Max-Planck-Institut
für Mathematik
in den Naturwissenschaften
Leipzig

Data-sparse approximation of non-local operators by $\mathcal{H}^2$-matrices

(revised version: August 2006)

by

Steffen Börm

Preprint no.: 44 2005

Data-sparse approximation of non-local operators by $\mathcal{H}^2$-matrices

Steffen Börm

August 10, 2006

Many of today's most efficient numerical methods are based on multilevel decompositions: The multigrid algorithm is based on a hierarchy of grids, wavelet techniques use a hierarchy of basis functions, while fast panel-clustering and multipole methods employ a hierarchy of clusters.

The high efficiency of these methods is due to the fact that the hierarchies are nested, i.e., that the information present on a coarser level is also present on finer levels, thus allowing efficient recursive algorithms.

$\mathcal{H}^2$-matrices employ nested local expansion systems in order to approximate matrices in optimal (or for some problem classes at least optimal up to logarithmic factors) order of complexity. This paper presents a criterion for the approximability of general matrices in the $\mathcal{H}^2$-matrix format and an algorithm for finding good nested expansion systems and constructing the approximation efficiently.

1 Introduction

In order to handle large systems of linear equations efficiently, the corresponding matrices have to be represented or at least approximated in a data-sparse form. In some situations, this is relatively simple: if most of the matrix entries are zero, it is sufficient to store the remaining non-zero entries.

For more general matrices, the task of finding a data-sparse representation can be far more complicated. It can even be impossible to solve, e.g., if the matrix coefficients are entirely random. We can only hope to find efficient representations if we can exploit special properties of the matrix. There are two important examples of non-local matrices allowing data-sparse approximations:

If the system results from the discretization of an elliptic partial differential equation, the matrix describing the system is sparse and corresponds in a suitable sense to a continuous problem, which makes it possible to employ multigrid and related algorithms in order to evaluate the inverse of the sparse matrix in optimal or near-optimal complexity, although it is a non-local operator.

If the system results from the discretization of an integral operator, it is possible to approximate the underlying kernel function by separable functions, i.e., tensor products like polynomials or multipole expansions, in order to construct blockwise low-rank approximations which can be used to speed up the evaluation of the dense matrix.

It can be shown [2] that both techniques are related: the densely populated matrix corresponding to one step of a multilevel iteration exhibits a blockwise low-rank structure which is very similar to that of panel-clustering [20, 22, 23] or multipole [21, 15, 16] methods.

The optimal order of complexity of modern multilevel algorithms relies on the fact that the grids of the underlying hierarchy are (at least approximately) nested, because this implies that the algorithm can be formulated in a recursive way which re-uses results of preceding levels and leads to an exponential decay of the number of operations required per level, so that the total complexity is dominated by the complexity of the operations on the finest level.

Nested structures can also be used to construct approximation schemes of optimal order of complexity for integral operators. This is especially simple for panel-clustering methods based on polynomial expansions of constant order [19, 11, 8], since here the expansion systems are already nested. For multipole techniques [15, 16], the general polynomials are replaced by harmonic functions.

The algebraic counterparts of panel-clustering techniques are hierarchical matrices ($\mathcal{H}$-matrices) [17, 18, 13, 6]: low-rank matrices take the place of separated expansions, the decomposition of the geometric domain is replaced by a decomposition of the index set. The resulting data-sparse representation makes it possible to perform algebraic operations like adding, multiplying or inverting matrices (approximatively) in almost optimal complexity.

$\mathcal{H}^2$-matrices [19, 7, 9, 3] are a specialization of $\mathcal{H}$-matrices which take advantage of nested expansion systems in order to reach the optimal order of complexity: both the number of elementary operations required for arithmetic operations [4] and the number of units of storage required for the representation of the matrix grow linearly in the number of degrees of freedom.

The high efficiency of $\mathcal{H}^2$-matrix techniques depends crucially on the nested structure of the expansion systems. In some situations, e.g., for polynomial expansion of fixed order [8], the construction of nested structures is trivial, but in general even their existence is not obvious.

This paper is organized as follows: Section 2 provides the basic framework of $\mathcal{H}^2$-matrices and cluster bases. Section 3 contains the fundamental approximation result: a general matrix can be approximated by an $\mathcal{H}^2$-matrix if and only if each element of a family of submatrices, called “total cluster basis”, can be approximated by low-rank matrices. Section 4 describes a practical algorithm which can be used to compute an $\mathcal{H}^2$-matrix approximation of a general matrix. The numerical experiments of Section 5 demonstrate that integral operators and solution operators of elliptic partial differential equations can indeed be approximated by $\mathcal{H}^2$-matrices.

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

We will now briefly recall the structure of $\mathcal{H}^2$-matrices [19, 7].

2.1 Block structure

The basic idea of hierarchical matrix techniques is to identify 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*

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

We note that the definition implies that $\hat{t} \subseteq \mathcal{I}$ holds for all clusters in $\mathcal{T}_{\mathcal{I}}$ and that $\mathcal{L}_{\mathcal{I}}$ corresponds to the disjoint partition $\{\hat{t} \ : \ t \in \mathcal{L}_{\mathcal{I}}\}$ of $\mathcal{I}$.

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 $\mathcal{T}_{\mathcal{J}}$. A set $P_{\text{near}} \subseteq P$ is the* nearfield *of P if $\operatorname{sons}(t) = \emptyset = \operatorname{sons}(s)$ holds for all $(t, s) \in P_{\text{near}}$. Each nearfield corresponds to a* farfield *$P_{\text{far}} := P \setminus P_{\text{near}}$. The blocks in P_{far} are called* admissible blocks*, the blocks in P_{near} are called* inadmissible blocks*.*

This definition implies that 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.

In the following, we will treat P_{far} and P_{near} as implicitly given for each block partition P.

Remark 2.4 (Admissibility condition) *In practice, an* admissibility condition *is used to identify those blocks which allow low-rank approximations. For discretized elliptic problems, the condition*

$$\max\{\operatorname{diam}(\Omega_t), \operatorname{diam}(\Omega_s)\} \leq \operatorname{dist}(\Omega_t, \Omega_s) \tag{1}$$

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.

The condition (1) ensures that the blocks t and s are far from the diagonal of the matrix, i.e., in a region where we can expect Green's function to be smooth or at least separable.

Remark 2.5 (Construction) *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.*

Given row and column cluster trees and an admissibility condition like (1), a block partition can be constructed recursively: we start with the block $(t, s) = (\operatorname{root}(\mathcal{T}_\mathcal{I}), \operatorname{root}(\mathcal{T}_\mathcal{J}))$. For each block, we check whether it satisfies the admissibility condition. If it does, the block is added to P_{far}. Otherwise, we investigate the subblocks corresponding to sons of t and of s. If t and s do not have any sons, the block (t, s) is added to P_{near}.

Under moderate assumptions, it is possible to prove that $\mathcal{T}_\mathcal{I}$, $\mathcal{T}_\mathcal{J}$ and $P = P_{\text{far}} \dot{\cup} P_{\text{near}}$ can be constructed efficiently and that the number of blocks in P resulting from this construction grows only linearly in the number of degrees of freedom [14].

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 range and image of these blocks 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

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

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 subblock $\hat{t} \times \hat{s}$ and zero everywhere else.

Definition 2.6 (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.

The definition does not require the matrices V_t to be of full rank, so their columns do not really form a basis, although the name "cluster basis" suggests this (a more fitting name might be "cluster frame"). This is only a practical consideration: in some applications, a system of vectors spanning the desired space can be constructed efficiently, but ensuring their linear independence would lead to unnecessary technical complications.

Definition 2.7 (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*

$$V_{t^+} = \sum_{t \in \operatorname{sons}(t^+)} V_t E_t \tag{2}$$

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.

The nested structure is the key difference between general hierarchical matrices and $\mathcal{H}^2$-matrices [19, 7, 8], since it allows us to construct very efficient algorithms by re-using information across the entire cluster tree.

Definition 2.8 ($\mathcal{H}^2$-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 a matrix $S_b \in \mathbb{R}^{k_t \times l_s}$ for each $b = (t, s) \in P_{\text{far}}$ satisfying*

$$\chi_t M \chi_s = V_t S_b (W_s)^\top, \tag{3}$$

*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

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

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

By replacing the low-rank representation $\chi_t A \chi_s = X_b Y_b^\top$ used in the context of general hierarchical matrices by the specialized representation (3), we can prepare cluster-related quantities before performing complicated operations. Due to the nested structure, the preparation of these quantities can be organized efficiently.

Remark 2.9 (Complexity) *In practice, $\mathcal{H}^2$-matrices are described by the coupling matrices $(S_b)_{b\in P_{\text{far}}}$ for the admissible blocks and the inadmissible parts $\chi_t M \chi_s$ for all $(t,s) \in P_{\text{near}}$.*

For $n := \max\{\#\mathcal{I}, \#\mathcal{J}\}$, we can use the results in [14] as in [7, 8] in order to show that the amount of storage required for coupling and inadmissible matrices is in $\mathcal{O}(n(k_* + l_*))$ if $k_t \le k_*$ holds for all $t \in \mathcal{T}_{\mathcal{I}}$ and $l_s \le l_*$ holds for all $s \in \mathcal{T}_{\mathcal{J}}$.

Due to the nested structure, the cluster basis $V = (V_t)_{t\in\mathcal{T}_{\mathcal{I}}}$ can be expressed by the family $E = (E_t)_{t\in\mathcal{T}_{\mathcal{I}}}$ of transfer matrices and the family $(V_t)_{t\in\mathcal{L}_{\mathcal{I}}}$ of cluster basis matrices corresponding only to leaf clusters. The amount of storage required for this representation is in $\mathcal{O}(nk_*)$.

We can combine these complexity estimates to conclude that the storage requirements of an $\mathcal{H}^2$-matrix are in $\mathcal{O}(n(k_* + l_*))$.

Remark 2.10 (Rank distributions) *The rank k_t depends on the desired precision $\epsilon \in \mathbb{R}_{>0}$ of the matrix approximation, and the precision will usually depend on the underlying discretization error, e.g., $\epsilon \sim n^{-\beta}$ for some $\beta \in \mathbb{R}_{>0}$.*

2.3 Orthogonal cluster bases and best approximations

Since we intend to approximate results of arithmetic operations, we need an efficient way of finding best approximations of arbitrary matrices in a given $\mathcal{H}^2$-matrix format. This problem is especially simple if the columns of the cluster basis matrices V_t are pairwise orthonormal.

