S(x-y)\, \mathrm{d}s_y \quad \text{(double layer potential)}, \end{aligned}$$

the solution of (1) is given via the *representation formula*

$$u(x) = \mathcal{V}\psi(x) - \mathcal{K}g(x), \quad x \in \mathbb{R}^d \setminus \partial\Omega,$$

where the Neumann data $\psi \in H^{-1/2}(\partial\Omega)$ is the solution of the boundary integral equation

$$\mathcal{V}\psi = \left(\frac{1}{2}\mathcal{I} + \mathcal{K}\right) g \quad \text{on } \partial\Omega. \tag{2}$$

If we extend the L^2-scalar product to a duality pairing between $H^{-1/2}(\partial\Omega)$ and $H^{1/2}(\partial\Omega)$, the boundary integral equation (2) can be stated in variational form as

$$(\mathcal{V}\psi, \psi')_{L^2} = (\left(\frac{1}{2}\mathcal{I} + \mathcal{K}\right) g, \psi')_{L^2}, \quad \psi' \in H^{-1/2}(\partial\Omega).$$

Due to the coercivity of the bilinear form $(\mathcal{V}\cdot,\cdot)_{L^2}$, the Riesz-Fischer theorem yields the existence of the unique solution $\psi \in H^{-1/2}(\partial\Omega)$.

In order to solve (2) or its variational form numerically, Galerkin discretisations are often used. In the case $d = 3$, the boundary $\partial\Omega$ is decomposed into a regular partition $\mathcal{T}$ consisting of N triangles. Let the set $\{\psi_1, \ldots, \psi_N\}$ denote a basis of the the space of piecewise constant functions $\mathcal{P}_0(\mathcal{T})$. Then for $\psi = \sum_{j=1}^{N} x_j\psi_j$, $x_j \in \mathbb{R}$, the discretisation of the variational problem reads

$$\begin{aligned}(\left(\frac{1}{2}\mathcal{I}+\mathcal{K}\right) g, \psi_i)_{L^2} = (\mathcal{V}\psi, \psi_i)_{L^2} &= \int_{\partial\Omega}\left[\int_{\partial\Omega} S(x-y)\psi(y)\,\mathrm{d}s_y\right]\psi_i(x)\,\mathrm{d}s_x \\ &= \sum_{j=1}^{N} x_j\left[\int_{\partial\Omega}\int_{\partial\Omega} S(x-y)\psi_j(y)\psi_i(x)\,\mathrm{d}s_y\,\mathrm{d}s_x\right], \quad i = 1, \ldots, N,\end{aligned}$$

which due to the Riesz-Fischer theorem is uniquely solvable, too. With

$$a_{ij} := \int_{\partial\Omega}\int_{\partial\Omega} S(x-y)\psi_j(y)\psi_i(x)\,\mathrm{d}s_y\,\mathrm{d}s_x \quad \text{and} \quad b_i := (\left(\frac{1}{2}\mathcal{I}+\mathcal{K}\right) g, \psi_i)_{L^2}, \quad i, j = 1, ..., N,$$

the above stated discretisation results in the system of linear equations

$$Ax = b, \quad A \in \mathbb{R}^{N\times N},\ b \in \mathbb{R}^N. \tag{3}$$

Since the matrix A is fully populated, our goal is to reduce the required storage for A and the numerical effort for its construction to logarithmic-linear complexity. To this end, hierarchical matrices will be employed; see Sect. 3. The efficiency of hierarchical matrices is based on a blockwise low-rank approximation. The construction of these low-rank approximations is often done via adaptive cross approximation; see Sect. 3.1. While in hierarchical matrix approximations all blocks are commonly approximated with the same prescribed accuracy, in this article we propose to construct the hierarchical matrix approximation in a block-adaptive way, i.e., in addition to the adaptivity of ACA on each block we add another level of adaptivity to the construction of the hierarchical matrix.

Although we consider the model problem (1), the method presented in this article can be applied in more general situations as long as the matrix A results from finite element discretisation of an integral representation of the problem. Examples are the Helmholtz equation, Lamé equations, the Lippmann-Schwinger equation, fractional diffusion problems, the Gauss transform and many others.

3 Hierarchical Matrices and the Adaptive Cross Approximation

This section gives a short overview of the *hierarchical matrix* ($\mathcal{H}$-matrix) structure and the ACA method, which we are going to extend. $\mathcal{H}$-matrices were introduced by Hackbusch [21] and Hackbusch and Khoromskij [22]. Their efficiency is based on a suitable hierarchical partitioning of the matrix into sub-blocks and the blockwise approximation by low-rank matrices.

We consider matrices $A \in \mathbb{R}^{N\times N}$ whose entries result from the discretisation of a non-local operator such as the boundary integral formulation of the boundary value problem described in Sect. 2. Although A is fully populated, it can often be approximated by low-rank matrices on suitable blocks $t\times s$ consisting of rows and columns $t, s \subset I := \{1, \ldots, N\}$, of A, i.e.

$$A_{ts} \approx UV^T, \quad U \in \mathbb{R}^{t\times r},\ V \in \mathbb{R}^{s\times r},$$

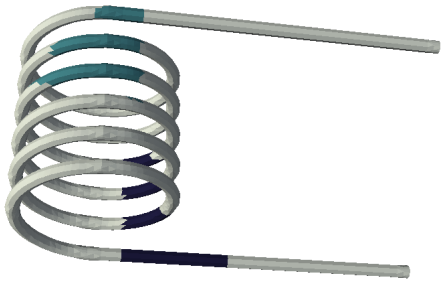

Figure 1: Two clusters X_t and X_s corresponding to an admissible block $t \times s$.

where the rank r is small compared with $|t|$ and $|s|$. If A discretises an integral representation or the inverse of second-order elliptic partial differential operators, then the condition

$$\min\{\operatorname{diam} X_t, \operatorname{diam} X_s\} < \beta \operatorname{dist}(X_t, X_s), \quad \beta > 0, \tag{4}$$

on the block $t \times s$ turns out to be appropriate; see [5]. Here, X_t is the union of the supports $X_i := \operatorname{supp} \psi_i$, $i \in t$, and

$$\operatorname{diam} X = \sup_{x,y \in X} |x - y| \quad \text{and} \quad \operatorname{dist}(X, Y) = \inf_{x \in X,\, y \in Y} |x - y|$$

denote the diameter and the distance of two bounded sets $X, Y \subset \mathbb{R}^d$. All blocks $t \times s$ satisfying this condition are called *admissible*; see Fig. 1.

A partition P of the matrix indices $I \times I$ is usually constructed using *cluster trees*, i.e. trees T_I satisfying the conditions

(1) I is the root of the tree,

(2) if $v \in T_I$ is not a leaf, then v has sons $v_1, \ldots, v_s \in T_I$ so that $v = v_1 \dot\cup \ldots \dot\cup v_s$.

