Constructing Nested Bases Approximations from the Entries of Non-local Operators

Mario Bebendorf, Raoul Venn

no. 491

Diese Arbeit ist mit Unterstützung des von der Deutschen Forschungsgemeinschaft getragenen Sonderforschungsbereichs 611 an der Universität Bonn entstanden und als Manuskript vervielfältigt worden.

Bonn, Februar 2011

Constructing Nested Bases Approximations from the Entries of Non-local Operators*

M. Bebendorf and R. Venn

Institute for Numerical Simulation, University of Bonn

February 14, 2011

In this article, a method for constructing nested bases approximations to large-scale fully populated discretizations of integral operators is introduced. The scheme uses only few of the matrix entries for approximating the whole matrix. In this sense, it is similar to the adaptive cross approximation method. However, its computational complexity is improved.

1 Introduction

Let $\Omega \subset \mathbb{R}^3$ be an m-dimensional manifold, $m = 2, 3$. We consider matrices $A \in \mathbb{R}^{I \times J}$

$$a_{ij} = \int_\Omega \int_\Omega \kappa(x,y)\psi_i(y)\varphi_j(x)\,\mathrm{d}\mu_y\,\mathrm{d}\mu_x, \quad i \in I,\ j \in J, \tag{1}$$

with index sets I and J, the m-dimensional measure μ and ansatz and trial functions φ_j, ψ_i, $i \in I$ and $j \in J$. For simplicity, κ, φ_j and ψ_i are assumed to be non-negative. The matrix A is usually fully populated, i.e., all of its $|I| \cdot |J|$ entries are non-zero.

Since $|I|$ and $|J|$ are assumed to be large, the aim of this article is to reduce the storage complexity of A in (1) to logarithmic-linear. A typical approach is to exploit smoothness or similar properties of the kernel function $\kappa : \Omega \times \Omega \to \mathbb{R}$. κ is commonly assumed to be *asymptotic smooth*, i.e., for $x \in X$ and $y \in Y$ it holds that

$$|\partial_x^\beta \partial_y^\alpha \kappa(x,y)| \le c_p\, p!\, |x-y|^{-p} |\kappa(x,y)| \quad \text{for all } \alpha, \beta \in \mathbb{N}_0^3, \tag{2}$$

where $p = |\alpha| + |\beta|$. This assumption is known to be valid if, for instance, κ is the singularity function of an elliptic differential operator; see for instance [7]. In this case, suitable sub-blocks of A can be approximated by a matrix of low rank., i.e. the rank k of the approximation

$$UV^T \approx A_{ts}, \quad U \in \mathbb{R}^{t \times k},\ V \in \mathbb{R}^{s \times k}, \tag{3}$$

*This work is supported by DFG collaborative research center SFB 611.

for each block $t \times s \in P$ from a suitable partition P of A depends logarithmically on the prescribed approximation accuracy. In addition to storing an approximation of A efficiently, it is obvious that this approach can be used to speed up the multiplication of A by a vector.

The local approximation with matrices of low rank is exploited by techniques such as the *mosaic skeleton method* [30] and *hierarchical matrices* [20, 23]. The *Barnes-Hut algorithm* [5], the *fast multipole method* [18, 19], *panel clustering* [25] and the interpolation technique [12] are based on similar ideas. Notice that the approximation by blockwise low-rank matrices is not restricted to integral operators. It is also applicable to inverses and the factors of the LU decomposition of second order elliptic boundary value problems even in the case of non-smooth coefficients; see [7].

The construction of approximations (3) is usually done via analytic approximations (interpolation, multipole expansion, etc.) of the kernel function κ. *Kernel independent fast multipole methods* [1, 31, 13] try to avoid explicit kernel expansions using equivalent densities. The *adaptive cross approximation* (ACA) method [6, 10, 8] is a purely algebraic procedure, which allows to construct approximations (3) from few of the original matrix entries. The analytic background of the approximation is used only in the convergence analysis. Although this method can be easily implemented and results in approximations of high-quality in the sense that the rank k of approximation usually differs only slightly from the optimal rank, the storage complexity of the approximation is $k(|I| + |J|) \log(|I| + |J|)$, whereas, for instance, the fast multipole method and its algebraic generalization $\mathcal{H}^2$-*matrices* [24] are able to achieve complexity $k(|I| + |J|)$. The reason for this is that the latter methods construct approximations (3) such that the vectors of U (and V) are chosen from the linear hull of a nested basis $U(t) \in \mathbb{R}^{t \times k}$ which depends on t but not on s.

Nested bases are usually constructed via analytic approximation, i.e. interpolation [24, 22] or multipole expansion. The aim of this article is to construct nested basis approximations by a method which is in the spirit of ACA, i.e. which is based on few of the matrix entries, and hence to bring together the ease of use and efficiency of ACA with the improved storage complexity of nested bases. The new method differs from the procedure presented in [11], which converts $\mathcal{H}$-matrices to $\mathcal{H}^2$-matrices. Our algorithm constructs nested bases approximations without intermediate $\mathcal{H}$-matrix approximation. To this end, we introduce a new form of the adaptive cross approximation, which is adapted towards numerically stable nested basis approximation. A related technique is used in [27, 4] for the approximation of higher-order tensors.

The article is organized as follows. In Sect. 2 we introduce properties of the matrix partition on which the approximation is based. Sect. 3 reviews the adaptive cross approximation method, which is used in Sect. 4 to construct nested bases approximations using only few of the original matrix entries. The approximation error and the computational complexity is analyzed. Finally, numerical experiments which demonstrate the improved efficiency of the new approach in particular when using the recompression procedure from [11] are presented in Sect. 5.

2 Matrix partitioning

The low-rank approximation (3) usually cannot be done globally. For instance, elliptic problems typically lead to singularities of $\kappa(x, y)$ for $x = y$. An example is the Coulomb potential

$\kappa(x,y) = |x-y|^{-1}$. Hence, the set of matrix indices $I \times J$ of A has to be partitioned suitably. In the case of elliptic problems, for the existence of approximations (3), the sub-block $t \times s$ has to satisfy the so-called *admissibility condition*

$$\max\{\text{diam } Y_t, \text{diam } X_s\} \leq \eta \, \text{dist}(Y_t, X_s) \tag{4}$$

with a given parameter $\eta > 0$ or $\min\{|t|, |s|\} \leq n_{\min}$ holds for a given $n_{\min} \in \mathbb{N}$. Here, we make use of the notation

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

and the supports

$$Y_t := \bigcup_{i \in t} \text{supp}\, \psi_i, \quad X_s := \bigcup_{j \in s} \text{supp}\, \varphi_j$$

of clusters $t \subset I$ and $s \subset J$, respectively. Equivalently to (4), the pair (Y_t, X_s) has to satisfy $X_s \subset \mathcal{F}(Y_t)$ and $Y_t \subset \mathcal{F}(X_s)$, where

$$\mathcal{F}(D) := \{x \in \mathbb{R}^3 : \eta \, \text{dist}(x, D) \geq \text{diam } D\}$$

denotes the *far field* of the bounded set $D \subset \mathbb{R}^3$. Notice that in order to satisfy (4), the supports of t and s have to be far enough away from each other.

On the other hand, a suitable partition P of the matrix indices $I \times J$ must be computable with logarithmic-linear complexity in order not to spoil the desired overall complexity. Searching the set of possible partitions of $I \times J$ for a partition P which guarantees (3) seems practically impossible since this set is considerably large. By restricting ourselves to blocks $t \times s$ made up from rows t and columns s which are generated by recursive subdivision, P can be found with almost linear complexity. The structure which describes the way I and J are subdivided into smaller parts is the *cluster tree*.

A tree T_I is called a *cluster tree* for an index set I if it satisfies the following conditions:

(i) I is the root of T_I;

(ii) if $t \in T_I$ is not a leaf, then t is a disjoint union of its sons $S_I(t) := \{t_1, t_2\} \subset T_I$.

We denote the set of leaves of the tree T_I by $\mathcal{L}(T_I)$. Each level of T_I contains a partition of the index set I. The depth of T_I, i.e. the maximum level of clusters $t \in T_I$ increased by one, will be denoted by $L(T_I)$. Cluster trees can be computed, for instance, by the *bounding box method* [14] or the *principal component analysis*; see, for instance, [7]. The latter methods take into account the geometric information associated with the matrix indices.

The construction of the desired partition P can be done no matter what the actual admissibility condition is. The partition is usually generated by recursive subdivision of $I \times J$. If (4) is not satisfied for a block $t \times s$, then (4) is checked for its sub-blocks $S_I(t) \times S_J(s)$. The recursion stops in blocks which satisfy (4) or which are small enough. The set of former blocks are denoted by P_{adm}, the set of latter blocks will be referred to as P_{nonadm}. The union of P_{adm} and P_{nonadm} constitutes the desired partition P of $I \times J$. The constructed partition has the property that for a given $t \in T_I$ a constantly bounded number $c^r_{\text{sp}}(t) := |\{s \subset J : t \times s \in P\}|$ of

blocks $t \times s$ appear in P. Similarly, given $s \in T_J$, the expression $c^c_{\rm sp}(s) := |\{t \subset I : t \times s \in P\}|$ is bounded by a constant. Hence, the expression

$$c_{\rm sp} := \max_{t \in T_I,\, s \in T_J} \{c^r_{\rm sp}(t), c^c_{\rm sp}(s)\} \tag{5}$$

is bounded independently of the sizes of I and J; see [17].

