Max-Planck-Institut
für Mathematik
in den Naturwissenschaften
Leipzig

Adaptive variable-rank approximation of general matrices

(revised version: February 2007)

by

Steffen Börm

Preprint no.: 114 2005

ADAPTIVE VARIABLE-RANK APPROXIMATION OF GENERAL DENSE MATRICES

STEFFEN BÖRM*

Abstract. In order to handle large dense matrices arising in the context of integral equations efficiently, panel-clustering approaches (like the popular multipole expansion method) have proven to be very useful. These techniques split the matrix into blocks, approximate the kernel function on each block by a degenerate expansion, and discretize this expansion in order to find an efficient low-rank approximation of each block.

Typical expansion schemes use the same expansion order in each block, and since the matrix approximation error has to be kept consistent with the discretization error, this allows us to handle $n \times n$ matrices by algorithms with a complexity of $\mathcal{O}(n \log^\alpha n)$ for $\alpha \geq 1$.

Recently, variable-order expansion schemes have been introduced, which choose different ranks for different blocks and have been demonstrated to reach a complexity of $\mathcal{O}(n)$ while keeping the matrix approximation error and the discretization error consistent.

This paper introduces an algorithm which can construct variable-rank approximations for general matrices without the need of an in-depth analysis of the underlying operators: the matrix is fed into the algorithm, and the algorithm approximates it up to an arbitrary precision.

Key words. Hierarchical matrices, data-sparse approximation, non-local operators, variable-order approximation

AMS subject classifications. 65F30,65N38

Acknowledgement. A significant part of this work has been conducted during the Special Radon Semester on Computational Mechanics, October 3 - December 16, 2005, and supported by RICAM, Austrian Academy of Sciences, Linz.

1. Introduction. We are interested in efficient algorithms for storing, and working with, a matrix $M \in \mathbb{R}^{\mathcal{I} \times \mathcal{I}}$, where $\mathcal{I}$ is a general index set of cardinality $n \in \mathbb{N}$. Storing M directly requires $\mathcal{O}(n^2)$ units of storage, i.e., the storage complexity depends quadratically on the number of degrees of freedom. Therefore this representation is only efficient for small matrices.

For large matrices, different techniques have been developed: many matrices resulting from the discretization of partial differential equations are *sparse*, i.e., the number of non-zero entries in each of their rows can be bounded by a constant. Storing only these non-zero entries and treating the remaining ones implicitly yields a representation which requires only $\mathcal{O}(n)$ units of storage.

Another type of large matrix arises from the discretization of integral equations: since the integral kernels typically have global support, the resulting matrices have $\mathcal{O}(n^2)$ non-zero entries and cannot be treated by sparse matrix techniques. There are different techniques for handling this type of matrix. Most prominent are panel-clustering and wavelet techniques. Panel-clustering techniques [16, 18] and the closely related multipole methods [17, 13] approximate the kernel function locally by degenerate expansions, thereby describing the interaction of subdomains very efficiently. Wavelet techniques [9, 8] use a Galerkin approach with special basis functions in order to ensure that the resulting matrix is essentially sparse, i.e., sparse up to entries of negligible size.

We will focus here on panel-clustering techniques. The basic idea is to split the index set $\mathcal{I} \times \mathcal{I}$ corresponding to the matrix M into subblocks $t \times s$, where $t, s \subseteq \mathcal{I}$, and approximate the subblocks $M|_{t \times s}$ by low-rank matrices $\widetilde{M}|_{t \times s}$. If we represent

*Max Planck Institute for Mathematics in the Sciences, Inselstraße 22–26, 04103 Leipzig, Germany

the low-rank matrices in an efficient factorized format, the resulting approximation $\widetilde{M}$ of M is called a *hierarchical matrix* [14, 4, 3] and requires only $\mathcal{O}(nk \log n)$ units of storage, where $k \in \mathbb{N}$ is an upper bound for the rank of the subblocks.

If we use a more efficient factorization of the subblocks, we arrive at the $\mathcal{H}^2$*-matrix* representation [15, 5], which will in general require only $\mathcal{O}(nk)$ units of storage.

In practical applications, the rank k controls the accuracy of the approximation: usually, we have a relationship of the kind $k \sim \log^\alpha(1/\epsilon)$, where $\epsilon \in \mathbb{R}_{>0}$ is the desired accuracy and $\alpha \in \mathbb{N}$ is a parameter depending on the approximation scheme.

When dealing with matrices resulting from the discretization of an integral or partial differential equation, we want the matrix approximation error to be proportional to the discretization error. If we assume that the discretization error behaves like $\epsilon \sim n^{-\beta}$, where n is the number of degrees of freedom and $\beta \in \mathbb{R}_{>0}$ is determined by the discretization scheme, this means that the rank k has to behave like $k \sim \beta \log^\alpha(n)$, which implies an effective complexity of $\mathcal{O}(n \log^{\alpha+1} n)$ for hierarchical matrices and $\mathcal{O}(n \log^\alpha n)$ for $\mathcal{H}^2$-matrices.

Especially for very large problems, the additional polylogarithmic factor leads to a significant increase in the computing time, therefore we would like to get rid of it and reach the *optimal* complexity $\mathcal{O}(n)$. For $\mathcal{H}^2$-matrices, this can be achieved by *variable-rank approximations* [18]: we use a low rank for small subblocks and a high rank for large ones. For a certain class of matrices, the rank can be chosen in such a way that a complexity of $\mathcal{O}(n)$ is sufficient for keeping the matrix approximation error consistent with the discretization error [18, 7, 19].

Obviously, methods of optimal complexity are very desirable when dealing with large problems, and we would like to have methods of this kind at our disposal for as many problem classes as possible. Unfortunately, the results given in [18, 7, 19] are closely connected to a special kind of matrix resulting from a special kind of discretization of a special kind of integral operator, and they rely on a careful analysis of the underlying continuous problem. For general applications, a purely algebraic method is desirable: we would like to have an algorithm which takes an arbitrary matrix and constructs an efficient variable-rank approximation of prescribed accuracy.

This paper is devoted to the construction of an algorithm matching this description. Since we will formulate it in the language of $\mathcal{H}^2$-matrices, we have to briefly recall the corresponding basic definitions. Then we introduce the algorithm and prove that it is efficient. A careful error analysis allows us to choose the truncation thresholds required by the algorithm in such a way that a prescribed accuracy is reached. First numerical experiments demonstrate the practical applicability of the new method.

2. $\mathcal{H}^2$-matrices.

We will now briefly recall the structure of $\mathcal{H}^2$-matrices [15, 5].

2.1. Block structure.

Hierarchical matrix techniques are based on detecting subblocks of the matrix which admit a data-sparse approximation. In order to find these *admissible* blocks efficiently, we introduce a hierarchy of subsets:

DEFINITION 2.1 (Cluster tree). *Let $\mathcal{I}$ be an index set. Let $\mathcal{T}$ be a labeled tree. We denote its root by* $\operatorname{root}(\mathcal{T})$*, the label of $t \in \mathcal{T}$ by $\hat{t}$, and the set of sons by* $\operatorname{sons}(t, \mathcal{T})$ *(or just* $\operatorname{sons}(t)$ *if this does not lead to ambiguity).*

$\mathcal{T}$ is a cluster tree *for $\mathcal{I}$ if it satisfies the following conditions:*

- $\widehat{\operatorname{root}(\mathcal{T})} = \mathcal{I}$.

- *If* $\operatorname{sons}(t) \neq \emptyset$ *holds for* $t \in \mathcal{T}$*, we have*

$$\hat{t} = \bigcup_{s \in \operatorname{sons}(t)} \hat{s} \quad \textit{and}$$
$$\hat{s}_1 \cap \hat{s}_2 = \emptyset \textit{ for all } s_1, s_2 \in \operatorname{sons}(t) \textit{ with } s_1 \neq s_2.$$

If $\mathcal{T}$ *is a cluster tree for* $\mathcal{I}$*, we will denote it by* $\mathcal{T}_\mathcal{I}$ *and call its nodes* clusters*. The set of* leaves *of* $\mathcal{T}_\mathcal{I}$ *is denoted by*

$$\mathcal{L}_\mathcal{I} := \{t \in \mathcal{T}_\mathcal{I} \ : \ \operatorname{sons}(t) = \emptyset\}.$$

The definition implies $\hat{t} \subseteq \mathcal{I}$ for all clusters $t \in \mathcal{T}_\mathcal{I}$. We can use induction to prove that the set $\mathcal{L}_\mathcal{I}$ of leaves of $\mathcal{T}_\mathcal{I}$ is a disjoint partition of the index set $\mathcal{I}$.

The *level* of a cluster $t \in \mathcal{T}_\mathcal{I}$ is defined by

$$\operatorname{level}(t) := \begin{cases} \operatorname{level}(t^+) + 1 & \text{if there is a } t^+ \in \mathcal{T}_\mathcal{I} \text{ with } t \in \operatorname{sons}(t^+), \\ 0 & \text{otherwise, i.e., if } t = \operatorname{root}(\mathcal{T}_\mathcal{I}). \end{cases}$$

For each $\ell \in \mathbb{N}_0$, we define the set

$$\mathcal{T}_\mathcal{I}^{(\ell)} := \{t \in \mathcal{T}_\mathcal{I} \ : \ \operatorname{level}(t) = \ell\}$$

of clusters with level ℓ. The *set of descendants* of a cluster $t \in \mathcal{T}_\mathcal{I}$ is defined by

$$\operatorname{sons}^*(t) := \begin{cases} \{t\} \cup \bigcup_{s \in \operatorname{sons}(t)} \operatorname{sons}^*(s) & \text{if } \operatorname{sons}(t) \neq \emptyset, \\ \{t\} & \text{otherwise.} \end{cases}$$

Using cluster trees, we can now define a partition of the matrix entries:

DEFINITION 2.2 (Block partition). *Let* $\mathcal{I}$ *and* $\mathcal{J}$ *be finite index sets, and let* $\mathcal{T}_\mathcal{I}$ *and* $\mathcal{T}_\mathcal{J}$ *be corresponding cluster trees. A set* $P \subseteq \mathcal{T}_\mathcal{I} \times \mathcal{T}_\mathcal{J}$ *is a* block partition *if* $\{\hat{t} \times \hat{s} \ : \ (t, s) \in P\}$ *is a disjoint partition of* $\mathcal{I} \times \mathcal{J}$*. We will call the elements of* P blocks*.*

The admissible blocks, i.e., those that can be treated by a data-sparse approximation, are picked from the elements of P:

DEFINITION 2.3 (Admissibility). *Let* P *be a block partition for* $\mathcal{T}_\mathcal{I}$ *and* $\mathcal{T}_\mathcal{J}$*. Let* $P_{\text{near}} \subseteq P$ *be such that for all* $(t, s) \in P_{\text{near}}$ *the equation* $\operatorname{sons}(t) = \emptyset = \operatorname{sons}(s)$ *holds. Then* P_{near} *is called a* nearfield *for* P*, and* $P_{\text{far}} := P \setminus P_{\text{near}}$ *is called the corresponding* farfield*. The blocks in* P_{far} *are called* admissible blocks*, the blocks in* P_{near} *are called* inadmissible blocks*. Whenever we introduce a block partition, we assume that matching sets* P_{near} *and* P_{far} *are implied.*

Due to this definition, inadmissible blocks correspond to leaves of the cluster trees, i.e., to small subsets of $\mathcal{I} \times \mathcal{J}$ which we can afford to store in the standard format. An efficient representation is only required for admissible blocks.