Definition 2.11 (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

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

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 [3].

2.4 Fast matrix-vector multiplication

Let M be an $\mathcal{H}^2$-matrix with cluster bases V and W for the cluster trees $\mathcal{T}_{\mathcal{I}}$ and $\mathcal{T}_{\mathcal{J}}$ and the block partition P. Let E and F be the families of transfer matrices for V and W.

The matrix-vector multiplication $y := Mx$ is split into four phases: First, we compute the auxiliary vectors

$$x_s := (W_s)^\top x$$

for all $s \in \mathcal{T}_{\mathcal{J}}$. This step is called the *forward transformation.* Then, we compute the auxiliary vectors

$$y_t := \sum_{b=(t,s)\in P_{\text{far}}} S_b x_s$$

for all $t \in \mathcal{T}_{\mathcal{I}}$. This phase handles the interaction of all admissible blocks. In the third step, the *backward transformation*, we accumulate the part of the result

$$y := \sum_{t\in\mathcal{T}_{\mathcal{I}}} V_t y_t$$

that corresponds to all admissible blocks of the matrix. In order to complete the multiplication, we add the inadmissible parts

$$y := y + \sum_{b=(t,s)\in P_{\text{near}}} \chi_t M \chi_s x.$$

Under standard assumptions, the second and last step can be performed in linear complexity, since they only involve relatively small matrices. In order to treat the forward transformation efficiently, we have to make use of the nested structure: $\text{sons}(s) \neq \emptyset$ implies

$$x_s = W_s^\top x = \sum_{s'\in\text{sons}(s)} F_{s'}^\top W_{s'}^\top x = \sum_{s'\in\text{sons}(s)} F_{s'}^\top x_{s'},$$

so we can compute $x_s = W_s^\top x$ using $2k_s(\sum_{s'\in\text{sons}(s)} k_{s'})$ operations instead of the $2k_s\#\hat{s}$ operations required by a naive approach, and we need to store W_s only for leaves of the cluster tree and can use the transfer matrices F_s for all other clusters.

This leads to a complexity of $\mathcal{O}(nl_*)$ for the forward transformation. Treating the backward transformation in a similar way leads to a total complexity of $\mathcal{O}(n(k_* + l_*))$ for the matrix-vector multiplication.

3 Existence of $\mathcal{H}^2$-matrix approximations

We have seen in the previous section that $\mathcal{H}^2$-matrices can provide us with data-sparse representations for densely populated matrices. Now we will investigate which kinds of matrices can be approximated efficiently by $\mathcal{H}^2$-matrices.

We fix index sets $\mathcal{I}$ and $\mathcal{J}$, corresponding cluster trees $\mathcal{T}_{\mathcal{I}}$ and $\mathcal{T}_{\mathcal{J}}$ and a block partition $P = P_{\text{far}}\dot{\cup}P_{\text{near}}$.

3.1 Left and right semi-uniform matrices

The admissible blocks of an $\mathcal{H}^2$-matrix are expressed in the form $V_t S_b W_s^\top$, i.e., they depend both on the row and column cluster basis. In order to simplify our analysis, we will restrict our attention to formats which depends only on one cluster basis.

Definition 3.1 (Left and right semi-uniform matrices) *Let V and W be nested cluster bases with rank distributions k and l. Let $M \in \mathbb{R}^{\mathcal{I}\times\mathcal{J}}$.*

If we can find a matrix $B_b \in \mathbb{R}^{\mathcal{J}\times k_t}$ with $\chi_s B_b = B_b$ for each $b = (t,s) \in P_{\mathrm{far}}$ satisfying

$$\chi_t M \chi_s = V_t B_b^\top, \tag{4}$$

the matrix M is called a left semi-uniform matrix*.*

If we can find a matrix $A_b \in \mathbb{R}^{\mathcal{I}\times l_s}$ with $\chi_t A_b = A_b$ for each $b = (t,s) \in P_{\mathrm{far}}$ satisfying

$$\chi_t M \chi_s = A_b W_s^\top, \tag{5}$$

the matrix M is called a right semi-uniform matrix*.*

The sets of left and right semi-uniform matrices are denoted by $\mathcal{H}^2(P,V,*)$ and $\mathcal{H}^2(P,*,W)$, respectively, where the symbol "$*$" means that either column or row basis are not required.

We can see that a matrix $M \in \mathbb{R}^{\mathcal{I}\times\mathcal{J}}$ is an $\mathcal{H}^2$-matrix if and only if M is both left and right semi-uniform.

If V and W are orthogonal, the best approximations (in the Frobenius norm) of an arbitrary matrix $M \in \mathbb{R}^{\mathcal{I}\times\mathcal{J}}$ can be expressed by orthogonal projections:

$$\begin{aligned}
\Pi^P_{V,*} &: \mathbb{R}^{\mathcal{I}\times\mathcal{J}} \to \mathcal{H}^2(P,V,*), & M &\mapsto \sum_{(t,s)\in P_{\mathrm{far}}} V_t V_t^\top M \chi_s + \sum_{(t,s)\in P_{\mathrm{near}}} \chi_t M \chi_s,\\
\Pi^P_{*,W} &: \mathbb{R}^{\mathcal{I}\times\mathcal{J}} \to \mathcal{H}^2(P,*,W), & M &\mapsto \sum_{(t,s)\in P_{\mathrm{far}}} \chi_t M W_s W_s^\top + \sum_{(t,s)\in P_{\mathrm{near}}} \chi_t M \chi_s.
\end{aligned}$$

The projection $\Pi^P_{V,*}$ maps an arbitrary matrix to $\mathcal{H}^2(P,V,*)$, the projection $\Pi^P_{*,W}$ maps it to $\mathcal{H}^2(P,*,W)$. Both projections commute, and their product $\Pi^P_{V,W} := \Pi^P_{V,*}\Pi^P_{*,W} = \Pi_{*,W}\Pi_{V,*}$ maps an arbitrary matrix to its best approximation in $\mathcal{H}^2(P,V,W)$.

Lemma 3.2 (Separation of cluster bases) *Let V and W be orthogonal cluster bases.*

$$\|M - \Pi^P_{V,W}M\|_F^2 \le \|M - \Pi^P_{V,*}M\|_F^2 + \|M - \Pi^P_{*,W}M\|_F^2$$

holds for all $M \in \mathbb{R}^{\mathcal{I}\times\mathcal{J}}$.

Proof. Let $M \in \mathbb{R}^{\mathcal{I}\times\mathcal{J}}$. Since $\Pi^P_{V,*}$ is an orthogonal projection, we have

$$\begin{aligned}
\|M - \Pi^P_{V,W}M\|_F^2 &= \|M - \Pi^P_{V,*}\Pi^P_{*,W}M\|_F^2\\
&= \|M - \Pi^P_{V,*}M + \Pi^P_{V,*}(M - \Pi^P_{*,W}M)\|_F^2
\end{aligned}$$

$$\begin{aligned} &= \|M - \Pi^P_{V,*}M\|_F^2 + 2\langle M - \Pi^P_{V,*}M, \Pi^P_{V,*}(M - \Pi^P_{*,W}M)\rangle_F \\ &\quad + \|\Pi^P_{V,*}(M - \Pi^P_{*,W}M)\|_F^2 \\ &\leq \|M - \Pi^P_{V,*}M\|_F^2 + \|M - \Pi^P_{*,W}M\|_F^2. \end{aligned}$$

■

This implies that we do not have to investigate $\mathcal{H}^2$-matrices directly but can construct row and column cluster bases independently.

Remark 3.3 *Let V and W be orthogonal cluster bases, and let $M \in \mathbb{R}^{\mathcal{I}\times\mathcal{J}}$. We find*

$$\begin{aligned} \|M - \Pi^P_{V,*}M\|_F^2 &= \|M\|_F^2 - \|\Pi^P_{V,*}M\|_F^2 \leq \|M\|_F^2 - \|\Pi^P_{*,W}\Pi^P_{V,*}M\|_F^2 \\ &= \|M\|_F^2 - \|\Pi^P_{V,W}M\|_F^2 = \|M - \Pi^P_{V,W}M\|_F^2 \end{aligned}$$

and can prove

$$\|M - \Pi^P_{*,W}M\|_F^2 \leq \|M - \Pi^P_{V,W}M\|_F^2$$

by a similar argument. Therefore restricting our attention to left and right semi-uniform matrices instead of dealing with $\mathcal{H}^2$-matrices directly will not lead to a significant loss of precision in our error estimates.

The projections into the spaces of left and right semi-uniform matrices are closely related:

Lemma 3.4 (Transposed matrices) *Let V and W be orthogonal cluster bases. We define the* transposed *block partition $P^\top$ for $\mathcal{J}$ and $\mathcal{I}$ by*

$$\begin{aligned} (s,t) \in P^\top &\Leftrightarrow (t,s) \in P \quad \text{and} \\ (s,t) \in P^\top_{\text{far}} &\Leftrightarrow (t,s) \in P_{\text{far}} \end{aligned}$$

for all $t \in \mathcal{T}_\mathcal{I}$ and $s \in \mathcal{T}_\mathcal{J}$. We have

$$(\Pi^P_{*,W}M)^\top = \Pi^{P^\top}_{W,*}M^\top$$

for all $M \in \mathbb{R}^{\mathcal{I}\times\mathcal{J}}$.

Proof. For $M \in \mathbb{R}^{\mathcal{I}\times\mathcal{J}}$, we find

$$\begin{aligned} (\Pi^P_{*,W}M)^\top &= \left(\sum_{(t,s)\in P_{\text{far}}} \chi_t M W_s W_s^\top + \sum_{(t,s)\in P_{\text{near}}} \chi_t M \chi_s \right)^\top \\ &= \sum_{(t,s)\in P_{\text{far}}} W_s W_s^\top M^\top \chi_t + \sum_{(t,s)\in P_{\text{near}}} \chi_s M^\top \chi_t \\ &= \sum_{(s,t)\in P_{\text{far}}} W_s W_s^\top M^\top \chi_t + \sum_{(s,t)\in P_{\text{near}}} \chi_s M^\top \chi_t \\ &= \Pi^{P^\top}_{W,*}M^\top. \end{aligned}$$

■

3.2 Total cluster bases

Due to Lemma 3.2, we can investigate row and column cluster bases independently. Lemma 3.4 implies that it is sufficient to consider only row cluster bases, since we can get results on column cluster bases by switching to the transposed matrix.