The ℓ-th level of T_I will be referred to as $T_I^{(\ell)}$, $\ell = 0, \ldots, L$, where L is the depth of T_I. The set of leaves of the tree T_I is denoted by $\mathcal{L}(T_I)$. Given a cluster tree T_I for I, a block cluster tree $T_{I \times I}$ for $I \times I$ is constructed by recursive subdivision descending the cluster tree T_I for the rows and for the columns concurrently until (4) is satisfied or one the clusters cannot be subdivided any further. Hence, a matrix partition consisting of blocks either satisfying (4) or being small can be found as the leaves $P := \mathcal{L}(T_{I \times I})$ of $T_{I \times I}$. P is the union of the sets of admissible and non-admissible blocks P_{adm} and $P_{\text{non-adm}}$, respectively. The *sparsity constant* c_{sp} is defined as

$$c_{\text{sp}} := \max \left\{ \max_{t \in T_I} c_{\text{sp}}^r(t), \max_{s \in T_I} c_{\text{sp}}^c(s) \right\},$$

where

$$c_{\text{sp}}^r(t) := |\{s \subset I : t \times s \in P\}|$$

denotes the maximum number of blocks $t \times s$ contained in P for a given cluster $t \in T_I$ and

$$c^c_{\mathrm{sp}}(s) := |\{t \subset I : t \times s \in P\}|$$

the maximum number of blocks $t \times s \in P$ for a cluster $s \in T_I$. For more details on cluster trees (in particular for their construction) the reader is referred to [8].

Given a matrix partition P, we define the set of $\mathcal{H}$-matrices with blockwise rank at most r by

$$\mathcal{H}(P, r) := \left\{ M \in \mathbb{R}^{I \times I} : \operatorname{rank} M_b \leq r \text{ for all } b \in P_{\mathrm{adm}} \right\}.$$

Due to the structure of hierarchical matrices and due to the fact that each block $t \times s$ has complexity $r(|t| + |s|)$, the complexity of the whole $\mathcal{H}$-matrix can be shown to be of the order $r\, N \log N$; see [8].

3.1 Matrix construction

There are several ways how low-rank approximations of a matrix block can be constructed. For matrices A resulting from the discretisation of non-local operators the *adaptive cross approximation* (ACA) method introduced in [4, 13, 10] constructs the low-rank approximation using only few of the original entries of A. Non-admissible blocks cannot be approximated but are small, so they are computed entry by entry. We focus on a single admissible block $A_{ts} \in \mathbb{R}^{t \times s}$ of A. With $R_0 := A_{ts}$ we define two sequences of vectors $u_r \in \mathbb{R}^t$ and $v_r \in \mathbb{R}^s$ for $r = 1, 2, 3, \ldots$ by the following algorithm.

Algorithm 1 Adaptive Cross Approximation (ACA)

```
Let r = 1; Z = ∅; ε_ACA > 0
repeat
    find i_r by some rule
    ṽ_r := A_{i_r,s}
    for l = 1, ..., r − 1 do ṽ_r := ṽ_r − (u_l)_{i_r} v_l
    end for
    Z := Z ∪ {i_r}
    if ṽ_r does not vanish then
        j_r := argmax_{j∈s} |(ṽ_r)_j|; v_r := (ṽ_r)^{−1}_{j_r} ṽ_r
        u_r := A_{t,j_r}
        for l = 1, ..., r − 1 do u_r := u_r − (v_l)_{j_r} u_l
        end for
        r := r + 1
    end if
until ‖u_{r+1}‖_2 ‖v_{r+1}‖_2 ≤ ε_ACA(1−β)/(1+ε_ACA) ‖S_r‖_F or Z = t
```

The matrix

$$S_r := \sum_{l=1}^{r} u_l v_l^T \tag{5}$$

has rank at most r and can be shown (under reasonable assumptions) to be an approximation of A_{ts} with remainder $R_r := A_{ts} - S_r$ having relative accuracy

$$\|R_r\|_F \leq \varepsilon_{\mathrm{ACA}} \|A_{ts}\|_F, \tag{6}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. The set Z collects all vanishing rows of the matrix R_r. The vectors u_r and $\tilde{v}_r$ are columns and rows of R_{r-1}, i.e.

$$u_r = (R_{r-1})_{tj_r}, \quad \tilde{v}_r = (R_{r-1})_{i_r s}.$$

The row indices i_r have to be chosen such that the Vandermonde matrix corresponding to the system in which the approximation error is to be estimated is non-singular; for details see [8]. If the i_r-th row of R_{r-1} is nonzero and hence is used as v_r after scaling $\tilde{v}_r$ with $1/(\tilde{v}_r)_{j_r}$, it is also added to Z since the i_r-th row of the following remainder R_r will vanish. It can be shown that the numerical effort of ACA is of the order $|Z|^2(|t| + |s|)$.

4 Error Estimators and the Block-Adaptive Cross Approximation (BACA)

The usual way of treating discrete problems (3) is to construct an hierarchical matrix approximation $\tilde{A}$ (e.g. via ACA) such that each block $\tilde{A}_b$, $b \in P$, satisfies (cf. (6))

$$\|A_b - \tilde{A}_b\|_F \leq \varepsilon \|A_b\|_F \quad \text{for all } b \in P. \tag{7}$$

Due to the independence of the blocks, this can be done in parallel; see [11]. While (7) guarantees local properties of the $\mathcal{H}$-matrix approximation, the global impact of this condition is difficult to estimate. Notice that (7) still implies the obvious global property

$$\|A - \tilde{A}\|_F \leq \varepsilon \|A\|_F,$$

however, for instance the eigenpairs of A and $\tilde{A}$ are not as accurate in general and one has to apply suitable techniques (see [9]) in order to be able to guarantee spectral equivalence of A and $\tilde{A}$. In other words, the low-rank approximation is usually constructed regardless of the importance of the respective block for global properties of the matrix.

$\tilde{A}$ obtained from usual ACA may not be the ideal approximation to A when the norm of the solution error of the associated solution is the measure. In order to find an $\mathcal{H}$-matrix approximation that is better suited for this problem, we employ techniques known from adaptivity together with suitable error estimators. The strategy of error estimation is well known in the context of numerical methods for partial differential or integral equations. There are several types of a posteriori error estimators. The method of this article is inspired by the $(h - h/2)$-version investigated in [17]. Adaptive methods usually focus on mesh refinement. Here, these strategies will be used to successively improve block-wise low-rank approximations, whereas the underlying grid (and the hierarchical block structure P) does not change.

In contrast to constructing the matrix approximation $\tilde{A}$ of A in the usual way (via ACA) and solving a single linear system $\tilde{A}x = b$, we build a sequence of $\mathcal{H}$-matrix approximations A_k of A and solve each linear system $A_k x_k = b$ for x_k. At first glance this seems to be more costly than solving $\tilde{A}x = b$ only once. However, the approximation of A_{k+1} can be steered by the approximate solution x_k, and x_{k+1} can be computed as an update of x_k. Furthermore, the accuracy of x_k can be low for small k and we can adapt it to the estimated error.