In practice, these blocks are identified by an *admissibility condition*. For matrices resulting from the discretization of elliptic problems, the condition

$$\max\{\operatorname{diam}(\Omega_t), \operatorname{diam}(\Omega_s)\} \leq \operatorname{dist}(\Omega_t, \Omega_s) \tag{2.1}$$

is frequently used, where Ω_t and Ω_s are suitable domains containing the supports of the basis functions or functionals corresponding to t and s.

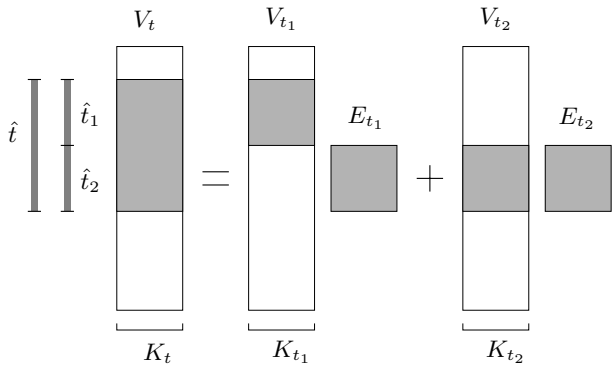

FIG. 2.1. *Nested cluster basis*

The condition (2.1) ensures that we are dealing with a region where we can expect Green's function to be smooth or at least separable. In the case $\mathcal{I} = \mathcal{J}$, this means that the block $\hat{t} \times \hat{s}$ lies "sufficiently far away" from the diagonal of the matrix.

If the indices in $\mathcal{I}$ and $\mathcal{J}$ correspond to locations in space, it is possible to construct good cluster trees $\mathcal{T}_\mathcal{I}$ and $\mathcal{T}_\mathcal{J}$ by binary space partitioning and a good block partition $P = P_{\text{far}} \dot{\cup} P_{\text{near}}$ by a simple recursion strategy [11, 12].

2.2. Factorized representation. Typical hierarchical matrices are defined based on the block partition P: for all admissible blocks $b = (t, s) \in P_{\text{far}}$, the corresponding matrix block $M|_{\hat{t}\times\hat{s}}$ is required to be of low rank and stored in an appropriate factorized form.

The $\mathcal{H}^2$-matrix format is a specialization of this representation: we require not only that admissible blocks correspond to low-rank matrix blocks, but also that the ranges of these blocks and their adjoints are contained in predefined spaces.

In order to simplify the presentation, we introduce a restriction operator $\chi_t : \mathcal{I} \to \mathcal{I}$ for each $t \in \mathcal{T}_\mathcal{I}$ by

$$(\chi_t)_{ij} = \begin{cases} 1 & \text{if } i = j \in \hat{t}, \\ 0 & \text{otherwise,} \end{cases} \qquad \text{for all } i, j \in \mathcal{I}.$$

Restriction operators $\chi_s : \mathcal{J} \to \mathcal{J}$ for $s \in \mathcal{T}_\mathcal{J}$ are defined in a similar fashion. For $t \in \mathcal{T}_\mathcal{I}$, $s \in \mathcal{T}_\mathcal{J}$, the matrix $\chi_t M \chi_s \in \mathbb{R}^{\mathcal{I}\times\mathcal{J}}$ is equal to M in the sub-block $\hat{t} \times \hat{s}$ and zero everywhere else.

DEFINITION 2.4 (Cluster basis). *Let $\mathcal{T}_\mathcal{I}$ be a cluster tree. A family $k = (k_t)_{t\in\mathcal{T}_\mathcal{I}}$ of integers is called* rank distribution. *For a given rank distribution k, a family $V = (V_t)_{t\in\mathcal{T}_\mathcal{I}}$ satisfying $V_t \in \mathbb{R}^{\mathcal{I}\times k_t}$ and $\chi_t V_t = V_t$ for all $t \in \mathcal{T}_\mathcal{I}$ is called* cluster basis *for $\mathcal{T}_\mathcal{I}$ with rank distribution k.*

We can see that this definition implies $(V_t)_{i\nu} = 0$ for all $t \in \mathcal{T}_\mathcal{I}$, $i \in \mathcal{I} \setminus \hat{t}$ and $\nu \in \{1, \ldots, k_t\}$, i.e., only matrix rows corresponding to indices in $\hat{t}$ can differ from zero.

DEFINITION 2.5 (Nested cluster bases). *Let $\mathcal{T}_\mathcal{I}$ be a cluster tree, and let V be a corresponding cluster basis with rank distribution k. Let $E = (E_t)_{t\in\mathcal{T}_\mathcal{I}}$ be a family of matrices satisfying $E_t \in \mathbb{R}^{k_t\times k_{t^+}}$ for each cluster $t \in \mathcal{T}_\mathcal{I}$ that has a father $t^+ \in \mathcal{T}_\mathcal{I}$. If the equation*

$$V_t = \sum_{t' \in \text{sons}(t)} V_{t'} E_{t'} \tag{2.2}$$

(cf. Figure 2.1) holds for all $t \in \mathcal{T}_\mathcal{I}$ *with* $\operatorname{sons}(t) \neq \emptyset$*, the cluster basis* V *is called* nested *with* transfer matrices E*.*

The case $t = \operatorname{root}(\mathcal{T}_\mathcal{I})$ is only included in order to avoid the necessity of treating a special case: we can see that the definition does not require the transfer matrix for the root of $\mathcal{T}_\mathcal{I}$ to satisfy any conditions. In practice, this matrix can be ignored completely.

If a cluster basis $V = (V_t)_{t \in \mathcal{T}_\mathcal{I}}$ is nested, it satisfies the recursive equation

$$V_t = \begin{cases} \sum_{t' \in \operatorname{sons}(t)} V_{t'} E_{t'} & \text{if } \operatorname{sons}(t) \neq \emptyset, \\ V_t & \text{otherwise,} \end{cases} \qquad \text{for all } t \in \mathcal{T}_\mathcal{I}, \tag{2.3}$$

i.e., we do not have to store the matrices V_t for clusters $t \in \mathcal{T}_\mathcal{I} \setminus \mathcal{L}_\mathcal{I}$ which are not leaves, since we can rely on the transfer matrices $E_{t'}$ for $t' \in \operatorname{sons}(t)$ instead.

The nested structure is the key difference between general hierarchical matrices and $\mathcal{H}^2$-matrices [15, 5, 6], since it allows us to construct very efficient algorithms by re-using information across the entire cluster tree.

DEFINITION 2.6 ($\mathcal{H}^2$-matrix). *Let* $\mathcal{T}_\mathcal{I}$ *and* $\mathcal{T}_\mathcal{J}$ *be cluster trees. Let* $P = P_{\text{far}} \dot\cup P_{\text{near}}$ *be a block partition. Let* V *and* W *be nested cluster bases for* $\mathcal{T}_\mathcal{I}$ *and* $\mathcal{T}_\mathcal{J}$ *with rank distributions* k *and* l*. Let* $M \in \mathbb{R}^{\mathcal{I} \times \mathcal{J}}$*. If we can find matrices* $(S_b)_{b \in P_{\text{far}}}$ *satisfying*

$$S_b \in \mathbb{R}^{k_t \times l_s}, \qquad \chi_t M \chi_s = V_t S_b W_s^\top \qquad \text{for all } b = (t,s) \in P_{\text{far}}, \tag{2.4}$$

the matrix M *is called an* $\mathcal{H}^2$-matrix *with* row cluster basis V *and* column cluster basis W*. The family* $S = (S_b)_{b \in P_{\text{far}}}$ *is called the family of* coupling matrices*.*

The set of all $\mathcal{H}^2$*-matrices with row cluster basis* V*, column cluster basis* W *and block partition* P *is denoted by* $\mathcal{H}^2(P, V, W)$*.*

This definition implies that each $\mathcal{H}^2$-matrix can be written in the form

$$M = \sum_{b=(t,s) \in P_{\text{far}}} V_t S_b W_s^\top + \sum_{b=(t,s) \in P_{\text{near}}} \chi_t M \chi_s, \tag{2.5}$$

since $P = P_{\text{far}} \dot\cup P_{\text{near}}$ defines a partition of $\mathcal{I} \times \mathcal{J}$.

2.3. Complexity. Let us now consider the storage complexity of the $\mathcal{H}^2$-matrix representation.

Block partitions constructed for standard situations have an important property: for each $t \in \mathcal{T}_\mathcal{I}$, there is only a limited number of blocks of the form (t, s). For cluster trees and block partitions constructed by geometric bisection, an explicit bound for this number can be given, and this bound does not depend on the number of degrees of freedom [11, 12].

DEFINITION 2.7 (Sparse partition). *Let* P *be a block partition. Let* $C_{\text{sp}} \in \mathbb{N}$*. The partition* P *is* C_{sp}-sparse *if we have*

$$\#\{s \in \mathcal{T}_\mathcal{J} \ : \ (t,s) \in P\} \leq C_{\text{sp}} \qquad \text{for all } t \in \mathcal{T}_\mathcal{I}, \tag{2.6}$$

$$\#\{t \in \mathcal{T}_\mathcal{I} \ : \ (t,s) \in P\} \leq C_{\text{sp}} \qquad \text{for all } s \in \mathcal{T}_\mathcal{J}. \tag{2.7}$$

The complexity of most $\mathcal{H}^2$-matrix algorithms is determined by the behaviour of the rank distributions $k = (k_t)_{t \in \mathcal{T}_\mathcal{I}}$ and $l = (l_s)_{s \in \mathcal{T}_\mathcal{J}}$. The total complexity of most algorithms is determined by suitably defined average values of k and l: the rank can be allowed to be large in a small number of clusters as long as its stays small

in the majority of clusters. In order to prove *optimal* bounds, we require that, as the rank increases polynomially, the number of clusters exceeding this rank decreases exponentially:

DEFINITION 2.8 (Polynomial rank). *Let $k = (k_t)_{t\in\mathcal{T}_\mathcal{I}}$ be a rank distribution. Let $\hat{k} := (\hat{k}_t)_{t\in\mathcal{T}_\mathcal{I}}$ be a family defined by*

$$\hat{k}_t := \begin{cases} \max\left\{k_t, \sum_{t'\in\operatorname{sons}(t)} k_{t'}\right\} & \text{if } \operatorname{sons}(t) \neq \emptyset, \\ \max\left\{k_t, \#\hat{t}\right\} & \text{otherwise,} \end{cases} \qquad \text{for all } t \in \mathcal{T}_\mathcal{I}. \tag{2.8}$$

*Let $\alpha, \beta \in \mathbb{N}_0$, $\gamma \in \mathbb{N}$ and $\xi \in \mathbb{R}_{>1}$. The rank distribution k is $(\alpha,\beta,\gamma,\xi)$-*polynomial *if we have*

$$\#\{t \in \mathcal{T}_\mathcal{I} \ : \ \hat{k}_t > (\alpha+\beta\ell)^\gamma\} \leq \xi^{-\ell}\#\mathcal{T}_\mathcal{I} \qquad \text{for all } \ell \in \mathbb{N}_0. \tag{2.9}$$

Most papers on variable-order techniques rely on a level-wise splitting of the set of clusters: a high rank is allowed only for low levels, and the number of clusters is expected to increase exponentially in the level number.

It is easily verified that Definition 2.8 is more general than level-wise splittings: we can simply identify the level number and the parameter ℓ.

The more general approach presented here is required in order to be able to handle local irregularities arising from non-smooth boundaries or non-uniform triangulations.