Our goal is to construct an intermediate row cluster basis, the *total cluster basis*, that can be used to represent a matrix $M \in \mathbb{R}^{\mathcal{I}\times\mathcal{J}}$ as a left semi-uniform matrix, but has, in general, a prohibitively high rank. Practically useful row cluster bases can be constructed by approximating the total cluster basis, and their approximation properties can be judged by how well they approximate it.

We require some preliminary definitions and results:

Definition 3.5 (Descendants and predecessors) *For all $t \in \mathcal{T}_{\mathcal{I}}$, we define the* set of descendants *recursively by*

$$\operatorname{sons}^*(t) := \begin{cases} \{t\} \cup \bigcup_{t' \in \operatorname{sons}(t)} \operatorname{sons}^*(t') & \textit{if } \operatorname{sons}(t) \neq \emptyset, \\ \{t\} & \textit{otherwise.} \end{cases}$$

The set of predecessors *is defined by*

$$\operatorname{pred}(t) = \{t^+ \in \mathcal{T}_{\mathcal{I}} \ : \ t \in \operatorname{sons}^*(t^+)\}.$$

Lemma 3.6 (Block rows) *For all $t \in \mathcal{T}_{\mathcal{I}}$, let*

$$\begin{aligned} \operatorname{row}(t) &:= \{s \in \mathcal{T}_{\mathcal{J}} \ : \ (t,s) \in P_{\mathrm{far}}\}, \\ \operatorname{row}^*(t) &:= \{s \in \mathcal{T}_{\mathcal{J}} \ : \ \textit{there is a } t^+ \in \operatorname{pred}(t) \textit{ with } (t^+,s) \in P_{\mathrm{far}}\}. \end{aligned}$$

For $t \in \mathcal{T}_{\mathcal{I}}$ and $s_1, s_2 \in \operatorname{row}^*(t)$ with $s_1 \neq s_2$, we have $\hat{s}_1 \cap \hat{s}_2 = \emptyset$, i.e., the index sets corresponding to the clusters in $\operatorname{row}^*(t) \supseteq \operatorname{row}(t)$ are pairwise disjoint.

For $t \in \mathcal{T}_{\mathcal{I}}$ and $t^+ \in \operatorname{pred}(t)$, we have $\operatorname{row}^*(t^+) \subseteq \operatorname{row}^*(t)$.

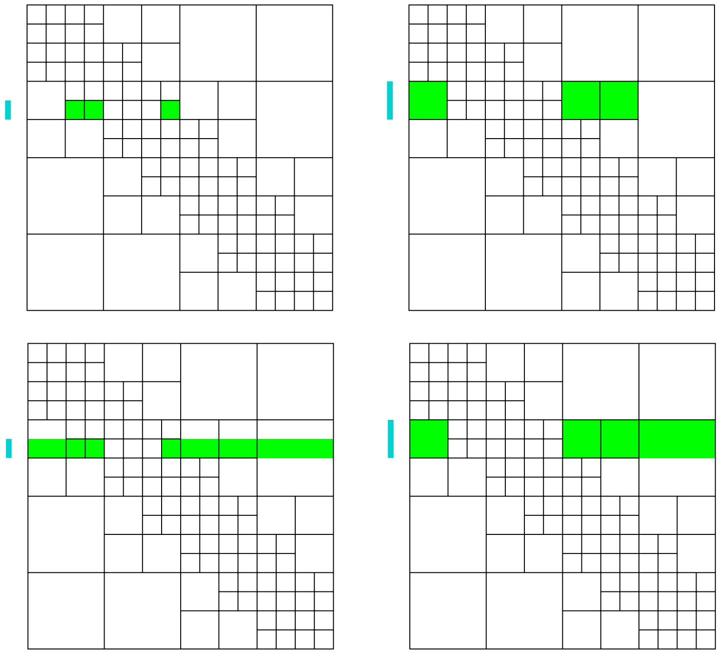
Proof. We will prove the first statement by contraposition. Let $t \in \mathcal{T}_{\mathcal{I}}$ and $s_1, s_2 \in \operatorname{row}^*(t)$ with $\hat{s}_1 \cap \hat{s}_2 \neq \emptyset$. By definition, there are $t_1^+, t_2^+ \in \operatorname{pred}(t)$ such that $s_1 \in \operatorname{row}(t_1^+)$ and $s_2 \in \operatorname{row}(t_2^+)$ hold. Since both t_1^+ and t_2^+ are predecessors of t, we can assume $t_2^+ \in \operatorname{sons}^*(t_1^+)$ without loss of generality, which implies $\hat{t}_2^+ \subseteq \hat{t}_1^+$. Let $j \in \hat{s}_1 \cap \hat{s}_2$ and $i \in \hat{t}_2^+ \subseteq \hat{t}_1^+$. Then we have $(j,i) \in (\hat{t}_1^+ \times \hat{s}_1) \cap (\hat{t}_2^+ \times \hat{s}_2)$, and since (t_1^+, s_1) and (t_2^+, s_2) are both elements of the block partition P, Definition 2.2 implies $(t_1^+, s_1) = (t_2^+, s_2)$, and therefore $s_1 = s_2$.

The second statement is a simple consequence of the fact that $\operatorname{pred}(t) \subseteq \operatorname{pred}(t')$ holds for all $t' \in \operatorname{sons}(t)$. ■

We define the matrices

$$M_t^0 := \sum_{r \in \operatorname{row}(t)} \chi_t M \chi_r \qquad \text{for all } t \in \mathcal{T}_{\mathcal{I}}.$$

Figure 1: Matrices M_t^0 (top) and M_t (bottom) for several clusters t appearing in a simple one-dimensional model problem



According to Lemma 3.6, all clusters s on the right-hand side of this definition correspond to disjoint index sets, and we find

$$M_t^0 \chi_s = \left(\sum_{r \in \mathrm{row}(t)} \chi_t M \chi_r \right) \chi_s = \sum_{r \in \mathrm{row}(t)} \chi_t M \chi_r \chi_s = \chi_t M \chi_s \qquad \text{for all } (t,s) \in P_{\mathrm{far}}, \quad (6)$$

i.e., the family $(M_t^0)_{t \in \mathcal{T}_\mathcal{I}}$ satisfies condition (4) for all $b = (t,s) \in P_{\mathrm{far}}$ with $B_b = \chi_s$. In order to ensure that the family is also nested, i.e., satisfies (2), we simply include all predecessors t^+ of a cluster t in the definition:

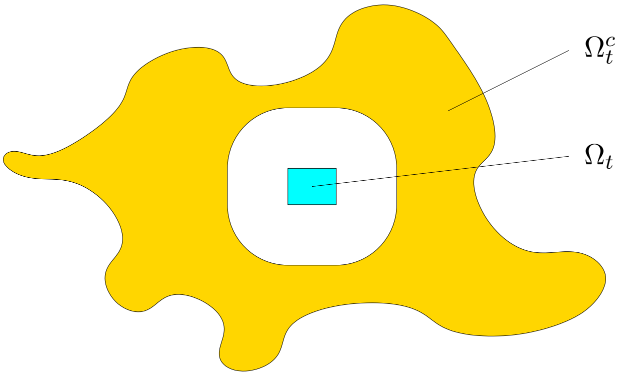
Definition 3.7 (Total cluster basis) *Let $M \in \mathbb{R}^{\mathcal{I} \times \mathcal{J}}$. The family $(M_t)_{t \in \mathcal{T}_\mathcal{I}}$ given by*

$$M_t := \sum_{t^+ \in \mathrm{pred}(t)} \sum_{r \in \mathrm{row}(t^+)} \chi_t M \chi_r = \sum_{r \in \mathrm{row}^*(t)} \chi_t M \chi_r \qquad \textit{for all } t \in \mathcal{T}_\mathcal{I}$$

is called the total cluster basis *corresponding to M.*

Applying Lemma 3.6 to M_t instead of M_t^0, we can prove $M_t \chi_s = \chi_t M \chi_s$ by the same arguments as in equation (6). Neglecting the technical detail of numbering $\mathcal{J}$, we can treat $(M_t)_{t \in \mathcal{T}_\mathcal{I}}$ as a nested cluster basis and conclude that M is a left semi-uniform matrix with respect to its total cluster basis.

Figure 2: Geometric interpretation of total cluster bases in 2D



In case of the simple one-dimensional model problem introduced, e.g., in [6, Chapter 1], Figure 1 illustrates the subblocks contributing to the matrices $(M_t^0)_{t\in\mathcal{T}_\mathcal{I}}$ and $(M_t)_{t\in\mathcal{T}_\mathcal{I}}$.

Remark 3.8 (Two-dimensional example) *Let us now consider a more interesting example: we assume that the clusters $t \in \mathcal{T}_\mathcal{I}$ correspond to a suitable hierarchy of subdomains Ω_t of a domain $\Omega \subseteq \mathbb{R}^2$, and that the admissibility condition (1) is used to determine the block partition P and the farfield P_{far}.*

Let $t \in \mathcal{T}_\mathcal{I}$, $t^+ \in \operatorname{pred}(t)$ and $r \in \operatorname{row}(t^+)$. Due to $\Omega_t \subseteq \Omega_{t^+}$ and (1), we have

$$\operatorname{dist}(\Omega_t, \Omega_r) \geq \operatorname{dist}(\Omega_{t^+}, \Omega_r) \geq \operatorname{diam}(\Omega_{t^+}) \geq \operatorname{diam}(\Omega_t).$$

This means $\Omega_r \subseteq \Omega_t^c$ for

$$\Omega_t^c := \{x \in \Omega \ : \ \operatorname{dist}(\Omega_t, x) \geq \operatorname{diam}(\Omega_t)\},$$

i.e. all clusters in $\operatorname{row}^*(t)$ *correspond to subdomains contained in the* geometric farfield *Ω_t^c. Geometrically speaking, the total cluster basis matrix M_t describes the flow of information from the geometric farfield Ω_t^c (depicted in Figure 2) into Ω_t.*

The total cluster basis is in general not useful for practical applications, since the number of columns of each of the matrices M_t is too large. The following result provides a hint on how to reduce the complexity:

Lemma 3.9 (Restrictions) *Let V be a nested cluster basis. For all $t \in \mathcal{T}_\mathcal{I}$ and all $t^* \in \operatorname{sons}^*(t)$, we define*

$$E_{t^*,t} := \begin{cases} E_{t^*,t'} E_{t'} & \text{if there is a } t' \in \operatorname{sons}(t) \text{ with } t^* \in \operatorname{sons}(t'), \\ I & \text{otherwise, i.e., if } \operatorname{sons}(t) = \emptyset \end{cases}$$

and find

$$\chi_{t^*} V_t = V_{t^*} E_{t^*,t}. \tag{7}$$

Proof. By induction over $\#\operatorname{sons}^*(t) \in \mathbb{N}$. Let $t \in \mathcal{T}_{\mathcal{I}}$ and $t^* \in \operatorname{sons}^*(t)$. If $\#\operatorname{sons}^*(t) = 1$ holds, we have $\operatorname{sons}(t) = \emptyset$ and therefore $t = t^*$. The definition of the cluster basis implies $\chi_{t^*} V_t = \chi_t V_t = V_t = V_t E_{t,t}$.