3 Adaptive cross approximation

The adaptive cross approximation (ACA) method was introduced for Nyström matrices [6] and extended to collocation matrices [10]. A version with refined pivoting strategy and a generalization of the method to Galerkin matrices was presented in [8]. We consider sub-blocks $t \times s$ of $A \in \mathbb{R}^{I \times J}$ defined in (1). If $t \times s \in P_{\rm nonadm}$, then all the entries of A_{ts} are stored without approximation. ACA treats each block A_{ts}, $t \times s \in P_{\rm adm}$, independently from all other blocks in P. The method constructs sequences of vectors $u_k = \hat{u}_k/(\hat{u}_k)_{i_k}$ and v_k from the following recursion

$$\hat{u}_k := A_{tj_k} - \sum_{\ell=1}^{k-1} (v_\ell)_{j_k} u_\ell \quad \text{and} \quad v_k := A_{i_k s} - \sum_{\ell=1}^{k-1} (u_\ell)_{i_k} v_\ell$$

with suitable row and column indices i_k and j_k satisfying $(\hat{u}_k)_{i_k} \neq 0$; see [7] for details. In particular, this means that only k of the original columns A_{tj_k} and rows $A_{i_k s}$ of the sub-block A_{ts} have to be computed. It is known (see [7]) that

$$UV^T = A_{t\sigma}(A_{\tau\sigma})^{-1} A_{\tau s}, \tag{6}$$

where $U := [u_1, \ldots, u_k]$, $V := [v_1, \ldots, v_k]$ and $\tau := \{i_1, \ldots, i_k\} \subset t$, $\sigma := \{j_1, \ldots, j_k\} \subset s$. Note that both τ and σ usually depend on both t and s. The number of operations required for constructing the approximation

$$UV^T \approx A_{ts}, \tag{7}$$

is of the order $k^2(|t| + |s|)$, while the storage required for UV^T is of the order $k(|t| + |s|)$.

The required rank of the approximation k to satisfy a prescribed accuracy ε can be found from inspecting the norms of $\hat{u}_k$ and v_k. In the case of asymptotically smooth kernels κ, the rank k can be guaranteed to depends logarithmically on ε. This follows from the fact that $\hat{u}_k = (R_{k-1})_{tj_k}$ and $v_k = (R_{k-1})_{i_k s}$ are columns and rows of the matrix $R_{k-1} \in \mathbb{R}^{t \times s}$ defined by the iteration $R_0 := A_{ts}$ and

$$R_k := R_{k-1} - \frac{(R_{k-1})_{tj_k}(R_{k-1})_{i_k s}}{(R_{k-1})_{i_k j_k}}, \quad k = 1, 2, \ldots.$$

This matrix recursion is related with the following analytic approximation of the kernel function κ: $r_0(x, y) = \kappa(x, y)$ and for $k = 1, 2, \ldots$

$$r_k(x, y) = \kappa(x, y) - \begin{bmatrix} \kappa(x_1, y) \\ \vdots \\ \kappa(x_k, y) \end{bmatrix}^T \xi^{(k)}(x), \qquad \xi^{(k)}(x) := M_k^{-T} \begin{bmatrix} \kappa(x, y_1) \\ \vdots \\ \kappa(x, y_k) \end{bmatrix} \in \mathbb{R}^k, \tag{8}$$

with points x_k and y_k chosen such that

$$M_k := \begin{bmatrix} \kappa(x_1, y_1) & \dots & \kappa(x_1, y_k) \\ \vdots & & \vdots \\ \kappa(x_k, y_1) & \dots & \kappa(x_k, y_k) \end{bmatrix} \in \mathbb{R}^{k \times k}$$

is non-singular. In [8] we proved that

$$|r_k(x, y)| \leq (\sigma_k + 1) \max_{z \in \{x, x_1, \dots, x_k\}} |\mathcal{E}_k[\kappa_z](y)|,$$

where $\mathcal{E}_k[f]$ denotes the Lagrange interpolation error for any system of k functions and

$$\sigma_k := \sup_{x \in X_s} \sum_{\nu=1}^{k} |\xi_\nu^{(k)}(x)|.$$

Approximations of type (8) were already considered by Micchelli and Pinkus in [26], where it is proved that such approximations are optimal with respect to the L^2-norm. Furthermore, Tyrtyshnikov et al. [15] investigated the matrix analogue of (8). If the pivoting points x_i, $i = 1, \dots, k$, are chosen such that M_k has maximal determinant in modulus then we obtain $\sigma_k \leq k$. In this case of so-called matrices of *maximum volume*, we also refer to the error estimates in [29] which are based on the technique of *exact annihilators*; see [3, 2]. In practice it is, however, difficult to find matrices of maximum volume. Finding the pivot i_k from choosing the maximum entry in modulus in the j_k-th column can be done efficiently. In this case, one can prove the conservative bound $\sigma_k \leq 2^k - 1$. Hence, the low-rank approximation UV^T can be regarded as quasi-optimal. Possible redundancies among the vectors u_ℓ, v_ℓ, $\ell = 1, \dots, k$, can be removed by orthogonalization; cf. [7]. Treating each block in P_{adm} this way, the total number of operations for the construction of an approximation to $A \in \mathbb{R}^{|I| \times |J|}$ is of the order $k^2(|I| + |J|) \log(|I| + |J|)$ and the total amount of storage required is of the order $k(|I| + |J|) \log(|I| + |J|)$.

ACA treats each block A_{ts}, $t \times s \in P_{\text{adm}}$, independently from all others. However, the low-rank approximation UV^T in (7) shares common properties with other blocks. In [9] we presented a modified technique (RACA) which exploits the existence of common column and row bases $\mathcal{U} = (U(t))_{t \in T_I}$ and $\mathcal{V} = (V(t))_{s \in T_J}$ of the low-rank approximations UV^T among the blocks, i.e.

$$A_{ts} \approx U(t) S(t, s) V(s)^T, \quad t \times s \in P_{\text{adm}}.$$

Notice that the matrices U and V in (7) depend on both t and s, whereas here only the (small) matrix $S(t, s) \in \mathbb{R}^{k \times k}$ depends on both t and s. In order to improve the storage complexity, in [9] the bases $\mathcal{U}$ and $\mathcal{V}$ were chosen such that they can be easily recomputed each time they are required. As a consequence, the number of operations did not improve compared with ACA. In order to store the cluster bases efficiently, an additional property has to be exploited. A cluster basis $\mathcal{U} = (U(t))_{t \in T_I}$ is called *nested* (see [24]) if for each $t \in T_I \setminus \mathcal{L}(T_I)$ there are transfer matrices $F_{t't} \in \mathbb{R}^{k_{t'} \times k_t}$ such that for the restriction of the matrix $U(t)$ to the rows t' it holds that

$$U(t)|_{t'} = U(t') F_{t't} \quad \text{for all } t' \in S_I(t). \tag{9}$$

Storing a nested cluster basis requires storing $U(t)$ for all leaf clusters $t \in \mathcal{L}(T_I)$ and the transfer matrices $F_{t't}$, $t' \in S_I(t)$, for all $t \in T \setminus \mathcal{L}(T_I)$. The nestedness of cluster bases is the reason for the improved complexity of fast multipole methods and $\mathcal{H}^2$-matrices.

4 Construction of nested approximations

Although ACA generates approximations of high quality, the size of the factors $A_{t\sigma}$ and $A_{\tau s}$ in (6) depends linearly on the number of rows $|t|$ and the number of columns $|s|$. To overcome this, our aim is to consider the approximation

$$A_{ts} \approx A_{t\sigma_t}(A_{\tau_t\sigma_t})^{-1}A_{\tau_t\sigma_s}(A_{\tau_s\sigma_s})^{-1}A_{\tau_s s} \tag{10}$$

instead of (6) and store only the small matrix $A_{\tau_t\sigma_s}$ for each block $t \times s \in P_{\mathrm{adm}}$. Here, $\tau_t \subset t$ and $\sigma_t \subset F(t)$ satisfying $|\tau_t| = |\sigma_t| = k$, represent t and its far field

$$F(t) := \bigcup\{s \in T_J : \exists \hat{t} \supset t : \hat{t} \times s \in P_{\mathrm{adm}}\} \subset \{j \in J : X_j \subset \mathcal{F}(Y_t)\},$$

respectively. The (large) matrices $A_{t\sigma_t}(A_{\tau_t\sigma_t})^{-1}$ and $(A_{\tau_s\sigma_s})^{-1}A_{\tau_s s}$ will be approximated by matrices $U(t) \in \mathbb{R}^{t\times\sigma_t}$ and $V(s) \in \mathbb{R}^{s\times\tau_s}$ which are constructed as nested bases. The aim of this section is to prove error estimates for the special type of low-rank approximation

$$A_{ts} \approx U(t)A_{\tau_t\sigma_s}V(s)^T.$$

Notice that one could also store $(A_{\tau_t\sigma_t})^{-1}A_{\tau_t\sigma_s}(A_{\tau_s\sigma_s})^{-1}$ for each block $t \times s \in P_{\mathrm{adm}}$ and represent $A_{t\sigma_t}$ and $A_{\tau_s s}$ recursively. This, however, leads to a numerically unstable algorithm, because the matrices $A_{\tau_t\sigma_t}$ are close to singular (their singular values decay exponentially as we shall see in Lemma 1) and its inverse can be treated in a stable way only as a product with $A_{t'\sigma_t}$, $t' \subset I$.