The following method will be based on *residual error estimators*. Notice that if $\mathcal{A} : V \to V'$ denotes the operator that is discretised by A and u_h refers to the Galerkin approximation of the exact solution to $\mathcal{A}u = f$, then for the residual error $\|f - \mathcal{A}u_h\|_{V'}$ we have

$$c_B\|u - u_h\|_V \leq \|f - \mathcal{A}u_h\|_{V'} = \sup_{\varphi \in V} \frac{|a(u - u_h, \varphi)|}{\|\varphi\|_V} \leq c_S\|u - u_h\|_V$$

provided that the continuous bilinear form $a : V \times V \to \mathbb{R}$ corresponding to $\mathcal{A}$, i.e. $|a(u, v)| \leq c_S\|u\|_V\|v\|_V$ for all $u, v \in V$, satisfies an *inf-sup condition*

$$\inf_{u \in V} \sup_{v \in V} \frac{|a(u, v)|}{\|u\|_V\|v\|_V} \geq c_B.$$

Hence, the solution error $\|u - u_h\|_V$ is equivalent with the residual error $\|f - \mathcal{A}u_h\|_{V'}$ and in the rest of this section we may focus on the residual error.

In the following algorithm, $\hat{A}_k$ denotes a better approximation of A than A_k, i.e. we assume that the *saturation condition*

$$\|\hat{A}_k x_k - Ax_k\| \leq c_{\text{sat}}\|A_k x_k - Ax_k\| \tag{8}$$

holds with $0 < c_{\text{sat}} < 1$. A natural choice for $\hat{A}_k$ is the improved approximation that results from A_k by adding a fixed number of additional steps of ACA for each admissible block and by setting $(\hat{A}_k)_b = A_b$ for all other blocks $b \in P_{\text{non-adm}}$. We define the matrix

$$W_k := A_k - \hat{A}_k.$$

Then $(W_k)_b = 0$ for all $b \in P_{\text{non-adm}}$.

Algorithm 2 Block-adaptive ACA

1. Start with a coarse $\mathcal{H}$-matrix approximation A_0 of A and set $k = 0$.
2. Solve the linear system $A_k x_k = b$ for x_k with residual error δ_k (use x_{k-1} as a starting vector of an iterative solver; $x_{-1} := 0$).
3. Given $0 < \theta < 1$, find a set of marked blocks $M_k \subset P_{\text{adm}}$ with minimal cardinality such that

$$\eta_k(M_k) \geq \theta\, \eta_k, \tag{9}$$

   where $\eta_k^2(M) := \sum_{t \times s \in M} \|(W_k)_{ts}(x_k)_s\|_2^2$ and $\eta_k := \eta_k(P_{\text{adm}})$.
4. Let

$$A_{k+1} = \begin{cases} (\hat{A}_k)_b, & b \in M_k, \\ (A_k)_b, & b \in P \setminus M_k. \end{cases}$$

5. If $\eta_k > \varepsilon_{\text{BACA}}$ increment k and go to 2.

Notice that Algorithm 1 contains a stopping parameter ε_{ACA}, which describes the desired accuracy of the low-rank approximation for the respective block. While it is difficult to choose ε_{ACA} so that the resulting solution of the linear system satisfies a prescribed accuracy, the parameter $\varepsilon_{\text{BACA}}$ in

Algorithm 2 gives an upper bound on the residual error $\|b - Ax_k\|_2$ of x_k as we shall see in the next Lemma 1.

It seems that two $\mathcal{H}$-matrices A_k and $\hat{A}_k$ are required for the previous algorithm. Since they are strongly related, it is actually sufficient to store only $\hat{A}_k$. Notice that the ACA steps required for the construction of A_{k+1} can be adopted from $\hat{A}_k$, and $\hat{A}_{k+1}$ can be computed from $\hat{A}_k$ by improving the blocks contained in M_k.

In addition to the locatability of the error estimator

$$\eta_k^2 = \sum_{t\times s\in P_{\mathrm{adm}}} \|(W_k)_{ts}(x_k)_s\|_2^2 \tag{10}$$

introduced in Algorithm 2 its reliability is a main issue for a successful adaptive procedure. In particular, the dependence on the residual error $\delta_k := \|b - A_k x_k\|_2$ of the solution x_k of $A_k x_k = b$ in Algorithm 2 has to be investigated.

Lemma 1. *Let assumption* (8) *be valid and let* $\delta_k \le \alpha\|W_k x_k\|_2$ *with some constant* $\alpha \ge 0$. *Then* η_k *is reliable, i.e., it holds*

$$\|b - Ax_k\|_2 \le \frac{1+\alpha(1+c_{\mathrm{sat}})}{1-c_{\mathrm{sat}}}\,\|W_k x_k\|_2 \le \sqrt{c_{\mathrm{sp}}L}\,\frac{1+\alpha(1+c_{\mathrm{sat}})}{1-c_{\mathrm{sat}}}\,\eta_k.$$

Proof. We make use of the decomposition of $A \in \mathcal{H}(P,k)$ into a sum of level matrices $A^{(\ell)}$, which contain the blocks $b \in P$ of A belonging to the ℓ-th level of the block cluster tree $T_{I\times I}$

$$A = \sum_{\ell=1}^{L} A^{(\ell)}.$$

Notice that $A^{(\ell)}$ is a Cartesian product block matrix. Due to $(\sum_{i=1}^n a_i)^2 \le n\sum_{i=1}^n a_i^2$ for all $a_i \in \mathbb{R}$, $i = 1,\dots,n$, we have

$$\begin{aligned}
\|W_k x_k\|_2^2 &\le \left(\sum_{\ell=1}^{L} \|W_k^{(\ell)} x_k\|_2\right)^2 \le L\sum_{\ell=1}^{L} \|W_k^{(\ell)} x_k\|_2^2 = L\sum_{\ell=1}^{L}\sum_{t\in T_I^{(\ell)}} \|\sum_{s:t\times s\in P} (W_k)_{ts}(x_k)_s\|_2^2 \\
&\le L\sum_{\ell=1}^{L}\sum_{t\in T_I^{(\ell)}} \left(\sum_{s:t\times s\in P} \|(W_k)_{ts}(x_k)_s\|_2\right)^2 \le c_{\mathrm{sp}} L\sum_{\ell=1}^{L}\sum_{t\in T_I^{(\ell)}}\sum_{s:t\times s\in P} \|(W_k)_{ts}(x_k)_s\|_2^2 \\
&= c_{\mathrm{sp}} L \sum_{t\times s\in P} \|(W_k)_{ts}(x_k)_s\|_2^2 = c_{\mathrm{sp}} L\,\eta_k^2.
\end{aligned}$$

The lemma follows from

$$\begin{aligned}
\|b - Ax_k\|_2 &\le \|b - A_k x_k\|_2 + \|W_k x_k\|_2 + \|\hat{A}_k x_k - Ax_k\|_2 \le (1+\alpha)\|W_k x_k\|_2 + c_{\mathrm{sat}}\|A_k x_k - Ax_k\|_2 \\
&\le (1+\alpha)\|W_k x_k\|_2 + c_{\mathrm{sat}}(\alpha\|W_k x_k\|_2 + \|b - Ax_k\|_2).
\end{aligned}$$

□

Although the efficiency of the error estimator η_k, i.e. $\eta_k \lesssim \|b - Ax_k\|_2$, can be observed in numerical experiments, its provability is still an open question. Nevertheless, the computable expression $\|W_k x_k\|_2$ can be used as a lower bound for $\|b - Ax_k\|_2$.