Let $n \in \mathbb{N}$ be such that (7) holds for all $t \in \mathcal{T}_{\mathcal{I}}$ with $\#\operatorname{sons}^*(t) = n$. Let $t \in \mathcal{T}_{\mathcal{I}}$ with $\#\operatorname{sons}^*(t) = n + 1$. Let $t^* \in \operatorname{sons}^*(t)$. If $t^* = t$, we can proceed as before. If $t^* \neq t$, there is a $t' \in \operatorname{sons}(t)$ with $t^* \in \operatorname{sons}^*(t')$, and we find

$$\chi_{t^*} V_t = \chi_{t^*} \sum_{t' \in \operatorname{sons}(t)} V_{t'} E_{t'} = \chi_{t^*} V_{t'} E_{t'}.$$

Due to $\#\operatorname{sons}(t') \leq \#\operatorname{sons}(t) - 1 = n$, we can apply the induction assumption to find $\chi_{t^*} V_{t'} = V_{t^*} E_{t^*,t'}$, which implies

$$\chi_{t^*} V_t = \chi_{t^*} V_{t'} E_{t'} = V_{t^*} E_{t^*,t'} E_{t'} = V_{t^*} E_{t^*,t},$$

and concludes the induction. ■

Let us assume that $M \in \mathcal{H}^2(P, V, *)$ is a left semi-uniform matrix. Let $t \in \mathcal{T}_{\mathcal{I}}$, and let $t^+ \in \operatorname{pred}(t)$ and $r \in \operatorname{row}(t^+)$. Since M is left semi-uniform and $b := (t^+, r) \in P_{\text{far}}$ holds, there is a matrix $B_b \in \mathbb{R}^{\mathcal{J} \times k_{t^+}}$ satisfying $\chi_{t^+} M \chi_r = V_{t^+} B_b^\top$. Due to $t \in \operatorname{sons}^*(t^+)$, Lemma 3.9 implies

$$\chi_t M \chi_r = \chi_t \chi_{t^+} M \chi_r = \chi_t V_{t^+} B_b^\top = V_t E_{t,t^+} B_b^\top,$$

i.e., not only admissible blocks, but also their restrictions can be expressed in terms of appropriate cluster bases. According to Definition 3.7, this means

$$M_t = \sum_{t^+ \in \operatorname{pred}(t)} \sum_{r \in \operatorname{row}(t^+)} \chi_t M \chi_r = V_t \left(\sum_{t^+ \in \operatorname{pred}(t)} \sum_{r \in \operatorname{row}(t^+)} E_{t,t^+} B_{(t^+,r)}^\top \right),$$

i.e., the range of M_t is contained in the range of V_t, therefore the rank of M_t is bounded by the rank of V_t, i.e., by k_t.

3.3 Approximability by left semi-uniform matrices

We have seen that the ranks of all total cluster basis matrices M_t are bounded if M is a left semi-uniform matrix M.

Now we will prove the converse, i.e., that M is left semi-uniform if the ranks of the total cluster basis matrices M_t are bounded. Theorem 3.13 even shows that M can be *approximated* efficiently by a left semi-uniform matrix if the total cluster basis matrices can be approximated by low-rank matrices.

Since we are interested in approximations, we reformulate the problem: instead of requiring that the rank of M_t is bounded, we only require that there is a matrix $W_t \in \mathbb{R}^{\mathcal{I} \times k_t}$ with $\chi_t W_t = W_t$ such that

$$\sup_{x \in \mathbb{R}^{\mathcal{J}}} \inf_{y \in \mathbb{R}^{k_t}} \frac{\|M_t x - W_t y\|_2}{\|x\|_2} \leq \epsilon_t \tag{8}$$

holds for a suitable tolerance $\epsilon_t \in \mathbb{R}_{\geq 0}$.

For $\epsilon_t = 0$, this inequality implies $\operatorname{rank}(M_t) \leq \operatorname{rank}(W_t) = k_t$. For larger values of ϵ_t, the "numerical rank" takes the place of the exact one.

In order to simplify the notation, we replace (8) by the equivalent condition

$$\inf_{Z \in \mathbb{R}^{\mathcal{J} \times k_t}} \|M_t - W_t Z^\top\|_2 \leq \epsilon_t. \tag{9}$$

Before we can prove the main result of this section, we require an alternative representation of the error: since a block $(t, s) \in P_{\text{far}}$ is not only represented by M_t, but, at least in part, also by M_{t^*} for any descendant $t^* \in \operatorname{sons}^*(t)$, replacing the total cluster basis by an approximation does not only introduce an error in the cluster t, but also additional errors for all $t^* \in \operatorname{sons}^*(t)$. If the total cluster basis is approximated by a nested orthogonal cluster basis, it is possible to express the error as an orthogonal sum of individual errors introduced in each cluster. In order to formulate the corresponding Theorem 3.11, we require the following notations for nested orthogonal cluster bases:

Definition 3.10 *Let $V = (V_t)_{t \in \mathcal{T}_\mathcal{I}}$ be a nested orthogonal cluster basis. We define the families $Q = (Q_t)_{t \in \mathcal{T}_\mathcal{I}}$ and $\widehat{V} = (\widehat{V}_t)_{t \in \mathcal{T}_\mathcal{I}}$ as follows: if $\operatorname{sons}(t) = \emptyset$, we let $Q_t = \chi_t$ and $\widehat{V}_t = V_t$. If $\operatorname{sons}(t) \neq \emptyset$, we let $\sigma := \# \operatorname{sons}(t)$, $\{t_1, \ldots, t_\sigma\} := \operatorname{sons}(t)$, and*

$$Q_t = \begin{pmatrix} V_{t_1} & \ldots & V_{t_\sigma} \end{pmatrix} \quad \text{and} \quad \widehat{V}_t = \begin{pmatrix} E_{t_1} \\ \vdots \\ E_{t_\sigma} \end{pmatrix}.$$

Note that the nested structure of V implies $V_t = Q_t \widehat{V}_t$ for all $t \in \mathcal{T}_\mathcal{I}$. Now we can construct a decomposition of the approximation error into "local" contributions: for each cluster $t \in \mathcal{T}_\mathcal{I}$, we compare the approximation of M_t by the cluster basis V_t with the approximation by Q_t, i.e., by the cluster bases corresponding to the sons of t.

Since V_t and Q_t are orthogonal, the best approximations of M_t are given by $V_t V_t^\top M_t$ and $Q_t Q_t^\top M_t$, respectively, and the error added by a cluster t is given by the expression $Q_t Q_t^\top M_t - V_t V_t^\top M_t$. These expression characterize the total approximation error.

Theorem 3.11 (Approximation error) *Let $M \in \mathbb{R}^{\mathcal{I} \times \mathcal{J}}$. Let $V = (V_t)_{t \in \mathcal{T}_\mathcal{I}}$ be a nested orthogonal cluster basis, let $Q = (Q_t)_{t \in \mathcal{T}_\mathcal{I}}$ be as in Definition 3.10. We define*

$$D_t := Q_t Q_t^\top M_t - V_t V_t^\top M_t \qquad \text{for all } t \in \mathcal{T}_\mathcal{I}.$$

Then we have

$$\|(M - \Pi_{V,*}^P M)x\|_2^2 = \sum_{t \in \mathcal{T}_\mathcal{I}} \|D_t x\|_2^2 \qquad \text{for all } x \in \mathbb{R}^\mathcal{J}.$$

Proof. We start by proving

$$(M_t - V_t V_t^\top M_t)\chi_s = \sum_{t^* \in \operatorname{sons}^*(t)} D_{t^*} \chi_s \tag{10}$$

for all $t \in \mathcal{T}_{\mathcal{I}}$, $t^+ \in \operatorname{pred}(t)$ and $s \in \operatorname{row}(t^+)$ by induction over $\#\operatorname{sons}^*(t)$. For $\#\operatorname{sons}^*(t) = 1$, we have $\operatorname{sons}(t) = \emptyset$ and (10) follows by definition.

Let now $n \in \mathbb{N}$ be such that (10) holds for all $t \in \mathcal{T}_{\mathcal{I}}$ with $\#\operatorname{sons}^*(t) \leq n$. Let $t \in \mathcal{T}_{\mathcal{I}}$ with $\#\operatorname{sons}^*(t) = n+1$. We have $\operatorname{sons}(t) \neq \emptyset$, and

$$Q_t Q_t^\top = \sum_{t' \in \operatorname{sons}(t)} V_{t'} V_{t'}^\top \tag{11}$$

implies

$$\begin{aligned} M_t - V_t V_t^\top M_t &= M_t - Q_t Q_t^\top M_t + Q_t Q_t^\top M_t - V_t V_t^\top M_t \\ &= \sum_{t' \in \operatorname{sons}(t)} \chi_{t'} M_t - \sum_{t' \in \operatorname{sons}(t)} V_{t'} V_{t'}^\top M_t + D_t. \end{aligned}$$

Let $t^+ \in \operatorname{pred}(t)$ and $s \in \operatorname{row}(t^+)$. Due to the definition of M_t, we have

$$\begin{aligned} M_t \chi_s - V_t V_t^\top M_t \chi_s &= \sum_{t' \in \operatorname{sons}(t)} (\chi_{t'} M_t \chi_s - V_{t'} V_{t'}^\top M_t \chi_s) + D_t \chi_s \\ &= \sum_{t' \in \operatorname{sons}(t)} (M_{t'} - V_{t'} V_{t'}^\top M_{t'}) \chi_s + D_t \chi_s. \end{aligned}$$

Since $\#\operatorname{sons}^*(t') \leq \#\operatorname{sons}^*(t) - 1 = n$, we can apply the induction assumption to prove

$$M_t \chi_s - V_t V_t^\top M_t \chi_s = \sum_{t' \in \operatorname{sons}(t)} \sum_{t^* \in \operatorname{sons}^*(t')} D_{t^*} \chi_s + D_t \chi_s = \sum_{t^* \in \operatorname{sons}^*(t)} D_{t^*} \chi_s \tag{12}$$

and conclude the induction.