LEMMA 2.9 (Storage of cluster bases). *Let $V = (V_t)_{t\in\mathcal{T}_\mathcal{I}}$ be a nested cluster basis with rank distribution $k = (k_t)_{t\in\mathcal{T}_\mathcal{I}}$ represented in the form (2.3). Let k be $(\alpha,\beta,\gamma,\xi)$-polynomial. Let $c_\mathcal{I} := \#\mathcal{T}_\mathcal{I}$ be the number of clusters in $\mathcal{T}_\mathcal{I}$. Then V requires $\mathcal{O}((\alpha+\beta)^{2\gamma}c_\mathcal{I})$ units of storage.*

Proof. Let $\hat{k}$ be defined as in (2.8). We define

$$\mathcal{C}_\ell := \begin{cases} \{t \in \mathcal{T}_\mathcal{I} \ : \ \hat{k}_t \leq \alpha^\gamma\} & \text{if } \ell = 0, \\ \{t \in \mathcal{T}_\mathcal{I} \ : \ (\alpha+\beta(\ell-1))^\gamma < \hat{k}_t \leq (\alpha+\beta\ell)^\gamma\} & \text{otherwise,} \end{cases} \tag{2.10}$$

for all $\ell \in \mathbb{N}_0$. We observe that $(\mathcal{C}_\ell)_{\ell=0}^\infty$ is a disjoint partition of $\mathcal{T}_\mathcal{I}$ and that Definition 2.8 implies

$$\#\mathcal{C}_\ell \leq \xi^{-(\ell-1)}c_\mathcal{I} = \xi\xi^{-\ell}c_\mathcal{I} \qquad \text{for all } \ell \in \mathbb{N}_0.$$

The cluster basis V is described by transfer matrices $(E_t)_{t\in\mathcal{T}_\mathcal{I}}$ for all clusters and the matrices $(V_t)_{t\in\mathcal{L}_\mathcal{I}}$ for leaf clusters. For all $t \in \mathcal{T}_\mathcal{I} \setminus \{\operatorname{root}(\mathcal{T}_\mathcal{I})\}$, the transfer matrix E_t requires $k_t k_{t^+}$ units of storage, where $t^+ \in \mathcal{T}_\mathcal{I}$ is the father of t. Therefore all transfer matrices require

$$\sum_{t\in\mathcal{T}_\mathcal{I}\setminus\{\operatorname{root}(\mathcal{T}_\mathcal{I})\}} k_t k_{t^+} = \sum_{t\in\mathcal{T}_\mathcal{I}} \sum_{t'\in\operatorname{sons}(t)} k_{t'} k_t \leq \sum_{t\in\mathcal{T}_\mathcal{I}} \hat{k}_t k_t \leq \sum_{t\in\mathcal{T}_\mathcal{I}} \hat{k}_t^2$$

units of storage. Due to the definition of $(\mathcal{C}_\ell)_{\ell=0}^\infty$, we have

$$\begin{aligned} \sum_{t\in\mathcal{T}_\mathcal{I}} \hat{k}_t^2 &= \sum_{\ell=0}^\infty \sum_{t\in\mathcal{C}_\ell} \hat{k}_t^2 \leq \sum_{\ell=0}^\infty \sum_{t\in\mathcal{C}_\ell} (\alpha+\beta\ell)^{2\gamma} \\ &\leq \xi c_\mathcal{I} \sum_{\ell=0}^\infty (\alpha+\beta\ell)^{2\gamma}\xi^{-\ell} \leq \xi c_\mathcal{I} \left(\alpha^{2\gamma} + (\alpha+\beta)^{2\gamma} \sum_{\ell=0}^\infty \ell^{2\gamma}\xi^{-\ell}\right), \end{aligned}$$

and since $\xi > 1$ holds, the sum can be bounded by a constant $C_{\mathrm{sum}} \in \mathbb{R}_{>0}$. This means

$$\sum_{t \in \mathcal{T}_{\mathcal{I}} \setminus \{\mathrm{root}(\mathcal{T}_{\mathcal{I}})\}} k_t k_{t^+} \leq \sum_{t \in \mathcal{T}_{\mathcal{I}}} \hat{k}_t^2 \leq (C_{\mathrm{sum}} + 1) \xi c_{\mathcal{I}} (\alpha + \beta)^{2\gamma}. \tag{2.11}$$

For all leaf clusters $t \in \mathcal{L}_{\mathcal{I}}$, the matrix V_t requires $(\#\hat{t})k_t$ units of storage, so the storage requirements for all leaf clusters can be bounded by

$$\sum_{t \in \mathcal{L}_{\mathcal{I}}} (\#\hat{t}) k_t \leq \sum_{t \in \mathcal{L}_{\mathcal{I}}} \hat{k}_t k_t \leq \sum_{t \in \mathcal{L}_{\mathcal{I}}} \hat{k}_t^2 \leq \sum_{t \in \mathcal{T}_{\mathcal{I}}} \hat{k}_t^2,$$

and we can proceed as in (2.11) in order to conclude the proof. ∎

LEMMA 2.10 (Storage of coefficients). *Let $V = (V_t)_{t \in \mathcal{T}_{\mathcal{I}}}$ and $W = (W_s)_{s \in \mathcal{T}_{\mathcal{J}}}$ be nested cluster bases with rank distributions $k = (k_t)_{t \in \mathcal{T}_{\mathcal{I}}}$ and $l = (l_s)_{s \in \mathcal{T}_{\mathcal{J}}}$. Let P be a block partition for $\mathcal{T}_{\mathcal{I}}$ and $\mathcal{T}_{\mathcal{J}}$. Let k and l be $(\alpha, \beta, \gamma, \xi)$-polynomial, and let P be C_{sp}-sparse. Let $c_{\mathcal{I}} := \#\mathcal{T}_{\mathcal{I}}$ and $c_{\mathcal{J}} := \#\mathcal{T}_{\mathcal{J}}$. Then the matrices $(S_b)_{b \in P_{\mathrm{far}}}$ and $(\chi_t M \chi_s)_{b \in P_{\mathrm{near}}}$ of a matrix M given in the form (2.5) require $\mathcal{O}((\alpha + \beta)^{2\gamma} c_{\mathcal{I}}^{1/2} c_{\mathcal{J}}^{1/2})$ units of storage.*

Proof. Let $\hat{k}$ be defined as in (2.8), let $\hat{l}$ be defined similarly for the rank distribution l.

Let us consider a block $b = (t, s) \in P$. If $b \in P_{\mathrm{far}}$, we store the matrix S_b, which requires $k_t l_s \leq \hat{k}_t \hat{l}_s$ units of storage. If $b \in P_{\mathrm{near}}$, we store the matrix $\chi_t M \chi_s$, which requires $(\#\hat{t})(\#\hat{s})$ units of storage. In this case, Definition 2.2 implies $t \in \mathcal{L}_{\mathcal{I}}$ and $s \in \mathcal{L}_{\mathcal{J}}$, i.e., we have $\#\hat{t} \leq \hat{k}_t$ and $\#\hat{s} \leq \hat{l}_s$, so the storage requirements are also bounded by $\hat{k}_t \hat{l}_s$.

Combining these estimates, we find that the coefficient matrices of all blocks require not more than

$$\begin{aligned}\sum_{b=(t,s) \in P} \hat{k}_t \hat{l}_s &\leq \left(\sum_{b=(t,s) \in P} \hat{k}_t^2 \right)^{1/2} \left(\sum_{b=(t,s) \in P} \hat{l}_s^2 \right)^{1/2} \\ &\leq \left(C_{\mathrm{sp}} \sum_{t \in \mathcal{T}_{\mathcal{I}}} \hat{k}_t^2 \right)^{1/2} \left(C_{\mathrm{sp}} \sum_{s \in \mathcal{T}_{\mathcal{J}}} \hat{l}_s^2 \right)^{1/2}\end{aligned}$$

units of storage. We can proceed as in (2.11) in order to conclude

$$\sum_{b=(t,s) \in P} \hat{k}_t \hat{l}_s \leq C_{\mathrm{sp}} (C_{\mathrm{sum}} + 1) \xi (c_{\mathcal{I}} c_{\mathcal{J}})^{1/2} (\alpha + \beta)^{2\gamma}.$$

This is the desired estimate. ∎

THEOREM 2.11 (Storage complexity). *Let $V = (V_t)_{t \in \mathcal{T}_{\mathcal{I}}}$ and $W = (W_s)_{s \in \mathcal{T}_{\mathcal{J}}}$ be nested cluster bases with rank distributions $k = (k_t)_{t \in \mathcal{T}_{\mathcal{I}}}$ and $l = (l_s)_{s \in \mathcal{T}_{\mathcal{J}}}$. Let P be a block partition for $\mathcal{T}_{\mathcal{I}}$ and $\mathcal{T}_{\mathcal{J}}$. Let k and l be $(\alpha, \beta, \gamma, \xi)$-polynomial, and let P be C_{sp}-sparse. Let $c_{\mathcal{I}} := \#\mathcal{T}_{\mathcal{I}}$ and $c_{\mathcal{J}} := \#\mathcal{T}_{\mathcal{J}}$. Then storing the matrix M, given in the form (2.5) with cluster bases given in the form (2.3), requires $\mathcal{O}((\alpha + \beta)^{2\gamma}(c_{\mathcal{I}} + C_{\mathrm{sp}} c_{\mathcal{I}}^{1/2} c_{\mathcal{J}}^{1/2} + c_{\mathcal{J}}))$ units of storage.*

Proof. Combine Lemma 2.9 with Lemma 2.10. ∎

For constant-order approximations, we have $\beta = 0$ and will usually use cluster trees with n/α^γ nodes. In this case, Theorem 2.11 yields a storage complexity of $\mathcal{O}(\alpha^\gamma n)$.

For variable-order approximations, we assume that α and β do not depend on n, and that the number of clusters is bounded by n. In this situation, Theorem 2.11 implies a storage complexity of $\mathcal{O}(n)$.

2.4. Orthogonal cluster bases and best approximations. A matrix format is defined by the partition P and the cluster bases V and W. Finding the best approximation of an arbitrary matrix in this format is simple if the columns of the cluster basis matrices V_t are pairwise orthonormal.

DEFINITION 2.12 (Orthogonal cluster basis). *Let V be a cluster basis for the cluster tree $\mathcal{T}_\mathcal{I}$. It is called* orthogonal *if $V_t^\top V_t = I$ holds for all $t \in \mathcal{T}_\mathcal{I}$.*

The orthogonality implies that $V_t V_t^\top$ is an orthogonal projection onto the image of V_t, since

$$\langle V_t V_t^\top x, V_t y\rangle = \langle V_t^\top V_t V_t^\top x, y\rangle = \langle V_t^\top x, y\rangle = \langle x, V_t y\rangle$$

holds for all $x \in \mathbb{R}^\mathcal{I}$ and $y \in \mathbb{R}^{k_t}$. Therefore $V_t V_t^\top M W_s W_s^\top$ is the best approximation of a matrix block $\chi_t M \chi_s$ in the bases V_t and W_s, and

$$\widetilde{M} := \sum_{b\in P_{\mathrm{far}}} V_t (V_t^\top M W_s) W_s^\top + \sum_{b\in P_{\mathrm{near}}} \chi_t M \chi_s \tag{2.12}$$

is the best approximation (in the Frobenius norm) of an arbitrary matrix $M \in \mathbb{R}^{\mathcal{I}\times\mathcal{J}}$ in the $\mathcal{H}^2$-matrix format defined by P, V and W.