Lemma 2. *Let* (8) *be valid and let* $\delta_k \le \alpha \|W_k x_k\|_2$ *with* $0 \le \alpha \le 1/2$. *Then*

$$\|b - Ax_k\|_2 \ge \frac{1 - \alpha(1 + c_{\text{sat}})}{1 + c_{\text{sat}}} \|W_k x_k\|_2.$$

Proof. The saturation assumption (8) yields

$$\begin{aligned}\|W_k x_k\|_2 &\le \|A_k x_k - Ax_k\|_2 + \|Ax_k - \hat{A}_k x_k\|_2 \le (1 + c_{\text{sat}})\|Ax_k - A_k x_k\|_2 \\ &\le (1 + c_{\text{sat}})(\|b - Ax_k\|_2 + \|b - A_k x_k\|_2) \le (1 + c_{\text{sat}})\|b - Ax_k\|_2 + (1 + c_{\text{sat}})\alpha \|W_k x_k\|_2.\end{aligned}$$

□

In the following lemma the convergence of the error estimator η_k is investigated. Its convergence is a consequence of the *Dörfler marking strategy* (9). To prove it, we use the obvious property

$$\lim_{k\to\infty} \|A_{k+1} - A_k\| = 0 = \lim_{k\to\infty} \|\hat{A}_{k+1} - \hat{A}_k\|. \tag{11}$$

Lemma 3. *Assume that the sequence* $\{\|A_k^{-1}\|_2\}_{k\in\mathbb{N}_0}$ *is bounded and that* $\delta_k \to 0$ *for* $k \to \infty$. *Then it holds that*

$$\eta_{k+1}^2 \le q\, \eta_k^2 + z_k, \tag{12}$$

where z_k *converges to zero and* $q := 1 - \frac{1}{2}\theta^2 < 1$. *Furthermore,* $\lim_{k\to\infty} \eta_k = 0$.

Proof. Using the definition of A_{k+1}, η_{k+1}^2 reads

$$\sum_{t\times s\in P} \|(W_{k+1})_{ts}(x_{k+1})_s\|_2^2 = \sum_{t\times s\in M_k} \|(\hat{A}_k - \hat{A}_{k+1})_{ts}(x_{k+1})_s\|_2^2 + \sum_{t\times s\notin M_k} \|(A_k - \hat{A}_k)_{ts}(x_{k+1})_s\|_2^2.$$

Due to Young's inequality and (9) we have for the second term

$$\begin{aligned}\sum_{t\times s\notin M_k} \|(W_k)_{ts}(x_{k+1})_s\|_2^2 &\le (1+\epsilon)\,\eta_k^2(P \setminus M_k) + (1 + 1/\epsilon) \sum_{t\times s\notin M_k} \|(W_k)_{ts}(x_{k+1} - x_k)_s\|_2^2 \\ &= (1+\epsilon)[\eta_k^2 - \eta_k^2(M_k)] + (1 + 1/\epsilon) \sum_{t\times s\notin M_k} \|(W_k)_{ts}(x_{k+1} - x_k)_s\|_2^2 \\ &\le (1+\epsilon)(1-\theta^2)\,\eta_k^2 + (1 + 1/\epsilon) \sum_{t\times s\notin M_k} \|(W_k)_{ts}(x_{k+1} - x_k)_s\|_2^2\end{aligned}$$

for all $\epsilon > 0$. The choice $\epsilon := \frac{1}{2}\theta^2/(1-\theta^2)$ and

$$\begin{aligned}
z_k &:= \sum_{t\times s\in M_k} \|(\hat{A}_k - \hat{A}_{k+1})_{ts}(x_{k+1})_s\|_2^2 + (1+1/\epsilon) \sum_{t\times s\notin M_k} \|(W_k)_{ts}(x_{k+1}-x_k)_s\|_2^2 \\
&\le c_{\mathrm{sp}} L \max_{t\times s\in P} \|(\hat{A}_k - \hat{A}_{k+1})_{ts}\|_2^2 \, \|x_{k+1}\|_2^2 \\
&\quad + (1+1/\epsilon) c_{\mathrm{sp}} L \max_{t\times s\in P} \|(W_k)_{ts}\|_2^2 \, \|A_{k+1}^{-1}\|_2^2 \, \|A_{k+1}x_{k+1} - A_{k+1}x_k\|_2^2 \\
&\le c_{\mathrm{sp}} L \Big[\|\hat{A}_k - \hat{A}_{k+1}\|_2^2 \, \|x_{k+1}\|_2^2 \\
&\quad + (1+1/\epsilon)\|W_k\|_2^2 \, \|A_{k+1}^{-1}\|_2^2 \left(\|A_{k+1}x_{k+1} - b\|_2^2 + \|(A_{k+1}-A_k)x_k\|_2^2 + \|A_k x_k - b\|_2^2 \right) \Big] \\
&\le c_{\mathrm{sp}} L \Big[\|\hat{A}_k - \hat{A}_{k+1}\|_2^2 \, \|x_{k+1}\|_2^2 \\
&\quad + (1+1/\epsilon)\|W_k\|_2^2 \, \|A_{k+1}^{-1}\|_2^2 \left(\delta_{k+1}^2 + \|A_{k+1}-A_k\|_2^2 \, \|x_k\|_2^2 + \delta_k^2 \right) \Big]
\end{aligned}$$

leads to the first part of the assertion due to the boundedness of $\|A_k^{-1}\|_2$, the convergence of δ_k to 0 and (11).

For the proof of the second part we pursue the ideas of the estimator reduction principle, which was introduced in [3]. Let $z > 0$ be a number with $z_k \le z$ for all k. With the estimator reduction (12) it follows that

$$\begin{aligned}
\eta_{k+1}^2 &\le q\,\eta_k^2 + z_k \le q^2\,\eta_{k-1}^2 + q\,z_{k-1} + z_k \le \ldots \le q^{k+1}\eta_0^2 + \sum_{i=0}^{k} q^{k-i} z_i \\
&\le q^{k+1}\eta_0^2 + z \sum_{l=0}^{k} q^l \le \eta_0^2 + \frac{z}{1-q}.
\end{aligned}$$

Hence, the sequence $\{\eta_k\}_{k\in\mathbb{N}_0}$ is bounded and we define $M := \limsup_{k\to\infty} \eta_k^2$. Using the estimator reduction (12) once more, leads to

$$M = \limsup_{k\to\infty} \eta_{k+1}^2 \le q \limsup_{k\to\infty} \eta_k^2 + \limsup_{k\to\infty} z_k = qM.$$

Hence, $M = 0$ and we obtain

$$0 \le \liminf_{k\to\infty} \eta_k \le \limsup_{k\to\infty} \eta_k = 0$$

and finally $\lim_{k\to\infty} \eta_k = 0$. □

Theorem 1. *The residuals $r_k := b - Ax_k$ of the sequence $\{x_k\}_{k\in\mathbb{N}}$ constructed by Algorithm 2 converge to zero.*

Proof. With Lemma 1 and Lemma 3 we obtain that

$$\|b - Ax_k\|_2 \le \sqrt{c_{\mathrm{sp}} L}\, \frac{1 + \alpha(1 + c_{\mathrm{sat}})}{1 - c_{\mathrm{sat}}}\, \eta_k \to 0.$$

□

5 Numerical Experiments

The numerical experiments are subdivided into three parts. At first we investigate the storage requirements of BACA. After that the quality of the proposed error estimator is discussed. Finally, the third part treats the acceleration of the matrix approximation and solution of the boundary value problem in comparison with ACA. In order to be able to compare these results with the results obtained from ACA, we prescribe the same error of u_h. Therefore,

$$e_h := \frac{\|u - u_h\|_{L^2}}{\|u\|_{L^2}}$$

describes the relative error of the solution u_h.