The approximation error is given by

$$\begin{aligned} M - \Pi_{V,*}^P M &= \sum_{(t,s) \in P_{\text{far}}} \chi_t M \chi_s - V_t V_t^\top M \chi_s = \sum_{t \in \mathcal{T}_{\mathcal{I}}} \sum_{s \in \operatorname{row}(t)} (M_t - V_t V_t^\top M_t) \chi_s \\ &\overset{(12)}{=} \sum_{t \in \mathcal{T}_{\mathcal{I}}} \sum_{s \in \operatorname{row}(t)} \sum_{t^* \in \operatorname{sons}^*(t)} D_{t^*} \chi_s = \sum_{t^* \in \mathcal{T}_{\mathcal{I}}} \sum_{t \in \operatorname{pred}(t^*)} \sum_{s \in \operatorname{row}(t)} D_{t^*} \chi_s \\ &= \sum_{t^* \in \mathcal{T}_{\mathcal{I}}} D_{t^*} = \sum_{t \in \mathcal{T}_{\mathcal{I}}} D_t. \end{aligned} \tag{13}$$

We will now investigate the properties of the matrices D_t appearing in this representation of the error.

Since V is nested, we have $V_t = Q_t \widehat{V}_t$ with $\widehat{V}_t$ as in Definition 3.10. This implies $V_t = Q_t \widehat{V}_t = Q_t Q_t^\top Q_t \widehat{V}_t = Q_t Q_t^\top V_t$ and we find

$$V_t^\top D_t = V_t^\top Q_t Q_t^\top M_t - V_t^\top V_t V_t^\top M_t = V_t^\top M_t - V_t^\top M_t = 0,$$

i.e., the range of D_t is perpendicular on that of V_t.

Let now $t \in \mathcal{T}_{\mathcal{I}}$ with $\operatorname{sons}(t) \neq \emptyset$, and let $t' \in \operatorname{sons}(t)$. Due to (11) and (2), we have

$$\chi_{t'} D_t = \chi_{t'} Q_t Q_t^\top M_t - \chi_{t'} V_t V_t^\top M_t = V_{t'} V_{t'}^\top M_t - V_{t'} E_{t'} V_t^\top M_t = V_{t'}(V_{t'}^\top M_t - E_{t'} V_t^\top M_t),$$

and Lemma 3.9 implies

$$\chi_{t^*} D_t = V_{t^*} E_{t^*,t'} (V_{t'}^\top M_t - E_{t'} V_t^\top M_t) \tag{14}$$

for all $t^* \in \operatorname{sons}^*(t) \setminus \{t\}$, i.e., the restriction of the range of D_t to proper descendants $t^* \neq t$ of t is contained in the range of V_{t^*}.

Let now $t, s \in \mathcal{T}_{\mathcal{I}}$ with $t \neq s$, and let $x, y \in \mathbb{R}^{\mathcal{J}}$. If $s \in \operatorname{sons}^*(t)$, we can find $t' \in \operatorname{sons}(t)$ with $s \in \operatorname{sons}^*(t')$ and get

$$D_s^\top D_t = D_s^\top \chi_s D_t \overset{(14)}{=} D_s^\top V_s E_{s,t'} (V_{t'}^\top M_t - E_{t'} V_t^\top M_t) = 0. \tag{15}$$

Due to symmetry, the same holds if $t \in \operatorname{sons}^*(s)$. If $s \notin \operatorname{sons}^*(t)$ and $t \notin \operatorname{sons}^*(s)$, the definition of the cluster tree implies $\hat{t} \times \hat{s} = \emptyset$, and since the support of $D_t x$ is contained in $\hat{t}$ and that of $D_s y$ in $\hat{s}$, we can conclude $D_s^\top D_t = 0$ for all $t, s \in \mathcal{T}_{\mathcal{I}}$ with $t \neq s$, i.e., the ranges of the matrices $(D_t)_{t \in \mathcal{T}_{\mathcal{I}}}$ are pairwise perpendicular.

This implies

$$\|(M - \Pi_{V,*}^P M)x\|_2^2 \overset{(13)}{=} \left\| \sum_{t \in \mathcal{T}_{\mathcal{I}}} D_t x \right\|_2^2 \overset{(15)}{=} \sum_{t \in \mathcal{T}_{\mathcal{I}}} \|D_t x\|_2^2$$

and concludes the proof. ■

With the precise expression for the approximation error provided by Theorem 3.11, we can now use the matrices $(W_t)_{t \in \mathcal{T}_{\mathcal{I}}}$ from condition (9) in order to construct a suitable nested orthogonal cluster basis.

Construction 3.12 *Let $k = (k_t)_{t \in \mathcal{T}_{\mathcal{I}}}$ be a rank distribution. Let $W = (W_t)_{t \in \mathcal{T}_{\mathcal{I}}}$ be a family of matrices satisfying $\chi_t W_t = W_t \in \mathbb{R}^{\mathcal{I} \times k_t}$ for all $t \in \mathcal{T}_{\mathcal{I}}$.*

Let $t \in \mathcal{T}_{\mathcal{I}}$. If $\operatorname{sons}(t) = \emptyset$, we compute the Householder factorization $V_t R = W_t$ in order to find an orthogonal matrix $V_t \in \mathbb{R}^{\mathcal{I} \times k_t}$ with $\chi_t V_t = V_t$ and $\operatorname{range}(W_t) \subseteq \operatorname{range}(V_t)$.

If $\operatorname{sons}(t) \neq \emptyset$, we assume that the matrices $V_{t'}$ corresponding to $t' \in \operatorname{sons}(t)$ have already been constructed, form Q_t as in Definition 3.10 and define

$$\widehat{W}_t := Q_t^\top W_t.$$

We compute the Householder factorization $\widehat{V}_t R = \widehat{W}_t$ in order to find an orthogonal matrix $\widehat{V}_t$ with $\operatorname{range}(\widehat{W}_t) \subseteq \operatorname{range}(\widehat{V}_t)$. Setting $V_t := Q_t \widehat{V}_t$ concludes the construction.

The transfer matrices $(E_t)_{t \in \mathcal{T}_{\mathcal{I}}}$ of the nested cluster basis V can be reconstructed from the matrices $\widehat{V}_t$ by applying Definition 3.10.

We can now prove that a matrix can be approximated by a left semi-uniform matrix if all cluster basis matrices $(M_t)_{t \in \mathcal{T}_{\mathcal{I}}}$ can be approximated by low-rank matrices:

Theorem 3.13 (Error bound) *Let $M \in \mathbb{R}^{\mathcal{I}\times\mathcal{J}}$. Let $W = (W_t)_{t\in\mathcal{T}_\mathcal{I}}$ be a family of matrices satisfying $\chi_t W_t = W_t \in \mathbb{R}^{\mathcal{I}\times k_t}$ for all $t \in \mathcal{T}_\mathcal{I}$. Let $V = (V_t)_{t\in\mathcal{T}_\mathcal{I}}$ be the nested orthogonal cluster basis with rank distribution $k = (k_t)_{t\in\mathcal{T}_\mathcal{I}}$ from Construction 3.12. Then we have*

$$\|M - \Pi^P_{V,*}M\|_2^2 \leq \sum_{t\in\mathcal{T}_\mathcal{I}} \inf_{Z\in\mathbb{R}^{\mathcal{J}\times k_t}} \|M_t - W_t Z^\top\|_2^2 \quad \textit{and}$$
$$\|M - \Pi^P_{V,*}M\|_F^2 \leq \sum_{t\in\mathcal{T}_\mathcal{I}} \inf_{Z\in\mathbb{R}^{\mathcal{J}\times k_t}} \|M_t - W_t Z^\top\|_F^2.$$

Proof. Due to Theorem 3.11, we can restrict our attention to the error operators D_t.

Let $t \in \mathcal{T}_\mathcal{I}$ and $x \in \mathbb{R}^\mathcal{J}$. If $\operatorname{sons}(t) = \emptyset$, we have

$$\|D_t x\|_2^2 = \|M_t x - V_t V_t^\top M_t x\|_2^2 = \inf_{y\in\mathbb{R}^{k_t}} \|M_t x - V_t y\|_2^2 \leq \inf_{Z\in\mathbb{R}^{\mathcal{J}\times k_t}} \|(M_t - W_t Z^\top)x\|_2^2,$$

since $\operatorname{range}(W_t) \subseteq \operatorname{range}(V_t)$.

If $\operatorname{sons}(t) \neq \emptyset$, we find

$$\begin{aligned}
\|D_t x\|_2^2 &= \|Q_t Q_t^\top M_t x - V_t V_t^\top M_t x\|_2^2 = \inf_{y\in\mathbb{R}^{k_t}} \|Q_t Q_t^\top M_t x - V_t y\|_2^2 \\
&= \inf_{y\in\mathbb{R}^{k_t}} \|Q_t Q_t^\top M_t x - Q_t \widehat{V}_t y\|_2^2 \leq \inf_{Z\in\mathbb{R}^{\mathcal{J}\times k_t}} \|Q_t Q_t^\top (M_t - Q_t \widehat{W}_t Z^\top)x\|_2^2 \\
&= \inf_{Z\in\mathbb{R}^{\mathcal{J}\times k_t}} \|Q_t Q_t^\top (M_t - Q_t Q_t^\top W_t Z^\top)x\|_2^2 = \inf_{Z\in\mathbb{R}^{\mathcal{J}\times k_t}} \|Q_t Q_t^\top (M_t - W_t Z^\top)x\|_2^2 \\
&\leq \inf_{Z\in\mathbb{R}^{\mathcal{J}\times k_t}} \|(M_t - W_t Z^\top)x\|_2^2,
\end{aligned}$$

due to $\operatorname{range}(\widehat{W}_t) \subseteq \operatorname{range}(\widehat{V}_t)$ and the orthogonality of Q_t.