If a non-nested cluster basis is given, an orthogonal counterpart can be constructed by simple Gram-Schmidt orthonormalization. If a nested cluster basis is given, it is possible to construct a nested orthogonal cluster basis by a modified orthonormalization algorithm in linear complexity [1].

3. Approximation algorithm. Let $M \in \mathbb{R}^{\mathcal{I}\times\mathcal{J}}$ be an arbitrary matrix, let $\mathcal{T}_\mathcal{I}$ and $\mathcal{T}_\mathcal{J}$ be cluster trees for $\mathcal{I}$ and $\mathcal{J}$, respectively, and let $P = P_{\mathrm{far}} \dot\cup P_{\mathrm{near}}$ be a matching block partition. Let $(\epsilon_{t,s})_{t\in\mathcal{T}_\mathcal{I}, s\in\mathcal{T}_\mathcal{J}}$ be a family of error tolerances in $\mathbb{R}_{>0}$ which will be specified later.

Our goal is to find suitable nested cluster bases $V = (V_t)_{t\in\mathcal{T}_\mathcal{I}}$ and $W = (W_s)_{s\in\mathcal{T}_\mathcal{J}}$ such that M can be approximated in the corresponding space $\mathcal{H}^2(P, V, W)$ of $\mathcal{H}^2$-matrices.

In order to construct V and W, we rely on a variant of the algorithm introduced in [5]. This algorithm creates orthogonal cluster bases by working recursively from the leaf clusters towards the root cluster.

We restrict our attention to V. Let $t \in \mathcal{T}_\mathcal{I}$. The matrix V_t has to be constructed in such a way that

$$\|\chi_t M \chi_s - V_t V_t^\top M \chi_s\|_2 \leq \epsilon_{t,s} \tag{3.1}$$

holds for all $s \in \mathcal{T}_\mathcal{J}$ with $b = (t, s) \in P_{\mathrm{far}}$.

Since we are looking for a nested cluster basis, our choice of the matrix V_t will also influence all predecessors of t, i.e., all clusters $t^+ \in \mathcal{T}_\mathcal{I}$ with $t \in \mathrm{sons}^*(t^+)$, therefore we have to ensure (3.1) also for all $s \in \mathcal{T}_\mathcal{J}$ for which a predecessor t^+ of t satisfying $b = (t^+, s) \in P_{\mathrm{far}}$ exists.

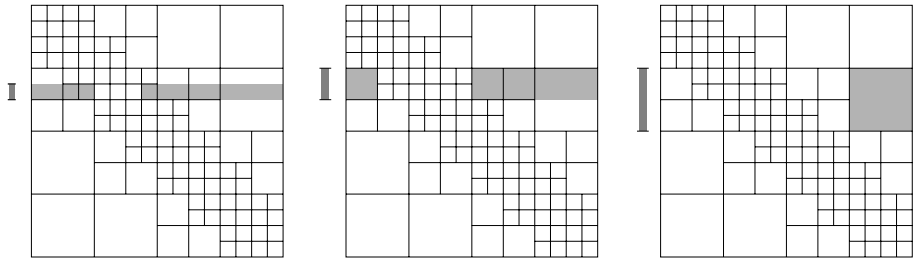

FIG. 3.1. *Matrices M_t for different clusters*

By collecting all relevant clusters $s \in \mathcal{T}_\mathcal{J}$ in the set

$$\operatorname{row}^*(t) := \{s \in \mathcal{T}_\mathcal{J} \ : \ \text{there exists } t^+ \in \mathcal{T}_\mathcal{I} \text{ with } t \in \operatorname{sons}^*(t^+) \\ \text{and } b = (t^+, s) \in P_{\mathrm{far}}\}$$

and introducing the matrix

$$M_t := \sum_{s \in \operatorname{row}^*(t)} \epsilon_{t,s}^{-1} \chi_t M \chi_s,$$

(cf. Figure 3.1) we can use the unified condition

$$\|M_t - V_t V_t^\top M_t\|_2 \leq 1 \tag{3.2}$$

as a weaker form of the conditions (3.1) for all $s \in \operatorname{row}^*(t)$. Our goal is to find orthogonal low-rank matrices V_t for all $t \in \mathcal{T}_\mathcal{I}$ which satisfy this condition.

For a leaf cluster $t \in \mathcal{L}_\mathcal{I}$, we can construct the optimal matrix V_t by computing the singular value decomposition

$$M_t = U_t \Sigma_t P_t^\top$$

of M_t, where the columns of U_t are the orthogonal left singular vectors, those of P_t are the orthogonal right singular vectors, and $\Sigma_t = \operatorname{diag}(\sigma_1, \ldots, \sigma_p)$ is a diagonal matrix containing the singular values $\sigma_1 \geq \ldots \geq \sigma_p \geq 0$ of M_t.

For any $k_t \in \{0, \ldots, p\}$, the first k_t left singular vectors, i.e., the first k_t columns of the matrix U_t, form an orthogonal matrix V_t satisfying

$$\|M_t - V_t V_t^\top M_t\|_2 \leq \begin{cases} \sigma_{k_t+1} & \text{if } k_t < p, \\ 0 & \text{otherwise,} \end{cases} \tag{3.3}$$

and $V_t V_t^\top M_t$ is the best rank k_t approximation of M_t [10, Theorem 2.5.3]. This means that we can construct an *optimal* matrix V_t satisfying the condition (3.2) by using

$$k_t := \begin{cases} \max\{i \in \{0, \ldots, p-1\} \ : \ \sigma_{i+1} < 1\} & \text{if } \sigma_p < 1, \\ p & \text{otherwise.} \end{cases}$$

Let us now consider the case of a cluster $t \in \mathcal{T}_\mathcal{I}$ which is not a leaf. We denote the number of sons by $\tau := \#\operatorname{sons}(t)$ and the sons by $\{t_1, \ldots, t_\tau\} := \operatorname{sons}(t)$. We assume that orthogonal matrices $V_{t_1}, \ldots, V_{t_\tau}$ have already been constructed. We are looking

for a nested cluster basis, so equation (2.2) has to hold, and this equation can be written as

$$V_t = \underbrace{\begin{pmatrix} V_{t_1} & \dots & V_{t_\tau} \end{pmatrix}}_{=:Q_t} \underbrace{\begin{pmatrix} E_{t_1} \\ \vdots \\ E_{t_\tau} \end{pmatrix}}_{=:\widehat{V}_t}. \tag{3.4}$$

Since Q_t is prescribed by the sons of t, we only have to compute $\widehat{V}_t$, i.e., the transfer matrices. The approximation error is given by

$$M_t - V_t V_t^\top M_t = M_t - Q_t \widehat{V}_t \widehat{V}_t^\top Q_t^\top M_t,$$

so multiplying with $Q_t^\top$ from the left and exploiting the fact that Q_t is orthogonal yields

$$Q_t^\top (M_t - V_t V_t^\top M_t) = Q_t^\top M_t - \widehat{V}_t \widehat{V}_t^\top Q_t^\top M_t = \bar{M}_t - \widehat{V}_t \widehat{V}_t^\top \bar{M}_t$$

with $\bar{M}_t := Q_t^\top M_t$. This problem is similar to the one encountered before, and we can use the singular value decomposition of $\bar{M}_t$ to construct an orthogonal matrix $\widehat{V}_t$ satisfying

$$\|\bar{M}_t - \widehat{V}_t \widehat{V}_t^\top \bar{M}_t\|_2 \leq 1.$$

The transfer matrices $E_{t_1}, \dots, E_{t_\tau}$ can be recovered by splitting $\widehat{V}_t$ as in (3.4).

Now all we need is an efficient method for computing $\bar{M}_t$. We introduce

$$\widehat{M}_{t,s} := V_t^\top M \chi_s$$

for all $t \in \mathcal{T}_\mathcal{I}$ and all $s \in \operatorname{row}^*(t)$. For $t \in \mathcal{T}_\mathcal{I}$ with $\operatorname{sons}(t) \neq \emptyset$, we let

$$\bar{M}_{t,s} := \begin{pmatrix} \widehat{M}_{t_1,s} \\ \vdots \\ \widehat{M}_{t_\tau,s} \end{pmatrix} \tag{3.5}$$

and observe

$$\bar{M}_t = \sum_{s \in \operatorname{row}^*(t)} \epsilon_{t,s}^{-1} \bar{M}_{t,s}.$$

The construction of the matrices $\widehat{M}_{t,s}$ can be accomplished by a recursion: if $\operatorname{sons}(t) \neq \emptyset$, the nested structure of the cluster basis implies

$$\widehat{M}_{t,s} = V_t^\top M \chi_s = \sum_{t' \in \operatorname{sons}(t)} E_{t'}^\top V_{t'}^\top M \chi_s = \sum_{t' \in \operatorname{sons}(t)} E_{t'}^\top \widehat{M}_{t',s} = \widehat{V}_t^\top \bar{M}_{t,s}. \tag{3.6}$$

Using this recursion to construct the matrices $\widehat{M}_{t,s}$ efficiently, we can now assemble the complete algorithm:

procedure BuildBasis(t);
if sons(t) $= \emptyset$ **then**
  $M_t := 0$;
  for $s \in \mathrm{row}^*(t)$ **do** $M_t := M_t + \epsilon_{t,s}^{-1} \chi_t M \chi_s$;
  Use the singular value decomposition of M_t to define k_t and construct V_t;
  for $s \in \mathrm{row}^*(t)$ **do** $\widehat{M}_{t,s} := V_t^\top M \chi_s$
else
  $\bar{M}_t := 0$;
  for $s \in \mathrm{row}^*(t)$ **do begin**
    Construct $\bar{M}_{t,s}$ as in (3.5);
    $\bar{M}_t := \bar{M}_t + \epsilon_{t,s}^{-1} \bar{M}_{t,s}$
  end;
  Use the singular value decomposition of $\bar{M}_t$ to define k_t and construct $\widehat{V}_t$;
  for $s \in \mathrm{row}^*(t)$ **do** $\widehat{M}_{t,s} := \widehat{V}_t^\top \bar{M}_{t,s}$
endif

THEOREM 3.1 (Complexity). *We assume that P is C_{sp}-sparse and that the rank distribution $k = (k_t)_{t \in \mathcal{T}_\mathcal{I}}$ is $(\alpha, \beta, \gamma, \xi)$-polynomial. Then the algorithm requires $\mathcal{O}(c_\mathcal{I}(\alpha+\beta)^{3\gamma} + c_\mathcal{I}(\#\mathcal{J})(\alpha+\beta)^{2\gamma})$ operations.*

Proof. Let $\hat{k} = (\hat{k}_t)_{t \in \mathcal{T}_\mathcal{I}}$ be defined as in (2.8), and let $(\mathcal{C}_\ell)_{\ell=0}^\infty$ be defined as in (2.10).

Let $t \in \mathcal{T}_\mathcal{I}$. If sons($t$) $= \emptyset$, we compute the matrix M_t, and this matrix has $\#\hat{t} \leq \hat{k}_t$ rows and $\#\mathcal{J}$ columns. If sons(t) $\neq \emptyset$, we compute the matrix $\bar{M}_t$, and this matrix has

$$m_t := \sum_{t' \in \mathrm{sons}(t)} k_t \leq \hat{k}_t$$

rows and $\#\mathcal{J}$ columns.

Since $\{\hat{s} \ : \ s \in \mathrm{row}^*(t)\}$ is a disjoint partition of $\mathcal{J}$, the construction of the matrices M_t and $\bar{M}_t$, respectively, can be accomplished in $\mathcal{O}(\hat{k}_t(\#\mathcal{J}))$ operations, and all matrices $\widehat{M}_{t,s}$ can be computed in $\mathcal{O}(k_t \hat{k}_t(\#\mathcal{J})) \subseteq \mathcal{O}(\hat{k}_t^2(\#\mathcal{J}))$ operations.