Let $t \in T_I$ and let $\{q_1, \ldots, q_{k_\varepsilon}\}$ be a basis of Π_p^3, where $k_\varepsilon := \dim \Pi_p^3 \sim p^3$ and $p \in \mathbb{N}$ is the smallest number such that $p \geq |\log_\eta \varepsilon|$. We assume that there is $\tau_t = \{i_1, \ldots, i_{k_\varepsilon}\} \subset t$ such that the following two conditions are satisfied.

(i) There are coefficients $\xi_{i\ell}$ such that

$$\frac{(\psi_i, q_j)_{L^2(\Omega)}}{\|\psi_i\|_{L^1}} = \sum_{\ell=1}^{k_\varepsilon} \xi_{i\ell}\frac{(\psi_{i_\ell}, q_j)_{L^2(\Omega)}}{\|\psi_{i_\ell}\|_{L^1}}, \quad i \in t,\ j = 1, \ldots, k_\varepsilon, \tag{11}$$

(ii) the matrix $A_{\tau_t F(t)}$ has full rank.

To see that these assumptions are reasonable, let $Z \in \mathbb{R}^{|t|\times k_\varepsilon}$ be defined by $Z_{ij} = (\psi_i, q_j)_{L^2(\Omega)}$, $i \in t$, $j = 1, \ldots, k_\varepsilon$. Then (i) follows from $\operatorname{rank} Z \leq k_\varepsilon$. The set τ_t is not unique. Usually, any sub-set of t having k_ε elements will do. To see this, assume for the time being that $\psi_i = \delta_{x_i}$. In this case, the matrix $(Z_{i_\ell j})_{\ell j}$ having entries $(\psi_{i_\ell}, q_j) = q_j(x_{i_\ell})$ is non-singular iff Π_p^3 is unisolvent with respect to the points x_{i_ℓ}, $\ell = 1, \ldots, k_\varepsilon$, i.e.

$$q(x_{i_\ell}) = 0, \quad \ell = 1, \ldots, k_\varepsilon,$$

for some $q \in \Pi_p^3$ implies that $q = 0$. The set of tupels $(x_{i_1}, \ldots, x_{i_{k_\varepsilon}})$ for which Π_p^3 is not unisolvent is known to be of measure zero; see [28]. Hence, we may assume that $\tau_t \subset t$ is chosen such that also (ii) is valid. Otherwise, the rank of $A_{tF(t)}$ would already be bounded by k_ε. A method for choosing τ_t and σ_t will be discussed later in this article.

The following lemma states the existence of low-rank approximations consisting of linear combinations of some of the block's rows. Notice that with the previous assumptions it is possible to guarantee that the rows τ_t used for the approximation of A_{ts} can be chosen independently of s. This will be crucial for this article.

Lemma 1. *Let assumption (i) be valid and let κ, φ_i and ψ_j in (1) be non-negative. Then there is $\Xi \in \mathbb{R}^{t\times\tau_t}$ and a constant $c_1 > 0$ such that*

$$\|A_{t's} - \Xi A_{\tau_t s}\|_F \leq c_1\varepsilon\|A_{t's}\|_F$$

for all $t' \subset t$ and $s \subset F(t)$.

Proof. The asymptotic smoothness (2) of κ guarantees that the Taylor series with respect to y about $y' \in Y_t$ converges exponentially, i.e. we have the decomposition

$$\kappa(x,y) = T_p(x,y) + R_p(x,y),$$

where $T_p(x,\cdot) \in \Pi_p^3$ and for $x \in X_s$ and $y \in Y_t$

$$|R_p(x,y)| \leq \tilde{c}_p \left(\frac{\operatorname{diam} Y_t}{\operatorname{dist}(X_s, Y_t)}\right)^p |\kappa(x,y')| \leq \tilde{c}_p\eta^p|\kappa(x,y')|$$

due to $X_s \subset \mathcal{F}(Y_t)$. Exchanging the roles of y and y', we obtain in particular

$$|\kappa(x,y')| \leq |\kappa(x,y)| + |R_1(x,y')| \leq (1+\tilde{c}_1\eta)|\kappa(x,y)|. \tag{12}$$

For simplicity we may assume that $\|\psi_i\|_{L^1} = 1$, $i \in I$. Then assumption (11) is equivalent with

$$\int_\Omega \left(\psi_i - \sum_{\ell=1}^{k_\varepsilon} \xi_{i\ell}\psi_{i_\ell}\right) q(y)\,\mathrm{d}\mu_y = 0$$

for all $q \in \Pi_p^3$. From

$$\begin{aligned}
a_{ij} - \sum_{\ell=1}^{k_\varepsilon} \xi_{i\ell}a_{i_\ell j} &= \int_\Omega\int_\Omega \left(\psi_i(y) - \sum_{\ell=1}^{k_\varepsilon} \xi_{i\ell}\psi_{i_\ell}(y)\right)\kappa(x,y)\varphi_j(x)\,\mathrm{d}\mu_y\,\mathrm{d}\mu_x \\
&= \int_\Omega\int_\Omega \left(\psi_i(y) - \sum_{\ell=1}^{k_\varepsilon} \xi_{i\ell}\psi_{i_\ell}(y)\right)[\kappa(x,y) - T_p(x,y)]\varphi_j(x)\,\mathrm{d}\mu_y\,\mathrm{d}\mu_x \\
&= \int_\Omega\int_\Omega \left(\psi_i(y) - \sum_{\ell=1}^{k_\varepsilon} \xi_{i\ell}\psi_{i_\ell}(y)\right)R_p(x,y)\varphi_j(x)\,\mathrm{d}\mu_y\,\mathrm{d}\mu_x
\end{aligned}$$

we see that

$$|a_{ij} - \sum_{\ell=1}^{k_\varepsilon} \xi_{i\ell}a_{i_\ell j}| \leq \tilde{c}_p\left(1 + \sum_{\ell=1}^{k_\varepsilon}|\xi_{i\ell}|\right)\eta^p \max_{\nu\in\tau_t\cup\{i\}} \int_\Omega\int_\Omega |\psi_\nu(y)||\kappa(x,y)||\varphi_j(x)|\,\mathrm{d}\mu_y\,\mathrm{d}\mu_x.$$

Estimate (12) implies

$$\int_\Omega |\psi_\nu(y)||\kappa(x,y)|\,\mathrm{d}\mu_y \le \max_{y\in\operatorname{supp}\psi_\nu}|\kappa(x,y)| \le c\min_{y\in\operatorname{supp}\psi_i}|\kappa(x,y)| \le c\int_\Omega |\psi_i(y)||\kappa(x,y)|\,\mathrm{d}\mu_y.$$

The non-negativity of ψ_i, φ_j, and κ leads to

$$|a_{ij} - \sum_{\ell=1}^{k_\varepsilon}\xi_{i\ell}a_{i_\ell j}| \le c\tilde{c}_p\left(1+\sum_{\ell=1}^{k_\varepsilon}|\xi_{i\ell}|\right)\eta^p|a_{ij}|.$$

The matrix $\Xi\in\mathbb{R}^{t'\times\tau_t}$ having the entries $\xi_{i\ell}$ guarantees that

$$\|A_{t's} - \Xi A_{\tau_t s}\|_F \le c\tilde{c}_p(1+\|\Xi\|_\infty)\eta^p\|A_{t's}\|_F.$$

□

Notice that Lemma 1 implies that the singular values of the matrix $A_{\tau_t\sigma_t}$ decay exponentially to zero as stated before. The following lemma uses the previous to establish the existence of low-rank approximations of ACA type.

Lemma 2. *Let assumptions (i) and (ii) be valid. Then there is $\sigma_t\subset F(t)$, $|\sigma_t|=|\tau_t|$, such that*

$$\|A_{t's} - A_{t'\sigma_t}(A_{\tau_t\sigma_t})^{-1}A_{\tau_t s}\|_F \le c_2\varepsilon\|A_{t'F(t)}\|_F \quad \text{for all } s\subset F(t),\ t'\subset t, \tag{13}$$

where $c_2 := c_1(1+\|(A_{\tau_t\sigma_t})^{-1}A_{\tau_t s}\|_F)$.

Proof. Let $\Xi\in\mathbb{R}^{t'\times\tau_t}$ be as in Lemma 1. According to assumption (ii) there is $\sigma_t\subset F(t)$, $|\sigma_t|=|\tau_t|$, such that $A_{\tau_t\sigma_t}$ is invertible. Then we have

$$A_{t's} - A_{t'\sigma_t}(A_{\tau_t\sigma_t})^{-1}A_{\tau_t s} = \{A_{t's}-\Xi A_{\tau_t s}\} - \{A_{t'\sigma_t}-\Xi A_{\tau_t\sigma_t}\}(A_{\tau_t\sigma_t})^{-1}A_{\tau_t s}$$

and hence with $c := \|(A_{\tau_t\sigma_t})^{-1}A_{\tau_t s}\|_F$

$$\begin{aligned}\|A_{t's} - A_{t'\sigma_t}(A_{\tau_t\sigma_t})^{-1}A_{\tau_t s}\|_F &\le \|A_{t's}-\Xi A_{\tau_t s}\|_F + c\|A_{t'\sigma_t}-\Xi A_{\tau_t\sigma_t}\|_F\\ &\le c_1\varepsilon(\|A_{t's}\|_F + c\|A_{t'\sigma_t}\|_F) \le c_2\varepsilon\|A_{t'F(t)}\|_F.\end{aligned}$$

The second last estimate follows from Lemma 1, because $s,\sigma_t\subset F(t)$. □

Remark. *Note that Lemma 2 holds true for every choice $\sigma_t\subset F(t)$ with $A_{\tau_t\sigma_t}$ invertible.*

Let $t'\subset t$. If the matrix

$$B_{t't} := A_{t'\sigma_t}(A_{\tau_t\sigma_t})^{-1} \tag{14}$$

is applied to columns $A_{\tau_t j}$, $j\in F(t)$, then $(B_{t't}A_{\tau_t j})_{\tau_t} = A_{t'\cap\tau_t j}$ and according to Lemma 2

$$\|A_{t'j} - B_{t't}A_{\tau_t j}\|_2 \le c_2\varepsilon\|A_{t'F(t)}\|_F,$$

i.e. $B_{t't}A_{\tau_t j}$ approximates $A_{t'j}$. Hence, $B_{t't}$ may be regarded as the algebraic form of an interpolation operator. The following lemma estimates the accuracy when expressing $B_{t't}$ by the product $B_{t't'}B_{\tau_{t'}t}$ of two interpolation operators. Notice that the estimate is valid only on "smooth functions" $A_{\tau_t s}$, $s\subset F(t)$.