In order to prove the error estimate for the operator norm, we combine our estimates with Theorem 3.11 and get

$$\|(M - \Pi^P_{V,*}M)x\|_2^2 \leq \sum_{t\in\mathcal{T}_\mathcal{I}} \inf_{Z\in\mathbb{R}^{\mathcal{J}\times k_t}} \|(M_t - W_t Z^\top)x\|_2^2 \leq \sum_{t\in\mathcal{T}_\mathcal{I}} \inf_{Z\in\mathbb{R}^{\mathcal{J}\times k_t}} \|M_t - W_t Z^\top\|_2^2 \|x\|_2^2,$$

so taking the supremum over all $x \in \mathbb{R}^\mathcal{J}$ yields our claim. For the error estimate in the Frobenius norm, we observe that

$$\|X\|_F^2 = \sum_{j\in\mathcal{J}} \|X e_j\|_2^2$$

holds for an arbitrary matrix $X \in \mathbb{R}^{\mathcal{I}\times\mathcal{J}}$, where $e_j \in \mathbb{R}^\mathcal{J}$ is the j-th unit vector. This implies

$$\begin{aligned}
\|M - \Pi^P_{V,*}M\|_F^2 &= \sum_{j\in\mathcal{J}} \|(M - \Pi^P_{V,*}M)e_j\|_2^2 \leq \sum_{j\in\mathcal{J}} \sum_{t\in\mathcal{T}_\mathcal{I}} \inf_{Z\in\mathbb{R}^{\mathcal{J}\times k_t}} \|(M_t - W_t Z^\top)e_j\|_2^2 \\
&= \sum_{t\in\mathcal{T}_\mathcal{I}} \inf_{Z\in\mathbb{R}^{\mathcal{J}\times k_t}} \|M_t - W_t Z^\top\|_F^2
\end{aligned}$$

and concludes the proof. ■

Corollary 3.14 *Let $M \in \mathbb{R}^{\mathcal{I}\times\mathcal{J}}$ with $\operatorname{rank}(M_t) \leq k_t$ for all $t \in \mathcal{T}_{\mathcal{I}}$. Then there is a nested orthogonal cluster basis $V = (V_t)_{t\in\mathcal{T}_{\mathcal{I}}}$ with rank distribution $k = (k_t)_{t\in\mathcal{T}_{\mathcal{I}}}$ satisfying $M = \Pi_{V,*}^{P} M$, i.e., if the total cluster basis matrices of M have bounded rank, M is a left semi-uniform matrix.*

Proof. Due to $\operatorname{rank}(M_t) \leq k_t$, we can find $W_t \in \mathbb{R}^{\mathcal{I}\times k_t}$ and $Z_t \in \mathbb{R}^{\mathcal{J}\times k_t}$ with $\chi_t W_t = W_t$ and $W_t Z_t^\top = M_t$. Theorem 3.13 concludes the proof. ■

We will now apply Theorem 3.13 to prove that integral and differential operators can be approximated by $\mathcal{H}^2$-matrices.

Remark 3.15 (Integral operators) *Let $\Omega \subset \mathbb{R}^d$ be a subdomain or submanifold, and let $(\varphi_i)_{i\in\mathcal{I}}$ be a finite element basis of a suitable function space on Ω. If the matrix $M \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ has the form*

$$M_{ij} = \int_\Omega \varphi_j(x) \int_\Omega \varphi_i(y) g(x,y)\,dy\,dx$$

for a suitable kernel function $g : \Omega\times\Omega \to \mathbb{R}$, we can apply [5, Corollary 4.7] to an m-th order interpolant of g in order to find a matrix W_t with rank $k_t = m^d$ which satisfies

$$\inf_{Z\in\mathbb{R}^{\mathcal{I}\times k_t}} \|M_t - W_t Z^\top\|_2 \lesssim |\Omega_t|^{1/2} \exp(-\alpha m),$$

where $\alpha \in \mathbb{R}_{>0}$ and $\Omega_t = \bigcup_{i\in\hat{t}} \operatorname{supp}(\varphi_i)$ is the support of all basis functions corresponding to indices in $\hat{t}$.

Theorem 3.13 requires us to sum over all t, and since the overlap of the supports of basis functions is limited, we find

$$\sum_{t\in\mathcal{T}_{\mathcal{I}}} \inf_{Z\in\mathbb{R}^{\mathcal{J}\times k_t}} \|M_t - W_t Z^\top\|_2^2 \lesssim \operatorname{depth}(\mathcal{T}_{\mathcal{I}})|\Omega| \exp(-2\alpha m).$$

Replacing polynomial interpolation by an expansion in spherical harmonics leads to a similar estimate with $k_t = m^{d-1}$. Using variable-order expansions [23, 22, 10, 9], we can eliminate the depth of the cluster tree $\mathcal{T}_{\mathcal{I}}$ from the error estimate and, applying Theorem 3.11 in a more sophisticated manner, even ensure that the approximation error stays proportional to the discretization error without sacrificing the linear relationship between the computational complexity and the number of degrees of freedom.

Remark 3.16 (Solution operators) *Let $\Omega \subseteq \mathbb{R}^d$ be a subdomain with Lipschitz boundary, and let $(\varphi_i)_{i\in\mathcal{I}}$ be a finite element basis of $H_0^1(\Omega)$. If the matrix $M \in \mathbb{R}^{\mathcal{I}\times\mathcal{I}}$ is the discrete solution operator corresponding to the Galerkin discretization of the elliptic partial differential operator*

$$L := -\operatorname{div} C(x) \operatorname{grad},$$

where $C : \Omega \to \mathbb{R}^{d\times d}$ maps each point $x \in \Omega$ to a coefficient matrix satisfying $0 < \alpha I \leq C(x) \leq \beta I$, we can use [1, Theorem 2.8] to construct rank-k_t-approximations of M_t, where $k_t \sim |\log(\epsilon)|^{d+1}$ and $\epsilon \in \mathbb{R}_{>0}$ is the desired precision.

We can proceed as in the case of integral operators to conclude that we can find an $\mathcal{H}^2$-matrix approximation of M.

4 Construction of $\mathcal{H}^2$-matrix approximations

Let $M \in \mathbb{R}^{\mathcal{I}\times\mathcal{J}}$. Using Theorem 3.13, we can ensure that M can be approximated by a left semi-uniform matrix, but we have yet to find an algorithm which computes a suitable cluster basis efficiently.

4.1 Approximation algorithm

We use a slightly modified version of the algorithm presented in [7]: the original algorithm uses Gram matrices and eigenvectors to compute the left singular vectors required for the construction of the matrices V_t. Since the computation of the Gram matrices can lead to numerical instabilities, the modified algorithm presented here relies on singular value decompositions of the total cluster basis matrices M_t and their orthogonal projections, thus reaching a higher accuracy and better compression.

The approximation algorithm is motivated by Theorem 3.11: for leaf cluster t, we have to find an orthogonal matrix $V_t \in \mathbb{R}^{\mathcal{I}\times k_t}$ such that $V_t = \chi_t V_t$ holds and the error $\|M_t - V_t V_t^\top M_t\|_2$ or $\|M_t - V_t V_t^\top M_t\|_F$ is as small as possible. This minimization problem can be solved by using the singular value decomposition of M_t: due to [12, Theorem 2.5.2], we can find orthogonal matrices P_1 and P_2 with

$$M_t = P_1 \operatorname{diag}(\sigma_1, \ldots, \sigma_l) P_2^\top,$$

where $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_l \geq 0$ are the singular values of M_t.

For a given rank $k_t \in \{1, \ldots, l\}$, we can construct a rank-k_t-approximation by setting

$$\widetilde{M}_t := P_1 \operatorname{diag}(\sigma_1, \ldots, \sigma_{k_t}, 0, \ldots, 0) P_2^\top,$$

and since P_1 and P_2 are orthogonal matrices, the approximation error is given by

$$\|M_t - \widetilde{M}_t\|_2 = \epsilon_{2,t} := \begin{cases} \sigma_{k_t+1} & \text{if } k_t < l, \\ 0 & \text{otherwise,} \end{cases} \tag{16a}$$

$$\|M_t - \widetilde{M}_t\|_F = \epsilon_{F,t} := \left(\sum_{i=k_t+1}^{l} \sigma_i^2 \right)^{1/2}. \tag{16b}$$

This means that we can control the approximation error by adapting the rank k_t.

We construct the matrix V_t by copying the first k_t columns of the matrix P_1 and

$$V_t V_t^\top M_t = V_t V_t^\top P_1 \operatorname{diag}(\sigma_1, \ldots, \sigma_l) P_2^\top = P_1 \operatorname{diag}(\sigma_1, \ldots, \sigma_{k_t}, 0, \ldots, 0) P_2^\top = \widetilde{M}_t, \tag{17}$$

yields that the corresponding orthogonal projection will indeed give us $\widetilde{M}_t$.

Now let us consider the case of non-leaf clusters t. Since we are looking for a nested cluster basis, we have to ensure $V_t = Q_t \widehat{V}_t$ (cf. Definition 3.10). Proceeding by recursion, we can compute the orthogonal cluster basis matrices $V_{t'}$, and since these matrices define Q_t, we only have to construct the remaining matrix $\widehat{V}_t$.

Since V_t is orthogonal,

$$I = V_t^\top V_t = \widehat{V}_t^\top Q_t^\top Q_t \widehat{V}_t = \widehat{V}_t^\top \widehat{V}_t$$

implies that $\widehat{V}_t$ also has to be orthogonal.

According to Theorem 3.11, the error of the matrix approximation will be small if we can ensure that the local errors D_t are small. In our case, we have $V_t = Q_t \widehat{V}_t$ and observe

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

We introduce the auxiliary matrix

$$\widehat{M_t} := Q_t^\top M_t$$

and conclude

$$D_t = Q_t (\widehat{M_t} - \widehat{V}_t \widehat{V}_t^\top \widehat{M_t}).$$

We are only interested in minimizing the norm

$$\|D_t x\|_2 = \|Q_t (\widehat{M_t} - \widehat{V}_t \widehat{V}_t^\top \widehat{M_t}) x\|_2 = \|(\widehat{M_t} - \widehat{V}_t \widehat{V}_t^\top \widehat{M_t}) x\|_2$$

for vectors $x \in \mathbb{R}^{\mathcal{J}}$, and since this norm does not depend on Q_t, we can base the computation of $\widehat{V}_t$ entirely on the auxiliary matrix $\widehat{M_t}$: as in the case of the leaf clusters, we have to find an orthogonal matrix $\widehat{V}_t$ with k_t columns that minimizes the error $\|\widehat{M_t} - \widehat{V}_t \widehat{V}_t^\top \widehat{M_t}\|_2$ or $\|\widehat{M_t} - \widehat{V}_t \widehat{V}_t^\top \widehat{M_t}\|_F$, and this problem can again be solved by computing the singular value decomposition of $\widehat{M_t}$. Since this matrix has only

$$m_t := \begin{cases} \sum_{t' \in \mathrm{sons}(t)} k_{t'} & \text{if } \mathrm{sons}(t) \neq \emptyset, \\ \#\hat{t} & \text{otherwise,} \end{cases}$$

rows and $\#\mathcal{J}$ columns, it can be handled very efficiently. Once $\widehat{V}_t$ has been computed, the transfer matrices $E_{t'}$ can be recovered using Definition 3.10.