Which leaves us to consider the computation of the singular value decomposition. Fortunately, we require it only up to machine accuracy, therefore $\mathcal{O}(\hat{k}_t^2(\#\mathcal{J}) + \hat{k}_t^3)$ operations suffice to find k_t and construct V_t and $\widehat{V}_t$, respectively.

This means that we can find a constant $C_{\mathrm{ad}} \in \mathbb{N}$ such that the number of operations required for a cluster t is bounded by $C_{\mathrm{ad}} \hat{k}_t^2 (\hat{k}_t + (\#\mathcal{J}))$. Therefore the number of operations for all clusters is bounded by

$$\begin{aligned}
\sum_{t \in \mathcal{T}_\mathcal{I}} C_{\mathrm{ad}} \hat{k}_t^2 (\hat{k}_t + (\#\mathcal{J})) &\leq C_{\mathrm{ad}} \sum_{\ell=0}^\infty \sum_{t \in \mathcal{C}_\ell} (\alpha + \beta\ell)^{3\gamma} + (\alpha+\beta\ell)^{2\gamma} (\#\mathcal{J}) \\
&\leq C_{\mathrm{ad}} \xi c_\mathcal{I} \left(\sum_{\ell=0}^\infty (\alpha+\beta\ell)^{3\gamma} \xi^{-\ell} + (\#\mathcal{J}) \sum_{\ell=0}^\infty (\alpha+\beta\ell)^{2\gamma} \xi^{-\ell} \right) \\
&\leq C_{\mathrm{ad}} C_{\mathrm{sum}} \xi c_\mathcal{I} ((\alpha+\beta)^{3\gamma} + (\alpha+\beta)^{2\gamma} (\#\mathcal{J}))
\end{aligned}$$

for a suitable constant C_{sum} depending only on γ and ξ. ∎

In the constant-order case, i.e., for $\beta = 0$ and $c_\mathcal{I} \lesssim n/\alpha^\gamma$, the algorithm has a complexity of $\mathcal{O}(n\alpha^2 + n^2\alpha)$. In the variable-order case, the complexity is in $\mathcal{O}(n^2)$. This is the optimal order of complexity, since n^2 matrix entries have to be processed.

If the matrix M is given in a data-sparse format, we can reduce the complexity of the algorithm. If M, e.g., is a hierarchical matrix, we can reach almost linear complexity [5]. If M is an $\mathcal{H}^2$-matrix, even linear complexity is possible [1].

4. Error analysis. The main difference between the variable-rank approximation algorithm and its predecessor introduced in [5] is the weighting strategy for the block matrices: in the original algorithm, the parameters $\epsilon_{t,s}$ appearing in the construction of M_t are all set to the same value, which leads to reasonable performance, but will not detect a variable-rank structure if it is present.

We will therefore base our choice of $\epsilon_{t,s}$ on a careful analysis of the error propagation in our algorithm.

4.1. Nested error estimate. Before we can investigate the global matrix approximation error, we first have to find a bound for the approximation error of single blocks. The major step is to control the error introduced by projecting a matrix block into a cluster basis.

THEOREM 4.1 (Nested error). *Let $V = (V_t)_{t\in\mathcal{T}_\mathcal{I}}$ be the nested orthogonal cluster basis constructed by the algorithm from Section 3. Then we have*

$$\|\chi_t M \chi_s - V_t V_t^\top M \chi_s\|_2 \leq \sum_{t^*\in\operatorname{sons}^*(t)} \epsilon_{t^*,s} \tag{4.1}$$

for all $t \in \mathcal{T}_\mathcal{I}$ and all $s \in \operatorname{row}^*(t)$.

Proof. We use the notation of (3.4) and extend the notation by letting $\widehat{V}_t := V_t$, $Q_t := I$ and $\bar{M}_{t,s} := \chi_t M \chi_s$ for all leaf clusters $t \in \mathcal{L}_\mathcal{I}$ and all $s \in \operatorname{row}^*(t)$.

We will first prove

$$\chi_t M \chi_s - V_t V_t^\top M \chi_s = \sum_{t^*\in\operatorname{sons}^*(t)} Q_{t^*}(I - \widehat{V}_{t^*}\widehat{V}_{t^*}^\top)\bar{M}_{t^*,s}. \tag{4.2}$$

for all $t \in \mathcal{T}_\mathcal{I}$ by induction over $\#\operatorname{sons}^*(t)$, the number of descendants of t.

If $\#\operatorname{sons}^*(t) = 1$, we have $\operatorname{sons}^*(t) = \{t\}$, which implies $\operatorname{sons}(t) = \emptyset$ by definition, i.e., $\bar{M}_{t,s} = \chi_t M \chi_s$ and $\widehat{V}_t = V_t$. In this case, equation (4.2) is trivial.

Let $n \in \mathbb{N}$, and assume that (4.1) holds for all $t \in \mathcal{T}_\mathcal{I}$ with $\#\operatorname{sons}^*(t) \leq n$. Let $t \in \mathcal{T}_\mathcal{I}$ with $\#\operatorname{sons}^*(t) = n+1$. This implies $\#\operatorname{sons}^*(t) > 1$, i.e., $\operatorname{sons}(t) \neq \emptyset$. We can split the error into a part corresponding to the approximation in the sons of t and a part corresponding to the approximation in t:

$$\begin{aligned}
\chi_t M \chi_s - V_t V_t^\top M \chi_s &= \chi_t M \chi_s - Q_t \widehat{V}_t \widehat{V}_t^\top Q_t^\top M \chi_s \\
&= \chi_t M \chi_s - Q_t Q_t^\top M \chi_s + Q_t Q_t^\top M \chi_s - Q_t \widehat{V}_t \widehat{V}_t^\top Q_t^\top M \chi_s \\
&= \sum_{t'\in\operatorname{sons}(t)} (\chi_{t'} M \chi_s - V_{t'} V_{t'}^\top M \chi_s) + Q_t(I - \widehat{V}_t\widehat{V}_t^\top) Q_t^\top M \chi_s \\
&= \sum_{t'\in\operatorname{sons}(t)} (\chi_{t'} M \chi_s - V_{t'} V_{t'}^\top M \chi_s) + Q_t(I - \widehat{V}_t\widehat{V}_t^\top) \bar{M}_t.
\end{aligned}$$

For all $t' \in \operatorname{sons}(t)$, we have $\#\operatorname{sons}^*(t') < \#\operatorname{sons}^*(t)$, i.e., $\#\operatorname{sons}^*(t) \leq n$, so we can apply the induction assumption in order to conclude that (4.2) holds for t.

Applying the triangle inequality to (4.2) yields

$$\|\chi_t M \chi_s - V_t V_t^\top M \chi_s\|_2 \leq \sum_{t^*\in\operatorname{sons}^*(t)} \|Q_{t^*}(I - \widehat{V}_{t^*}\widehat{V}_{t^*}^\top)\bar{M}_{t^*,s}\|_2,$$

and we conclude the proof using the orthogonality of the matrices Q_{t^*}. ∎

The result of Theorem 4.1 can be improved: by the arguments used in step 3 of the proof of [1, Theorem 4], we can show that the ranges of all terms in the sum (4.2) are pairwise orthogonal, therefore we could use Pythagoras' identity instead of the triangle inquality to bound the error.

4.2. Blockwise error estimate. For all $b = (t, s) \in P_{\mathrm{far}}$, we use the optimal coefficient matrix $S_b := V_t^\top M W_s$ and find

$$\begin{aligned}
\|\chi_t M \chi_s - V_t S_b W_s^\top\|_2 &= \|\chi_t M \chi_s - V_t V_t^\top M W_s W_s^\top\|_2 \\
&= \|\chi_t M \chi_s - V_t V_t^\top M \chi_s + V_t V_t^\top M \chi_s - V_t V_t^\top M W_s W_s^\top\|_2 \\
&\leq \|\chi_t M \chi_s - V_t V_t^\top M \chi_s\|_2 + \|V_t V_t^\top M \chi_s - V_t V_t^\top M W_s W_s^\top\|_2 \\
&\leq \|\chi_t M \chi_s - V_t V_t^\top M \chi_s\|_2 + \|\chi_t M \chi_s - \chi_t M W_s W_s^\top\|_2 \\
&= \|\chi_t M \chi_s - V_t V_t^\top M \chi_s\|_2 + \|\chi_s M^\top \chi_t - W_s W_s^\top M^\top \chi_t\|_2. \qquad (4.3)
\end{aligned}$$

Until now, we have only investigated the influence of the row cluster basis $V = (V_t)_{t \in \mathcal{T}_\mathcal{I}}$, i.e., the first term in this estimate.

We can handle the column cluster basis $W = (W_s)_{s \in \mathcal{T}_\mathcal{J}}$ by applying the algorithm from Section 3 to the transposed matrix $M^\top$, the corresponding transposed block partition given by

$$P^\top := \{(s, t) \ : \ (t, s) \in P\}, \qquad P_{\mathrm{far}}^\top = \{(s, t) \ : \ (t, s) \in P_{\mathrm{far}}\},$$

and a family $(\epsilon_{s,t}^\top)_{s \in \mathcal{T}_\mathcal{J}, t \in \mathcal{T}_\mathcal{I}}$ of error tolerances in $\mathbb{R}_{>0}$.

Combining (4.3) with Theorem 4.1 yields

$$\|\chi_t M \chi_s - V_t S_b W_s^\top\|_2 \leq \sum_{t^* \in \mathrm{sons}^*(t)} \epsilon_{t^*,s} + \sum_{s^* \in \mathrm{sons}^*(s)} \epsilon_{s^*,t}^\top. \qquad (4.4)$$

4.3. Global error estimate. The best approximation of the matrix M in the $\mathcal{H}^2$-matrix format described by the block partition P, the row cluster basis $V = (V_t)_{t \in \mathcal{T}_\mathcal{I}}$ and the column cluster basis $W = (W_s)_{s \in \mathcal{T}_\mathcal{J}}$ is given by

$$\widetilde{M} = \sum_{b=(t,s) \in P_{\mathrm{far}}} V_t S_b W_s^\top + \sum_{b=(t,s) \in P_{\mathrm{near}}} \chi_t M \chi_s$$

with $S_b = V_t^\top M W_s$ for all $b = (t, s) \in P_{\mathrm{far}}$.