Lemma 3. *Let $t' \in T_I$ satisfy $t' \subset t$. Then for all $s \subset F(t)$ it holds that*

$$\|[B_{t't} - B_{t't'}B_{\tau_{t'}t}]A_{\tau_t s}\|_F \leq c_3\varepsilon\|A_{t'J}\|_F.$$

Proof. From

$$[B_{t't} - B_{t't'}B_{\tau_{t'}t}]A_{\tau_t s} = A_{t's} - B_{t't'}A_{\tau_{t'}s} - (A_{t's} - B_{t't}A_{\tau_t s}) + B_{t't'}[A_{\tau_{t'}s} - B_{\tau_{t'}t}A_{\tau_t s}]$$

it follows using Lemma 2 that

$$\begin{aligned}\|[B_{t't} - B_{t't'}B_{\tau_{t'}t}]A_{\tau_t s}\|_F &\leq c_2\varepsilon\left(\|A_{t'F(t')}\|_F + \|A_{t'F(t)}\|_F + \|B_{t't'}\|_F\|A_{\tau_{t'}F(t)}\|_F\right)\\ &\leq c_2(2 + \|B_{t't'}\|_F)\varepsilon\|A_{t'J}\|_F.\end{aligned}$$

□

Similar results as for the row clusters t can be obtained for column clusters s provided assumptions analogous to (i) and (ii) are made. In particular, for a given $s \in T_J$ this defines clusters $\sigma_s \subset s$ and $\tau_s \subset F'(s)$, $|\tau_s| = |\sigma_s|$, where

$$F'(s) := \bigcup\{t \in T_I : \exists \hat{s} \supset s : t \times \hat{s} \in P_{\text{adm}}\}.$$

For $s' \subset s$ we investigate the matrix

$$C_{s's} := (A_{\tau_s\sigma_s})^{-1}A_{\tau_s s'}. \tag{15}$$

Due to the analogy, we omit the proofs.

Lemma 4. *Let assumptions analogous to (i) and (ii) be valid. Then there is $\tau_s \subset F'(s)$, $|\tau_s| = |\sigma_s|$, such that*

$$\|A_{ts'} - A_{t\sigma_s}(A_{\tau_s\sigma_s})^{-1}A_{\tau_s s'}\|_F \leq c_4\varepsilon\|A_{Is'}\|_F$$

for all $t \subset F'(s)$ and $s' \subset s$.

Lemma 5. *Let $s' \in T_J$ satisfy $s' \subset s$. Then for all $t \subset F'(s)$ it holds that*

$$\|A_{t\sigma_s}[C_{s's} - C_{s's'}C_{\sigma_{s'}s}]\|_F \leq c_5\varepsilon\|A_{Is'}\|_F.$$

4.1 Construction of nested row and column spaces

The construction of nested bases is usually done by analytic constructions. Fast multipole methods [18, 19] are based on multipole expansions which have to be adapted to the respective kernel function κ. $\mathcal{H}^2$-matrix approximations are usually constructed via interpolation; see [24, 22]. In this section, we construct the nested bases with a purely algebraic technique which is based on the original matrix entries and thus avoids explicit kernel expansions. In this sense the presented construction is in the class of kernel independent fast multipole methods; see [1, 31, 13]. We define a nested basis $\mathcal{U}$ consisting of matrices $U(t) \in \mathbb{R}^{t\times\sigma_t}$ for each $t \in T_I$ in a recursive manner starting from the leaves of T_I. For leaf clusters $t \in \mathcal{L}(T_I)$ set

$$U(t) := B_{tt},$$

where B_{tt} is defined in (14). Assume that matrices $U(t')$ have already been constructed for the sons $t' \in S_I(t)$ of $t \in T_I \setminus \mathcal{L}(T_I)$. Then in view of (9) we define

$$U(t)|_{t'} := U(t')B_{\tau_{t'}t}, \quad t' \in S_I(t). \tag{16}$$

Before we will prove in the following theorem that $U(t)$ approximates $A_{t\sigma_t}(A_{\tau_t\sigma_t})^{-1}$ for all $t \in T_I$, let us investigate the complexity of the construction of $\mathcal{U}$. Since the set of leaf clusters $\mathcal{L}(T_I)$ constitutes a partition of I and for each cluster $t \in \mathcal{L}(T_I)$ the $|t| \times k_t$-matrix B_{tt} with $k_t \le \min\{k_\varepsilon, |t|\}$ has to be stored, at most $k_\varepsilon|I|$ units of storage are required for all $U(t)$, $t \in \mathcal{L}(T_I)$. Similarly, the number of arithmetical operations for computing all matrices $B_{tt} = A_{t\sigma_t}(A_{\tau_t\sigma_t})^{-1}$, $t \in \mathcal{L}(T_I)$, is of the order

$$\sum_{t\in\mathcal{L}(T_I)} k_t^2|t| + k_t^3 \le \sum_{t\in\mathcal{L}(T_I)} 2k_t^2|t| \le 2k_\varepsilon^2|I|.$$

Additionally, storing the matrices $B_{\tau_{t'}t} \in \mathbb{R}^{k_{t'}\times k_t}$ requires

$$\sum_{t\in T_I\setminus\mathcal{L}(T_I)} \sum_{t'\in S_I(t)} k_{t'}k_t \le 2\sum_{t\in T_I} \min\{k_\varepsilon^2, |t|^2\} \le 6k_\varepsilon|I|$$

due to $\max\{k_{t'}, k_t\} \le \min\{k_\varepsilon, |t|\}$ and the estimate (see [7, Lemma 1.21], [21])

$$\sum_{t\in T_I} \min\{c, |t|^2\} \le 3\sqrt{c}|I|. \tag{17}$$

The number of operations for computing the matrices $B_{\tau_{t'}t} = A_{\tau_{t'}\sigma_t}(A_{\tau_t\sigma_t})^{-1}$ is bounded by

$$\sum_{t\in T_I\setminus\mathcal{L}(T_I)} \sum_{t'\in S_I(t)} k_{t'}k_t^2 + k_t^3 \le 2k_\varepsilon \sum_{t\in T_I\setminus\mathcal{L}(T_I)} \sum_{t'\in S_I(t)} k_t^2 \le 4k_\varepsilon \sum_{t\in T_I} \min\{k_\varepsilon^2, |t|^2\} \le 12k_\varepsilon^2|I|.$$

As a consequence, $\mathcal{O}(k_\varepsilon^2|I|)$ arithmetical operations and $\mathcal{O}(k_\varepsilon|I|)$ units of storage are required for the construction of the nested basis $\mathcal{U}$ defined in (16).

Theorem 1. *Let $t \in T_I$ and let $\ell = L(T_t)$ denote the depth of the cluster tree T_t. Then*

$$\|[U(t) - A_{t\sigma_t}(A_{\tau_t\sigma_t})^{-1}]A_{\tau_t s}\|_F \le c_6\varepsilon\|A_{tJ}\|_F \quad \textit{for all } s \subset F(t),$$

where

$$c_6 := \frac{\sqrt{2}c_3(\sqrt{2}c_B)^{\ell-1}}{\sqrt{2c_B^2 - 1}}$$

and $c_B := \max\{\|(A_{\tau_t\sigma_t})^{-1}A_{\tau_t s}\|_F,\ s \subset F(t),\ t \in T_I \setminus \mathcal{L}(T_I)\}$.