In order to reach a useful algorithm, we need an efficient method for computing $\widehat{M_t}$. Using $\widehat{M_t} = Q_t^\top M_t$ directly leads to a complexity of $\mathcal{O}(\#\hat{t}\#\mathcal{J})$, which is acceptable for small clusters, but not for large ones. In the case $\mathrm{sons}(t) = \emptyset$, we have $\widehat{M_t} = M_t$ and can safely assume that the number of rows of M_t is small enough. In the case $\mathrm{sons}(t) \neq \emptyset$, we note that

$$\chi_{t'} M_t = M_{t'} \chi_t^c$$

holds, where

$$\chi_t^c := \sum_{t^+ \in \mathrm{pred}(t)} \sum_{r \in \mathrm{row}(t^+)} \chi_r,$$

corresponds to the restriction of $M_{t'}$ to the clusters in $\mathrm{row}^*(t)$. This means that we can recover the total cluster basis of t from the total cluster bases of its sons.

Algorithm 1 Construction of an adaptive cluster basis

```
procedure RowClusterBasis($t$);
if $\mathrm{sons}(t) = \emptyset$ then
  $\widehat{M}_t := \chi_t M \chi_t^c$
else
  for $t' \in \mathrm{sons}(t)$ do
    RowClusterBasis($t'$)
  end for;
  $\widehat{M}_t := \begin{pmatrix} \bar{M}_{t_1}\chi_t^c \\ \vdots \\ \bar{M}_{t_\sigma}\chi_t^c \end{pmatrix}$
end if;
Construct $\widehat{V}_t$ from the singular value decomposition of $\widehat{M}_t$;
$\bar{M}_t := \widehat{V}_t^\top \widehat{M}_t$
```

We set $\sigma := \#\,\mathrm{sons}(t)$ and $\{t_1, \ldots, t_\sigma\} := \mathrm{sons}(t)$, and rewriting $\widehat{M}_t$ in the form

$$\widehat{M}_t = Q_t^\top M_t = \begin{pmatrix} V_{t_1}^\top M_t \\ \vdots \\ V_{t_\sigma}^\top M_t \end{pmatrix} = \begin{pmatrix} V_{t_1}^\top M_{t_1}\chi_t^c \\ \vdots \\ V_{t_\sigma}^\top M_{t_\sigma}\chi_t^c \end{pmatrix}$$

suggests the solution: after finding a new cluster basis matrix V_t, we compute

$$\bar{M}_t := V_t^\top M_t = \widehat{V}_t^\top Q_t^\top M_t = \widehat{V}_t^\top \widehat{M}_t$$

and then use

$$\widehat{M}_t = \begin{pmatrix} \bar{M}_{t_1}\chi_t^c \\ \vdots \\ \bar{M}_{t_\sigma}\chi_t^c \end{pmatrix}$$

to compute $\widehat{M}_t$ in $\mathcal{O}(m_t \# \mathcal{J})$ operations. The resulting recursive procedure is given in Algorithm 1.

Remark 4.1 (Complexity) *We assume that there is an upper bound k_* for the rank and the size of leaf clusters, i.e., that $k_t \leq k_*$ holds for all $t \in \mathcal{T}_\mathcal{I}$ and $\#\hat{t} \leq k_*$ holds for all leaves $t \in \mathcal{L}_\mathcal{I}$. We also assume that there is an upper bound σ_* for the number of sons, i.e., that $\#\,\mathrm{sons}(t) \leq \sigma_*$ holds for all $t \in \mathcal{T}_\mathcal{I}$.*

For leaf clusters $t \in \mathcal{T}_\mathcal{I}$, the construction of $\widehat{M}_t$ requires not more than $\mathcal{O}(\#\hat{t}\#\mathcal{J})$ operations, computing the singular value decomposition of this matrix can be accomplished in $\mathcal{O}((\#\hat{t})^2\#\mathcal{J}) \subseteq \mathcal{O}(k_*\#\hat{t}\#\mathcal{J})$ operations, and the matrix $\bar{M}_t$ can be formed in $\mathcal{O}(k_t\#\hat{t}\#\mathcal{J})$ operations. Summing over all leaf clusters, we get a complexity of $\mathcal{O}(k_*\#\mathcal{I}\#\mathcal{J})$.

For non-leaf clusters $t \in \mathcal{T}_\mathcal{I}$, the construction of $\widehat{M}_t$ requires $\mathcal{O}(m_t\#\mathcal{J})$ operations, the singular value decomposition can be computed in $\mathcal{O}(m_t^2\#\mathcal{J}) \subseteq \mathcal{O}(\sigma_*^2 k_*^2\#\mathcal{J})$ operations, *and the construction of $\bar{M}_t$ requires $\mathcal{O}(k_t m_t \# \mathcal{J}) \subseteq \mathcal{O}(\sigma_* k_*^2 \# \mathcal{J})$ operations. Summing over all clusters, we get a complexity of $\mathcal{O}(\sigma_*^2 k_*^2 \# \mathcal{T}_{\mathcal{I}} \# \mathcal{J})$.*

We can conclude that Algorithm 1 has a complexity of $\mathcal{O}(k_*(\#\mathcal{I} + \sigma_*^2 k_* \# \mathcal{T}_{\mathcal{I}}) \# \mathcal{J})$ for a general matrix. Assuming $\# \mathcal{T}_{\mathcal{I}} \in \mathcal{O}(\#\mathcal{I}/k_*)$, we can bound this by $\mathcal{O}(k_* \# \mathcal{I} \# \mathcal{J})$.

If the matrix M is not given in standard notation, but in a data-sparse format, the complexity of Algorithm 1 can be reduced significantly: the techniques of [7, Section 6] can be used to reach almost linear complexity if M is represented as a hierarchical matrix, and the approach of [3, Section 4] can be used to reach "true" linear complexity if M is represented as an $\mathcal{H}^2$-matrix.

4.2 Error control

Since our algorithm is closely related to Construction 3.12 and the techniques used in Theorem 3.11, it is relatively simple to derive error bounds.

Theorem 4.2 (Precision) *Let $V = (V_t)_{t \in \mathcal{T}_{\mathcal{I}}}$ be the orthogonal nested cluster basis constructed by Algorithm 1. We have*

$$\|M - \Pi^P_{V,*} M\|_2^2 \leq \sum_{t \in \mathcal{T}_{\mathcal{I}}} \epsilon_{2,t}^2 \quad \textit{and} \quad \|M - \Pi^P_{V,*} M\|_F^2 = \sum_{t \in \mathcal{T}_{\mathcal{I}}} \epsilon_{F,t}^2.$$

Proof. The matrices D_t from Theorem 3.11 are given by

$$\begin{aligned} D_t &= Q_t Q_t^\top M_t - V_t V_t^\top M_t = Q_t Q_t^\top M_t - Q_t \widehat{V}_t \widehat{V}_t^\top Q_t^\top M_t \\ &= Q_t (I - \widehat{V}_t \widehat{V}_t^\top) \widehat{M}_t \overset{(17)}{=} Q_t (\widehat{M}_t - \widetilde{M}_t), \end{aligned}$$

so we have

$$\|D_t\|_2 = \epsilon_{2,t} \quad \text{and} \quad \|D_t\|_F = \epsilon_{F,t}$$

due to (16) and can proceed as in the proof of Theorem 3.13. ■

We can also demonstrate that the error estimate for the approximation constructed by Algorithm 1 is at least as good as the error estimate of Theorem 3.13:

Remark 4.3 *Let $V = (V_t)_{t \in \mathcal{T}_{\mathcal{I}}}$ be as before and let $W = (W_t)_{t \in \mathcal{T}_{\mathcal{I}}}$ be as in Theorem 3.13. We find*

$$\begin{aligned} \epsilon_{2,t} &= \|\widehat{M}_t - \widetilde{M}_t\|_2 = \|\widehat{M}_t - \widehat{V}_t \widehat{V}_t^\top \widehat{M}_t\|_2 = \|Q_t^\top M_t - \widehat{V}_t \widehat{V}_t^\top Q_t^\top M_t\|_2 \\ &= \|Q_t Q_t^\top M_t - Q_t \widehat{V}_t \widehat{V}_t^\top Q_t^\top M_t\|_2 = \|Q_t Q_t^\top (M_t - V_t V_t^\top M_t)\|_2 \leq \|M_t - V_t V_t^\top M_t\|_2. \end{aligned}$$

Due to [12, Theorem 2.5.3], we get

$$\epsilon_{2,t} \leq \|M_t - V_t V_t^\top M_t\|_2 \leq \inf_{Z \in \mathbb{R}^{\mathcal{J} \times k_t}} \|M_t - W_t Z^\top\|_2$$

and conclude that the cluster basis constructed by our algorithm is at least as good as the one from Theorem 3.13. A similar estimate holds for the Frobenius norm.

Table 1: Recompression of cluster bases on the sphere Γ_S with cubic interpolation and recompression tolerance $\epsilon = 10^{-3}$

n	Build	Bld/n	Mem	Mem/n	MVM	Error
2048	11	5.4	7.5	3.7	0.01	5.9_{-4}
8192	46	5.6	34.9	4.3	0.11	6.5_{-4}
32768	186	5.7	147.4	4.5	0.47	6.9_{-4}
131072	760	5.8	607.9	4.6	2.15	7.0_{-4}
524288	3245	6.2	2685.9	5.2	9.46	7.4_{-4}

Table 2: Recompression of cluster bases on the cube Γ_C with quartic interpolation and recompression tolerance $\epsilon = 10^{-3}$

n	Build	Bld/n	Mem	Mem/n	MVM	Error
3072	27	9.1	12.8	4.3	0.03	2.9_{-4}
12288	113	9.2	46.6	3.9	0.14	4.3_{-4}
49152	452	9.2	168.0	3.5	0.51	5.6_{-4}
196608	1791	9.1	609.4	3.2	1.91	6.4_{-4}
786432	7097	9.0	2245.3	2.9	6.96	7.0_{-4}

5 Numerical experiments

We will now apply Algorithm 1 to two typical classes of densely populated matrices: discretized integral operators and the solution operators of discretized elliptic partial differential equations.