The approximation steps are executed without any parallelisation in both cases. We use the conjugate gradient method as a solver for the arising linear systems of equations without preconditioning. In the case of BACA, the accuracy of the iterative solver is adapted to the size of $\|W_k x_k\|_2$; see Lemma 1. For all the experiments that are based on ACA the accuracy of CG is set to 10^{-8}.

As numerical examples we consider a family of boundary value problems, in which the singularity of the right-hand side is moved towards the boundary of the computational domain Ω, i.e.

$$-\Delta u = 0 \qquad \text{in } \Omega := B_1(0), \tag{13a}$$

$$u(x) = S(x - p_i) \qquad \text{on } \partial\Omega,\ i = 1, 2, 3, 4, \tag{13b}$$

for $p_i = (x_i, 0, 0)^T$ with $x_1 = 10.0$, $x_2 = 1.5$, $x_3 = 1.1$, $x_4 = 1.05$. In these tests we use the following parameters: the minimal block size $b_{\min} = 15$, the blockwise accuracy $\varepsilon_{\text{ACA}} = 10^{-6}$ of ACA, and the admissibility parameter $\beta = 0.8$.

5.1 Storage Reduction

We start with a uniform decomposition of the unit ball into 642 points and 1280 triangles for the boundary value problems (13). Note that ACA does not take into account the right-hand side. Hence the approximation does not depend on the choice of p_i. So, we obtain the left picture of Fig. 2 after the application of ACA to the discrete integral operator in each of the cases $i = 1, \ldots, 4$.

The proposed algorithm BACA provides the results shown in Tab. 1, in which the parameter $\theta = 0.9$ was used. Here, the ranks of $\hat{A}_k$ are ahead of A_k by two ACA steps. The parameter α from Lemma 1 is set to 100. Compared with the approximation obtained from ACA, we obtain lower storage requirements for the matrix approximation $\hat{A}_k$.

The storage reduction and the improved compression rate observed in Tab. 1 are also visible from Fig. 2, where the respective approximation $\hat{A}_k$ of A with its blockwise ranks is shown (green blocks). Therein, red blocks are constructed entry-wise without approximation.

Example (13) has been chosen in order to investigate different right-hand sides which lead to structural differences in the respective solution. Notice that the influence of the singularity on the solution is stronger the smaller the distance of p_i to the boundary of Ω becomes. Hence, for p_i close to the boundary certain parts of the discrete integral operator are more important for the accuracy of the solution than others. Even for a relatively large distance, i.e. $p_1 = (10, 0, 0)^T$, smaller ranks and hence a better compression rate than ACA can be observed; see Fig. 2. This effect is even

i	blockwise rank A_0	$\varepsilon_{\text{BACA}}$	BACA $\|b - Ax_k\|_2$	e_h	storage (MB)	compr. (%)	ACA e_h	storage (MB)	compr. (%)
1	6	1e-08	1.89e-08	0.002	3.43	54.9	0.002	3.85	61.5
2	4	5e-06	6.08e-06	0.033	2.86	45.7	0.033	3.85	61.5
3	3	1e-04	1.48e-04	0.264	2.40	38.4	0.264	3.85	61.5
4	2	5e-04	6.40e-04	0.951	2.10	33.5	0.951	3.85	61.5

Table 1: Numerical results of BACA for four positions of the singularity.

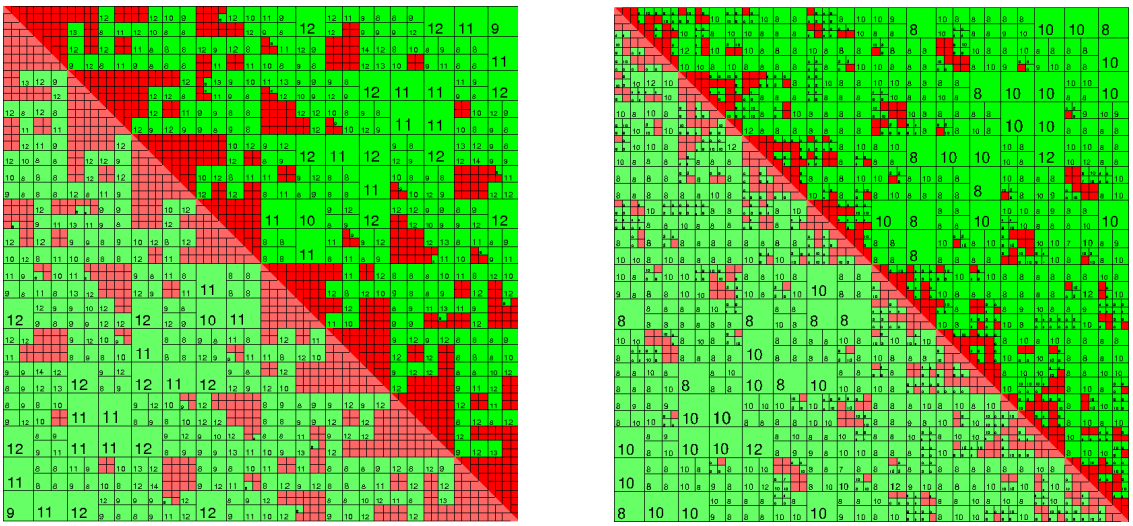

Figure 2: Approximation of the blocks via ACA (left) and via BACA in the case p_1 (right).

stronger if p_i approaches the boundary. The average ranks are shown in Tab. 2. So Tab. 1 and Fig. 2 indicate that the proposed error estimator detects those matrix blocks which are important for the respective problem; see Fig. 3. Therefore, BACA is able to construct an $\mathcal{H}$-matrix approximation that is particularly suited to solve a linear system for a specific right-hand side.

The improved storage requirements can be observed also for finer triangulations of the boundary. Tab. 3 shows the results for the boundary value problem (13) in the case $i = 3$ for several numbers of degrees of freedom N. In addition to the compression rate and the storage requirements, the number of computed entries for the construction of $\hat{A}_k$ are presented. Since the solution u_h is expected to be more accurate the larger N is, the accuracy $\varepsilon_{\text{BACA}}$ of the error estimator has to be adapted to N. All other parameters remain unchanged. In comparison, the results of ACA applied to the boundary value problem (13) in the case $i = 3$ can be seen in Tab. 4.

Hence, BACA requires significantly less original matrix entries than the usual construction via ACA of the $\mathcal{H}$-matrix approximation to obtain approximately the same relative error of u_h.

matrix	ACA	BACA			
		p_1	p_2	p_3	p_4
A_k	12.54	7.44	5.00	3.71	3.08
$\hat{A}_k$	—	8.52	6.62	5.38	5.01

Table 2: Average ranks.