Proof. The assertion is proved by induction. It is obviously true for leaf clusters. Assume

that it is valid for the sons $S_I(t)$ of $t \in T_I \setminus \mathcal{L}(T_I)$. Then

$$\begin{aligned}
\|[U(t) - A_{t\sigma_t}(A_{\tau_t\sigma_t})^{-1}]A_{\tau_t s}\|_F^2 &= \sum_{t'\in S_I(t)} \|[U(t)|_{t'} - A_{t'\sigma_t}(A_{\tau_t\sigma_t})^{-1}]A_{\tau_t s}\|_F^2 \\
&= \sum_{t'\in S_I(t)} \|[U(t')B_{\tau_{t'}t} - B_{t't}]A_{\tau_t s}\|_F^2 \\
&\le 2\sum_{t'\in S_I(t)} \|[U(t') - A_{t'\sigma_{t'}}(A_{\tau_{t'}\sigma_{t'}})^{-1}]A_{\tau_{t'}\sigma_t}(A_{\tau_t\sigma_t})^{-1}A_{\tau_t s}\|_F^2 + \|[B_{t't} - B_{t't'}B_{\tau_{t'}t}]A_{\tau_t s}\|_F^2 \\
&\le 2\sum_{t'\in S_I(t)} c_B^2\|[U(t') - A_{t'\sigma_{t'}}(A_{\tau_{t'}\sigma_{t'}})^{-1}]A_{\tau_{t'}\sigma_t}\|_F^2 + c_3^2\varepsilon^2\|A_{t'J}\|_F^2 \\
&= 2c_3^2\varepsilon^2\|A_{tJ}\|_F^2 + 2c_B^2\sum_{t'\in S_I(t)} \|[U(t') - A_{t'\sigma_{t'}}(A_{\tau_{t'}\sigma_{t'}})^{-1}]A_{\tau_{t'}\sigma_t}\|_F^2.
\end{aligned}$$

The last estimate follows from Lemma 3. Setting

$$\alpha_{t'} = \|[U(t') - A_{t'\sigma_{t'}}(A_{\tau_{t'}\sigma_{t'}})^{-1}]A_{\tau_{t'}\sigma_t}\|_F^2,$$

we obtain the recurrence relation

$$\alpha_{t'} \le 2c_3^2\varepsilon^2\|A_{t'J}\|_F^2 + 2c_B^2\sum_{t''\in S_I(t')} \alpha_{t''}, \quad t' \in T_I \setminus \mathcal{L}(T_I).$$

With $\alpha_{t'} = 0$ for $t' \in \mathcal{L}(T_I)$ this leads to

$$\alpha_{t'} \le 2c_3^2\frac{(2c_B^2)^{\ell-1} - 1}{2c_B^2 - 1}\,\varepsilon^2\|A_{t'J}\|_F^2,$$

where ℓ denotes the depth of the sub-tree $T_{t'}$. Hence,

$$\begin{aligned}
\|[U(t) - A_{t\sigma_t}(A_{\tau_t\sigma_t})^{-1}]A_{\tau_t s}\|_F^2 &\le 2c_3^2\varepsilon^2\|A_{tJ}\|_F^2 + 2c_B^2\sum_{t'\in S_I(t)} \alpha_{t'} \\
&\le 2c_3^2\varepsilon^2\frac{(2c_B^2)^{\ell} - 1}{2c_B^2 - 1}\|A_{tJ}\|_F^2,
\end{aligned}$$

which is the assertion. □

Remark. *The size of c_B depends on the choice of τ_t and σ_t. We have already mentioned that $A_{\tau_t\sigma_t}$ is close to singular, because $\tau_t \subset t$ and $\sigma_t \subset F(t)$. Nevertheless, the norm of products of the form $A_{t'\sigma_t}(A_{\tau_t\sigma_t})^{-1}$ or $(A_{\tau_t\sigma_t})^{-1}A_{\tau_t s}$ is bounded by a small constant c_B; see the discussion on σ_k in Sect. 3. Computations show that $c_B \sim |\tau_t| \sim |\log\varepsilon|^3$. Rigorous estimates for c_B will be proved in a forthcoming article.*

Similarly, for $s \in \mathcal{L}(T_J)$ we set $V(s) = C_{ss}^T$, where C_{ss} is defined in (15). Assume that matrices $V(s')$ have already been constructed for the sons $s' \in S_J(s)$ of $s \in T_J \setminus \mathcal{L}(T_J)$. Then we define

$$V(s)|_{s'} := V(s')C_{\sigma_{s's}}^T, \quad s' \in S_J(s). \tag{18}$$

The complexity estimates on the column basis $\mathcal{U}$ also apply to the row bases $\mathcal{V} := (V(s))_{s\in T_J}$, i.e. $\mathcal{O}(k_\varepsilon^2|J|)$ arithmetical operations and $\mathcal{O}(k_\varepsilon|J|)$ units of storage are required for $\mathcal{V}$.

Theorem 2. *Let $s \in T_J$ and let $\ell = L(T_s)$ denote the depth of the cluster tree T_s. Then there is $c_7 > 0$ such that*

$$\|A_{t\sigma_s}[(A_{\tau_s\sigma_s})^{-1}A_{\tau_s s} - V(s)^T]\|_F \leq c_7\varepsilon\|A_{Is}\|_F$$

for all $t \subset F'(s)$.

4.2 Matrix approximation

Let $t \times s \in P_{\text{adm}}$. Using the previously constructed bases $\mathcal{U}$ and $\mathcal{V}$, we will employ

$$U(t)S(t,s)V(s)^T,$$

where

$$S(t,s) := A_{\tau_t\sigma_s}, \tag{19}$$

as an approximation to A_{ts}. The amount of storage required for $S(t,s)$ and the number of arithmetical operations for its computation can be estimated by

$$\begin{aligned}\sum_{t\times s\in P} k_t k_s &\leq \frac{1}{2}\sum_{t\times s\in P} k_t^2 + k_s^2 \leq \frac{1}{2}\sum_{t\in T_I}\sum_{s:t\times s\in P}\min\{k_\varepsilon^2, |t|^2\} + \frac{1}{2}\sum_{s\in T_J}\sum_{t:t\times s\in P}\min\{k_\varepsilon^2, |s|^2\}\\ &\leq \frac{c_{\text{sp}}}{2}\left(\sum_{t\in T_I}\min\{k_\varepsilon^2, |t|^2\} + \sum_{s\in T_J}\min\{k_\varepsilon^2, |s|^2\}\right)\\ &\leq \frac{3}{2}c_{\text{sp}}k_\varepsilon(|I| + |J|)\end{aligned}$$

due to (17). Notice that this also includes the construction of A_{ts}, $t \times s \in P_{\text{nonadm}}$.

Theorem 3. *Let $t \times s \in P_{\text{adm}}$. For the approximation error it holds that*

$$\|A_{ts} - U(t)S(t,s)V(s)^T\|_F \leq (c_2 + c_6\|C_{ss}\|_F)\varepsilon\|A_{tJ}\|_F + (c_4\|B_{tt}\|_F + c_7\|U(t)\|_F)\varepsilon\|A_{Is}\|_F. \tag{20}$$

Proof. We have that

$$A_{ts} - B_{tt}S(t,s)C_{ss} = A_{ts} - B_{tt}A_{\tau_t s} + B_{tt}\left[A_{\tau_t s} - A_{\tau_t\sigma_s}C_{ss}\right].$$

From Lemma 2 it follows that $\|A_{ts} - B_{tt}A_{\tau_t s}\| \leq c_2\varepsilon\|A_{tJ}\|_F$, and from Lemma 4 we have that $\|A_{\tau_t s} - A_{\tau_t\sigma_s}C_{ss}\|_F \leq c_4\varepsilon\|A_{Is}\|_F$. Therefore,

$$\|A_{ts} - B_{tt}S(t,s)C_{ss}\|_F \leq \varepsilon(c_2\|A_{tJ}\|_F + c_4\|B_{tt}\|_F\|A_{Is}\|_F).$$

Furthermore, Theorem 1 and Theorem 2 yield

$$\begin{aligned}\|U(t)S(t,s)V(s)^T &- B_{tt}S(t,s)C_{ss}\|_F\\ &\leq \|U(t)\|_F\|S(t,s)[V(s)^T - C_{ss}]\|_F + \|C_{ss}\|_F\|[U(t) - B_{tt}]S(t,s)\|_F\\ &\leq \|U(t)\|_F c_7\varepsilon\|A_{Is}\|_F + \|C_{ss}\|_F c_6\varepsilon\|A_{tJ}\|_F,\end{aligned}$$

which proves the assertion. □

Gathering the previously analyzed steps, we obtain the following algorithm for the construction of the approximation of $A \in \mathbb{R}^{I \times J}$.

1. Generate the entries of all blocks A_{ts}, $t \times s \in P_{\text{nonadm}}$;
2. Generate the nested basis $\mathcal{U}$ via (16) and the pivotal indices τ_t, σ_t for each $t \in T_I$;
3. Generate the nested basis $\mathcal{V}$ via (18) and the pivotal indices τ_s, σ_s for each $s \in T_J$;
4. Generate $S(t,s)$ for all blocks $t \times s \in P_{\text{adm}}$ via (19).

As previously analyzed, these four steps require $\mathcal{O}((|I|+|J|)|\log\varepsilon|^6)$ arithmetical operations and $\mathcal{O}((|I|+|J|)|\log\varepsilon|^3)$ units of storage. Notice that for boundary integral operators in $\mathbb{R}^3$ it can be expected that $k_t \sim p^2 = \lceil \log_\eta \varepsilon \rceil^2$ due to the adaptivity when constructing the nested bases via ACA, which leads to complexities $\mathcal{O}((|I|+|J|)|\log\varepsilon|^4)$ and $\mathcal{O}((|I|+|J|)|\log\varepsilon|^2)$, respectively. Hence, the new method saves one logarithmic factor compared with the approximation by $\mathcal{H}$-matrices via ACA.