In order to bound the spectral norm of the approximation error, we use the following generalization of a result given in [11]:

THEOREM 4.2 (Global spectral error). *Let $V = (V_t)_{t \in \mathcal{T}_\mathcal{I}}$ and $W = (W_s)_{s \in \mathcal{T}_\mathcal{J}}$ be cluster bases. Let P be a C_{sp}-sparse block partition. Let $(\epsilon_{\mathcal{I},t})_{t \in \mathcal{T}_\mathcal{I}}$ and $(\epsilon_{\mathcal{J},s})_{s \in \mathcal{T}_\mathcal{J}}$ be families in $\mathbb{R}_{\geq 0}$ satisfying*

$$\|\chi_t M \chi_s - V_t S_b W_s^\top\|_2 \leq \epsilon_{\mathcal{I},t}^{1/2} \epsilon_{\mathcal{J},s}^{1/2} \qquad \text{for all } b = (t, s) \in P_{\mathrm{far}}. \qquad (4.5)$$

Then we have

$$\|M - \widetilde{M}\|_2 \leq C_{\mathrm{sp}} \sum_{\ell=0}^{\infty} \max\left\{\epsilon_{\mathcal{I},t}, \epsilon_{\mathcal{J},s} \ : \ t \in \mathcal{T}_\mathcal{I}^{(\ell)}, s \in \mathcal{T}_\mathcal{J}^{(\ell)}\right\}. \qquad (4.6)$$

Proof. Let $E := M - \widetilde{M} \in \mathbb{R}^{\mathcal{I}\times\mathcal{J}}$, let $u \in \mathbb{R}^{\mathcal{J}}$, $v := Eu \in \mathbb{R}^{\mathcal{I}}$, and introduce

$$\epsilon_\ell := \max\left\{\epsilon_{\mathcal{I},t}, \epsilon_{\mathcal{J},s} \ : \ t \in \mathcal{T}_{\mathcal{I}}^{(\ell)}, s \in \mathcal{T}_{\mathcal{J}}^{(\ell)}\right\} \qquad \text{for all } \ell \in \mathbb{N}_0.$$

By definition of P and $\widetilde{M}$, we have

$$E = \sum_{b=(t,s)\in P_{\mathrm{far}}} \chi_t M \chi_s - V_t S_b W_s, \qquad \|\chi_t E \chi_s\|_2 \leq \epsilon_{\mathcal{I},t}^{1/2} \epsilon_{\mathcal{J},s}^{1/2},$$

and find

$$\begin{aligned}
\|Eu\|_2^2 = \langle Eu, Eu\rangle_2 = \langle Eu, v\rangle_2 &= \sum_{b=(t,s)\in P_{\mathrm{far}}} \langle \chi_t E \chi_s u, v\rangle_2 \\
&\leq \sum_{b=(t,s)\in P_{\mathrm{far}}} \|\chi_t E \chi_s u\|_2 \|\chi_t v\|_2 \leq \sum_{b=(t,s)\in P_{\mathrm{far}}} \|\chi_t E \chi_s\|_2 \|\chi_s u\|_2 \|\chi_t v\|_2 \\
&\leq \sum_{b=(t,s)\in P_{\mathrm{far}}} \epsilon_{\mathcal{I},t}^{1/2} \epsilon_{\mathcal{J},s}^{1/2} \|\chi_s u\|_2 \|\chi_t v\|_2 \\
&\leq \left(\sum_{b=(t,s)\in P_{\mathrm{far}}} \epsilon_{\mathcal{J},s} \|\chi_s u\|_2^2\right)^{1/2} \left(\sum_{b=(t,s)\in P_{\mathrm{far}}} \epsilon_{\mathcal{I},t} \|\chi_t v\|_2^2\right)^{1/2}.
\end{aligned}$$

Using the sparsity assumption, the first sum can be bounded by

$$\begin{aligned}
\sum_{b=(t,s)\in P_{\mathrm{far}}} \epsilon_{\mathcal{J},s} \|\chi_s u\|_2^2 &\leq C_{\mathrm{sp}} \sum_{s\in\mathcal{T}_{\mathcal{J}}} \epsilon_{\mathcal{J},s} \|\chi_s u\|_2^2 \\
&\leq C_{\mathrm{sp}} \sum_{\ell=0}^{\infty} \epsilon_\ell \sum_{s\in\mathcal{T}_{\mathcal{J}}^{(\ell)}} \|\chi_s u\|_2^2 \leq C_{\mathrm{sp}} \sum_{\ell=0}^{\infty} \epsilon_\ell \|u\|_2^2.
\end{aligned}$$

Applying a similar argument to the second sum yields

$$\|Eu\|_2^2 \leq C_{\mathrm{sp}} \left(\sum_{\ell=0}^{\infty} \epsilon_\ell\right) \|u\|_2 \|v\|_2 = C_{\mathrm{sp}} \left(\sum_{\ell=0}^{\infty} \epsilon_\ell\right) \|u\|_2 \|Eu\|_2,$$

and this implies our claim. □

4.4. Error control. Let $\hat{\epsilon} \in \mathbb{R}_{>0}$ be given. We want the global error $\|M - \widetilde{M}\|_2$ in the spectral norm to be bounded by $\hat{\epsilon}$.

In order to keep the presentation simple, we will focus only on the simple case in which the error tolerances are connected to the level of clusters and blocks, but not to individual clusters.

We assume that P is level-consistent, i.e., that

$$\mathrm{level}(t) = \mathrm{level}(s) \qquad \text{holds for all } b = (t,s) \in P_{\mathrm{far}},$$

and we introduce

$$P_{\mathrm{far}}^{(\ell)} := \{b = (t,s) \in P_{\mathrm{far}} \ : \ \mathrm{level}(t) = \mathrm{level}(s) = \ell\}$$

for all $\ell \in \mathbb{N}_0$. If we let

$$\begin{aligned}
p &:= \max\{\ell \in \mathbb{N}_0 \ : \ P_{\mathrm{far}}^{(\ell)} \neq \emptyset\} \quad \text{and} \\
\epsilon_\ell &:= \max\{\|\chi_t M \chi_s - V_t S_b W_s^\top\|_2 \ : \ b = (t,s) \in P_{\mathrm{far}}^{(\ell)}\},
\end{aligned}$$

Theorem 4.2 takes the form

$$\|M - \widetilde{M}\|_2 \leq C_{\mathrm{sp}} \sum_{\ell=0}^{p} \epsilon_\ell.$$

By picking a parameter $\zeta_1 \in \mathbb{R}_{>1}$ and requiring

$$\epsilon_\ell \leq \hat{\epsilon}_\ell := C_1 \zeta_1^{\ell-p} \qquad \text{with } C_1 := \frac{\zeta_1 - 1}{C_{\mathrm{sp}} \zeta_1} \hat{\epsilon}, \tag{4.7}$$

we get

$$\begin{aligned}\|M - \widetilde{M}\|_2 &\leq C_{\mathrm{sp}} \sum_{\ell=0}^{p} \epsilon_\ell \leq C_{\mathrm{sp}} \frac{\zeta_1 - 1}{C_{\mathrm{sp}} \zeta_1} \sum_{\ell=0}^{p} \zeta_1^{\ell-p} \hat{\epsilon} = C_{\mathrm{sp}} \frac{\zeta_1 - 1}{C_{\mathrm{sp}} \zeta_1} \sum_{\ell=0}^{p} \zeta_1^{-\ell} \hat{\epsilon} \\ &\leq C_{\mathrm{sp}} \frac{\zeta_1 - 1}{C_{\mathrm{sp}} \zeta_1} \sum_{\ell=0}^{\infty} \zeta_1^{-\ell} \hat{\epsilon} = \frac{\zeta_1 - 1}{\zeta_1} \frac{\zeta_1}{\zeta_1 - 1} \hat{\epsilon} = \hat{\epsilon},\end{aligned}$$

so we "only" have to ensure that (4.7) holds, i.e.,

$$\|\chi_t M \chi_s - V_t S_b W_s^\top\|_2 \leq \hat{\epsilon}_\ell \qquad \text{for all } b = (t,s) \in P_{\mathrm{far}}^{(\ell)} \text{ and } \ell \in \mathbb{N}_0.$$

Due to estimate (4.4), we have

$$\|\chi_t M \chi_s - V_t S_b W_s^\top\|_2 \leq \sum_{t^* \in \mathrm{sons}^*(t)} \epsilon_{t^*,s} + \sum_{s^* \in \mathrm{sons}^*(s)} \epsilon_{s^*,t}^\top.$$

If we let

$$\begin{aligned}\epsilon_{\ell^*,\ell} := \max\{ \max\{ \epsilon_{t^*,s} \ &: \ b = (t,s) \in P_{\mathrm{far}}^{(\ell)}, t^* \in \mathrm{sons}^*(t), \mathrm{level}(t^*) = \ell^* \}, \\ \max\{ \epsilon_{s^*,t}^\top \ &: \ b = (t,s) \in P_{\mathrm{far}}^{(\ell)}, s^* \in \mathrm{sons}^*(s), \mathrm{level}(s^*) = \ell^* \} \},\end{aligned}$$

this bound takes the form

$$\|\chi_t M \chi_s - V_t V_t^\top M \chi_s\|_2 \leq \sum_{\ell^*=0}^{\infty} \epsilon_{\ell^*,\ell} \left(\# \left(\mathrm{sons}^*(t) \cap \mathcal{T}_{\mathcal{I}}^{(\ell^*)} \right) + \# \left(\mathrm{sons}^*(s) \cap \mathcal{T}_{\mathcal{J}}^{(\ell^*)} \right) \right).$$

We fix $C_{\mathrm{son}} \in \mathbb{N}$ satisfying

$$\# \mathrm{sons}(t) \leq C_{\mathrm{son}}, \qquad \# \mathrm{sons}(s) \leq C_{\mathrm{son}} \qquad \text{for all } t \in \mathcal{T}_{\mathcal{I}}, s \in \mathcal{T}_{\mathcal{J}}.$$

A simple induction proves

$$\# \left(\mathrm{sons}^*(t) \cap \mathcal{T}_{\mathcal{I}}^{(\ell^*)} \right) \leq \begin{cases} C_{\mathrm{son}}^{\ell^*-\ell} & \text{if } \ell^* \geq \ell, \\ 0 & \text{otherwise} \end{cases} \qquad \text{for all } \ell, \ell^* \in \mathbb{N}_0 \text{ and all } t \in \mathcal{T}_{\mathcal{I}}^{(\ell)},$$

$$\# \left(\mathrm{sons}^*(s) \cap \mathcal{T}_{\mathcal{J}}^{(\ell^*)} \right) \leq \begin{cases} C_{\mathrm{son}}^{\ell^*-\ell} & \text{if } \ell^* \geq \ell, \\ 0 & \text{otherwise} \end{cases} \qquad \text{for all } \ell, \ell^* \in \mathbb{N}_0 \text{ and all } s \in \mathcal{T}_{\mathcal{J}}^{(\ell)},$$

and we get

$$\|\chi_t M \chi_s - V_t V_t^\top M \chi_s\|_2 \leq 2 \sum_{\ell^*=\ell}^{\infty} \epsilon_{\ell^*,\ell} C_{\mathrm{son}}^{\ell^*-\ell}.$$

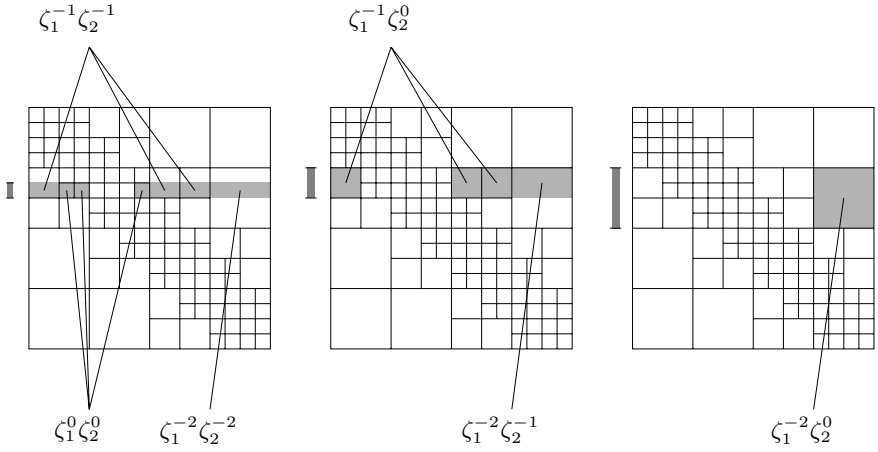

FIG. 4.1. *Choice of weights for* $p = 5$

We pick a second parameter $\zeta_2 \in \mathbb{R}_{>1}$ satisfying $\zeta_2 > C_{\mathrm{son}}$ and require

$$\epsilon_{\ell^*,\ell} \leq \hat{\epsilon}_{\ell^*,\ell} := C_2 \zeta_2^{\ell-\ell^*} \qquad \text{with } C_2 := \frac{\zeta_2 - C_{\mathrm{son}}}{2\zeta_2}\hat{\epsilon}_\ell, \tag{4.8}$$

which yields

$$\begin{aligned}\|\chi_t M \chi_s - V_t S_b W_s^\top\|_2 &\leq 2 \sum_{\ell^*=\ell}^{\infty} \epsilon_{\ell^*,\ell} C_{\mathrm{son}}^{\ell^*-\ell} \leq \frac{\zeta_2 - C_{\mathrm{son}}}{\zeta_2} \sum_{\ell^*=\ell}^{\infty} \left(\frac{C_{\mathrm{son}}}{\zeta_2}\right)^{\ell^*-\ell} \hat{\epsilon}_\ell \\ &\leq \frac{\zeta_2 - C_{\mathrm{son}}}{\zeta_2} \frac{1}{1 - C_{\mathrm{son}}/\zeta_2} \hat{\epsilon}_\ell = \hat{\epsilon}_\ell.\end{aligned}$$