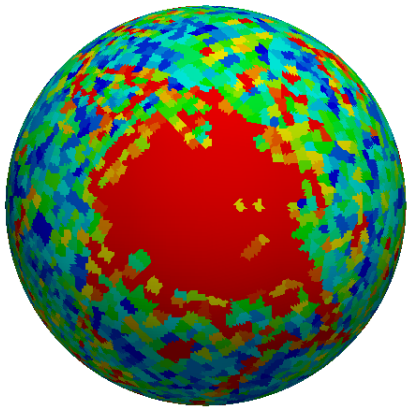

Figure 3: Regions corresponding to large ranks (red).

5.2 Quality of the error estimator

In the following tests we validate the error estimator η_k introduced in (10). At first the reliability statement of Lemma 1 together with the lower bound $\|W_k x_k\|_2$ of Lemma 2 is investigated. Again, the boundary element method combined with BACA is tested with the problem described in (13) and a triangulation with 14338 points and 28672 triangles. The accuracy $\varepsilon_{\text{BACA}}$ of the error estimator is set to 10^{-7}. We start BACA with a coarse approximation A_0 which has been generated by applying four ACA steps to each block. Tab. 5 and in Fig. 4 contain the results for $\alpha = 1/2$ and three refinement parameters θ. The ranks of $\hat{A}_k$ are ahead of A_k by three ACA steps.

The proposed error estimator η_k estimates the residual error $\|b - Ax_k\|_2$ in an appropriate way. The expression $\frac{1}{2}\|W_k x_k\|_2$ can serve as a lower bound for the residual error. Tab. 5 shows that the conver-

N	$\varepsilon_{\text{BACA}}$	$\|b - Ax_k\|_2$	e_h	number of computed entries	storage (MB)	compr. (%)
1 280	1e-04	4.14e-04	0.264	3.22e05	2.48	39.7
7 168	1e-05	2.31e-05	0.077	3.32e06	25.47	13.0
28 672	1e-06	2.56e-06	0.023	2.10e07	160.98	5.1
114 688	1e-06	2.13e-06	0.007	8.76e07	837.92	1.7

Table 3: Numerical results of BACA for p_3 and several N.

N	e_h	number of computed entries	storage (MB)	compr. (%)
1 280	0.264	5.01e05	3.85	61.5
7 168	0.077	4.91e06	37.53	19.8
28 672	0.023	2.66e07	203.70	6.5
114 688	0.007	1.35e08	1034.56	2.1

Table 4: Numerical results of ACA for p_3 and several N.

gence with respect to k is the faster the refinement parameter θ is closer to 1. A larger parameter θ in (9) leads to a larger set M_k of marked blocks, which is likely to produce redundant information in A_{k+1}. Conversely, small refinement parameters θ usually lead to matrix approximations with lower storage requirements.

	η_k			$\|b - Ax_k\|_2$		
k	$\theta = 0.6$	$\theta = 0.7$	$\theta = 0.9$	$\theta = 0.6$	$\theta = 0.7$	$\theta = 0.9$
1	1.36e-05	1.36e-05	1.36e-05	1.52e-05	1.41e-05	1.07e-05
2	1.08e-05	9.77e-06	6.07e-06	1.31e-05	1.19e-05	5.68e-06
3	8.49e-06	6.96e-06	2.71e-06	1.18e-05	9.45e-06	3.16e-06
4	6.78e-06	4.98e-06	1.20e-06	9.78e-06	7.05e-06	1.59e-06
5	5.41e-06	3.57e-06	5.60e-07	8.24e-06	5.55e-06	6.68e-07
6	4.33e-06	2.58e-06	2.56e-07	6.87e-06	4.22e-06	3.46e-07
7	3.47e-06	1.85e-06	1.19e-07	5.93e-06	3.39e-06	1.91e-07
8	2.80e-06	1.32e-06	5.96e-08	4.90e-06	2.42e-06	1.04e-07
9	2.24e-06	9.52e-07	—	4.15e-06	1.85e-06	—
10	1.79e-06	6.96e-07	—	3.53e-06	1.55e-06	—
11	1.44e-06	5.04e-07	—	3.11e-06	1.03e-06	—
12	1.15e-06	3.64e-07	—	2.37e-06	6.84e-07	—
13	9.31e-07	2.65e-07	—	1.99e-06	5.19e-07	—
14	7.53e-07	1.92e-07	—	1.70e-06	4.11e-07	—
15	6.07e-07	1.40e-07	—	1.34e-06	3.19e-07	—

Table 5: Numerical results of BACA for different θ.

The numerical results confirm the theoretical findings of Lemma 1 and Lemma 2. The proposed error estimator η_k is reliable and $\frac{1}{2}\|W_k x_k\|_2$ can be used as a lower bound on the residual error.

5.3 Acceleration of the matrix approximation

The results represent only a part of our aims of the extension of ACA. As already mentioned in the introduction, we are also interested in an acceleration of the matrix approximation and an acceleration of the solution process. Since the computation of matrix entries is by far the most time-consuming part of the boundary element method and we have seen that BACA requires less entries, we can expect improved computational times.

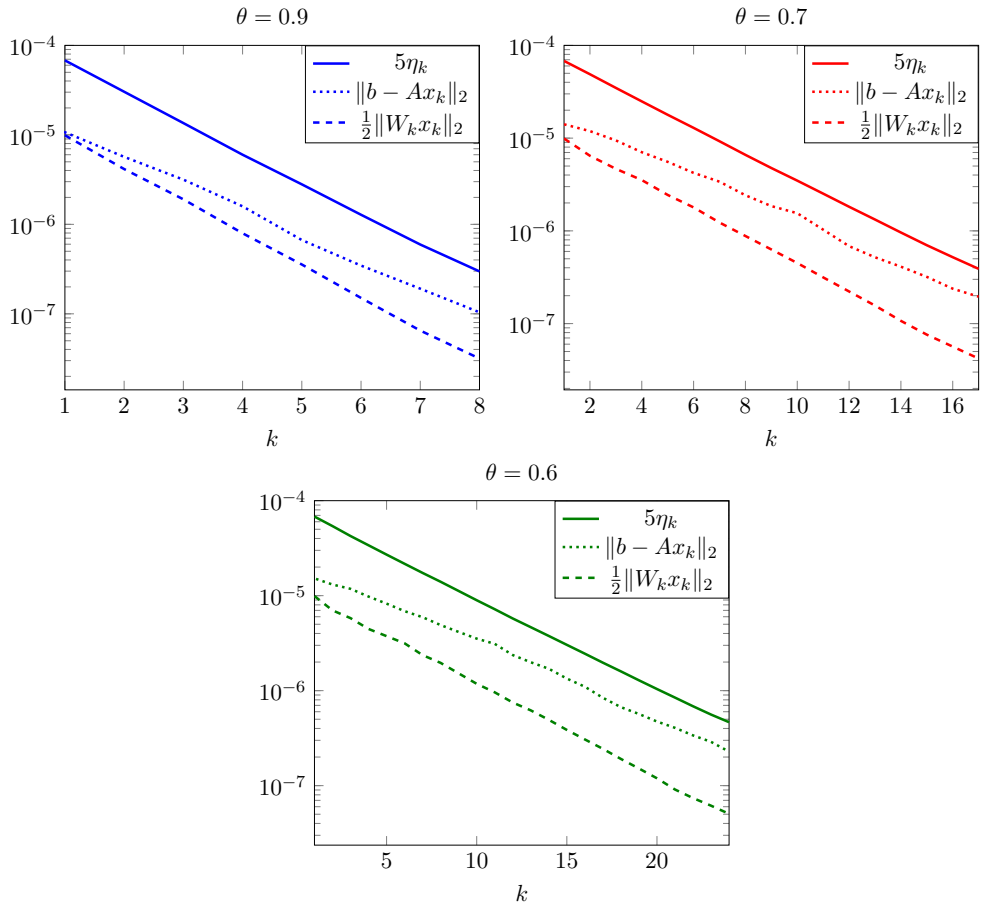

Figure 4: The error estimator, the residual, and the lower bound $\|W_k x_k\|_2$ for several parameters θ.