4.3 The choice of representor sets τ_t and σ_t

While the pivots τ and σ in ACA (6) can be found adaptively for each block $t \times s$, τ_t and σ_t in (10) have to be known a-priory. For a given set $t \subset I$, the pivots $\tau_t \subset t$ representing t and the far field pivots $\sigma_t \subset F(t)$ representing $F(t)$ have to be chosen such that the matrix $A_{\tau_t\sigma_t}$ is invertible in a numerically stable way, i.e. the norms of $A_{t\sigma_t}(A_{\tau_t\sigma_t})^{-1}$ and $(A_{\tau_t\sigma_t})^{-1}A_{\tau_t s}$ are small for all $s \subset F(t)$. In this section, two possible ways of constructing τ_t and σ_t will be presented. Both methods start from rich enough sets $\tilde{\tau}_t \subset t$ and $\tilde{\sigma}_t \subset F(t)$, i.e. $|\tilde{\tau}_t|, |\tilde{\sigma}_t| \geq k_\varepsilon$, such that $A_{\tilde{\tau}_t\tilde{\sigma}_t}$ has full rank. In a second step, suitable subsets $\tau_t \subset \tilde{\tau}_t$ and $\sigma_t \subset \tilde{\sigma}_t$ with $k_t := |\tau_t| = |\sigma_t| \leq k_\varepsilon$ and the above mentioned properties are chosen from $\tilde{\tau}_t$ and $\tilde{\sigma}_t$, respectively:

1. apply ACA with accuracy ε to the matrix $A_{\tilde{\tau}_t\tilde{\sigma}_t}$ (likewise with global pivoting);
2. use the row pivots $\{i_1, \ldots, i_{k_t}\}$ and column pivots $\{j_1, \ldots, j_{k_t}\}$ chosen by ACA as the representor sets τ_t and σ_t, respectively.

Constructing τ_t and σ_t from larger sets has several advantages. First of all, we can adapt the rank $k_t \leq k_\varepsilon$ to the properties of the cluster basis and hence improve the storage costs. Secondly, ACA automatically guarantees that the matrix $A_{\tau_t\sigma_t}$ is invertible, which is crucial for the interpolation process, i.e. for the definition of the matrix $B_{t't}$. Furthermore, by slightly modifying the ACA algorithm one can compute the LU decomposition of $A_{\tau_t\sigma_t}$ as a byproduct. Therefore, the expressions $A_{t'\sigma_t}A_{\tau_t\sigma_t}^{-1}$ and $A_{\tau_t\sigma_t}^{-1}A_{\tau_t s}$ can then be evaluated simply by applying forward and backward substitution.

Note that reducing the representor sets via ACA does not change the error analysis in Sect. 4.1. Inspecting the proof of Theorem 3 and its preceding lemmas shows that any choice of sets $\tau_t \subset t$ and $\sigma_t \subset F(t)$ satisfying

$$\|A_{t's} - A_{t'\sigma_t}A_{\tau_t\sigma_t}^{-1}A_{\tau_t s}\|_F \leq c\varepsilon\|A_{tF(t)}\|_F \quad \text{for all } s \subset F(t),\ t' \subset t, \tag{21}$$

instead of (13) in Lemma 2 yields a nested cluster basis with an approximation property similar to (20). One can see that replacing (13) by (21) will change only the constant in Theorem 1. The following lemma shows that the estimate (21) can be preserved.

Lemma 6. *Assume that* (13) *holds for* $\tau_t \subset t$ *and* $\sigma_t \subset F(t)$, *i.e.*

$$\|A_{t's} - A_{t'\sigma_t} A_{\tau_t\sigma_t}^{-1} A_{\tau_t s}\|_F \leq \varepsilon \|A_{t'F(t)}\|_F \quad \text{for all } s \subset F(t),\ t' \subset t. \tag{22}$$

Then for any pair of subsets $\xi_t \subset \tau_t$ *and* $\chi_t \subset \sigma_t$ *satisfying that* $A_{\xi_t\chi_t}$ *is invertible and*

$$\|A_{\tau_t\sigma_t} - A_{\tau_t\chi_t} A_{\xi_t\chi_t}^{-1} A_{\xi_t\sigma_t}\|_F \leq \varepsilon \|A_{\tau_t\sigma_t}\|_F, \tag{23}$$

we have that (21) *holds with* $c := 1 + c_B(\|B_{t't}\|_F + \|A_{\xi_t\chi_t}^{-1} A_{\xi_t\sigma_t}\|_F) + \|A_{t'\chi_t} A_{\xi_t\chi_t}^{-1}\|_F$.

Proof. This can be easily seen from the equation

$$\begin{aligned} A_{t's} - A_{t'\chi_t}(A_{\xi_t\chi_t})^{-1} A_{\chi_t s} = {} & A_{t's} - A_{t'\sigma_t}(A_{\tau_t\sigma_t})^{-1} A_{\tau_t s} + B_{t't}(A_{\tau_t\sigma_t} - A_{\tau_t\chi_t} A_{\xi_t\chi_t}^{-1} A_{\xi_t\sigma_t}) C_{st} \\ & - (A_{t'\chi_t} - B_{t't} A_{\tau_t\chi_t}) A_{\xi_t\chi_t}^{-1} A_{\xi_t\sigma_t} C_{st} - A_{t'\chi_t} A_{\xi_t\chi_t}^{-1} (A_{\xi_t s} - B_{\xi_t t} A_{\tau_t s}). \end{aligned}$$

Using (22) and (23), we obtain the assertion. □

In the numerical results two methods for constructing the latter sets $\tilde{\tau}_t$ and $\tilde{\sigma}_t$ will be compared. Both are motivated by the relation of ACA with interpolation.

Geometric construction

Let $t \subset I$. Our aim is to select a sub-set $\chi(t) \subset t$, $|\chi(t)| = \min\{k_\varepsilon, |t|\}$, representing t in some sense. If $k_\varepsilon \geq |t|$, then we choose $\chi(t) := t$. In the other case, let $M(t) := \{m_i,\ i \in t\} \subset \mathbb{R}^3$ be a set of points associated with the supports X_i, $i \in t$. A box $Q_M \subset \mathbb{R}^3$ with minimal side lengths containing M can, for instance, be computed by the principal component analysis of the points in $M(t)$. The main directions of Q_M are the orthogonal eigenvectors of the covariance matrix

$$C_t := \sum_{i \in t} (m_i - c_M)(m_i - c_M)^T,$$

where $c_M := \frac{1}{|t|} \sum_{i \in t} m_i$ denotes the centroid of $M(t)$. Q_M is then discretized using a tensor grid G with $k_\varepsilon = \lceil \log_\eta \varepsilon \rceil^3$ points (e.g. $\lceil \log_\eta \varepsilon \rceil$ Chebyshev nodes in each spatial direction), and we define

$$\chi(t) := \{i_j \in t : m_{i_j} \text{ is closest to a grid point } j = 1, \ldots, k_\varepsilon \text{ of } G\}.$$

Remark. *In order to guarantee that a chosen point* m_{i_j} *is not used twice, we remove it from* M. *This guarantees that* $|\chi(t)| = \min\{k_\varepsilon, |t|\}$ *and the construction of* $\chi(t)$ *can be done with* $\mathcal{O}(k_\varepsilon |t|)$ *operations.*

The method from above defines $\tilde{\tau}_t := \chi(t)$ for each $t \in T_I$. A naive approach to choose the far field pivots would be to define $\tilde{\sigma}_t$ analogously:

$$\tilde{\sigma}_t := \chi(F(t)).$$

This would, however, lead to an $\mathcal{O}(|I| \cdot |J|)$ algorithm due to the fact that $|F(t)| = \mathcal{O}(|J|)$. Instead, setting $P_{\text{far}}(t) := \bigcup\{s \in T_J : t \times s \in P_{\text{adm}}\}$, we exploit the recursive relation (starting from the clusters t with $P_{\text{far}}(t) \neq \emptyset$)

$$F(t) = F(t^f) \cup P_{\text{far}}(t),$$

where t^f denotes the father of t in T_I. We set

$$\tilde{\sigma}_t := \chi(\sigma_{t^f} \cup P_{\text{far}}(t)).$$

Since the cardinality of $P_{\text{far}}(t)$ is of the order $c_{\text{sp}}|t|$, where c_{sp} has been defined in (5), the latter approach yields an $\mathcal{O}(k_\varepsilon|I|)$ algorithm.

Recursive construction

Another way of constructing $\tilde{\tau}_t$ is based on a recursion. For the leaves set $\tilde{\tau}_t := t$ if $|t|$ is small enough. Otherwise, we use the geometric construction from above. Assume that $\tilde{\tau}_{t'}$ has already been constructed for the sons $t' \in S_I(t)$ of $t \in T_I \setminus \mathcal{L}(T_I)$. Let

$$\tilde{\tau}_t := \bigcup_{t' \in S_I(t)} \tau_{t'} \subset t.$$

Numerical experiments confirm that this method is generally faster than the first one, since the sets of pivots taken into account for the computation of the cluster basis are smaller. On the other hand, it turns out that the latter method may lead to a lower accuracy. This is due to the fact that too small subsets τ_t of t in Lemma 1 imply a less reliable approximation of the kernel function κ.