By combining (4.8) with (4.7), we can prove that choosing

$$\begin{aligned}\epsilon_{t^*,s} &\leq \frac{(\zeta_1 - 1)(\zeta_2 - C_{\mathrm{son}})\hat{\epsilon}}{2C_{\mathrm{sp}}\zeta_1\zeta_2} \zeta_1^{\mathrm{level}(s)-p} \zeta_2^{\mathrm{level}(s)-\mathrm{level}(t^*)} && \text{for all } t^* \in \mathcal{T}_\mathcal{I}, s \in \mathcal{T}_\mathcal{J}, \\ \epsilon_{s^*,t}^\top &\leq \frac{(\zeta_1 - 1)(\zeta_2 - C_{\mathrm{son}})\hat{\epsilon}}{2C_{\mathrm{sp}}\zeta_1\zeta_2} \zeta_1^{\mathrm{level}(t)-p} \zeta_2^{\mathrm{level}(t)-\mathrm{level}(s^*)} && \text{for all } s^* \in \mathcal{T}_\mathcal{J}, t \in \mathcal{T}_\mathcal{I}\end{aligned}$$

(cf. Figure 4.1) with $\zeta_1 > 1$ and $\zeta_2 > C_{\mathrm{son}}$ will guarantee that the global error is bounded by $\hat{\epsilon}$.

4.5. Model problem. Let us now investigate how we can fulfill the conditions (4.8) and (4.7) using a cluster basis with polynomial rank distribution for a simple model problem.

Let $p \in \mathbb{N}$ and $n := 2^p$. We consider the matrix $G \in \mathbb{R}^{n\times n}$ defined by

$$G_{ij} := \int_{(i-1)/n}^{i/n} \int_{(j-1)/n}^{j/n} \log|x - y| \, dy \, dx \qquad \text{for all } i, j \in \{1, \ldots, n\}. \tag{4.9}$$

We define the cluster tree $\mathcal{T}_\mathcal{I}$ for the index set $\mathcal{I} := \{1, \ldots, n\}$ by successive bisection: if a cluster t corresponds to an index set $\hat{t} = \{a, \ldots, b\}$ with more than one element, it has two sons s_1 and s_2 with $\hat{s}_1 = \{a, \ldots, \lfloor (a+b)/2 \rfloor - 1\}$ and $\hat{s}_2 = \{\lfloor (a+b)/2 \rfloor, \ldots, b\}$.

The support of a cluster t with $\hat{t} = \{a, \ldots, b\}$ is given by $\Omega_t = [(a-1)/n, b/n]$, and two clusters $t, s \in \mathcal{T}_{\mathcal{I}}$ are admissible if

$$\max\{\operatorname{diam}(\Omega_t), \operatorname{diam}(\Omega_s)\} \leq \operatorname{dist}(\Omega_t, \Omega_s)$$

holds.

We will now investigate the condition (3.1). Let $(t^+, s) \in P_{\text{far}}$, and let $t \in \operatorname{sons}^*(t^+)$. We let $x_0 \in \Omega_t$ be the center of Ω_t, and due to

$$\frac{\partial^\nu g}{\partial x^\nu}(x_0, y) = \frac{(-1)^{\nu-1}(\nu-1)!}{(x_0-y)^\nu} \qquad \text{for all } \nu \in \mathbb{N}, \tag{4.10}$$

the Taylor expansion of $g(x,y) := \log|x-y|$ in the x variable is given by

$$\tilde{g}(x,y) := \log|x_0-y| + \sum_{\nu=1}^{k_t-1} \frac{(-1)^{\nu-1}}{\nu} \frac{(x-x_0)^\nu}{(x_0-y)^\nu}.$$

Replacing g by $\tilde{g}$ in (4.9) yields

$$\begin{aligned}\tilde{G}_{ij} := &\underbrace{\int_{(i-1)/n}^{i/n} 1\,dx}_{=:A_{t,i0}} \underbrace{\int_{(j-1)/n}^{j/n} \log|x_0-y|\,dy}_{=:B_{t,s,0j}} \\ &+ \sum_{\nu=1}^{k_t-1} \underbrace{\int_{(i-1)/n}^{i/n} (x-x_0)^\nu\,dx}_{=:A_{t,i\nu}} \underbrace{\int_{(j-1)/n}^{j/n} \frac{(-1)^{\nu-1}}{\nu(x_0-y)^\nu}\,dy}_{=:B_{t,s,\nu j}} = (A_t B_{t,s})_{ij},\end{aligned}$$

i.e., we have found a rank k_t approximation of $G|_{\hat{t}\times\hat{s}}$.

Let us now take a look at the approximation error. Due to (4.10), we have

$$|g(x,y) - \tilde{g}(x,y)| \leq \left(\frac{\operatorname{diam}(\Omega_t)}{2\operatorname{dist}(\Omega_{t^+}, \Omega_s)}\right)^{k_t}$$

for all $x \in \Omega_t$ and all $y \in \Omega_s$. Let $\ell := \operatorname{level}(t^+) = \operatorname{level}(s)$ and $\ell^* := \operatorname{level}(t)$. Since we have constructed the cluster tree by bisection, we find

$$\operatorname{diam}(\Omega_t) = 2^{\ell-\ell^*} \operatorname{diam}(\Omega_{t^+}), \tag{4.11}$$

and the admissibility of (t^+, s) implies

$$\frac{\operatorname{diam}(\Omega_t)}{2\operatorname{dist}(\Omega_{t^+}, \Omega_s)} = 2^{\ell-\ell^*} \frac{\operatorname{diam}(\Omega_{t^+})}{2\operatorname{dist}(\Omega_{t^+}, \Omega_s)} \leq 2^{\ell-\ell^*-1},$$

so we get the error bound

$$|g(x,y) - \tilde{g}(x,y)| \leq (2^{\ell-\ell^*-1})^{k_t} = 2^{-k_t(\ell^*-\ell+1)},$$

which holds uniformly for all $x \in \Omega_t$ and all $y \in \Omega_s$.

For all $u, v \in \mathbb{R}^{\mathcal{I}}$, this means

$$\begin{aligned}
|\langle \chi_t (G - \tilde{G}) \chi_s u, v \rangle| &\leq \sum_{i \in \hat{t}} \sum_{j \in \hat{s}} |u_j| |v_i| \int_{(i-1)/n}^{i/n} \int_{(j-1)/n}^{j/n} |g(x,y) - \tilde{g}(x,y)| \, dy \, dx \\
&\leq \frac{1}{n^2} 2^{-k_t(\ell^* - \ell + 1)} \sum_{i \in \hat{t}} |v_i| \sum_{j \in \hat{s}} |u_j| \\
&\leq \frac{1}{n} \frac{(\#\hat{t})^{1/2} (\#\hat{s})^{1/2}}{n} 2^{-k_t(\ell^* - \ell + 1)} \|\chi_s u\|_2 \|\chi_t v\|_2 \\
&= \frac{1}{n} |\Omega_t|^{1/2} |\Omega_s|^{1/2} 2^{-k_t(\ell^* - \ell + 1)} \|\chi_s u\|_2 \|\chi_t v\|_2. \qquad (4.12)
\end{aligned}$$

By construction, we have $|\Omega_t| = 2^{-\ell^*}$, $|\Omega_s| = 2^{-\ell}$, and conclude

$$\|\chi_t G \chi_s - A_t B_{t,s}\|_2 \leq \frac{1}{n} 2^{-k_t(\ell^* - \ell + 1)} 2^{-\ell^*/2} 2^{-\ell/2}. \qquad (4.13)$$

According to the previous section, we have to satisfy an error estimate of the type

$$\|\chi_t G \chi_s - A_t B_{t,s}\|_2 \leq C \zeta_1^{\ell - p} \zeta_2^{\ell - \ell^*}.$$

In order to do this, we let $k_t := \alpha + \beta(p - \operatorname{level}(t)) = \alpha + \beta(p - \ell^*)$ for suitable parameters $\alpha, \beta \in \mathbb{N}$ and can see that

$$2^{-k_t(\ell^* - \ell + 1)} = 2^{-(\alpha + \beta(p - \ell^*))(\ell^* - \ell + 1)} \leq 2^{-\alpha - \beta(p - \ell^*) - \alpha(\ell^* - \ell)} = 2^{-\alpha} 2^{\beta(\ell^* - p)} 2^{\alpha(\ell - \ell^*)}$$

holds, so choosing $\alpha \geq 1/2 + \log_2 \zeta_2 + \log_2 \zeta_1$ and $\beta \geq 1 + \log_2 \zeta_1$ ensures

$$\begin{aligned}
2^{-k_t(\ell^* - \ell + 1)} &\leq 2^{-\alpha} 2^{\beta(\ell^* - p)} 2^{\alpha(\ell - \ell^*)} \leq 2^{-\alpha} \zeta_1^{\ell^* - p} \zeta_2^{\ell - \ell^*} 2^{\ell^* - p} 2^{(\ell - \ell^*)/2} \zeta_1^{\ell - \ell^*} \\
&= 2^{-\alpha} \zeta_1^{\ell - p} \zeta_2^{\ell - \ell^*} 2^{\ell^* - p} 2^{(\ell - \ell^*)/2}
\end{aligned}$$

For this choice of k_t, the error bound (4.13) yields

$$\|\chi_t M \chi_s - A_t B_{t,s}\|_2 \leq \frac{1}{n} 2^{-\alpha} 2^{-p} \zeta_1^{\ell - p} \zeta_2^{\ell - \ell^*} = \frac{2^{-\alpha}}{n^2} \zeta_1^{\ell - p} \zeta_2^{\ell - \ell^*}. \qquad (4.14)$$

If necessary, we can increase α in order to ensure $2^{-\alpha}/n^2 \leq C$.

In standard error estimates for the constant-rank case, we will only see a factor of $1/n$, which corresponds to the scaling of the basis functions in the L^2-norm, so we have to increase α to keep the matrix error consistent with the discretization error. In our estimate (4.14), we get a factor $1/n^2$, which means that the L^2-error of the matrix approximation will behave like $1/n$, i.e., will be consistent with the discretization error *without* any modification of α or β.