5.1 Integral operators

Our first example is the classical double layer potential operator

$$\mathcal{K}[u](x) := \int_\Gamma \frac{\langle x-y, n(y)\rangle}{4\pi\|x-y\|^3} u(y)\,dy,$$

where Γ is a Lipschitz surface. We approximate Γ by $n \in \mathbb{N}$ planar patches $\{\Gamma_1, \ldots, \Gamma_n\}$ and discretize it by a Galerkin method with piecewise constant basis functions. The resulting matrix $K \in \mathbb{R}^{n\times n}$ is given by

$$K_{ij} := \int_{\Gamma_i}\int_{\Gamma_j} \frac{\langle x-y, n(y)\rangle}{4\pi\|x-y\|^3}\,dy\,dx$$

for all $i, j \in \{1, \ldots, n\}$. We approximate K in two steps: first, we use local polynomial expansions [8] to construct a non-optimal approximation of K, then we apply an adapted variant [3] of Algorithm 1 to construct an $\mathcal{H}^2$-matrix $\widetilde{K}$.

The Tables 1 and 2 contain the results for the matrix K, where Γ is the three-dimensional unit sphere or the surface of the cube $[-1,1]^3$, respectively. The columns of the tables are interpreted as follows:

- The column "n" gives the number of degrees of freedom.
- The columns "Build" and "Build/n" give the time in seconds for the construction of the $\mathcal{H}^2$-matrix and the time per degree of freedom in milliseconds.
- The columns "Mem" and "Mem/n" give the storage requirements in MB for the $\mathcal{H}^2$-matrix and the storage requirements per degree of freedom in KB.
- The column "MVM" gives the time in seconds for the computation of the matrix-vector product.
- The column "Error" gives the relative spectral error $\|K - \widetilde{K}\|_2/\|K\|_2$ estimated by a power iteration.

We can see that the approximation error is bounded and that computing time and storage requirements grow proportionally to the number n of degrees of freedom.

5.2 Solution operators of elliptic partial differential equations

In the second example, we consider elliptic partial differential operators

$$\mathcal{L}[u](x) := -\operatorname{div} \sigma(x) \operatorname{grad} u(x)$$

in a Lipschitz domain $\Omega \subseteq \mathbb{R}^2$, where $\sigma : \Omega \to \mathbb{R}^{2\times 2}$ maps each $x \in \Omega$ to a symmetric positive definite coefficient matrix $\sigma(x)$.

We discretize $\mathcal{L}$ by a finite element method using standard nodal basis functions $(\varphi_i)_{i=1}^n$ on a triangulation of Ω and get a sparse matrix $L \in \mathbb{R}^{n\times n}$ given by

$$L_{ij} = \int_\Omega \langle \operatorname{grad} \varphi_i(x), \sigma(x) \operatorname{grad} \varphi_j(x)\rangle \, dx$$

for all $i, j \in \{1, \ldots, n\}$. Since L is sparse, it can be treated efficiently by standard techniques.

The corresponding inverse matrix L^{-1} is not sparse, but we can apply our algorithm to approximate it by an $\mathcal{H}^2$-matrix $S \approx L^{-1}$. Since computing L^{-1} directly for interesting problem dimensions would take too much time, we use an $\mathcal{H}$-matrix approximation instead, which can be computed using $\mathcal{H}$-matrix arithmetics [17, 14]. We then apply an efficient variant [7] of Algorithm 1 to convert the $\mathcal{H}$-matrix approximation into an $\mathcal{H}^2$-matrix.

Table 3 contains the results of a numerical experiment for the simple case of Poisson's equation in $[-1,1]^2$ with $\sigma \equiv 1$. The columns of this Table and Table 4 are interpreted as follows:

- "n", "Mem", "Mem/n", "MVM" are the same as in the case of the integral operators.

Table 3: Approximation of the solution operator of the elliptic partial differential equation with $\sigma \equiv 1$

n	$\mathcal{H}$Inv	$\mathcal{H}$Mem	Conv	Mem	Mem/n	MVM	InvErr
1024	< 1	2.1	< 1	1.6	1.6	< 0.01	4.3_{-4}
4096	5	13.5	5	8.4	2.0	< 0.01	8.6_{-4}
16384	45	84.9	35	40.5	2.5	0.07	9.5_{-4}
65536	387	516.6	207	183.0	2.8	0.34	7.4_{-4}
262144	2922	2970.6	1205	785.1	3.0	1.51	9.4_{-4}
1048576	20272	16457.3	6906	3253.5	3.1	6.64	9.8_{-4}

- "$\mathcal{H}$Inv" gives the time in seconds required by the $\mathcal{H}$-matrix inversion algorithm.
- "$\mathcal{H}$Mem" gives the storage requirements in MB for the $\mathcal{H}$-matrix.
- "Conv" gives the time in seconds for the conversion of the $\mathcal{H}$-matrix into an $\mathcal{H}^2$-matrix approximation.
- "InvErr" gives the inversion error $\|I - SL\|_2$ estimated by a power iteration.

We can see that the approximation error is bounded and that the storage requirements for the $\mathcal{H}^2$-matrix approximation and the time for the matrix-vector multiplication seem to be proportional to $\mathcal{O}(n \log n)$, where n is the number of degrees of freedom. The logarithmic factor can be explained by the fact that the approximation tolerance has to be proportional to n^{-1} in order to compensate for the growth of the condition number of the matrix, which is proportional to n.

We also observe that $\mathcal{H}^2$-matrices require significantly less storage than $\mathcal{H}$-matrices and that this advantage becomes more and more pronounced as the problem size increases.

Next, we investigate three problems with variable coefficients σ: in the first problem, we separate the square $[-1,1]^2$ into four quarters and switch the coefficients between 1 and 100:

$$\sigma_1(x) := \begin{cases} 100 & \text{if } x \in [-1,0) \times [-1,0) \text{ or } x \in [0,1] \times [0,1], \\ 1 & \text{otherwise.} \end{cases}$$

In the second problem, we separate the lower and upper half of the square by a strip with high conductivity:

$$\sigma_2(x) := \begin{cases} 100 & \text{if } x_2 \in [0, 1/16), \\ 1 & \text{otherwise.} \end{cases}$$

In the third problem, we introduce anisotropic coefficients in the lower half of the square:

$$\sigma_3(x) := \begin{cases} I & \text{if } x_2 \in [-1,0), \\ \operatorname{diag}(100,1) & \text{otherwise.} \end{cases}$$

Table 4: Approximation of the solution operator of the elliptic partial differential equation with variable σ

	Quartered, σ_1		Strip, σ_2		Anisotropy, σ_3	
n	Mem/n	MVM	Mem/n	MVM	Mem/n	MVM
1024	1.56	< 0.01	1.55	< 0.01	1.55	< 0.01
4096	2.09	0.01	2.10	0.01	2.11	0.01
16384	2.58	0.07	2.62	0.08	2.69	0.07
65536	2.98	0.36	3.05	0.37	3.24	0.39
262144	3.29	1.64	3.43	1.67	3.68	1.80
1048576	3.56	7.25	3.65	7.38	4.08	8.12

Table 4 only gives the storage requirements and times per matrix-vector-multiplication, since the run times and inversion errors are similar to those reported in Table 3. We again observe a log-linear growth of the storage requirements and the time for the matrix-vector multiplication, which again seems to be caused by the growth of the condition number.

Although the equations with varying coefficients are more complicated than the simple Poisson equation, the storage requirements and algorithmic complexity are only slightly higher, which fits the theoretical framework we have presented: the higher condition number of the former can be compensated easily, since the approximation error decreases exponentially if the rank is increased.

References

[1] M. Bebendorf and W. Hackbusch, *Existence of $\mathcal{H}$-matrix approximants to the inverse FE-matrix of elliptic operators with L^∞-coefficients*, Numerische Mathematik, 95 (2003), pp. 1–28.

[2] S. Börm, *$\mathcal{H}^2$-matrices — multilevel methods for the approximation of integral operators*, Comput. Visual. Sci., 7 (2004), pp. 173–181.

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

[4] ——, *$\mathcal{H}^2$-matrix arithmetics in linear complexity*, Computing, 77 (2006), pp. 1–28.

[5] S. Börm and L. Grasedyck, *Hybrid cross approximation of integral operators*, Numerische Mathematik, 101 (2005), pp. 221–249.

[6] S. Börm, L. Grasedyck, and W. Hackbusch, *Hierarchical Matrices.* Lecture Note 21 of the Max Planck Institute for Mathematics in the Sciences, 2003.

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

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

[9] S. Börm, M. Löhndorf, and J. M. Melenk, *Approximation of integral operators by variable-order interpolation*, Numerische Mathematik, 99 (2005), pp. 605–643.

[10] S. Börm and S. A. Sauter, *BEM with linear complexity for the classical boundary integral operators*, Mathematics of Computation, 74 (2005), pp. 1139–1177.

[11] K. Giebermann, *Multilevel approximation of boundary integral operators*, Computing, 67 (2001), pp. 183–207.

[12] G. H. Golub and C. F. V. Loan, *Matrix Computations*, Wiley-Interscience, New York, 1984.

[13] L. Grasedyck, *Theorie und Anwendungen Hierarchischer Matrizen*, PhD thesis, Universität Kiel, 2001.

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

[15] L. Greengard and V. Rokhlin, *A fast algorithm for particle simulations*, Journal of Computational Physics, 73 (1987), pp. 325–348.

[16] ——, *A new version of the fast multipole method for the Laplace in three dimensions*, in Acta Numerica 1997, Cambridge University Press, 1997, pp. 229–269.

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

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

[19] W. Hackbusch, B. Khoromskij, and S. 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.

[20] W. Hackbusch and Z. P. Nowak, *On the fast matrix multiplication in the boundary element method by panel clustering*, Numerische Mathematik, 54 (1989), pp. 463–491.

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

[22] S. Sauter, *Variable order panel clustering (extended version)*, Preprint 52/1999, Max-Planck-Institut für Mathematik, Leipzig, Germany, 1999.

[23] ——, *Variable order panel clustering*, Computing, 64 (2000), pp. 223–261.