4.4 Application of the approximation to a vector

Since the constructed approximation is similar to an $\mathcal{H}^2$-matrix, the matrix-vector multiplication $y := y + Ax$ of a matrix A by a vector $x \in \mathbb{R}^J$ can be done by the usual three-phase algorithm (cf. [24]). The following algorithm is a consequence of the decomposition

$$Ax \approx \sum_{t \times s \in P_{\text{nonadm}}} A_{ts} x_s + \sum_{t \times s \in P_{\text{adm}}} U(t) S(t,s) V(s)^T x_s.$$

1. *Forward transform*
   In this first phase, transformed vectors $\hat{x}(s) := V(s)^T x_s$ are computed for all $s \in T_J$. Exploiting the nestedness (18) of the cluster bases $\mathcal{V}$, one has the following recursive relation
   $$\hat{x}(s) = V(s)^T x_s = \sum_{s' \in S_J(s)} C_{\sigma_{s's}} V(s')^T x_{s'} = \sum_{s' \in S_J(s)} C_{\sigma_{s's}} \hat{x}(s'),$$
   which has to be applied starting from the vectors $\hat{x}(s) := V(s)^T x_s$, $s \in \mathcal{L}(T_J)$.

2. *Far field interaction*
   In the second phase, the products $S(t,s)\hat{x}(s)$ are computed and summed up over all clusters s satisfying $t \times s \in P_{\mathrm{adm}}$:

$$\hat{y}(t) := \sum_{s:t\times s\in P_{\mathrm{adm}}} S(t,s)\hat{x}(s), \quad t \in T_I.$$

3. *Backward transform*
   The third phase transforms the vectors $\hat{y}(t)$ to the target vector y. The nestedness (16) of the cluster basis $\mathcal{U}$ provides the following recursion, which descends the cluster tree T_I:
   (a) Compute $\hat{y}(t') := \hat{y}(t') + B_{t't}\hat{y}(t)$ for all $t' \in S_I(t)$;
   (b) Compute $y_t := y_t + U(t)\hat{y}(t)$ for all clusters $t \in \mathcal{L}(T_I)$.

4. *Near field interaction*
   For all $t \times s \in P_{\mathrm{nonadm}}$ compute $y_t := y_t + A_{ts}x_s$.

The number of operations required to perform step 1 to 4 is of the order $\mathcal{O}(k_\varepsilon(|I| + |J|))$; see [21].

5 Numerical experiments

In this section, we test the algorithm by approximating matrices $A \in \mathbb{R}^{I\times I}$ arising from the discretization of the classical single layer potential operator on two boundaries "cylinder" and "hinge", i.e., the matrix entries are defined as in (1) with piecewise constant basis functions $\varphi_i = \psi_i$, $i \in I = J$, and the singularity function

$$\kappa(x,y) = \frac{1}{4\pi|x-y|}.$$

Due to symmetry, it suffices to compute the upper triangular part of the matrix approximant, where for admissible blocks $t \times s$ we have $A_{ts} \approx U(t)S(t,s)U(s)^H$. For the computation of the representor sets τ_t and σ_t one of the two methods explained in Sect. 4.3 is used. The geometric one will be referred to as "ACAgeo", the recursive method will be labeled "ACAmerge".

For partitioning $I \times I$, the minimal cluster size $n_{\min} = 15$ and the parameter $\eta = 0.8$ is used. When comparing the nested basis approach with the standard ACA approach, we choose the value $\eta = 1.1$ (in terms of computational costs) for computing $\mathcal{H}$-matrices via ACA. In the following tables, the achieved accuracy (labeled "acc.") of an $\mathcal{H}^2$-matrix computed by nested ACA is the relative error in spectral norm between the latter matrix and the $\mathcal{H}$-matrix obtained by the standard ACA. Furthermore, the compression rate (labeled "compr."), i.e. the reduction in memory over entrywise storage, and the CPU time of a single matrix-vector multiplication (labeled "MV mult.") will be presented. All computations were performed using the A$\mathcal{H}$MED-library[1] on a single core of an Intel Xeon processor running at 2.53 GHz.

Table 1 and Table 2 compare the $\mathcal{H}^2$-matrices obtained from the methods ACAmerge and ACAgeo. ACAmerge shows a slight advantage in speed, which becomes clearer for larger numbers of degrees of freedom.

Table 1: Comparison of ACAmerge and ACAgeo with $\varepsilon = 10^{-3}$

	ACAmerge			ACAgeo		
Dofs	time	compr.	acc.	time	compr.	acc.
3136 (cylinder)	4.0s	13.77%	3.8_{-3}	4.0s	13.67%	2.3_{-3}
12288 (cylinder)	13.5s	3.11%	4.4_{-3}	13.7s	3.12%	1.2_{-2}
24448 (hinge)	35.1s	2.10%	6.3_{-3}	35.6s	2.10%	8.8_{-4}
97792 (hinge)	109.5s	0.41%	3.1_{-2}	112.4s	0.42%	4.5_{-2}

Table 2: Comparison of ACAmerge and ACAgeo with $\varepsilon = 10^{-4}$

	ACAmerge			ACAgeo		
Dofs	time	compr.	acc.	time	compr.	acc.
3136 (cylinder)	7.7s	22.57%	4.4_{-6}	7.9s	22.49%	2.8_{-6}
12288 (cylinder)	25.71s	4.91%	5.1_{-4}	27.1s	4.97%	5.5_{-4}
24448 (hinge)	61.8s	3.13%	2.7_{-3}	65.0s	3.18%	3.7_{-3}
97792 (hinge)	186.5s	0.59%	3.3_{-3}	199.4s	0.59%	2.9_{-4}

In Tables 3 and 4, one observes the improvement of nested ACA over standard ACA in building time, compression rate and performance of the matrix-vector multiplication. For small numbers of degrees of freedom the standard ACA approach might perform better.

Table 3: Comparison of standard and nested ACA with $\varepsilon = 10^{-3}$

Dofs		time	compr.	MV mult.	acc.
3136 (cylinder)	nested:	4.00s	13.77%	0.0031s	3.8_{-3}
	standard:	4.21s	13.18%	0.0028s	
12288 (cylinder)	nested:	13.54s	3.11%	0.0119s	3.2_{-3}
	standard:	22.83s	4.45%	0.0143s	
24448 (hinge)	nested:	35.05s	2.10%	0.0306s	4.5_{-3}
	standard:	60.96s	2.90%	0.0364s	
97792 (hinge)	nested:	109.5s	0.41%	0.1028s	7.1_{-3}
	standard:	302.6s	0.87%	0.1940s	

Finally, we compare whether the nested ACA approximant can be further improved in terms of storage costs. For this purpose we use the recompression procedure [11] after building the matrix approximation. In Table 5 and Table 6 we compare storage costs and computation time of ACA and nested ACA after coarsening the $\mathcal{H}$-matrix using the technique in [16] and recompressing the $\mathcal{H}^2$-matrix, respectively. We observe that the storage costs of nested ACA can be improved by a factor 2, whereas recompressing the $\mathcal{H}$-matrix obtained from standard

[1] see `http://bebendorf.ins.uni-bonn.de/AHMED.html`

Table 4: Comparison of standard and nested ACA with $\varepsilon = 10^{-4}$

Dofs		time	compr.	MV mult.	acc.
3136 (cylinder)	nested:	7.73s	22.57%	0.0051s	4.4_{-6}
	standard:	5.70s	18.44%	0.0038s	
12288 (cylinder)	nested:	25.7s	4.91%	0.0175s	5.1_{-4}
	standard:	31.4s	6.43%	0.0209s	
24448 (hinge)	nested:	61.8s	3.13%	0.0436s	2.7_{-3}
	standard:	86.4s	4.36%	0.0562s	
97792 (hinge)	nested:	186.5s	0.59%	0.1406s	3.3_{-3}
	standard:	435.8s	1.34%	0.3011s	

ACA leads to only small improvements.

In conclusion, the improvement in memory requirements of nested ACA in combination with $\mathcal{H}^2$-matrix recompression is significant compared with the standard ACA approach, especially for large numbers of degrees of freedom.

Table 5: Recompression of matrix obtained from standard and nested ACA, $\varepsilon = 10^{-3}$

Dofs		time	compr.	MV mult.	acc.
3136 (cylinder)	nested:	0.04s	7.28%	0.0018s	3.8_{-3}
	standard:	0.17s	11.76%	0.0023s	
12288 (cylinder)	nested:	0.13s	1.87%	0.0081s	3.5_{-3}
	standard:	0.62s	4.06%	0.0124s	
24448 (hinge)	nested:	0.29s	1.38%	0.0227s	1.1_{-2}
	standard:	1.94s	2.49%	0.0296s	
97792 (hinge)	nested:	0.99s	0.29%	0.0812s	3.7_{-2}
	standard:	6.81s	0.77%	0.1639s	

References

[1] Christopher R. Anderson. An implementation of the fast multipole method without multipoles. *SIAM J. Sci. Statist. Comput.*, 13(4):923–947, 1992.

[2] M.-B. A. Babaev. Best approximation by bilinear forms. *Mat. Zametki*, 46(2):21–33, 158, 1989.

[3] M.-B. A. Babaev. Exact annihilators and their applications in approximation theory. *Trans. Acad. Sci. Azerb. Ser. Phys.-Tech. Math. Sci.*, 20(1, Math. Mech.):17–24, 233, 2000.

Table 6: Recompression of matrix obtained from standard and nested ACA, $\varepsilon = 10^{-4}$

Dofs		time	compr.	MV mult.	acc.
3136 (cylinder)	nested:	0.08s	7.71%	0.0021s	2.5_{-3}
	standard:	0.24s	17.15%	0.0033s	
12288 (cylinder)	nested:	0.25s	1.95%	0.0086s	2.7_{-3}
	standard:	1.06s	5.99%	0.0187s	
24448 (hinge)	nested:	0.55s	1.43%	0.2400s	2.9_{-3}
	standard:	2.10s	3.86%	0.0478s	
24448 (hinge)	nested:	1.67s	0.30%	0.0871s	3.3_{-3}
	standard:	11.25s	1.21%	0.2619s	

[4] J. Ballani, L. Grasedyck, and M. Kluge. Black box approximation of tensors in hierarchical tucker format. Technical Report 57, Max Planck Institute MiS, 2010.