We owe the additional factor $1/n$ to the fact that the clusters shrink rapidly enough when the level is increased: in (4.11), we exploit the fact that the diameters of their supports decrease by a fixed factor, and in (4.13), we make use of the fact that the Lebesgue measure of the supports also decreases. In the one-dimensional model case, both properties coincide.

Our error estimate (4.14) matches the requirements of (3.1) with (4.7) and (4.8), and by adapting C, ζ_1 and ζ_2, we can ensure that

$$\|M_t - A_t B_t\|_2 \leq 1$$

		SLP			DLP		
n	$\hat{\epsilon}$	Mem	M/n	ϵ	Mem	M/n	ϵ
256	1.5_{-5}	119.4	0.47	3.5_{-6}	102.2	0.40	2.0_{-7}
512	3.8_{-6}	247.3	0.48	1.2_{-6}	207.1	0.40	5.0_{-8}
1024	9.5_{-7}	504.1	0.49	2.7_{-7}	413.8	0.40	1.3_{-8}
2048	2.4_{-7}	1007.7	0.49	8.0_{-8}	827.8	0.40	3.3_{-9}
4096	6.0_{-8}	2027.4	0.49	2.2_{-8}	1654.1	0.40	8.2_{-10}
8192	1.5_{-8}	4062.2	0.50	6.2_{-9}	3308.8	0.40	2.1_{-10}
16384	3.7_{-9}	8307.7	0.51	1.2_{-9}	6606.3	0.40	5.2_{-11}
32768	9.3_{-10}	16654.0	0.51	3.0_{-10}	13223.1	0.41	1.3_{-11}

TABLE 5.1
Approximation results for the unit disc

also holds for

$$B_t := \sum_{s \in \mathrm{row}^*(t)} \epsilon_{t,s}^{-1} B_{t,s},$$

i.e., we have found a rank k_t approximation of M_t which satisfies (3.2). If $\mathrm{sons}(t) \neq \emptyset$, the orthogonality of Q_t implies

$$\|\bar{M}_t - Q_t^\top A_t B_t\|_2 = \|Q_t Q_t^\top (M_t - A_t B_t)\|_2 \leq \|M_t - A_t B_t\|_2 \leq 1$$

i.e., we have also found a rank k_t approximation of $\bar{M}_t$. Since our algorithm computes the *optimal* approximations of M_t and $\bar{M}_t$, respectively, it will construct a cluster basis which is at least as good as the one we have found using the Taylor expansion.

5. Numerical experiments. We apply the algorithm of Section 3 to the boundary integral operators

$$\mathcal{V}[u](x) := -\frac{1}{2\pi} \int_\Gamma \log \|x-y\| u(y)\, dy, \quad \mathcal{K}[u](x) := \frac{1}{2\pi} \int_\Gamma \frac{\langle x-y, n(y)\rangle}{\|x-y\|^2} u(y)\, dy,$$

where Γ is the boundary curve of a two-dimensional domain Ω and n is the outward normal vector of Ω on Γ.

$\mathcal{V}$ and $\mathcal{K}$ are the classical single and double layer potential operators. We discretize them by using Galerkin's method with n piecewise constant basis functions on a polygonal approximation of Γ in order to get matrices V_n and K_n.

These matrices are then compressed using our algorithm with a prescribed precision of $\hat{\epsilon} := n^{-2}$, which corresponds to a convergence of $\mathcal{O}(h)$ in the L^2-norm, and parameters $\zeta_1 = \zeta_2 = 3$.

In the first experiment, we investigate the behaviour of our method for the unit disc $\Omega = \{x \in \mathbb{R}^2 \ : \ \|x\|_2 < 1\}$. In this situation, i.e., for a smooth boundary, the optimal complexity of variable-order schemes has been proven [18, 7, 19], so we only have to check whether our algorithm is able to recover this optimal behaviour. The numerical results are listed in Table 5.1. The first columns give the number of degrees of freedom n and the prescribed accuracy $\hat{\epsilon} = n^{-2}$. The columns labeled "Mem" contain the total storage requirements in KB, the columns labeled "M/n" the storage requirements per degree of freedom, and the columns labeled "ϵ" give

		SLP			DLP		
n	$\hat{\epsilon}$	Mem	M/n	ϵ	Mem	M/n	ϵ
256	1.5_{-5}	161.8	0.63	1.3_{-6}	164.3	0.64	4.0_{-6}
512	3.8_{-6}	320.4	0.63	2.3_{-7}	330.2	0.64	2.2_{-6}
1024	9.5_{-7}	651.2	0.64	1.0_{-7}	665.1	0.65	1.5_{-7}
2048	2.4_{-7}	1330.5	0.65	1.5_{-8}	1320.9	0.64	3.3_{-8}
4096	6.0_{-8}	2722.2	0.66	4.1_{-9}	2608.8	0.64	1.7_{-8}
8192	1.5_{-8}	5449.7	0.67	9.5_{-10}	5114.7	0.62	8.4_{-9}
16384	3.7_{-9}	11007.8	0.67	2.3_{-10}	10020.0	0.61	2.7_{-10}
32768	9.3_{-10}	22353.6	0.68	6.0_{-11}	19664.6	0.60	1.4_{-10}

TABLE 5.2
Approximation results for the unit square

the approximation error. We can see that ϵ is always below $\hat{\epsilon}$ and that the storage requirements grow only linearly, so our method works as predicted.

In our second experiment, we apply our algorithm to the unit square $\Omega = [-1, 1]^2$ with a uniform mesh (a graded mesh yields similar results if the cluster tree is constructed in the correct way). In this situation, a standard variable-order approximation scheme will not work due to the reduced smoothness of the integral kernel. The numbers in Table 5.2 show that our adaptive scheme can handle this case. This is due to the fact that our algorithm can increase the rank close to the edges of Ω in order to ensure the desired accuracy without giving up the overall optimal complexity.

In a third experiment, we consider the *recompression* of a variable-order approximation constructed by the explicit approach described in [7]: Approximating the kernel function of the single layer potential by reinterpolation yields an $\mathcal{H}^2$-matrix satisfying $\|G - \widetilde{G}\|_2 \lesssim h^3$. Since the approximation scheme is based on general polynomial interpolation, it cannot take the special properties of the geometry and the kernel function into account, therefore the resulting ranks will not be optimal. In order to improve the efficiency, we apply a suitably modified variant (cf. [2, Chapter 6.6]) of our algorithm to the initial approximation constructed by interpolation. Due to the fact that the original approximation is already available as an $\mathcal{H}^2$-matrix, the recompression can be accomplished in $\mathcal{O}(n)$ operations. Table 5.3 gives the results of this experiment: The columns "n", "$\hat{\epsilon}$", "M/n", and "ϵ" have the same meaning as before, the total storage requirements in the column "Mem" are now given in MBytes instead of KBytes, the new columns "Build" and "MVM" contain the time (in seconds, measured on one UltraSPARC IIIcu processor running at 900 MHz in a SunFire 6800 computer) for the construction of the approximation (including the time-consuming quadrature of singular integrals for the nearfield) and for one matrix-vector multiplication. We can see that time and storage requirements are in $\mathcal{O}(n)$ and that the approximation error behaves like $\mathcal{O}(h^3)$.

The first experiment shows that the algorithm works as expected, the second experiment demonstrates that it even works in situations where the standard analysis does not apply, and the third experiment indicates that the algorithm can be used to improve the efficiency of other compression schemes. Using the approach described in [2, Chapters 6.5 and 6.7], it can be combined with the very flexible and general hierarchical matrix techniques [14] to reduce storage requirements without sacrificing the "black box" nature of some of these methods.

n	$\hat{\epsilon}$	Build	Mem	M/n	MVM	ϵ
512	5.5_{-4}	1.0	1.7	3.5	< 0.01	2.9_{-4}
2048	7.1_{-5}	6.9	7.3	3.6	0.02	1.2_{-4}
8192	8.9_{-6}	43.1	30.7	3.8	0.18	6.0_{-6}
32768	1.1_{-6}	267.9	142.9	4.5	0.84	6.7_{-7}
131072	1.4_{-7}	1574.9	590.5	4.6	3.49	8.2_{-8}
524288	1.8_{-8}	8271.4	2449.4	4.8	15.60	9.8_{-9}
2097152	2.2_{-9}	38640.7	9921.5	4.8	65.74	1.2_{-9}

TABLE 5.3
Approximation results for the unit sphere

REFERENCES

[1] STEFFEN BÖRM, *Approximation of integral operators by $\mathcal{H}^2$-matrices with adaptive bases*, Computing, 74 (2005), pp. 249–271.

[2] ———, *$\mathcal{H}^2$-Matrices — An efficient tool for the treatment of dense matrices*, Habilitationsschrift, Christian-Albrechts-Universität zu Kiel, 2006.

[3] S. BÖRM, L. GRASEDYCK, AND W. HACKBUSCH, *Hierarchical Matrices*. Lecture Note 21 of the Max Planck Institute for Mathematics in the Sciences, 2003.

[4] ———, *Introduction to hierarchical matrices with applications*, Engineering Analysis with Boundary Elements, 27 (2003), pp. 405–422.

[5] S. BÖRM AND W. HACKBUSCH, *Data-sparse approximation by adaptive $\mathcal{H}^2$-matrices*, Computing, 69 (2002), pp. 1–35.

[6] ———, *$\mathcal{H}^2$-matrix approximation of integral operators by interpolation*, Applied Numerical Mathematics, 43 (2002), pp. 129–143.

[7] STEFFEN BÖRM, MAIKE LÖHNDORF, AND JENS MARKUS MELENK, *Approximation of integral operators by variable-order interpolation*, Numerische Mathematik, 99 (2005), pp. 605–643.

[8] WOLFGANG DAHMEN, HELMUT HARBRECHT, AND REINHOLD SCHNEIDER, *Compression techniques for boundary integral equations — Asymptotically optimal complexity estimates*, SIAM J. Num. Anal., 43 (2006), pp. 2251–2271.

[9] W. DAHMEN AND R. SCHNEIDER, *Wavelets on manifolds I: Construction and domain decomposition*, SIAM Journal of Mathematical Analysis, 31 (1999), pp. 184–230.

[10] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, Johns Hopkins University Press, London, 1996.

[11] L. GRASEDYCK, *Theorie und Anwendungen Hierarchischer Matrizen*, doctoral thesis, Universität Kiel, 2001.

[12] L. GRASEDYCK AND W. HACKBUSCH, *Construction and arithmetics of $\mathcal{H}$-matrices*, Computing, 70 (2003), pp. 295–334.

[13] L. GREENGARD AND V. ROKHLIN, *A fast algorithm for particle simulations*, Journal of Computational Physics, 73 (1987), pp. 325–348.

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

[15] WOLFGANG HACKBUSCH, BORIS KHOROMSKIJ, AND STEFAN A. SAUTER, *On $\mathcal{H}^2$-matrices*, in Lectures on Applied Mathematics, H. Bungartz, R. Hoppe, and C. Zenger, eds., Springer-Verlag, Berlin, 2000, pp. 9–29.

[16] WOLFGANG HACKBUSCH AND ZENON PAUL NOWAK, *On the fast matrix multiplication in the boundary element method by panel clustering*, Numerische Mathematik, 54 (1989), pp. 463–491.

[17] V. ROKHLIN, *Rapid solution of integral equations of classical potential theory*, Journal of Computational Physics, 60 (1985), pp. 187–207.

[18] S. A. SAUTER, *Variable order panel clustering*, Computing, 64 (2000), pp. 223–261.

[19] J. TAUSCH, *The variable order fast multipole method for boundary integral equations of the second kind*, Computing, 72 (2004), pp. 267–291.