We consider again the family of boundary value problems (13). The ranks of $\hat{A}_k$ are ahead of A_k by two ACA steps. and we choose $\alpha = 100$. The ratio of the time required for BACA and the time of the ACA can be observed in the Figs. 5 and 6 for two different triangulations. The two figures show that we obtain a lower time consumption even in the case of p_1, where the structural differences in the right-hand side are low. If the singularity is moved closer to the boundary of Ω, BACA speeds up continuously.

One of the biggest advantages of ACA is its linear logarithmic complexity. In order to observe this behaviour, the sphere discretised with 642 points and 1280 triangles is considered. We keep the resulting geometry and increase the number of triangles N for the boundary value problem with p_1. The time columns of Tab. 6 show the expected complexity of ACA and also of the new BACA. The growth factor of N is four and the growth factor of the computational time is between five and six.

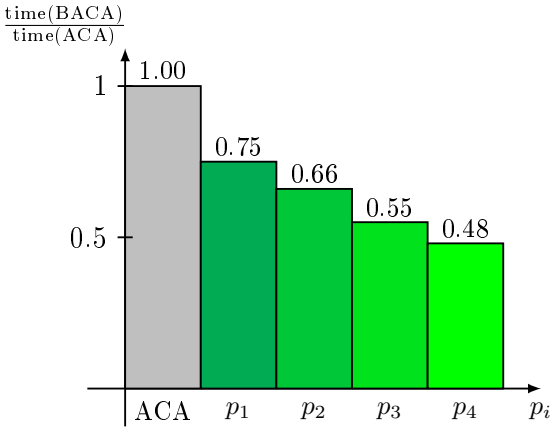

Figure 5: Triangulation with 7 168 triangles

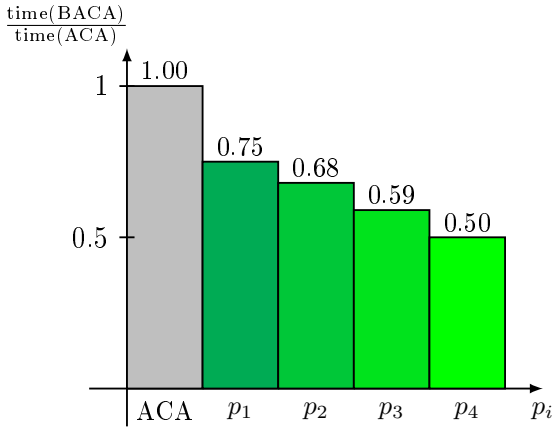

Figure 6: Triangulation with 28 672 triangles

N	BACA			ACA	
	$\|b - Ax_k\|_2$	e_h	time (s)	e_h	time (s)
1 280	1.58e-08	0.002	0.27	0.002	0.31
5 120	1.49e-08	0.002	1.46	0.002	1.83
20 480	1.08e-08	0.003	7.34	0.003	10.20
81 920	4.22e-09	0.004	38.15	0.004	54.15

Table 6: Numerical results of BACA with p_1 and $\varepsilon_{\text{BACA}} = 5 \cdot 10^{-8}$.

Effects on partially overrefined surfaces

In the following tests we apply ACA and BACA to triangulations which are partially overrefined. The boundary of the ellipsoid $\Omega = \{x \in \mathbb{R}^3 \ : \ x_1^2 + x_2^2 + x_3^2/9 = 1\}$ is more refined on the mid-region than on the remaining geometry; see Fig. 7.

The numerical results of BACA for the ellipsoid (p_2, $\varepsilon_{\text{BACA}} = 10^{-6}$) with a growing number of triangles are contained in Tab. 7. The accuracy of the block approximation for ACA is $\varepsilon_{\text{ACA}} = 10^{-6}$. The minimal block size is also given in Tab. 7, which we increase with the growing size of the problem. In order to compare the results, we keep the relative error at the same level.

Inspecting the time columns of Tabs. 7 and 8 reveals that BACA is significantly faster than ACA. For the largest problem BACA is more than twice as fast as ACA. The reason for the acceleration is that the solution does not benefit from the local over-refinement. While the solver based on ACA is not able to detect this, the combination of solution and error estimation used in BACA leads to lower number of CG iterations and computational time. For the largest problem the number of CG iterations required for the solution based on ACA is more than twice as large as for the solution based on BACA. We obtain no significant difference in the storage requirements after the application of the two algorithms. Hence, the observed speed-up results from the coupling of the iterative solver and the error estimator rather than from the reduction of the storage requirements. In addition to these facts, a decreasing number of required iterations k can be observed with a growing number of

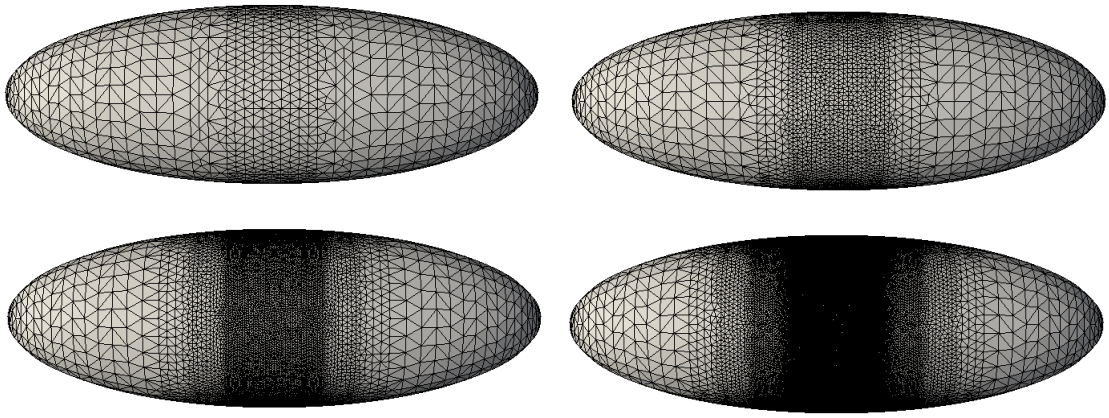

Figure 7: Meshes on the considered ellipsoid.

unknowns N.