[5] J. Barnes and P. Hut. A hierarchical $\mathcal{O}(n \ln n)$ force calculation algorithm. *Nature*, 324:446–449, 1986.

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

[7] 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, 2008. ISBN 978-3-540-77146-3.

[8] M. Bebendorf and R. Grzhibovskis. Accelerating Galerkin BEM for Linear Elasticity using Adaptive Cross Approximation. *Mathematical Methods in the Applied Sciences*, 29:1721–1747, 2006.

[9] M. Bebendorf and S. Kunis. Recompression techniques for adaptive cross approximation. *Journal of Integral Equations and Applications*, 21(3):331–357, 2009.

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

[11] S. Börm. Construction of data-sparse $\mathcal{H}^2$-matrices by hierarchical compression. *SIAM J. Sci. Comp.*, 31(3):1820–1839, 2009.

[12] A. Brandt and A. A. Lubrecht. Multilevel matrix multiplication and fast solution of integral equations. *J. Comput. Phys.*, 90(2):348–370, 1990.

[13] B. Engquist and L. Ying. Fast directional multilevel algorithms for oscillatory kernels. *SIAM J. Sci. Comput.*, 29(4):1710–1737 (electronic), 2007.

[14] K. Giebermann. Multilevel approximation of boundary integral operators. *Computing*, 67:183–207, 2001.

[15] S. A. Goreinov, E. E. Tyrtyshnikov, and N. L. Zamarashkin. A theory of pseudoskeleton approximations. *Linear Algebra Appl.*, 261:1–21, 1997.

[16] L. Grasedyck. Adaptive recompression of $\mathcal{H}$-matrices for BEM. *Computing*, 74:205–223, 2005.

[17] L. Grasedyck and W. Hackbusch. Construction and arithmetics of $\mathcal{H}$-matrices. *Computing*, 70:295–334, 2003.

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

[19] L. F. Greengard and V. Rokhlin. A new version of the fast multipole method for the Laplace equation in three dimensions. In *Acta numerica, 1997*, volume 6 of *Acta Numer.*, pages 229–269. Cambridge Univ. Press, Cambridge, 1997.

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

[21] W. Hackbusch and S. Börm. Data-sparse approximation by adaptive $\mathcal{H}^2$-matrices. *Computing*, 69(1):1–35, 2002.

[22] W. Hackbusch and S. Börm. $\mathcal{H}^2$-matrix approximation of integral operators by interpolation. *Appl. Numer. Math.*, 43(1-2):129–143, 2002. 19th Dundee Biennial Conference on Numerical Analysis (2001).

[23] 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.

[24] W. Hackbusch, B. N. Khoromskij, and S. A. Sauter. On $\mathcal{H}^2$-matrices. In H.-J. Bungartz, R. H. W. Hoppe, and Ch. Zenger, editors, *Lectures on Applied Mathematics*, pages 9–29. Springer-Verlag, Berlin, 2000.

[25] 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.

[26] C. A. Micchelli and A. Pinkus. Some problems in the approximation of functions of two variables and n-widths of integral operators. *J. Approx. Theory*, 24(1):51–77, 1978.

[27] I. V. Oseledets and E. E. Tyrtyshnikov. TT-Cross Approximation for Multidimensional Arrays. *Linear Algebra Appl.*, 432(5):70–88, 2010.

[28] Th. Sauer and Y. Xu. On multivariate Lagrange interpolation. *Math. Comp.*, 64(211):1147–1170, 1995.

[29] J. Schneider. Error estimates for two-dimensional Cross Approximation. Technical Report 5, Max-Planck-Institute MiS, 2009.

[30] E. E. Tyrtyshnikov. Mosaic-skeleton approximations. *Calcolo*, 33(1-2):47–57 (1998), 1996. Toeplitz matrices: structures, algorithms and applications (Cortona, 1996).

[31] Lexing Ying, George Biros, and Denis Zorin. A kernel-independent adaptive fast multipole algorithm in two and three dimensions. *J. Comput. Phys.*, 196(2):591–626, 2004.

Bestellungen nimmt entgegen:

Sonderforschungsbereich 611
der Universität Bonn
Endenicher Allee 60
D - 53115 Bonn

Telefon: 0228/73 4882
Telefax: 0228/73 7864
E-Mail: astrid.link@ins.uni-bonn.de http://www.sfb611.iam.uni-bonn.de/

Verzeichnis der erschienenen Preprints ab No. 460

460. Brenier, Yann; Otto, Felix; Seis, Christian: Upper Bounds on Coarsening Rates in Demixing Binary Viscous Liquids

461. Bianchi, Alessandra; Bovier, Anton; Ioffe, Dmitry: Pointwise Estimates and Exponential Laws in Metastable Systems Via Coupling Methods

462. Basile, Giada; Bovier, Anton: Convergence of a Kinetic Equation to a Fractional Diffusion Equation; erscheint in: Review Markov Processes and Related Fields

463. Bartels, Sören; Roubíček, Tomáš: Thermo-Visco-Elasticity with Rate-Independent Plasticity in Isotropic Materials Undergoing Thermal Expansion

464. Albeverio, Sergio; Torbin, Grygoriy: The Ostrogradsky-Pierce Expansion: Probability Theory, Dynamical Systems and Fractal Geometry Points of View

465. Capella Kort, Antonio; Otto, Felix: A Quantitative Rigidity Result for the Cubic to Tetragonal Phase Transition in the Geometrically Linear Theory with Interfacial Energy

466. Philipowski, Robert: Stochastic Particle Approximations for the Ricci Flow on Surfaces and the Yamabe Flow

467. Kuwada, Kazumasa; Philipowski, Robert: Non-explosion of Diffusion Processes on Manifolds with Time-dependent Metric; erscheint in: Mathematische Zeitschrift

468. Bacher, Kathrin; Sturm, Karl-Theodor: Ricci Bounds for Euclidean and Spherical Cones

469. Bacher, Kathrin; Sturm, Karl-Theodor: Localization and Tensorization Properties of the Curvature-Dimension Condition for Metric Measure Spaces

470. Le Peutrec, Dorian: Small Eigenvalues of the Witten Laplacian Acting on p-Forms on a Surface

471. Wirth, Benedikt; Bar, Leah; Rumpf, Martin; Sapiro, Guillermo: A Continuum Mechanical Approach to Geodesics in Shape Space

472. Berkels, Benjamin; Linkmann, Gina; Rumpf, Martin: An SL (2) Invariant Shape Median

473. Bartels, Sören; Schreier, Patrick: Local Coarsening of Triangulations Created by Bisections

474. Bartels, Sören: A Lower Bound for the Spectrum of the Linearized Allen-Cahn Operator Near a Singularity

475. Frehse, Jens; Löbach, Dominique: Improved Lp-Estimates for the Strain Velocities in Hardening Problems

476. Kurzke, Matthias; Melcher, Christof; Moser, Roger: Vortex Motion for the Landau-Lifshitz-Gilbert Equation with Spin Transfer Torque

477. Arguin, Louis-Pierre; Bovier, Anton; Kistler, Nicola: The Genealogy of Extremal Particles of Branching Brownian Motion

478. Bovier, Anton; Gayrard, Véronique: Convergence of Clock Processes in Random Environments and Ageing in the p-Spin SK Model

479. Bartels, Sören; Müller, Rüdiger: Error Control for the Approximation of Allen-Cahn and Cahn-Hilliard Equations with a Logarithmic Potential

480. Albeverio, Sergio; Kusuoka, Seiichiro: Diffusion Processes in Thin Tubes and their Limits on Graphs

481. Arguin, Louis-Pierre; Bovier, Anton; Kistler, Nicola: Poissonian Statistics in the Extremal Process of Branching Brownian Motion

482. Albeverio, Sergio; Pratsiovyta, Iryna; Torbin, Grygoriy: On the Probabilistic, Metric and Dimensional Theories of the Second Ostrogradsky Expansion

483. Bulíček, Miroslav; Frehse, Jens: C^{α}-Regularity for a Class of Non-Diagonal Elliptic Systems with p-Growth

484. Ferrari, Partik L.: From Interacting Particle Systems to Random Matrices

485. Ferrari, Partik L.; Frings, René: On the Partial Connection Between Random Matrices and Interacting Particle Systems

486. Scardia, Lucia; Zeppieri, Caterina Ida: Line-Tension Model as the Γ–Limit of a Nonlinear Dislocation Energy

487. Bolthausen, Erwin; Kistler, Nicola: A Quenched Large Deviation Principle and a Parisi Formula for a Perceptron Version of the Grem

488. Griebel, Michael; Harbrecht, Helmut: Approximation of Two-Variate Functions: Singular Value Decomposition Versus Regular Sparse Grids

489. Bartels, Sören; Kruzik, Martin: An Efficient Approach of the Numerical Solution of Rate-independent Problems with Nonconvex Energies

490. Bartels, Sören; Mielke, Alexander; Roubicek, Tomas: Quasistatic Small-strain Plasticity in the Limit of Vanishing Hardening and its Numerical Approximation

491. Bebendorf, Mario; Venn, Raoul: Constructing Nested Bases Approximations from the Entries of Non-local Operators