N	$b_{\min}$	$\varepsilon_{\text{BACA}}$	$\|b - Ax_k\|_2$	e_h	k	time (s)	CG iterations	storage (MB)
3 452	15	1e-06	1.39e-06	0.034	10	1.22	134	11.61
8 574	30	1e-06	1.16e-06	0.023	10	4.89	273	42.10
30 642	60	1e-06	1.04e-06	0.012	6	34.80	500	227.07
125 948	120	1e-06	1.21e-06	0.006	6	569.36	1119	1608.32

Table 7: Numerical results BACA for the ellipsoid.

N	$b_{\min}$	e_h	k	time (s)	CG iterations	storage (MB)
3 452	15	0.034	10	1.54	196	13.67
8 574	30	0.023	10	6.91	371	48.37
30 642	60	0.012	6	66.63	909	257.59
125 948	120	0.006	6	1206.28	2828	1674.62

Table 8: Numerical results ACA for the ellipsoid.

The ratio of the time required for ACA and for BACA can be seen in Fig. 8. The curves stabilize at a certain constant, which seems to depend on the geometry and the number and the location of refined areas.

References

[1] M. Ainsworth and C. Glusa. Aspects of an adaptive finite element method for the fractional Laplacian: a priori error estimates, efficient implementation and multigrid solver. *Comput.*

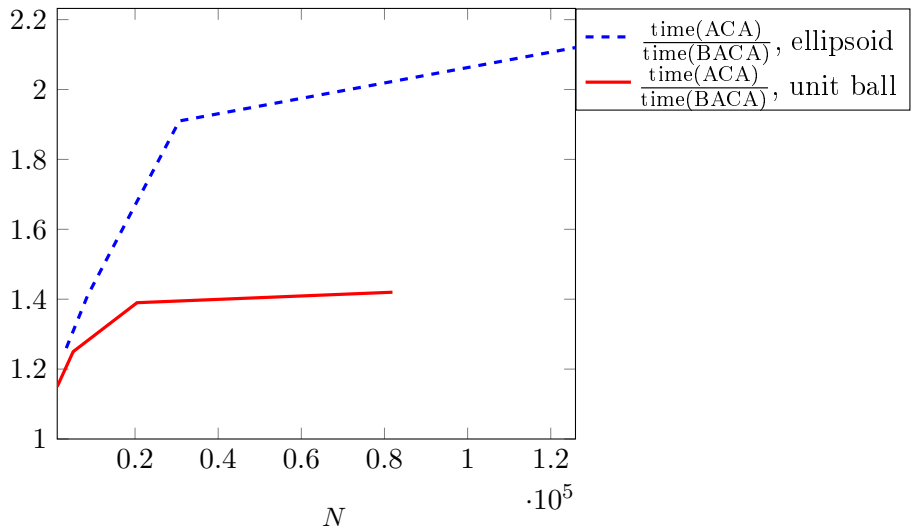

Figure 8: The ratio of the time required for BACA and for ACA for two geometries.

Methods Appl. Mech. Eng, 327:4–35, 2017.

[2] M. Aurada, M. Feischl, T. Führer, M. Karkulik, and D. Praetorius. Efficiency and optimality of some weighted-residual error estimator for adaptive 2D boundary element methods. *Comput. Methods Appl. Math.*, 13(3):305–332, 2013.

[3] A. Aurada, S. Ferraz-Leite, and D. Praetorius. Estimator reduction and convergence of adaptive BEM. *Appl. Numer. Math.*, 62(6):787–801, 2012.

[4] M. Bebendorf. Approximation of boundary element matrices. *Numer. Math.*, 86(4):565–589, 2000.

[5] M. Bebendorf. Efficient inversion of Galerkin matrices of general second-order elliptic differential operators with nonsmooth coefficients. *Math. Comp.*, 74(251):1179–1199, 2005.

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

[7] M. Bebendorf, Approximate inverse preconditioning of finite element discretisations of elliptic operators with nonsmooth coefficients. *SIAM J. Matrix Anal. Appl.*, 27(4):909–929, 2006.

[8] M. Bebendorf. Hierarchical Matrices: A Means to Efficiently Solve Elliptic Boundary Value Problems. Volume 63 of *Lecture Notes in Computational Science and Engineering (LNCSE)*, Springer, Berlin, 2008.

[9] M. Bebendorf, M. Bollhöfer, and M. Bratsch. On the spectral equivalence of hierarchical matrix preconditioners for elliptic problems. *Math. Comp.*, 85(302):2839–2861, 2016.

[10] M. Bebendorf and R. Grzhibovskis. Accelerating Galerkin BEM for linear elasticity using adaptive cross approximation. *Math. Meth. Appl. Sci.*, 29(4):1721–1747, 2006.

[11] M. Bebendorf and R. Kriemann, Fast parallel solution of boundary integral equations and related problems. *Comput. Visual. Sci.*, 8(3-4):121–135, 2005.

[12] M. Bebendorf, Y. Maday, and B. Stamm. Comparison of some reduced representation approximations. In A. Quarteroni and G. Rozza, editors, *Reduced Order Methods for modeling and computational reduction*, volume 8 of *Springer MS&A series*, pages 67–100. 2014.

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

[14] E. Di Nezza, G. Palatucci, and E. Valdinoci. Hitchhiker's guide to the fractional Sobolev spaces. *Bull. Sci. math.*, 136(5):521–573, 2012.

[15] M. D'Elia and M. Gunzburger. The fractional Laplacian operator on bounded domains as a special case of the nonlocal diffusion operator. *Comput. Math. Appl.*, 66(7):1245–1260., 2013,

[16] W. Dörfler. A convergent adaptive algorithm for Poisson's equation. *SIAM J. Numer. Anal.*, 33(3):1106–1124, 1996.

[17] S. Ferraz-Leite and D. Praetorius. Simple a posteriori error estimators for the h-version of the boundary element method. *Computing*, 83(4):135–162, 2008.

[18] T. Gantumur. Adaptive boundary element methods with convergence rates. *Numer. Math.*, 124(3):471–516, 2013.

[19] L. Grasedyck and W. Hackbusch. Constructions and arithmetics of $\mathcal{H}$-matrices. *Computing*, 70(4):295–334, 2003.

[20] L.F. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. Comput. Phys.*, 73(2):325–348, 1987.

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

[22] W. Hackbusch and B.N. Khoromskij. A sparse $\mathcal{H}$-matrix arithmetic. Part II: Application to multi-dimensional problems. *Computing*, 64(1):21–47, 2000.

[23] M. Karkulik, G. Of, and D. Praetorius. Convergence of adaptive 3D BEM for weakly singular integral equations based on isotropic mesh-refinement. *Numerical Methods for Partial Differential Equations*, 29(6):2081–2106, 2013.

[24] Y. Saad. Iterative Methods for Sparse Linear Systems. PWS Publishing Company, Boston (1996).

[25] S.A. Sauter and C. Schwab. Boundary Element Methods. *Springer Series in Computational Mathematics 39*, Springer, Berlin Heidelberg, 2011.

[26] O. Steinbach. Numerical Approximation Methods for Elliptic Boundary Value Problems. Springer, New York, 2008.

[27] E.E. Tyrtyshnikov. Mosaic-skeleton approximations. *Calcolo*, 33(1-2):47–57, Toeplitz matrices: structures algorithms and applications, Cortona, 1996.